INTERNATIONAL JOURNAL OF INNOVATIONS IN SCIENCE AND TECHNOLOGY

IJIST
P ISSN : 2618-1630
E ISSN : 2709-6130

International Conference on Innovations in Computing Technologies and Information Sciences

29-30 April 2024
UET PESHAWAR

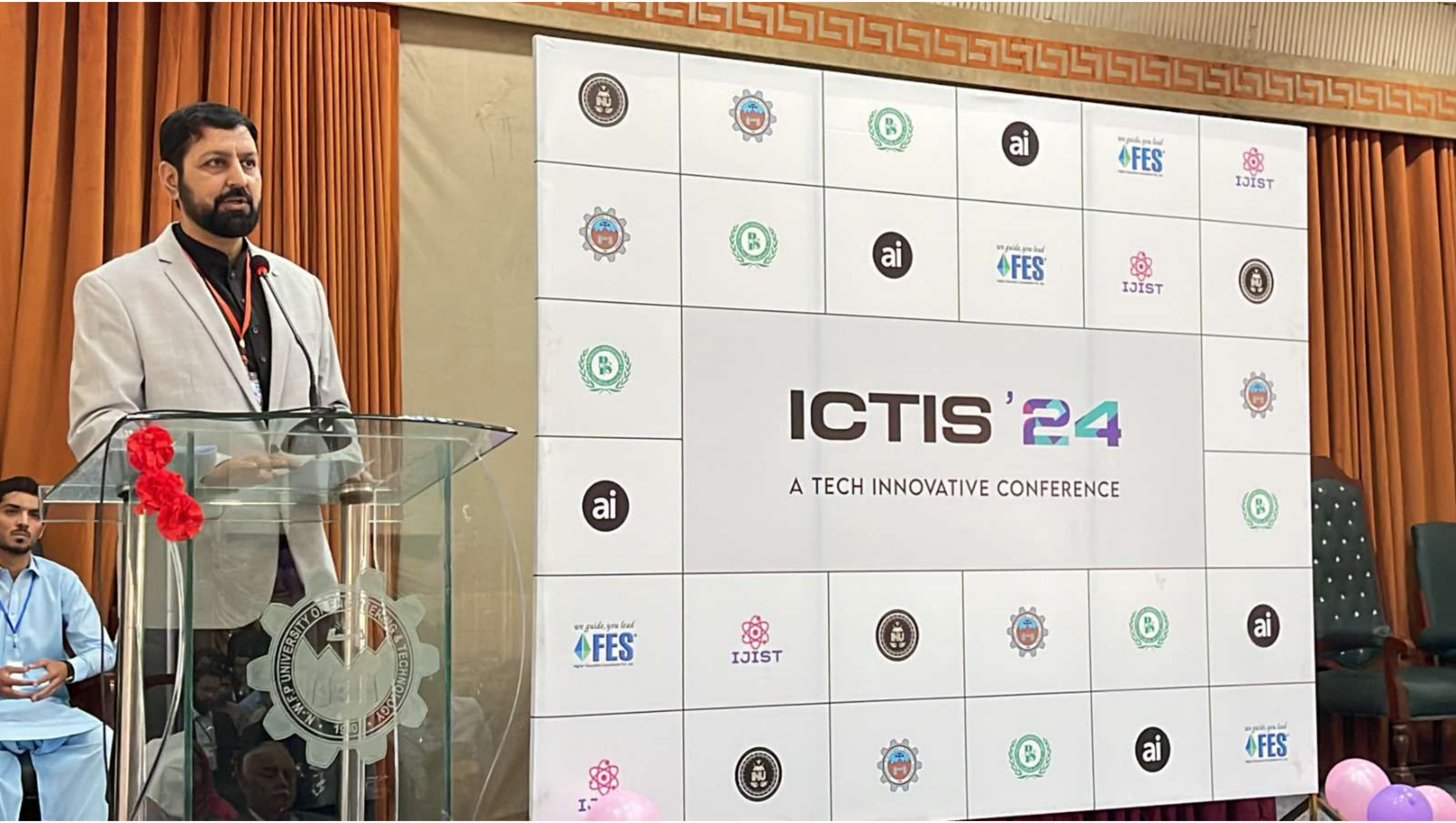
Journal.50sea.com

Special Thanks

DR. NASRU MINALLAH
Assistant Professor
UET Peshawar

SYED ATIF NAWAZ
Business Development
Manager
UET Peshawar

ICTIS '24
A TECH INNOVATIVE CONFERENCE
ai
we guide, you lead
FES
IJIST

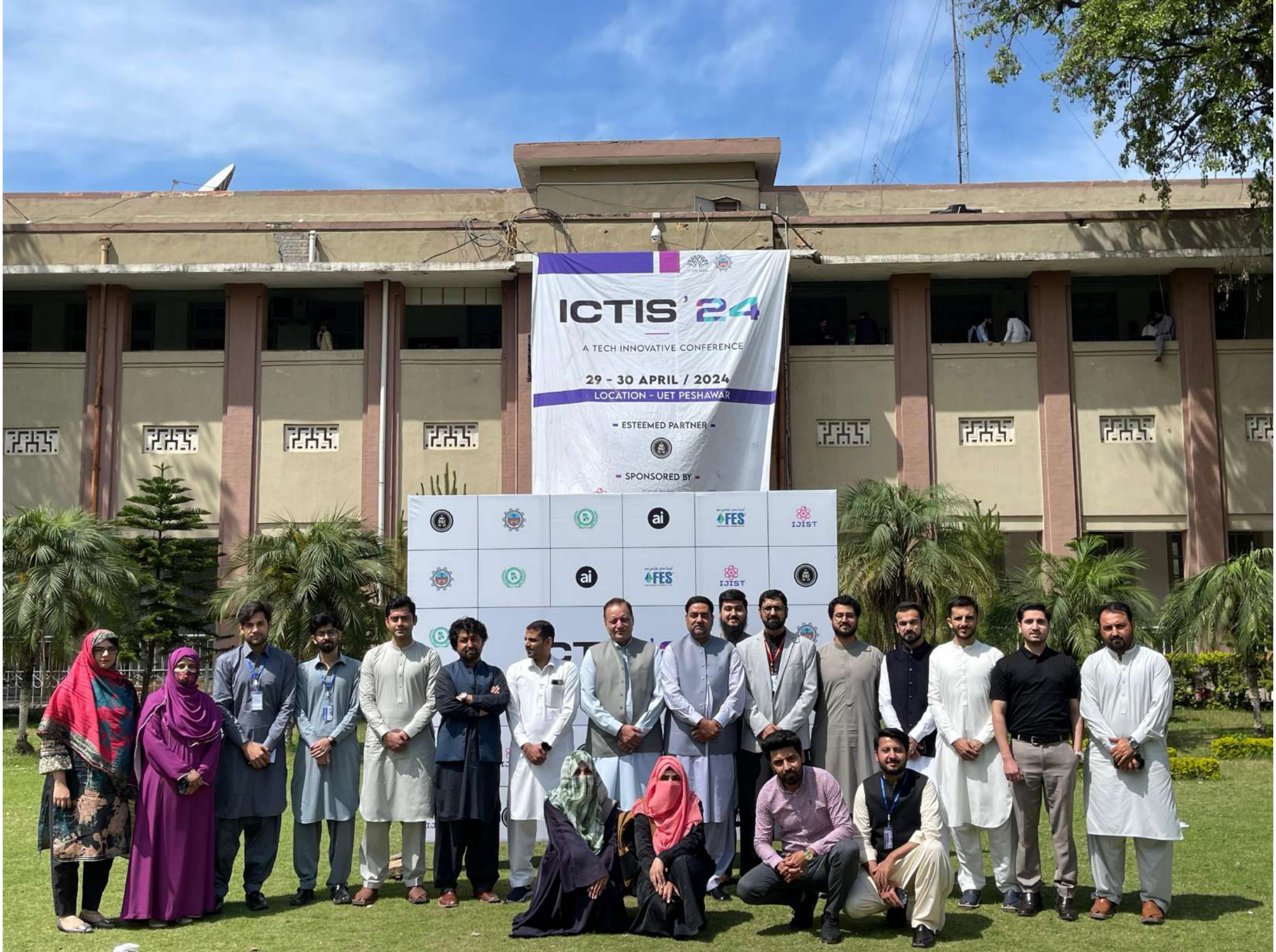

ICTIS '24
A TECH INNOVATIVE CONFERENCE
ai
FES
we guide, you lead
IJIST
N.W.F.P UNIVERSITY OF ENGINEERING & TECHNOLOGY

ICT
'24
A TECH IN
NFERENCE
we guide, you lead
FES
Higher Education Consultants Pvt. Ltd.
IJIST
ai
INU

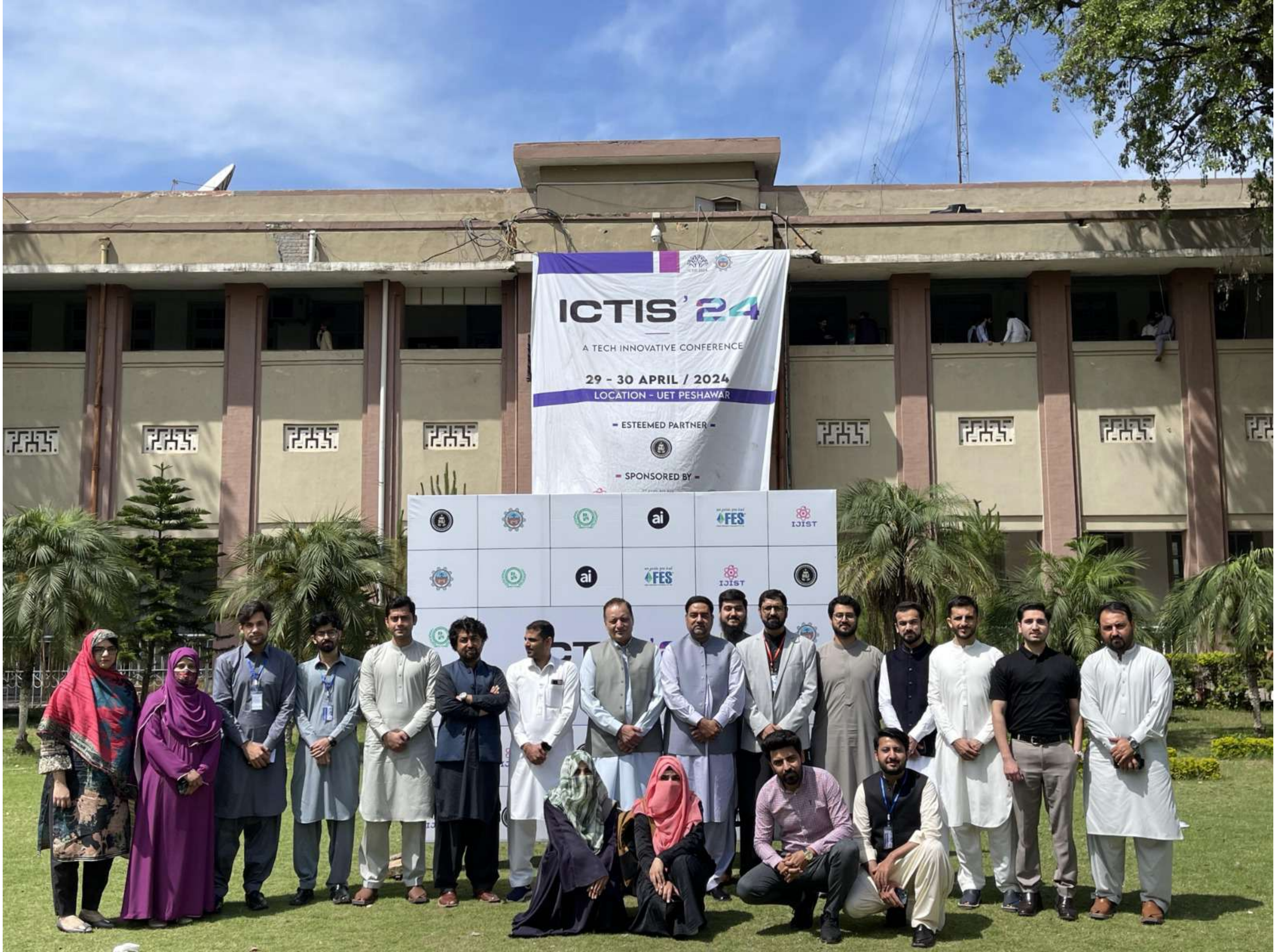
ICTIS '24
A TECH INNOVATIVE CONFERENCE
ai
we guide, you lead
FES
IJIST
WALL OF FAME

ai
FES
IJIST
ai
FES
IJIST

ICTIS '24
A TECH INNOVATIVE CONFERENCE
29 - 30 APRIL / 2024
LOCATION - UET PESHAWAR
ESTEEMED PARTNER
SPONSORED BY
ai
FES
IJIST
ICTIS '24


ICTIS '24
A TECH INNOVATIVE CONFERENCE
29 - 30 APRIL / 2024
LOCATION - UET PESHAWAR
ESTEEMED PARTNER
SPONSORED BY
ai
FES
IJIST

ICTIS'24
A TECH INNOVATIVE CONFERENCE
29 - 30 APRIL / 2024
LOCATION - UET PESHAWAR
ESTEEMED PARTNER
SPONSORED BY
IJIST
FES
ai
ICTIS'24


ICTIS' 24
A TECH INNOVATIVE CONFERENCE
29 - 30 APRIL / 2024
LOCATION - UET PESHAWAR
ESTEEMED PARTNER
SPONSORED BY
ai
FES
IJIST

ICTIS 2024
ICTIS '24
A TECH INNOVATIVE CONFERENCE
29 - 30 APRIL / 2024
LOCATION - UET PESHAWAR
ESTEEMED PARTNER
SPONSORED BY
ai
we guide, you lead
FES
IJIST
ICTIS '24

TABLE OF CONTENT

RESEARCH & INNOVATION DIVISION
IJIST

Detection of Holes in Point Clouds Using Statistical Technique

Zain ul Abideen[1], Hamza Ali[1], Muhammad Sajjad[1], Muhammad Abeer Irfan[1], Atif Jan[2], Yasir Saleem[1]

[1]Departement of Computer Systems Engineering University of Engineering and Technology, Peshawar, Pakistan

[2]Department of Electrical Engineering University of Engineering and Technology, Peshawar, Pakistan

***Correspondence**:zainikhan3434@gmail.com, hamzaali.dcse@gmail.com, muhammadsajad2710@gmail.com, abeer.irfan@uetpeshawar.edu.com, atifjan@uetpeshawar.edu.pk, yasirsaleem@uetpeshawar.edu.pk ,

Citation | Abideen. Z. U, Ali. H, Sajjad. M, Irfan. M. A, Jan. A, Saleem. Y, "Detection of Holes in Point Clouds Using Statistical Technique", IJIST, Special Issue pp 1-8, May 2024

Received | May 04, 2024, **Revised |** May 09, 2024, **Accepted |** May 15, 2024, **Published |** May 20, 2024.

A point cloud is a dynamic, three-dimensional geometric representation of data that has different qualities for every point, including geometry, normal vectors, and color. However, holes that often occur during the 3D point cloud collection process provide an immense obstruction to the analysis and reconstruction of point clouds. Thus, detecting these holes is a crucial initial step toward obtaining precise and comprehensive representations of the real surfaces. Although there are several methods available for hole detection and filling, the problem is exacerbated by their shortcomings, which include high computation complexity or limited effectiveness. Our method is based on a sequence of basic but efficient statistical techniques. Our method is based on a sequence of basic but efficient statistical techniques. First, we find the mean distances between each point using the K Nearest Neighbors (KNN) technique. Next, we can categorize normal points and points that belong to holes and borders by using this mean as a threshold. Our method's simplicity and low computational resource needs offer significant advantages over other approaches

Keywords: Point clouds; Hole detection; Surface reconstruction; Statistical analysis.

Introduction:

The evolution of laser scanning technology has democratized the process of digitizing objects, enabling precise and efficient data acquisition at scale. This advancement has catalyzed the emergence of reverse engineering; a process wherein digital models are generated from point cloud data as a valuable tool in various industries. A point cloud is a collection of points of data plotted in 3D space, using a 3D laser scanner. Such as, when scanning a building, every virtual point would correspond to a real spot on the metalwork, wall, window, or any other surface where the laser beam comes into contact. A point cloud can be defined as "A Point Cloud is an unordered set of 3-dimensional points in a frame of reference (Cartesian coordinate system) on the surface of objects." The point cloud is a set of discrete points obtained by scanning the object's surface with a three-dimensional (3D) scanning device. The point cloud can directly and stereoscopically represent the geometric features of the object's surface. Various additional details can be used along with 3D coordinates to represent them. A few are listed below:

- RGB color information associated with each point.
- Normal to surface at each point.
- Information about meshes (vertices & edges).

3D laser scanner technology is the foundation of 3D point cloud data. These electronic devices use visible light and lidar-based technology to gather detailed information about a specific area. The performance of the scanning device in use has a major impact on the final point cloud's clarity and accuracy. Nowadays, there are various sensors available for 3D scanning which are based on various principles some of which are mentioned below:

- Structured Light Cameras
- Time-of-flight sensors
- Stereo Vision
- Microsoft Kinect Cameras

Point cloud is of paramount importance in 3D object reconstruction, and object recognition with its high precision and fast processing speed [1][2]. It is also useful in Simultaneous Localization and Mapping (SLAM) and Preservation & restoration. It is possible to scan the complete surface of an object using high-fidelity scanners, but holes may appear in the final integrated model due to occlusions and maneuverability of the scanner while scanning. The holes in 3D objects as shown in Figure 1 may also occur due to factors such as occlusion, low surface reflectance coefficients, high grazing angles, missing parts in the original object, limited number of range scans from various viewing directions, laser range scanner functionality, etc.

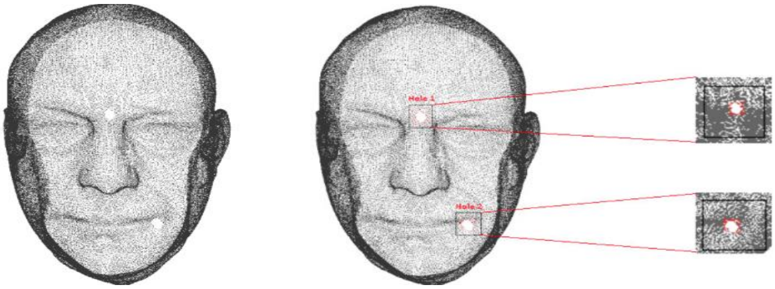


Figure 1: Holes in point clouds [3].

For example, some object portions may be absent because of accessibility or visibility issues, or because of unique physical characteristics of the scanned surface (transparency, reflectivity, etc.). This produces holes in the dataset, which do not correspond to any holes in the object. If the point cloud's boundary and hole points are not properly processed, it will influence subsequent research, such as point set simplification [4][5], point cloud registration [6], and hole repair [7]. It is worth mentioning that not all holes appearing in the estimated model are virtual, i.e. due to missing reconstructions of featureless surfaces [8]. A physical hole that is a part of the structure of the object being reconstructed may also be classified as a hole. Unfortunately, just looking at the point cloud, it is almost impossible to differentiate between these two types of holes (real and virtual). These real holes should be left as they are during the filling process. These days, a high-quality reconstruction of objects is an essential demand by many applications. Therefore, surface completion or hole-filling has grown in importance in the process of reconstructing 3D images. However, no hole filling can be applied without the detection of holes in point clouds. So, hole detection in the point cloud is an important component. Usually, 3D image reconstruction methods produce unstructured point clouds, in other words, the object's surface is not explicitly encoded with any connectivity information which makes the problem of the detection of holes on the surface an ill-defined problem [9]. In this paper, we introduce a statistical approach intended to identify holes within 3D point cloud data by using geometrical features in three-dimensional space. Our method relies on the principle of K-nearest neighbors (KNN) to identify neighboring points, followed by the computation of their average Euclidean distance. This average distance serves as a threshold, enabling the detection of irregularities in the point cloud. By comparing the distances between points, we can effectively pinpoint regions where data is missing, indicating the presence of holes.

Literature Review:

In this section, we study some existing methods for the detection of the hole boundary in the 3D point cloud. As we know the detection of point cloud boundaries and hole points is a vital task. In the state of the art of boundary extracting methods, different methods have been studied and developed to extract the boundary. These methods fall into two categories: (1) The methods executed on the grid; and (2) The methods executed directly on the data points of the point cloud. The former methods triangulate the point cloud and look for the adjacent triangular meshes. If there is no adjacent triangle, the associated triangle is considered as the boundary of the hole [4]. Authors in [10] subdivided the methods on meshes into volume-based methods and surface-based methods. In an early study [8], a seed boundary is selected on the point cloud grid to search for the next boundary according to the half-edge data structure. When the closed loop is reached to complete the boundary detection of holes, the search is finished. Many studies have recently focused on the methods of direct hole detection, which do not need to triangulate the point cloud in advance and can save a lot of time. In [9] the authors computed boundary probability for each point and classified the points in the point set into boundary or interior points through the application of the angle criterion. The coherence of the points is leveraged to extract a boundary loop that represents a simple hole boundary. As the point cloud is usually unstructured, therefore, to enhance the efficiency of neighborhood searches kD-tree, octree, or BSP-tree can be utilized. Along with the angle criterion authors in [9] also presented some other criteria to determine whether a point is a boundary point or not i.e. half disc criterion, shape criterion, and various enhancements based on these criteria. The author in [11], finds the neighborhood of each point in the point cloud and approximates the direction of the normal for detection of hole boundary in the point cloud. After finding the neighborhood of points, the author detected the candidate boundary points and created boundary polygons from boundary points. Following the detection of boundary points of the hole, an algebraic surface patch is fitted to the neighborhood and sample auxiliary points. Surfaces like spheres, cylinders, and planes are all recreated by this process. The author in [10] extracted the polygonal hole boundary with the Mean Shift approach to find out the vertices that have similar geometry properties with the hole region in the neighborhood of the boundary.

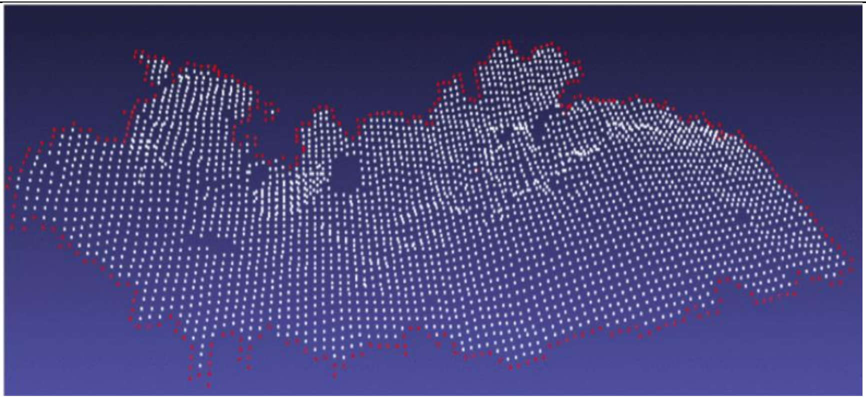


Figure 2: Detection of the surficial boundary of the point cloud described in [12].

Authors in [13] use the Distance-Centroid technique, which employs maximum squared distance and K-Nearest neighbors to detect the boundary points. The points that do not belong in the hole boundary are filtered out using Statistical Outlier Removal (SoR). To cluster out the different holes, the author used region-grow segmentation. The author in [14] proposed a method based on triangular mesh models which aims to find solid holes in 3D models. The author grouped the interconnected coplanar triangles and extracted the contour of the model using the boundaries of the nearby planes. Then, the author uses the extracted contour to form several disjoint clusters of model vertices and detect holes by analyzing the relationship between the clusters and plane. The author in [15] also uses the triangular mesh for the detection of holes in a point cloud. The author uses boundary edge tracking to automatically identify holes, an edge is a boundary edge if it is part of only one triangle; otherwise, it is an interconnected edge that is part of multiple triangles. Authors in [12] extracted exterior boundary points of the surface S and further classified them into exterior boundary points and inter points. Then looping is done for all points of S, if each of them is not an exterior boundary point, and one of its neighbors (in 4-connectivity) is empty, it is determined as a boundary point of a hole. The boundaries detected by the author are shown in Figure 2. The methods for boundary extraction based on the computation of convex hull have also been widely applied. The author in [16] proposed a method that involves modifying the convex hull to detect the boundary of the point cloud. For every point on the boundary p_b, the computation is repeated to calculate the smallest angle between p_b and its neighboring points for finding the next boundary points, but this method is time-consuming. The author in [17] computed the convex hull of a group of points, starting from an initial core point and its neighbors, and referred to it as a cluster. Then, by repeating the procedure and using the convex hulls of the core points as new core points, this cluster is extended repeatedly until each convex hull of a core point is joined with the cluster's main body. When all exterior points have been identified and the final boundary has been reached, clusters will not be combined or expanded further. The author in [18] proposed a method to detect the boundary of a surface of 3D points set. The author triangulated the convex hull C of the surface first and C is then optimized by computing and removing some outward triangles to obtain the exterior boundary of the surface. The author in [19] introduces a new method for identifying and fixing holes in point clouds. The author first calculated the density and 2D projection points of the point cloud. Then, the author used Euclidean distance to find the boundary points of the holes. It was followed by a preprocessing segmentation algorithm that helped to identify the boundary points more accurately. However, the author failed to detect the boundary of the whole point cloud simultaneously, so the method failed to detect the holes at boundary points.

Methodology:

Our proposed methodology consists of two steps first we find K-Nearest neighbors and then we calculate Euclidean distances. These two steps are employed to compare each point within the point cloud. A threshold is settled that facilitates the identification of boundary and hole points. If the distance between points exceeds the predefined threshold, then the point is a potential boundary point, and if it is not then it is a normal point. The threshold has been set as the mean of measured distances between points in the point cloud.

Dataset:

We have used the Kaggle digit dataset the "3D MNIST Dataset". The dataset includes 3D point clouds, which are sets of (x, y, z) coordinates produced from about 5,000 images in the original 2D MNIST dataset [20]. The point clouds' maximum dimension range is one, and their mean is zero.

K-Nearest Neighbors (KNN):

K-nearest neighbors refer to k () points close to the query point in the spatial distance. K-nearest neighbors search can be described as; given a point set S containing n points, for query point p_i ($p_i \in S$) there is a subset set C containing k points (not including point p_i), C ∈ *S*, k < n and for any $p_1 \in C$, $p_2 \in S$ - *C*, the following equation 1 can be met [21].

$$\text{dist}\,(p_i, p_1) \leq \text{dist}\,(p_i, p_2) \qquad (1)$$

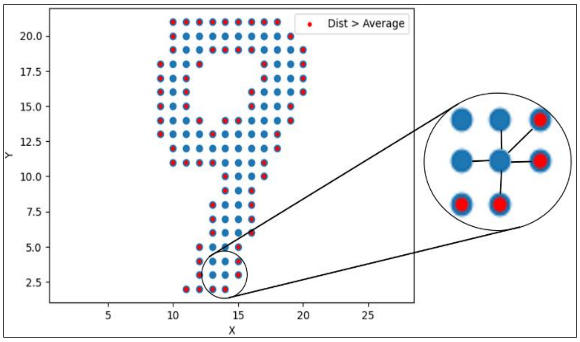


Figure 3: k-nearest neighbor in two-dimension

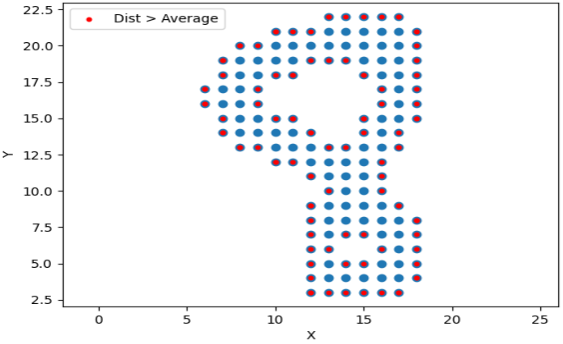


Figure 4: Boundary detection in the Kaggle digit

The k-nearest neighbors in two dimensions are shown in Figure 3, where the points are distributed in the two-dimensional space dispersedly, p is a query point and five points in the circle are the k (k=5) nearest neighbors. When the number of points is small, k-nearest neighbors can be obtained by calculating the query point to all other points' Euclidean distance directly, sorting the distance values, and then taking the first k values as the k-nearest neighbors [21].

Euclidean Distance:

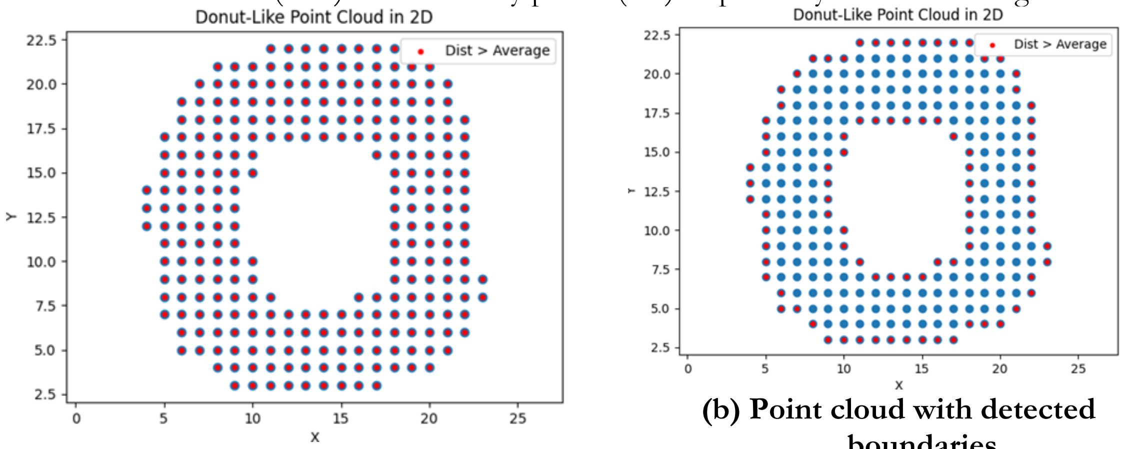
In our approach, we used KNN (K-Nearest Neighbors) for selecting the number of neighbors based on their Euclidean distances in 3D space. Next, we compute the distance between each point of the point cloud, identifying points with distances greater than this threshold as potential hole boundary points or point cloud boundaries. This process not only allows for effective hole detection but also provides insights into the overall structure and boundary of the point cloud. The Euclidean distances for 3-dimensional space are calculated in Equation 2.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (2)$$

Euclidean distance is applied to express the distance between two points on account of its convenience and simplicity of use. Moreover, it offers the advantage of requiring less computational power compared to more complex learning algorithms. We calculated the mean of the point cloud points while considering the mean of the K Nearest Neighbors (KNN). This involves computing distances between the points using KNN and then aggregating the mean distances of the neighbors. Subsequently, the average of these mean distances serves as the threshold, effectively separating points as hole or boundary points and normal points.

Results and Discussion:

As previously discussed, various approaches have been employed for hole detection, each with its own set of advantages and limitations. However, our approach, as detailed in this paper, takes a unique path toward hole detection, primarily relying on statistical techniques such as Euclidean distances. The Euclidian distances between neighboring points are given in Table 1. The distances between 5 points concerning each other are calculated using equation 2 and are recorded in the table. A small value of Euclidean distance between the points means the points are near to each other. For example, the distance between N1 and N5 is 0.11 and the distance between N1 and N3 is 0.22 meaning that N5 is nearest to N1 as compared to N3. The results obtained from this technique have proven to be highly satisfactory, as illustrated in Figure 5. Through this simple approach, we have achieved effective hole detection and boundary delineation within the point cloud. In Figure 4 our proposed method is applied to the Kaggle digit dataset and the points in the point cloud have been classified. The blue points show normal points while the red points are considered boundary points. In Figure 5a donut-like point cloud is shown. Before, all the points in the donut are red after looping through our method the points are classified as normal (blue) and boundary points (red) respectively as shown in Figure 5b.



(a) Input Point cloud

(b) Point cloud with detected boundaries

Figure 5: Hole boundary detection in point cloud

Table 1: Euclidean distances between neighboring points (k = 5)

5- Nearest Neighbors	N1	N2	N3	N4	N5
N1	0	0.12	0.22	0.18	0.11
N2	0.12	0	0.13	0.17	0.14
N3	0.22	0.13	0	0.15	0.16
N4	0.18	0.17	0.15	0	0.15
N5	0.11	0.14	0.11	0.15	0

Conclusion:

Hole detection in point clouds is a crucial initial step toward obtaining precise and comprehensive representations of the real surfaces. Our technique for the detection of holes is simple but effective and it takes a unique path towards hole detection, primarily relying on statistical techniques such as Euclidean distances and K Nearest Neighbors (KNN). For now, our method is limited to the detection of hole boundaries and point cloud boundaries in surficial point clouds, but in future work, we go on to modify this statistical technique and make it robust for the accurate detection of holes in complex point clouds.

References:

[1] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, “A novel local surface feature for 3D object recognition under clutter and occlusion,” Inf. Sci. (Ny)., vol. 293, pp. 196–213, Feb. 2015, doi: 10.1016/J.INS.2014.09.015.

[2] G. Sansoni, M. Trebeschi, and F. Docchio, “State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation,” Sensors 2009, Vol. 9, Pages 568-601, vol. 9, no. 1, pp. 568–601, Jan. 2009, doi: 10.3390/S90100568.

[3] R. A. Tabib et al., “Learning-Based Hole Detection in 3D Point Cloud Towards Hole Filling,” Procedia Comput. Sci., vol. 171, pp. 475–482, Jan. 2020, doi: 10.1016/J.PROCS.2020.04.050.

[4] H. Song and H. Y. Feng, “A progressive point cloud simplification algorithm with preserved sharp edge data,” Int. J. Adv. Manuf. Technol., vol. 45, no. 5–6, pp. 583–592, Nov. 2009, doi: 10.1007/S00170-009-1980-4/METRICS.

[5] H. Han, X. Han, F. Sun, and C. Huang, “Point cloud simplification with preserved edge based on normal vector,” Opt. - Int. J. Light Electron Opt., vol. 126, no. 19, pp. 2157–2162, Oct. 2015, doi: 10.1016/J.IJLEO.2015.05.092.

[6] J. Yang, Z. Cao, and Q. Zhang, “A fast and robust local descriptor for 3D point cloud registration,” Inf. Sci. (Ny)., vol. 346–347, pp. 163–179, Jun. 2016, doi: 10.1016/J.INS.2016.01.095.

[7] Y. Quinsat and C. lartigue, “Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics,” Int. J. Adv. Manuf. Technol., vol. 81, no. 1–4, pp. 411–421, Oct. 2015, doi: 10.1007/S00170-015-7185-0/METRICS.

[8] O. H. Nader H. Aldeeb, “Detection and Classification of Holes in Point Clouds”, [Online]. Available: https://www.semanticscholar.org/paper/Detection-and-Classification-of-Holes-in-Point-Aldeeb-Hellwich/2feb6ee457c21a3565763fa4b472cd6164a38d1e

[9] “Detecting holes in point set surfaces”.

[10] Q. He, S. Zhang, X. Bai, and X. Zhang, “Hole filling based on local surface approximation,” ICCASM 2010 - 2010 Int. Conf. Comput. Appl. Syst. Model. Proc., vol. 3, 2010, doi: 10.1109/ICCASM.2010.5620018.

[11] P. Chalmovianský and B. Jüttler, “Filling Holes in Point Clouds,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2768, pp. 196–212, 2003, doi: 10.1007/978-3-540-39422-8_14.

[12] V. S. Nguyen, T. H. Trinh, and M. H. Tran, “Hole Boundary Detection of a Surface of 3D Point Clouds,” Proc. - 2015 Int. Conf. Adv. Comput. Appl. ACOMP 2015, pp. 124–129, Feb. 2016, doi: 10.1109/ACOMP.2015.12.

[13] “Automatic Hole Detection for Selective Hole Filling in Point Cloud Data - Results.” Accessed: May 05, 2024. [Online]. Available: https://www.researchgate.net/publication/316975536_Automatic_Hole_Detection_for_Selective_Hole_Filling_in_Point_Cloud_Data_-_Results

[14] Y. Wang, R. Liu, F. Li, S. Endo, T. Baba, and Y. Uehara, “AN EFFECTIVE HOLE DETECTION METHOD FOR 3D MODELS”.

[15] J. Wang and M. M. Oliveira, “Filling holes on locally smooth surfaces reconstructed from point clouds,” Image Vis. Comput., vol. 25, no. 1, pp. 103–113, Jan. 2007, doi: 10.1016/J.IMAVIS.2005.12.006.

[16] A. Sampath and J. Shan, “Building boundary tracing and regularization from airborne lidar point clouds,” Photogramm. Eng. Remote Sensing, vol. 73, no. 7, pp. 805–812, 2007, doi: 10.14358/PERS.73.7.805.

[17] S. Suer, S. Kockara, and M. Mete, “An improved border detection in dermoscopy images for density based clustering,” BMC Bioinformatics, vol. 12, no. SUPPL. 10, pp. 1–10, Oct. 2011, doi: 10.1186/1471-2105-12-S10-S12/FIGURES/9.

[18] G. J. HUANG Xianfeng, CHENG Xiaoguang, ZHANG Fan, “SIDE RATIO CONSTRAIN BASED PRECISE BOUNDARY TRACING ALGORITHM FOR DISCRETE POINT CLOUDS”, [Online]. Available: https://www.isprs.org/proceedings/xxxvii/congress/3b_pdf/69.pdf

[19] C. Zhang, H. Zhou, and J. Duan, “A method for identifying and repairing holes on the surface of unorganized point cloud,” Measurement, vol. 210, p. 112575, Mar. 2023, doi:

10.1016/J.MEASUREMENT.2023.112575.

[20] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." Accessed: May 05, 2024. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[21] D. Li and A. Wang, "Improved KNN algorithm for scattered point cloud," Proc. 2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2017, pp. 1865–1869, Sep. 2017, doi: 10.1109/IAEAC.2017.8054336.

Visually: Assisting the Visually Impaired People Through AI-Assisted Mobility

Muhammad Arsalan Kamran*, Alishba Orakzai, Umama Noor, Yasir Saleem Afridi, Madiha Sher.

Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, Pakistan.

Correspondence: Muhammad Arsalan Kamran arsalankamran80@gmail.com

Citation | Kamran. M. A, Orakzai. A, Noor. U, Afridi. Y. S, Sher. M, "Visually: Assisting the Visually Impaired People Through AI-Assisted Mobility", IJIST, Special Issue pp 9-17, May 2024

Received | May 05, 2024, **Revised |** May 10, 2024, **Accepted |** May 16, 2024, **Published |** May 20, 2024.

This research introduces "Visually", a revolutionary mobile application that aims to address the complications that visually impaired people come across in their daily lives. By deploying advanced deep learning models for real-time object detection, facial recognition, and currency identification with voice outputs for each feature, the "Visually" application strives to enhance the autonomy, independence, and mobility of visually impaired people. The system undergoes thorough training on a diverse dataset, incorporating augmentation techniques to enhance the robustness of the models. The project's multifaceted objectives include a user-friendly interface, real-time object detection, multi-modal recognition, Text-to-Speech audio output, and an overarching aim of enriching the lives of visually impaired individuals. Driven by the global prevalence of visual impairment and the demand for cost-effective solutions, "Visually" is aligned with international efforts for accessibility and inclusivity. For cross-platform compatibility, the machine learning models have been integrated whilst being deployed with TensorFlow Lite. With Offline availability, the application ensures accessibility even in rural areas with limited network connectivity. To make a substantial societal impact "Visually" aims to contribute to a more inclusive and equitable society, by transforming the way visually impaired individuals navigate around the environment. Positioned at the intersection of technology, accessibility, and empowerment, the "Visually" project is poised to bring about positive change for a community that frequently encounters unique challenges in their daily lives.

Keywords: Visually Impaired, Assistive Technology, Mobility Aids, Navigation Assistance, Vision Impairment, AI-Assisted Mobility

Introduction:

In the realm of assistive technology, individuals with visual impairment face challenges every day that significantly impact their self-sufficiency, independence, and mobility. Despite the various existing technological aids, the challenge of navigating independently in unacquainted environments remains a pervasive issue, often requiring reliance on sighted assistance. This paper aims to address such challenges through the introduction of "Visually," an innovative mobile application developed to empower the visually impaired community. The primary goal of the project is to offer a real-time vision system, incorporating object detection, face recognition, and currency recognition features. These functionalities are integrated into the "Visually" application, incorporating deep learning models trained on a diverse dataset carefully curated for the unique needs of the visually impaired. This introduction outlines the background, research problem, and significance of the "Visually" project, which aims to ameliorate the daily experiences of visually impaired individuals by enhancing their independence, accessibility, and mobility. The main contributions of this research include addressing the challenges faced by visually impaired individuals, the global prevalence of visual impairment, and the development of an affordable and accessible solution with the potential to have a positive societal impact. The following sections will delve into the technical details of the model architectures, training methods, and the expected impact of the "Visually" application on its users.

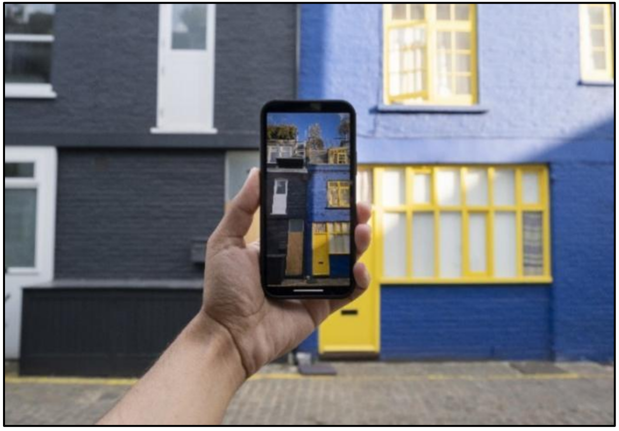


Figure 1: Navigation support using mobile application. [1]

Figure 1 illustrates the navigation support provided by the mobile application developed in the "Visually" project. The image shows a person using a smartphone application to navigate their surroundings, with the application providing real-time information about the user's environment through object detection and voice output.

Related Work:

Microsoft's "Seeing AI":

The landscape for assistive technologies for visually impaired individuals has witnessed significant contributions from notable projects and research endeavors. One of the major initiatives is Microsoft's "Seeing AI," a cloud-powered application that integrates Machine Learning (ML) to describe the user's surroundings and convert it into voice output, including features like text recognition, scenic descriptions, and more. "Seeing AI" can be considered a valuable benchmark for understanding the widespread usage beneficial for visually impaired users, influencing the development decisions and design considerations for the "Visual" application [2].

Notify (Currency Recognition):

Furthermore, "Notify", an application made in India for currency detection focuses on verifying accurate financial handling for visually impaired users, making it a worthy solution for the requirement [3].

Cutting-Edge Object Detection Techniques:

In the field of advanced object detection, Mahendran and his team explore cutting-edge solutions based on real-time vision systems using deep learning and point cloud processing. By investigating these techniques, the team helps the application recognize objects quickly and accurately. Their research is crucial for refining the goals of the "Visually" application, especially in computer vision and object recognition [4].

Deep-See Face Framework:

The DEEP-SEE FACE framework introduces an intriguing approach using convolutional neural networks (CNNs) for real-time facial recognition. This technology provides valuable insights for developing facial recognition models, focusing on features like hard negative mining and acoustic communication. These aspects align with "Visual's" goals of improving user understanding and communication, making the application more user-friendly and effective for visually impaired individuals [5].

Currency Recognition System:

A notable innovation is a currency recognition system designed for the visually impaired, enabling real-time identification of currency notes and obstacle-aware navigation. This system offers essential considerations for integrating an efficient and accurate currency recognition feature into the "Visually" application, ensuring that users can manage their finances independently and safely [6].

Summary:

In summary, the insights from these different sources serve as guiding principles for the development of the "Visually" application. Each project and research paper provides unique perspectives and considerations, collectively shaping innovative, user-centric solutions aimed at transforming the lives of visually impaired individuals. Through the integration of these technologies, "Visually" aims to empower users, enhancing their independence and quality of life.

Material and Methods:

The project's research design was meticulously crafted to ensure the effective implementation of its goals. The development of the "Visually" project follows a thorough and systematic methodology, covering all stages from data collection to model deployment. This approach is detailed in the following sections, providing a glimpse into the research design, data acquisition process, model training, and deployment strategies. Fig 2 demonstrates the development stages for the visual application.

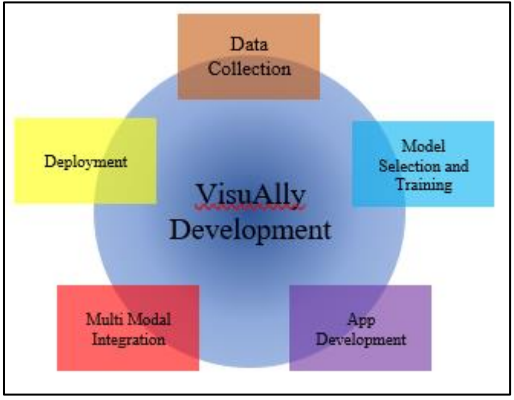


Figure 2: Development Stages for Visually App

Data Collection:

The "Visually" Project is built upon a rich and extensive dataset, specifically tailored and designed to meet the unique requirements of visually impaired people. The dataset includes a wide variety of indoor and outdoor objects, with a focus on objects essential for daily navigation. To ensure the model, that is to be trained on this dataset, is adaptable to various real-world scenarios, the dataset undergoes careful augmentation and labeling. Techniques such as rotation, scaling, and flipping the images are employed to improve the model's ability to learn different features of the images and enhance its accuracy in identifying objects from different angles and lighting conditions.

Model Architecture and Training:

The core of the "Visually" application is its deep learning models, which utilize the "You Only Look Once (YOLO)" architecture known for its ability to detect objects in real-time videos with high accuracy. The YOLO model is pre-trained on a diverse dataset but can be self-trained on specific datasets to detect specific objects. This model is trained on the augmented dataset using advanced optimization methods, such as stochastic gradient descent. During training, the focus is on minimizing detection loss while enhancing the model's accuracy in recognizing and pinpointing objects in the user's environment.

Table 1: Differences between Yolo Architecture and Mobile Net SSD

Features	YOLOv5	Mobile Net SSD
Number of Layers	Backbone: 53 layers Additional: Depends on variant	Backbone: 13 layers Additional: 6 layers
Optimizer Used	Adam	Adam
Number of Epochs	Around 300	Around 200
Dataset	COCO dataset	VOC dataset
Images Used	Around 118K images	7,000 images
Average Accuracy	mAP of 50.1 on COCO test set	mAP of 72.7 on VOC test set
Frames per Second	60+ FPS on a GPU	Over 100 FPS on modern hardware
Model Size	27 MB	23 MB

Table 1 compares two popular object detection models, YOLOv5 and Mobile Net SSD, across various metrics. YOLOv5 is a state-of-the-art object detection model known for its efficiency and accuracy. It features a highly optimized architecture, enabling it to achieve real-time inference speeds on GPU hardware. YOLOv5 has gained popularity for its simplicity, ease of use, and impressive performance on a variety of datasets. It offers a range of pre-trained models of different sizes (e.g., YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x), allowing users to choose the right balance between speed and accuracy for their specific application. Additionally, YOLOv5 is actively maintained and updated, ensuring compatibility with the latest advancements in object detection research. Overall, YOLOv5 is a compelling choice for developers seeking a powerful and efficient object detection solution.

Multi-Modal Recognition:

The "Visually" application offers a comprehensive user experience by integrating various recognition features such as object, face, and currency recognition. Face recognition utilizes the weights of the pre-trained models and performs personalized training to achieve precise identification. Currency recognition involves the development of a specialized dataset and model, providing users with information about the currency notes they come across [7].

Text-to-Speech (TTS) Integration:

To enhance communication with visually impaired people, the "Visually" application provides an audio output feature powered by Text-to-Speech (TTS) technology. This feature not only describes the objects but also provides contextual information, such as the proximity to the obstacles, identifying which feature the user is currently using, and giving an audio output of the recognized faces, etc.

Deployment:

To achieve compatibility across Android and IOS platforms, all machine/deep learning models in the application are converted into TensorFlow Lite (tflite) versions. The application is developed using the Flutter framework, which utilizes the tflite_flutter_helper package for efficient image processing and input preprocessing.

Offline Functionality:

Recognizing the importance of accessibility in areas with limited network coverage, "Visually" emphasizes offline functionality. By using tflite files, all the models will be stored on the device which will allow the user to make use of the features even without internet services. Face embeddings and emergency contacts are stored locally, reducing the reliance on a constant internet connection. This approach aligns with the project's goal of creating a user-friendly, adaptive, and accessible solution for visually impaired individuals, ensuring that the application effectively addresses their unique challenges and contributes to positive societal impact [8].

Results and Discussion:

Table 2 provides a detailed breakdown of the accuracy achieved by the "Visual" application in detecting various objects essential for visually impaired individuals. The table showcases the impressive accuracy rates for detecting different objects, highlighting the application's proficiency in real-time object recognition. With a focus on key objects like people, laptops, water bottles, furniture, cars, motorcycles, and houseplants, the "Visually" application demonstrates high accuracy levels, ranging from 90% to 99%. These accuracy rates underscore the application's ability to swiftly and accurately identify a diverse range of objects, enhancing users' spatial awareness, safety, and overall navigation experience.

By utilizing the YOLO-based object detection model, "Visually" can swiftly and accurately identify a wide range of objects in real-time, such as obstacles, furniture, and other navigation aids. This feature offers users instant information about their environment, improving their spatial awareness and safety. Our thorough assessment of "Visual's" object detection abilities revealed exceptional performance, demonstrating the reliability and effectiveness of the integrated technologies.

Laptops Detection:

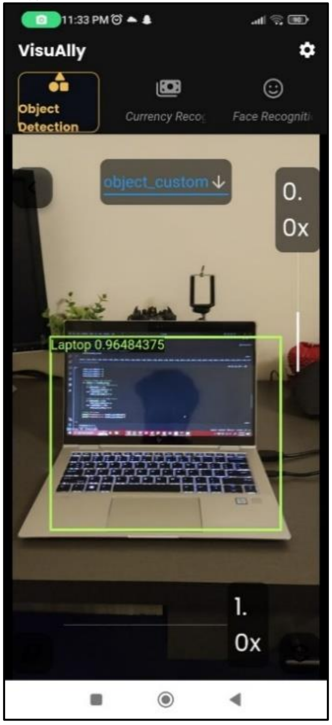


Figure 3: Laptop Detected

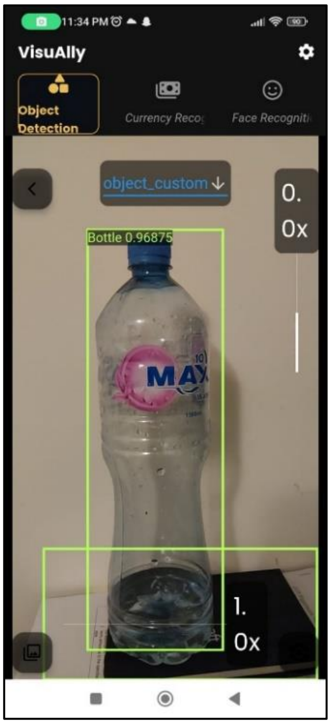


Figure 4: Bottle Detected

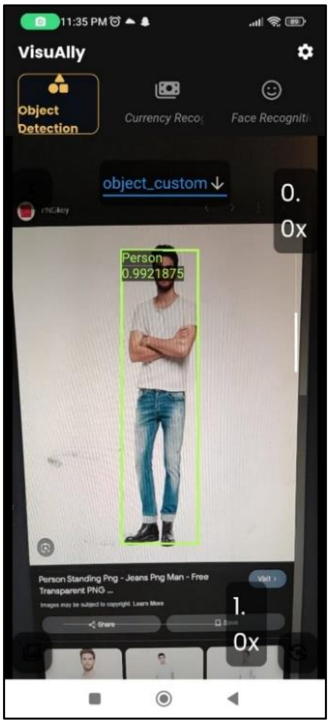


Figure 5: Person Detected

Figure 3 shows the snapshot of the application detecting a laptop. The model demonstrated exceptional proficiency in identifying laptops, achieving an impressive accuracy of 96%. This high level of accuracy is essential for users who rely on the application to navigate environments where laptops are prevalent.

Bottle Detection:

The model detected everyday objects, like water bottles, with exceptional accuracy, reaching 96%, as demonstrated in Fig 4. This ensures that the application can help a user in assisting with daily tasks such as locating personal items or identifying objects in their immediate vicinity.

Table 2: Accuracy achieved by Visually Application

Object Detected	Model Accuracy
Person	99%
Laptop	96%
Water Bottle	96%
Furniture	90%
Car	98%
Motorcycle	98%
Houseplants	96%

Person Detection:

Recognizing individuals in front of the user is a critical aspect of the application's functionality. In Fig 5, the detection of a person can be seen. Our model achieved an impressive accuracy of 99% in person detection, ensuring users are aware of the presence of others in their surroundings.

Car Detection:

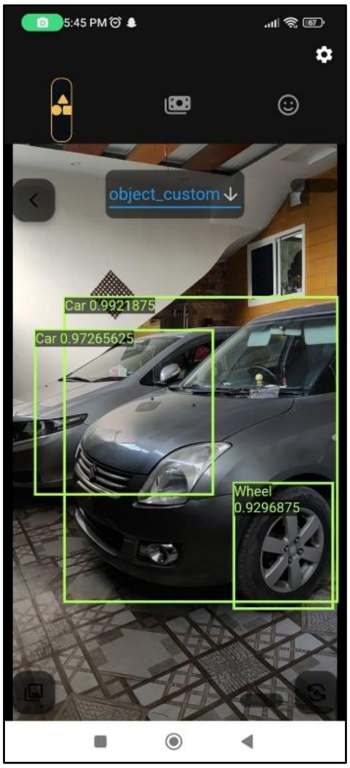


Figure 6: Car Detected

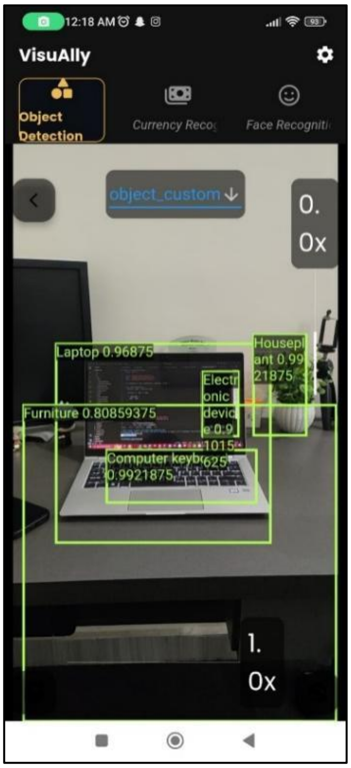


Figure 7: Multi Objects Detected

The most important objects to be detected in daily navigation are vehicles. Our model successfully detects cars in front of the user with a remarkable accuracy of 99%, making daily navigation easy for visually impaired people. Fig 6 demonstrates the snapshot of our application where it is detecting vehicles.

Multi-Object Detection:

The application's ability to detect multiple objects simultaneously further enhances its utility. With a focus on accuracy, the application provides users with detailed information about their environment, allowing for more informed navigation decisions. Fig 7 shows the multi-object detection capabilities of Visually. Moreover, TTS API is utilized to announce each object to the user that is detected, providing detailed information about their environment and further enhancing the application's usability.

Discussion:

The "Visually" application enhances the mobility of visually impaired individuals through the integration of YOLO v5 for real-time object detection and Google's ML Kit for text-to-speech conversion. These technologies offer several key benefits:

Immediate Environmental Feedback:

By incorporating real-time object recognition technology, visually impaired people may navigate safely and autonomously by being promptly informed about their surroundings. Their improved spatial awareness makes them more competent navigators.

High Precision Object Identification:

The remarkable accuracy of YOLO v5 ensures accurate information about the surroundings is provided and informs the user about the items in their immediate environment, hence enhancing their spatial comprehension.

Efficient Navigation:

The application vocalizes the names of detected items, especially in unfamiliar locations, reducing users' cognitive load and allowing them to focus more on their surroundings.

Enhanced Safety:

The safety of visually impaired people is improved by accurate object detection and TTS announcements, which notify them of potential hazards or obstructions in their route. Because of the trustworthy information given to visually impaired people, they may navigate with more confidence thanks to this proactive approach to safety.

User-Friendly Interface:

The user-friendly interface of the application makes it easier even for consumers with little technological background. The incorporation of these technologies into a Flutter application results in an intuitive user interface that is simple to use. This accessibility is essential to guarantee that a broad spectrum of users, regardless of their level of technical expertise, can use the application efficiently.

In conclusion, the integration of YOLO v5 and Google's ML Kit in the "VisuAlly" application greatly enhances the mobility and independence of individuals with visual impairments. This integration allows them to confidently explore their surroundings, ensuring they have the necessary information for safe and effective navigation through the combination of real-time object detection and text-to-speech conversion.

Competitive Analysis:

“Visually” aims at providing an extensive solution to enhance the independence and mobility of visually impaired and blind people, exhibiting several significant advantages in comparison to the already available apps designed for the same objective. This discussion concentrates on elucidating the project’s results and implications, including the comparison of results with the available solutions, whilst highlighting the significant contribution of “Visually” to the field.

Comparison with Existing Apps:

“Visually” discerns itself from the multiple existing apps for visually impaired people through its encompassing approach and multi-modal features. Current apps mostly focus on singular functionalities such as object detection or navigation, whereas “Visually” incorporates real-time object detection with face recognition and currency recognition, making it an extensive solution catering to meeting the multiple needs of users. Additionally, features like offline mode, user-friendly design, and continuous user feedback integration make "Visually" unique [9].

Other popular apps like Be My Eyes and Seeing AI offer useful features such as remote assistance and text recognition. However, they may not offer the same depth of functionalities and offline capabilities as "Visually." The use of YOLO architecture for object detection in "Visually" enhances its real-time processing capabilities, setting it apart from other solutions.

Significance of Results and Contribution to the Field:

The "Visually" project represents a significant advancement in assistive technology, specifically tailored to address the complex challenges faced by visually impaired individuals in their daily lives. Through its versatile and integrated approach, the application not only enhances users' independence but also reduces their reliance on others, fostering a sense of empowerment. One of the key strengths of "Visually" lies in its offline functionality, which ensures accessibility in a variety of environments, thereby filling a crucial gap in existing solutions. Moreover, the project's significance goes beyond its technical capabilities; it embodies a commitment to inclusivity, affordability, and user-centered design principles.

By aligning with global initiatives for accessibility, "Visually" acknowledges the widespread impact of visual impairment and strives to make a meaningful difference in the lives of affected individuals. The integration of user feedback further enhances the project's value, ensuring that the application evolves in response to the evolving needs of its users [10].

Conclusion:

"Visually" emerges as a groundbreaking assistive technology, designed to address daily challenges for visually impaired individuals. Utilizing advanced deep learning models, including real-time object detection, face recognition, and currency recognition, the application aims to transform the landscape of independence and mobility.

One of the key features of "Visually" is its commitment to a user-friendly interface and multi-modal recognition, including Text-to-Speech audio. This comprehensive approach is further enhanced by rigorous training on a diverse dataset, ensuring the application's adaptability to various real-world scenarios and reflecting its focus on meeting user needs. In line with global accessibility initiatives, "Visually" places a strong emphasis on affordability and offline functionality, which are essential for users in diverse environments. By combining the YOLO architecture for real-time processing with a holistic approach, the application surpasses existing solutions in its field.

What sets "Visually" apart from other apps is its offline capabilities, user-friendly design, and the integration of continuous user feedback. Beyond its technical capabilities, the application embodies inclusivity and user-centric design principles, contributing to a more equitable society. In conclusion, "Visually" represents a significant step towards empowerment for visually impaired individuals, redefining how they navigate the world and fostering inclusivity. With its responsive development cycle, the application is poised to have a lasting impact, evolving to meet the changing needs of its users and contributing to a more accessible world.

Acknowledgment: We acknowledge the usage of the GenAI platform for refining and enhancing the writing quality of this manuscript intended for submission at the ICTIS'24 conference.

References:

[1] "Seeing AI - Talking Camera for the Blind." Accessed: May 06, 2024. [Online]. Available: https://www.seeingai.com/

[2] "Noteify: Indian Currency Recognition App", [Online]. Available: https://github.com/chandran-jr/Noteify

[3] J. K. Mahendran, D. T. Barry, A. K. Nivedha, and S. M. Bhandarkar, "Computer vision-based assistance system for the visually impaired using mobile edge artificial intelligence," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., pp. 2418–2427, Jun. 2021, doi: 10.1109/CVPRW53098.2021.00274.

[4] B. Mocanu, R. Tapu, and T. Zaharia, "DEEP-SEE FACE: A Mobile Face Recognition System Dedicated to Visually Impaired People," IEEE Access, vol. 6, pp. 51975–51985, Sep. 2018, doi: 10.1109/ACCESS.2018.2870334.

[5] "Currency Recognition System Using Image Processing: Libyan Banknote as a Case Study ." Accessed: May 06, 2024. [Online]. Available: http://www.warse.org/IJETER/static/pdf/file/ijeter171022022.pdf

[6] R. C. Joshi, S. Yadav, M. K. Dutta, and C. M. Travieso-Gonzalez, "Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People," Entropy 2020, Vol. 22, Page 941, vol. 22, no. 9, p. 941, Aug. 2020, doi: 10.3390/E22090941.

[7] J. Madake, S. Bhatlawande, A. Solanke, and S. Shilaskar, "A Qualitative and Quantitative Analysis of Research in Mobility Technologies for Visually Impaired People," IEEE Access, vol. 11, pp. 82496–82520, 2023, doi: 10.1109/ACCESS.2023.3291074.

[8] "REAL TIME OBJECT DETECTION WITH SPEECH RECOGNITION USING TENSORFLOW LITE." Accessed: May 06, 2024. [Online]. Available: https://www.researchgate.net/publication/359393141_REAL_TIME_OBJECT_DETECTION_WITH_SPEECH_RECOGNITION_USING_TENSORFLOW_LITE

[9] N. Alsharabi, "Real-Time Object Detection Overview: Advancements, Challenges, and Applications," مجلة جامعة عمران , vol. 3, no. 6, pp. 12–12, Nov. 2023, doi: 10.59145/JAUST.V3I6.73.

[10] K. Vaishnavi, G. P. Reddy, T. B. Reddy, N. C. S. Iyengar, and S. Shaik, "Real-time Object Detection Using Deep Learning," J. Adv. Math. Comput. Sci., vol. 38, no. 8, pp. 24–32, Jun. 2023, doi: 10.9734/JAMCS/2023/V38I81787.

Modeling of Post-Myocardial Infarction and Its Solution Through Artificial Neural Network

Naheed Ali, Dr. Noor Badshah
Dept. of Basic Sciences and Islamiat University of Engineering and Technology Peshawar, Pakistan
***Correspondence**: naheedali581@gmail.com, noorbadshah@uetpeshawar.edu.pk,

Citation | Ali. N, Badshah. N, "Modeling of Post-Myocardial Infarction and Its Solution Through Artificial Neural Network", IJIST, Special Issue pp 18-29, May 2024
Received| May 02, 2024, **Revised|** May 08, 2024, **Accepted|** May 14, 2024, **Published|** May 20, 2024.

Cardiovascular diseases, particularly myocardial infarction (MI) constitute a significant health concern globally. A myocardial infarction, which is commonly known as a heart attack, happens when a part of the heart muscle doesn't get enough blood because of a blockage. Studying MI is complex and it requires looking at it from different angles. In recent years the fusion of mathematical modeling and artificial intelligence (AI) techniques has emerged as a promising avenue for understanding the complexities associated with MI. The primary goal of this study is to provide an AI-based solution for a new nonlinear mathematical model related to myocardial infarction phenomena. To obtain the solution we will use a well-known deep learning technique, known as artificial neural networks (ANNs) with the combination of the optimization technique Levenberg-Marquardt back propagation (LMB). This combined method is referred to as ANNs-LMB. The results obtained from the model using ANNs-LMB are compared with a reference dataset constructed through the adaptive MATLAB solver ode45. The numerical performance is validated through a reduction in mean square error (MSE). The MSE is around 10^{-6} and the obtained results by ANNs-LMB almost overlapped with the reference dataset, which shows the accuracy and efficiency of the proposed methodology.

Keywords: Artificial Neural Network; Myocardial Infarction; Mathematical Modeling.

Introduction:

Myocardial Infarction, often referred to as a heart attack, stands as a significant contributor to morbidity and mortality globally. Myocardial infarction causes 17.1 million deaths per year throughout the world [1]. Based on the latest statistics from the World Health Organization (WHO) on the incidence of heart attacks in Pakistan, it was reported that 240,720 individuals lost their lives due to heart attacks in the year 2020. Smoking, inadequate physical activity, excessive body weight, elevated cholesterol levels, high blood pressure, and an unhealthy diet leading to elevated blood sugar are all factors that contribute to the risk of experiencing a myocardial infarction [2]. After a heart attack, the blockage in blood vessels stops oxygen and nutrients from reaching the heart muscle downstream. This causes damage to the heart, leading to a series of events like cell death, inflammation, and changes in the heart's structure, resulting in scars, stiffness, and altered function. People who've had a heart attack often face serious heart complications later on [3].

To respond to MI, the left ventricular (LV) of the heart will change its structural and functional behavior i.e. LV size, shape, and function called the remodeling of LV [4]. Because of the limited availability of experimental data and the biological complexity of LV remodeling, the understanding of the MI mechanism is a very complex process. By representing interaction of the factors such as blood flow, tissue oxygenation and cellular response through mathematical models can provide effective and valuable predictive capabilities. Previous studies [4][5][6], have explored the remodeling of left ventricular by using mathematical modeling. The approach employed in these articles involves using the numerical Runge-Kutta method with computer assistance to obtain the numerical solutions and collect various pieces of information. These investigations include exploring the important roles played by cytokines in the development of macrophages and other cells. Additionally, Zeigler et al. [7], has been investigated a mathematical model for fibrosis, using an ordinary differential equations (ODEs) framework to predict the behavior of collagen formation, breakdown, and aggregation. By using different assumptions, there exist some other papers [8][9], that include mathematical models to investigate the behavior of the heart after myocardial infarction, but all have some specific limitations. Agent-based models [10][11] have been employed to study tissue fibrosis, while biomechanical models [12][13] are also present in the literature. However, there is a scarcity of studies focusing on ODE models. Dealing with changes in the heart after myocardial infarction is always a challenge.

Various disciplines, including health, biology, physics, chemistry, civil and mechanical engineering, and economics, extensively use mathematical models [14][15][16][17][18][19]. In particular, there is a notable emphasis on combining these models with deep learning techniques, especially focusing on multilayer neural networks. In 2018, Side et al. [20] emphasized the crucial role of mathematical science in preventing the spread of illnesses. In addressing the spread of viruses, a mathematical model can be implemented, as in 2021 Umar et al. [21] highlighted the significant role of mathematics in exploring disease outbreaks, spread, and predictive patterns, particularly in the field of epidemiology. To obtain numerical outcomes for these models, stochastic solvers based on artificial neural networks along with optimization techniques are employed. Sabir et al. [22] present applications of artificial neural networks with the combination of Levenberg-Marquardt backpropagation for COVID-19 in 2022. In 2023 Haider et al. [23] proposed a system of ODEs for the study of hepatitis B virus (HBV) through deep learning techniques i.e. artificial neural network with the combination of Levenberg-Marquardt backpropagation. In this paper, we apply a deep learning methodology, specifically leveraging a widely recognized approach known as artificial neural networks, to infarction. In the diagram, cells are depicted by boxes with black color, while green boxes contain cytokines and specific proteins. Two types of arrows are used: black arrows signify the physical transfer of cells between groups, for instance, the transition of M_1 to M_2 macrophages and vice versa. On the other hand, blue arrows denote interactions between distinct cell populations, like the release of cytokines by macrophages. The dotted black line represents the consumption rate of M_d by M_1.

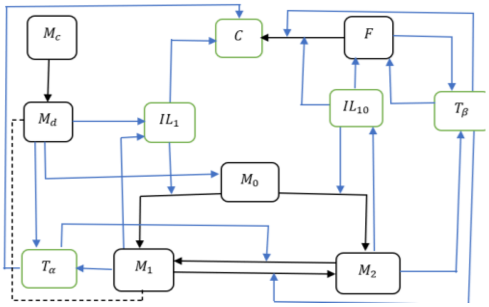


Figure 1: Diagram illustrating the cellular and molecular dynamics following a myocardial investigation of the phenomenon of myocardial infarction.

In the diagram, cells are depicted by boxes with black color, while green boxes contain cytokines and specific proteins. Two types of arrows are used: black arrows signify the physical transfer of cells between groups, for instance, the transition of M_1 to M_2 macrophages and vice versa. On the other hand, blue arrows denote interactions between distinct cell populations, like the release of cytokines by macrophages. The dotted black line represents the consumption rate of M_d by M_1.

The goal of this study is to introduce numerical simulations of the remodeling of MI through a nonlinear system of ODEs, including different compartments of the post-MI phenomenon. The solutions of the system are obtained using artificial neural networks methodology supported by an optimization technique called Levenberg-Marquardt backpropagation. Furthermore, an analysis of various components of ANNs-LMB is conducted to assess the proposed methodology's efficacy in achieving high accuracy and optimal performance. This method is suggested as an artificial intelligence-based approach for solving complex types of ODE systems with known initial conditions [24][25][26]. Some salient geographies of the designed study are given as follows:

- The MI mathematical model presented in this study is a modification of the mathematical model proposed by Lafci et al. [3]. We add two more cytokines: Transforming Growth Factor beta (T_β), and Tumor Necrosis Factor alpha (T_α), in the nonlinear system of ODEs proposed by Lafci et al. [3]. TGF-β is involved in the regulation of cell growth, differentiation, apoptosis, immune responses, and other cellular functions. It acts as a signaling molecule in various tissues and cell types, influencing both physiological and pathological processes [27], and is produced by alternatively activated macrophages (M_2) and fibroblasts (F). TNF-α is a cytokine involved in inflammation and immune system regulation. TNF-alpha is produced mainly by activated macrophages (M_1) and damaged cardiomyocytes (M_d) and can induce fever, inflammation, and cell death in certain tissues [28].
- Detailed descriptions of all compartments of the MI model are provided.
- The AI-based solutions of the model are performed by using a deep learning technique, ANNs-LMB in MATLAB.

Mathematical Model:

By incorporating T_β and T_α, and introducing some modifications to the mathematical model proposed in [3], our enhanced model is formulated as:

$$\frac{dM_c}{dt} = -k_1 M_c \qquad (1)$$

$$\frac{dM_d}{dt} = k_1 M_c - k_2 M_1 M_d - \mu_1 M_d \qquad (2)$$

$$\frac{dIL_1}{dt} = k_3M_d + k_4M_1\frac{c_1}{c_1+IL_{10}} - d_{IL_1}IL_1 \quad (3)$$

$$\frac{dIL_{10}}{dt} = k_5M_2\frac{c_2}{c_2+IL_{10}} - d_{IL_{10}}IL_{10} \quad (4)$$

$$\frac{dM_0}{dt} = k_6M_d - k_7M_0\frac{IL_1}{IL_1+c_{IL_1}} - k_8M_0\frac{IL_{10}}{IL_{10}+c_{IL_{10}}} - \mu M_0 \quad (5)$$

$$\frac{dM_1}{dt} = k_7M_0\frac{IL_1}{IL_1+c_{IL_1}} + \tau_1M_2\frac{T_\alpha}{T_\alpha+c_{T_\alpha}} - k_9M_1\frac{T_\beta}{T_\beta+c_{T_\beta}} - \mu M_1 \quad (6)$$

$$\frac{dM_2}{dt} = k_8M_0\frac{IL_{10}}{IL_{10}+c_{IL_{10}}} + k_9M_1\frac{T_\beta}{T_\beta+c_{T_\beta}} - \tau_1M_2\frac{T_\alpha}{T_\alpha+c_{T_\alpha}} - \mu M_2. \quad (7)$$

$$\frac{dC}{dt} = k_{10}F\frac{IL_{10}}{IL_{10}+c_3} + \alpha_1F\frac{T_\beta}{T_\beta+\beta_1} - k_{11}C\frac{IL_1}{IL_1+c_4} - \tau_2C\frac{T_\alpha}{T_\alpha+c_4} - d_cC \quad (8)$$

$$\frac{dF}{dt} = k_{12}F\frac{IL_{10}}{IL_{10}+c_5} + \beta_2F\frac{T_\beta}{T_\beta+\beta_1} - d_FF \quad (9)$$

$$\frac{dT_\beta}{dt} = \alpha_2F + \alpha_3M_2 - d_{T_\beta}T_\beta \quad (10)$$

$$\frac{dT_\alpha}{dt} = (\tau_3M_1 + \tau_4M_d)\frac{c_6}{c_6+T_\beta} - d_{T_\alpha}T_\alpha \quad (11)$$

The model's parameters, along with their descriptions, values, and units are listed in Table 1. This mathematical model captures the cellular and molecular dynamics associated with MI. It is derived from the depicted interactions in Figure 1, which serves as a flow diagram illustrating the system dynamics after MI, specifically focusing on scenarios of post-MI without any medical interventions.

Equation 1 shows how the number of heart muscle cells (M_c) changes over time. Equation 2 explains how the number of damaged heart muscle cells (M_d) changes over time. It goes up as healthy cells M_c damage and decreases at a rate of k_2 and μ_1. Equation 3 illustrates how the concentration of interleukin 1 cytokines (IL_1) changes over time. These cytokines are released by both damaged heart muscle cells and a specific type of immune cells, classically activated macrophages (M_1). The impact of the inhibition by interleukin 10 cytokines (IL_{10}) is modeled as a decreasing function, where c_1 signifies the strength of inhibition. Equation 4 outlines the temporal evolution of IL_{10}. These cytokines are released by specific types of immune cells, alternatively activated macrophages (M_2). A decreasing function is used to depict the inhibition of IL_{10} by IL_1. Equation 5 explains how the quantity of monocytes (M_0) changes over time. It rises because of M_d and declines due to two factors: the differentiation of M_0 into M_1 and M_2, and a constant emigration rate. The transition of M_0 into M_1 is stimulated by interleukin 1, while the transition into M_2 is promoted by interleukin 10 cytokines. Equation 6 delineates how the density of M_1 changes over time. It increases when M_0 differentiates into M_1 and M_2 transferred into M_1 because of T_α, and decreases when M_1 transfer to M_2 by stimulation of T_β and emigration. Equation 7 portrays the temporal evolution of the density of M_2. It increases when M_0 activates M_2 and when M_1 shifts to M_2, and it decreases due to emigration. Equation 8 shows the variation in the density of collagen (C) over time. It increases as fibroblasts (F) produce collagen in response to stimulation by IL_{10} and T_β. On the other hand, it decreases due to degradation caused by the presence of IL_1, T_α and a constant decay rate represented by d_c. Equation 9 characterizes how the density of fibroblasts changes over time. It increases through stimulation by IL_{10} and T_β but decreases due to death or emigration, represented by the rate d_F. Equation 10 details the rate of change of transforming growth factor-β over time. It is secreted by both F and M_2. Equation 11 represents the change of T_α over time. It produces by M_1 and M_d with constant rates τ_3 and τ_4 respectively. A decreasing function is used to represent the inhibition of T_α by T_β. In this equation d_{T_α} shows the decay rate of T_α by considering its half-life time.

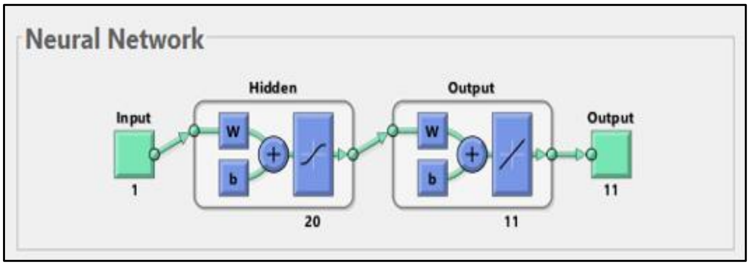


Figure 2: Designed A NNs-LMB

Table 1: MODEL'S PARAMETERS

Parameters	Description:	Values
k_1	Death rate of M_c	0.3
k_2	Rate at which M_d are consumed by M_1	0.003
k_3	Rate of Secretion of IL_1 by M_d	0.0004
k_4	Rate of Secretion of IL_1 by M_1	0.0005
k_5	Rate of Secretion of IL_{10} by M_2	0.0005
k_6	Rate of recruitment of M_0 based on M_d	0.4
k_7	Activation rate of IL_1 to activate M_1	0.7
k_8	Rate of activation of IL_{10} to activate M_2	0.3
k_9	Rate of transition from state M_1 to M_2	0.075
k_{10}	C production rate by F	26×10^5
k_{11}	Degradation rate of C by IL_1	0.0003
k_{12}	Fibroblasts growth rate	0.25
c_1	Effectiveness of IL_{10} inhibition on IL_1	2.5
c_2	Effectiveness of IL_1 inhibition on IL_{10}	10
c_3	Effectiveness of IL_{10} inhibition on F	5
c_4	Effectiveness of IL_1 and T_α on C	10
c_5	Impact of promoting of IL_{10} on F	2.5
c_6	Effectiveness of T_β inhibition on T_α	0.0007
τ_1	Transition rate of M_2 to M_1 because of T_α	0.7
τ_2	Degradation rate of C by T_α	0.0003
τ_3	Rate at which M_1 produces T_α	0.0007
τ_4	Rate at which M_d produces T_α	0.000005
α_1	Stimulation rate of transition of F to C by T_β	10
α_2	Secretion rate of T_β by F	0.0167
α_3	Rate at which M_2 produces T_β	0.0144
β_1	Effectiveness of T_β promotion on F	0.00316
β_2	Stimulation rate of T_β on F	0.03
c_{IL_1}	Impact of promoting of IL_1 on M_1	10
$c_{IL_{10}}$	Impact of promoting of IL_{10} on M_2	5
c_{T_β}	Effectiveness of T_β promotion on M_2	0.00316
c_{T_α}	Impact of promoting of T_α on M_1	10
d_{IL_1}	Decay rate of IL_1 considering its half-life time	0.2
$d_{IL_{10}}$	Decay rate of IL_{10} considering its half-life time	0.2

d_C	The decay rate of C by some enzymes	0.002
d_F	Emigration rate of F	0.02
d_{T_β}	Decay rate of T_β considering its half-life time	4.06
d_{T_α}	Decay rate of T_α considering its half-life time	0.5
μ	M_0, M_1 and M_2 emigration rate	0.2
μ_1	Removal rate of M_d	0.002

Methodology:

The two-step approach to the stochastic numerical procedure for the MI mathematical model is used. The first step involves offering comprehensive and detailed explanations of the computational stochastic numerical procedure, which is centered on ANNs-LMB. The second step encompasses the implementation procedures that bolster the stochastic numerical computation for the MI nonlinear mathematical model. The implementation of ANNs-LMB is employed to analyze and utilize the computational stochastic numerical outcomes of the mathematical model for myocardial infarction. We use MATLAB for the implementation of ANNs-LMB to obtain the results.

The "MATLAB" implementation follows a specific structure depicted in Figure 2, comprising a single input layer, hidden layers, and output layers. The configuration involves 20 hidden neurons, n-fold cross-validation, a log-sigmoid activation function, 20000 epochs, and the Levenberg-Marquardt optimization algorithm. It is important to highlight that the label data for input and targets are obtained from the standard numerical solution i.e. MATLAB solver command ode45.

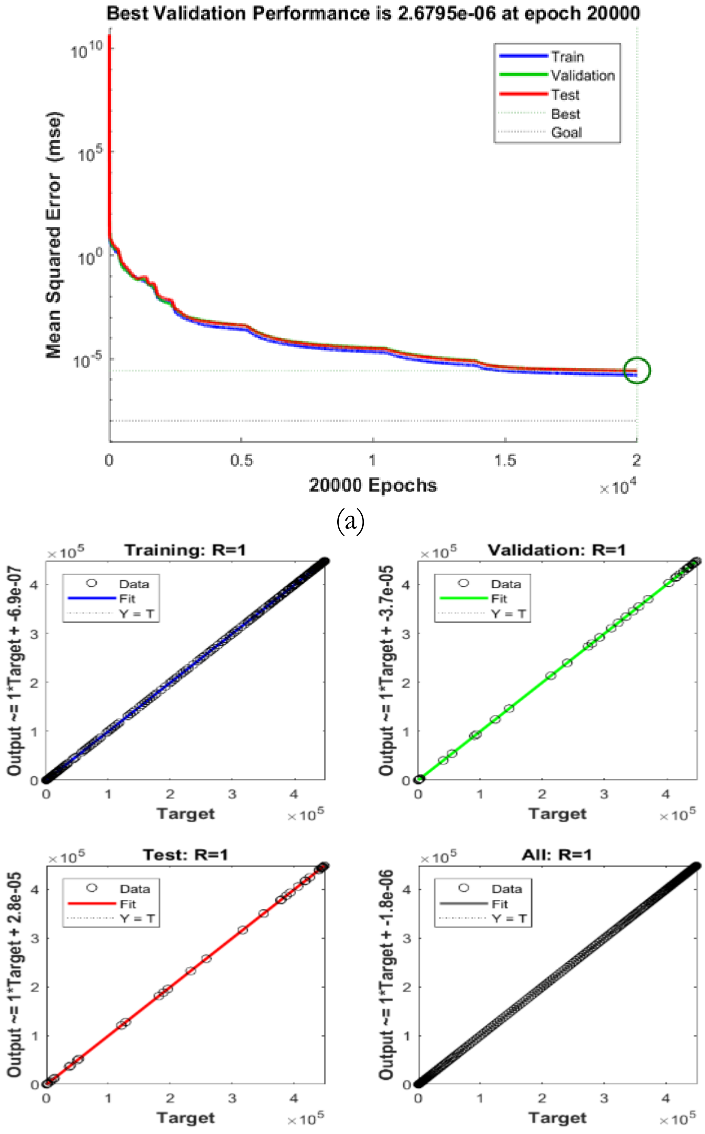
Table 2: Model's Initial Conditions

Variables	Values	Units
$M_c(0)$	400	cells/mL
$M_d(0)$	0	cells/mL
$IL_1(0)$	0.00001	pg/mL
$IL_{10}(0)$	0.000001	pg/mL
$M_0(0)$	0.02	cells/mL
$M_1(0)$	0	cells/mL
$M_2(0)$	0	cells/mL
$C(0)$	839.5	pg/mL
$F(0)$	1	cells/mL
$T_\beta(0)$	0.054	pg/mL
$T_\alpha(0)$	0.00001	pg/mL

Numerical Simulations:

The parameters used in the numerical simulations are presented in Table 1, providing descriptions and values for each parameter. The initial values of all compartments used in the model are illustrated in Table 2. Numerical outcomes for the nonlinear dynamical model of myocardial infarction within the input range [0, 60] are obtained through the ANNs-LMB methodology. The results are generated by using MATLAB and depicted in Figures 3-6. The graphs illustrating calculated results for the MI mathematical model are presented in Figures 3-6. In particular,

In particular, Figure 3 offers insights into the performance, error histogram with 20 bins, and regressions of the applied methodology. Specifically, Figure 3(a) displays the calculated mean square error, measures for the best curves during training, validation, and testing with optimal performance achieved at epoch 20000, which is 2.6795×10^{-6}. These visual representations underscore the successful convergence and precision achieved by the used methodology. Moreover, in Figure 3(c), correlation measures are presented, highlighting the regression performances. The correlation performances, expressed as the coefficient of determination (R^2 values), predominantly approach 1, underscoring the precision in solving the model. These plots encompass training, validation, testing and collectively indicating the accuracy of the scheme. Finally, fitting curves are depicted in Figure 4 to show the comparison between training, validation, and testing of the used methodology.



(a)

(b)

Figure 3: The performance of the used methodology ANNs-LMB to solve the MI mathematical model is presented in (a). Error histogram, and regression measure through ANNs-LMB are shown in (b) and (c) respectively.

Figures 5 and 6 display comparison plots of the solutions obtained by using the ANNs-LMB methodology and the true solutions (reference dataset constructed through MATLAB solver ode45) for the nonlinear dynamical system associated with myocardial infarction. Figure 5(a) compares the ANNs-LMB solution with the exact solution of the cardiomyocytes. We can observe that the solution obtained by ANNs-LMB and the exact solution are almost overlapped. Similarly, comparisons of dead cardiomyocytes, monocytes, macrophages, and fibroblasts are presented in Figure 5(b)-5(f). The comparison of cytokines and proteins after myocardial infarction is presented in Figure 6. These plots reveal a nearly perfect overlap between the exact solutions and those obtained by ANNs-LMB, underscoring the precision and effectiveness of the designed ANNs-LMB in solving the nonlinear system of differential equations related to the phenomena of myocardial infarction.

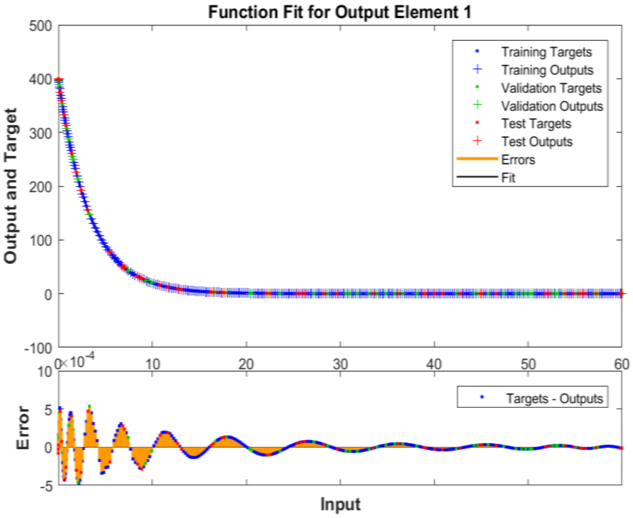
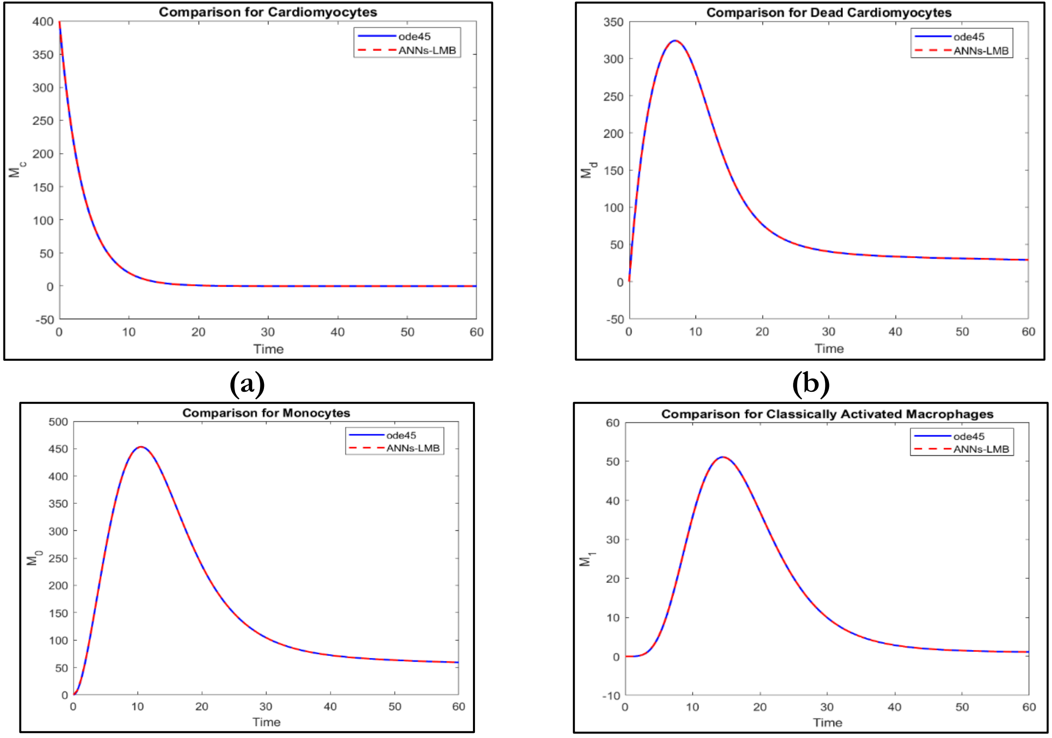


Figure 4: Function fit for output

Discussion:

A mathematical model is constructed to encompass critical interactions among cardiac cells, immune responses, and matrix proteins following myocardial infarction. Our model represents a notable improvement over earlier mathematical models, as highlighted in the work by Lafci et al. [3]. It stands out for considering significant biological factors, explicitly addressing the change of cardiomyocytes, the behavior of fibroblasts, and the deposition of fibrotic collagen in the context of post-myocardial infarction. Through numerical simulations, we tested the model's ability to describe events following a heart attack. The model's accurate predictability enhances our understanding of left ventricular remodeling after a heart attack. To derive solutions for the model, we used the deep learning strategy known as ANNs-LMB.

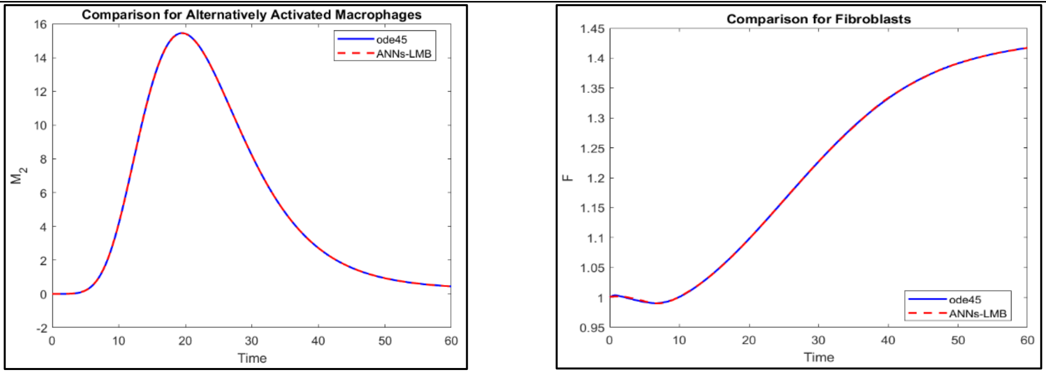


(a)

(b)

(c)

(d)

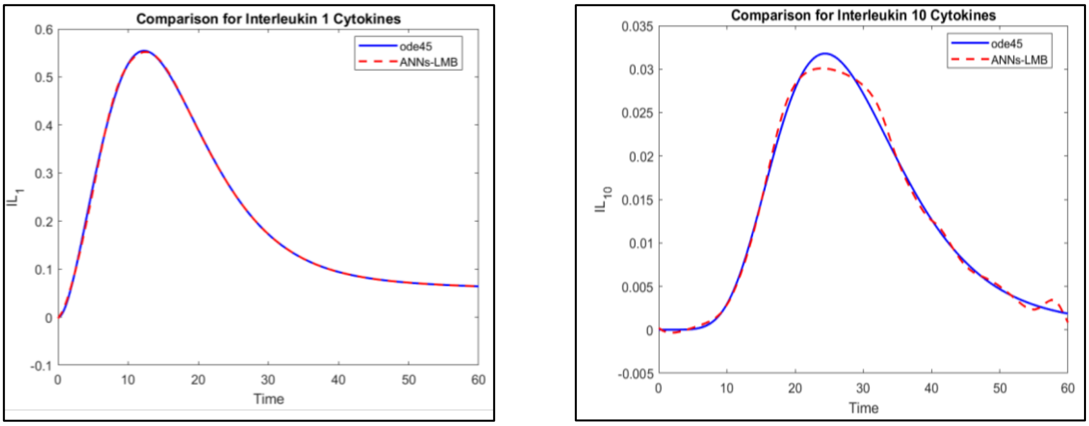


(e) (f)

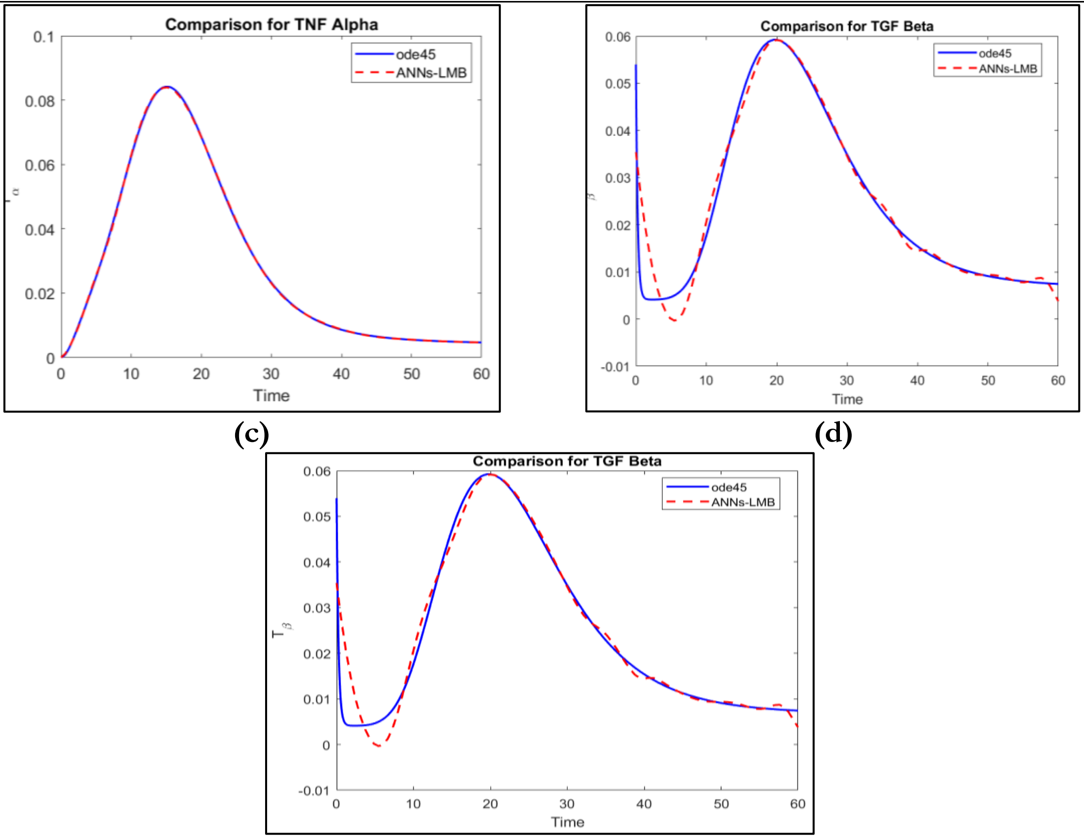
Figure 5: Dynamical behavior of cells after MI

Many papers on mathematical models lack visual representations, such as plots, that illustrate the evolution of various populations and species over time. While notable exceptions include the works of Jin et al. [4] and Wang et al. [5], Lafci et al.'s paper [3] stands out by providing insightful plots of evolution. To enhance understanding, we introduced changes to Lafci et al.'s model, particularly by adding two new compartments, TGF-β and TNF-α, and then solving the modified model by using a deep learning strategy, specifically ANNs-LMB.

Existing models related to MI, capture various aspects but often neglect key components. For example, Wang's model considers the monocytes and macrophage relationship, inhibitory and synthesizing bio factors, yet overlooks collagen, fibroblasts, and cardiomyocytes. Jin's model includes multiple elements but omits the behavior of cardiomyocytes. Zeigler et al. [7] primarily focus on fibroblast and collagen concentrations post-MI. Our study distinguishes itself by incorporating two additional compartments into the existing model proposed by Lafci et al., which plays a role in left ventricular remodeling after myocardial infarction. Furthermore, we obtained solutions for the myocardial infarction mathematical model by applying the deep learning strategy ANNs-LMB. Despite its thoroughness, a limitation stems from the lack of clinical data to derive certain unknown parameters. This model's limitation can be addressed in the future with more detailed data, allowing for a more precise representation of post-MI biological processes. Additionally, in this work, we used eleven compartments of the MI phenomena, a simplification compared to the real-world scenario. Future improvements may involve including more compartments to reduce this limitation.



(a) (b)



(c) (d)

(e)

Figure 6: Dynamical behavior of cytokines and proteins after MI.

Conclusions:

The objective of the current study is to apply deep learning strategies for the investigation of myocardial infarction through mathematical modeling. Several parameters in the mathematical model proposed by Lafci et al., for myocardial infarction are modified. The primary alteration involves the addition of two more compartments, namely Transforming Growth Factor beta and Tumor Necrosis Factor-alpha. Results of the nonlinear dynamical MI mathematical model are obtained by using a deep learning technique ANNs-LMB. The mathematical model is dependent on eleven dimensions.

Validation, testing, and training processes are conducted utilizing ANNs-LMB for the MI mathematical model. The numerical solutions derived from the model are compared with a reference dataset constructed through MATLAB. The outcomes demonstrate a notable overlapping with the reference dataset, underscoring the accuracy of the used methodology. Additionally, the results are further validated through the reduction of MSE. To evaluate the precision, reliability, and efficiency of the approach, various analyses, including MSE, error histograms, and regressions are used in this study.

References:

[1] T. I. Siddiqui, A. K. K. S., and D. K. Dikshit, “Platelets and Atherothrombosis: Causes, Targets and Treatments for Thrombosis,” Curr. Med. Chem., vol. 20, no. 22, pp. 2779–2797, Jun. 2013, doi: 10.2174/09298673113202200004.

[2] R. Hajar, “Risk Factors for Coronary Artery Disease: Historical Perspectives,” Heart Views, vol. 18, no. 3, p. 109, 2017, doi: 10.4103/HEARTVIEWS.HEARTVIEWS_106_17.

[3] M. Lafci Büyükkahraman, G. K. Sabine, H. V. Kojouharov, B. M. Chen-Charpentier, S. R. McMahan, and J. Liao, “Using models to advance medicine: mathematical modeling

of post-myocardial infarction left ventricular remodeling," Comput. Methods Biomech. Biomed. Engin., vol. 25, no. 3, pp. 298–307, 2022, doi: 10.1080/10255842.2021.1953487.

[4] Y. F. Jin, H. C. Han, J. Berger, Q. Dai, and M. L. Lindsey, "Combining experimental and mathematical modeling to reveal mechanisms of macrophage-dependent left ventricular remodeling," BMC Syst. Biol., vol. 5, no. 1, pp. 1–14, May 2011, doi: 10.1186/1752-0509-5-60/FIGURES/6.

[5] Y. Wang et al., "Mathematical modeling and stability analysis of macrophage activation in left ventricular remodeling post-myocardial infarction.," BMC Genomics, vol. 13 Suppl 6, no. 6, pp. 1–8, Oct. 2012, doi: 10.1186/1471-2164-13-S6-S21/FIGURES/4.

[6] U. Pagalay, L. Handayani, and A. Azzam, "Dynamics of Macrofages and Cytokines after Myocardial Infarction," Jun. 2019, doi: 10.4108/EAI.2-5-2019.2284674.

[7] A. C. Zeigler, A. R. Nelson, A. S. Chandrabhatla, O. Brazhkina, J. W. Holmes, and J. J. Saucerman, "Computational Model Predicts Paracrine and Intracellular Drivers of Fibroblast Phenotype After Myocardial Infarction," bioRxiv, p. 840017, Nov. 2019, doi: 10.1101/840017.

[8] L. C. Lee, G. S. Kassab, and J. M. Guccione, "Mathematical modeling of cardiac growth and remodeling," Wiley Interdiscip. Rev. Syst. Biol. Med., vol. 8, no. 3, pp. 211–226, May 2016, doi: 10.1002/WSBM.1330.

[9] N. Moise and A. Friedman, "A mathematical model of immunomodulatory treatment in myocardial infarction," J. Theor. Biol., vol. 544, p. 111122, Jul. 2022, doi: 10.1016/J.JTBI.2022.111122.

[10] A. D. Rouillard and J. W. Holmes, "Mechanical regulation of fibroblast migration and collagen remodelling in healing myocardial infarcts," J. Physiol., vol. 590, no. 18, pp. 4585–4602, Sep. 2012, doi: 10.1113/JPHYSIOL.2012.229484.

[11] S. M. Rikard et al., "Multiscale Coupling of an Agent-Based Model of Tissue Fibrosis and a Logic-Based Model of Intracellular Signaling," Front. Physiol., vol. 10, p. 479813, Dec. 2019, doi: 10.3389/FPHYS.2019.01481/BIBTEX.

[12] P. Sáez and E. Kuhl, "Computational modeling of acute myocardial infarction," Comput. Methods Biomech. Biomed. Engin., vol. 19, no. 10, pp. 1107–1115, Jul. 2016, doi: 10.1080/10255842.2015.1105965.

[13] E. Cutrì, A. Meoli, G. Dubini, F. Migliavacca, T. Y. Hsia, and G. Pennati, "Patient-specific biomechanical model of hypoplastic left heart to predict post-operative cardio-circulatory behaviour," Med. Eng. Phys., vol. 47, pp. 85–92, Sep. 2017, doi: 10.1016/J.MEDENGPHY.2017.06.024.

[14] M. Lo Schiavo, B. Prinari, J. A. Gronski, and A. V. Serio, "An artificial neural network approach for modeling the ward atmosphere in a medical unit," Math. Comput. Simul., vol. 116, pp. 44–58, Oct. 2015, doi: 10.1016/J.MATCOM.2015.04.006.

[15] J. L. G. Guirao, "On the stochastic observation for the nonlinear system of the emigration and migration effects via artificial neural networks," Int. J. Math. Comput. Eng., vol. 1, no. 2, pp. 177–186, Dec. 2023, doi: 10.2478/IJMCE-2023-0014.

[16] M. Shoaib, R. Kainat, M. Ijaz Khan, B. C. Prasanna Kumara, R. Naveen Kumar, and M. A. Zahoor Raja, "Darcy-Forchheimer entropy based hybrid nanofluid flow over a stretchable surface: intelligent computing approach," Waves in Random and Complex Media, Sep. 2022, doi: 10.1080/17455030.2022.2122627.

[17] Z. Sabir, R. Sadat, M. R. Ali, S. Ben Said, and M. Azhar, "A numerical performance of the novel fractional water pollution model through the Levenberg-Marquardt backpropagation method," Arab. J. Chem., vol. 16, no. 2, p. 104493, Feb. 2023, doi: 10.1016/J.ARABJC.2022.104493.

[18] T. Botmart, Z. Sabir, M. A. Z. Raja, W. weera, R. Sadat, and M. R. Ali, "Stochastic procedures to solve the nonlinear mass and heat transfer model of Williamson nanofluid past over a stretching sheet," Ann. Nucl. Energy, vol. 181, p. 109564, Feb. 2023, doi: 10.1016/J.ANUCENE.2022.109564.

[19] W. Weera et al., "Fractional Order Environmental and Economic Model Investigations Using Artificial Neural Network," Comput. Mater. Contin., vol. 74, no. 1, pp. 1735–1748, Sep. 2022, doi: 10.32604/CMC.2023.032950.

[20] "Stability Analysis Susceptible, Exposed, Infected, Recovered (SEIR) Model for Spread Model for Spread of Dengue Fever in Medan", [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/954/1/012018

[21] M. Umar et al., "Numerical Investigations through ANNs for Solving COVID-19 Model," Int. J. Environ. Res. Public Heal. 2021, Vol. 18, Page 12192, vol. 18, no. 22, p. 12192, Nov. 2021, doi: 10.3390/IJERPH182212192.

[22] Z. Sabir, M. A. Z. Raja, S. E. Alhazmi, M. Gupta, A. Arbi, and I. A. Baba, "Applications of artificial neural network to solve the nonlinear COVID-19 mathematical model based on the dynamics of SIQ," J. Taibah Univ. Sci., vol. 16, no. 1, pp. 874–884, Dec. 2022, doi: 10.1080/16583655.2022.2119734.

[23] Q. Haider, A. Hassan, and S. M. Eldin, "Artificial neural network scheme to solve the hepatitis B virus model," Front. Appl. Math. Stat., vol. 9, p. 1072447, Mar. 2023, doi: 10.3389/FAMS.2023.1072447/BIBTEX.

[24] Z. Sabir et al., "Artificial neural network scheme to solve the nonlinear influenza disease model," Biomed. Signal Process. Control, vol. 75, p. 103594, May 2022, doi: 10.1016/J.BSPC.2022.103594.

[25] S. Mall and S. Chakraverty, "Comparison of Artificial Neural Network Architecture in Solving Ordinary Differential Equations," Adv. Artif. Neural Syst., vol. 2013, pp. 1–12, Dec. 2013, doi: 10.1155/2013/181895.

[26] Y. Wen, T. Chaolu, and X. Wang, "Solving the initial value problem of ordinary differential equations by Lie group based neural network method," PLoS One, vol. 17, no. 4, p. e0265992, Apr. 2022, doi: 10.1371/JOURNAL.PONE.0265992.

[27] M. Bujak and N. G. Frangogiannis, "The role of TGF-β signaling in myocardial infarction and cardiac remodeling," Cardiovasc. Res., vol. 74, no. 2, pp. 184–195, May 2007, doi: 10.1016/J.CARDIORES.2006.10.002/2/74-2-184-FIG4.GIF.

[28] D. I. Jang et al., "The Role of Tumor Necrosis Factor Alpha (TNF-α) in Autoimmune Disease and Current TNF-α Inhibitors in Therapeutics," Int. J. Mol. Sci. 2021, Vol. 22, Page 2719, vol. 22, no. 5, p. 2719, Mar. 2021, doi: 10.3390/IJMS22052719.

Honey Adulteration Detection through Hyperspectral Imaging and Machine Learning

Hazrat Usman[1], Anna Amjad[1], Maryam Mahsal Khan[1], Sumayyea Salahuddin[2]

[1]Department of Computer Science (CECOS University of IT and Emerging Sciences, Peshawar, Pakistan).

[2]Department of Computer Systems Engineering (University of Engineering and Technology, Peshawar, Pakistan).

*__Correspondence__: Maryam Mahsal Khan, maryam.khan@cecos.edu.pk,

Citation | Usman. H, Amjad. A, Khan. M. M, Salahuddin. S, "Honey Adulteration Detection through Hyperspectral Imaging and Machine Learning", IJIST, Special Issue pp 30-36, May 2024

Received| May 02, 2024, **Revised|** May 09, 2024, **Accepted|** May 13, 2024, **Published|** May 20, 2024.

Introduction/Importance of Study:

The purity and authenticity of honey are paramount for ensuring consumer trust and maintaining the integrity of the honey industry. There is a pressing need for advanced and efficient detection methods to increase the prevalence of honey adulteration.

Novelty statement:

Our research provides a solution to the challenge of predicting the change in adulterated honey properties through hyperspectral imaging and advanced machine learning algorithms, filling a critical gap in existing methodologies.

Material and Method:

A publicly available dataset with spectral features, extracted through hyperspectral imaging, across different classes of honey and adulteration levels has been examined and various machine learning models were developed to identify honey adulteration concentration and type of honey. The dataset was balanced and a five-fold cross-validation technique was used to train the machine learning models.

Result and Discussion:

Random forest was found to perform better in three identified scenarios i.e. (a) type of honey (b) adulteration level (c) both (a, b); with a maximum average accuracy of 99.69% performing better than the one reported in the literature (95%). For both single-output and multiple-output ML models, the trend in feature importance was observed. The single model identifying the class of honey utilized low and mid-frequency spectra while the multi-model used mid-frequency spectrum only.

Concluding Remarks:

The proposed approach aims to provide an accurate and cost-effective solution to address the challenges associated with honey adulteration, contributing to the enhancement of honey quality assessment and consumer confidence.

Keywords: Honey Fraud Detection; Hyperspectral imaging; Machine Learning; Random Forest; XG Boost.

Introduction:

Honey has been since ancient times valued for its unique flavors and nutritional qualities. Honey production is a profitable market. Due to the increase in demand for honey because of population growth, honey producers are tempted to commit fraud. Pure honey is diluted with common adulterants, like sugar syrups and other sweets. The honey contamination results not only in quality compromise but also severe health issues. Conventional approaches to honey adulteration are unable to identify the various adulterants in the honey. This emphasizes the need to develop and design sophisticated techniques that guarantee quick and accurate findings. In this setting, hyperspectral imaging becomes an effective tool that provides a non-destructive way to obtain detailed spectral information from honey samples over a wide range of wavelengths. With the use of this technology, complex chemical fingerprints may be extracted, making it possible to distinguish between genuine and fake honey. However, advanced analytical tools like machine learning are needed to fully utilize hyperspectral data. Research in this area is being done by the current study.

This work outlines a methodical approach that includes gathering hyperspectral data from honey samples, optimizing data quality through preprocessing, and utilizing multiple machine learning algorithms to achieve robust detection. One of the outcomes of the research is a flexible and all-encompassing strategy that can protect the integrity of the honey business. The goal of the suggested method is to combine machine learning and hyperspectral imaging to transform the detection of adulterated honey. By combining these technologies, this study aims to provide a smart and effective solution that will guarantee the production and consumption of unadulterated honey for years to come. The current study aims to develop an intelligent system that can detect adulteration in honey using machine learning algorithms across three scenarios.

- Identify the class or type of honey.
- Identify the level of adulteration of sugar syrup, in the honey class.
- Identify both the type of honey and the level of adulteration in one go.

Literature Review:

Existing Quality Assurance Techniques:

Most honey's plant sources are categorized chemically; however, more conventional methods still include honey specialists tasting and smelling the honey. Pollen analysis and assays for specific components that make up different types of honey are among the chemical measures [1]. Numerous techniques have been put up to identify the adulteration of honey with sugar. High-pressure liquid chromatography (HPLC) [2], deuterium nuclear magnetic resonance (NMR) spectroscopy [3][4], mass spectroscopy of the carbon isotope ratio [5][6], and FTIR spectroscopy [7][8] can be used to detect adulterated honey.

The detection of honey adulteration with cane sugar using Fourier transform infrared (FTIR) spectroscopy has been the subject of several studies [9]. These studies assessed adulteration in cane sugar concentrations ranging from 0.5 to 25%. In [10], a single variety of honey was utilized to estimate the sugar concentration with an accuracy of 93.75% utilizing statistical techniques and artificial neural networks. When three different varieties of honey were used to classify adulteration, the classification accuracy was less than 80% [9]. These studies demonstrate that it is feasible to anticipate adulteration in honey by combining spectroscopic and machine learning approaches; yet, the capacity to forecast sugar content across a variety of honey varieties has to be enhanced.

By extending spectroscopy and enabling the use of spatial information in addition to spectral information, hyperspectral imaging is a potential method for ensuring the quality of food [11]. Instead of only capturing the spectrum at one spot on the item, spatial information enables the image to highlight certain flaws like bruising on fruit at a specific area [12]. Numerous food quality applications, such as those involving meat, fish, fruit, vegetables, and cereals, have made use of hyperspectral imaging [5].

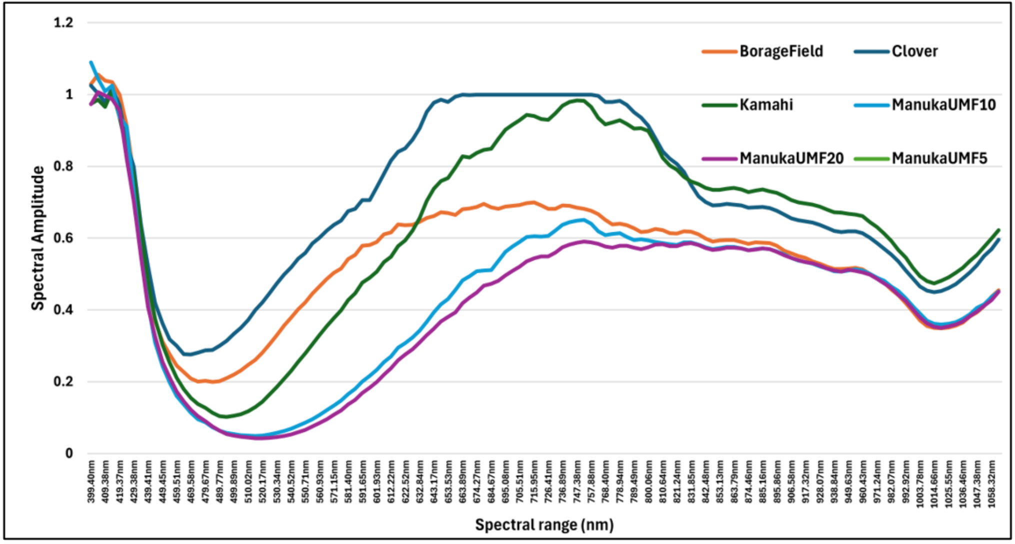


Figure 1. Hyperspectral response of 6 different types of honey classes with an adulteration concentration level of 50%.

A hyperspectral imaging technique has been developed to determine the botanical source of honey [13][14][15][16]. With 90% accuracy, the botanical ancestry of 21 different types of honey was predicted [16]. These techniques used a class embodiment autoencoder (CEAE) and support vector machines (SVM) to classify the data, which was obtained via a hyperspectral imaging system as detailed in [17]. In [13], 52 samples from five distinct types of honey were identified botanically with a 90% classification accuracy in a small data set.

Adulteration Dataset:

The dataset [18] comprises 12 unique honey products sourced from seven different brands and characterized by 11 botanical origin labels. Six independent samples were collected for each type of honey, with an equal distribution between Manuka honey, a high-quality honey variant from New Zealand, and other types of New Zealand honey. Throughout the dataset creation process, images of all honey varieties were captured at varying sugar concentrations (5, 10, 25, 50). The detailed composition of the dataset is presented in Table 1. Figure 1 shows the hyperspectral response of six different types of honey classes with a honey adulteration level of 50%. The drift in the spectrum indicates that the effect of adulteration on the spectral response of honey is different and non-linear, where AI would aid in interpreting the trends accordingly.

Material and Methods:

Table 1: The overall makeup of the adulterated honey data set from each brand and botanical origins label of honey. taken from [18]

Class	Adulteration Concentration					
	0%	5%	10%	25%	50%	Sum
Clover	150	150	300	300	300	1200
Multi Floral	150	150			150	450
ManukaUMF5		150	150	150	150	600
ManukaUMF15		150	150	150	150	600
ManukaUMF20		150	150	150	150	600
ManukaUMF10		150	150	150	125	575
Manuka Blend		150		150	150	450
Borage Field	150	150	150	150	150	750
Kamahi	150	150	150	150	150	750

Rewarewa	150	150	150			450
Manuka Blend	150	150	150	150	150	750
Manuka	150	300	300	300	300	1350

Figure 2 shows the methodology of the current study in which we have focused on the detection of the following three scenarios. Given a honey sample,

- Task A: To identify the class of honey.
- Task B: To identify the adulteration concentration level.
- Task C: To identify both the class and adulteration concentration level.

The representation of samples is not balanced in all these tasks. In order to reduce bias in the generated machine learning (ML) models, we have then employed balancing the dataset via a popular and common technique called Synthetic Minority Oversampling Technique (SMOTE) [19]. In this manner, accurate measurement of ML performance metrics can be recorded accordingly.

- TaskA represents 12 Honey classes (first column in Table 1). After balancing, each class represented 1350 samples.
- For TaskB, honey samples consist of adulteration concentration levels of 0%, 5%, 10%, 25%, and 50% in the dataset. After balancing the set a total of 1950 samples per class were generated.
- Finally, Task C is to identify honey type and adulteration concentration level while using one ML model. After balancing 300 samples per class were generated.

The dataset created in this manner for all three tasks is then forwarded to the ML model for creation and evaluation. A five-fold cross-validation was used to train and test the ML models. From Figure 3, given a dataset, three different representations of the dataset are generated that are then balanced by using SMOTE. Machine learning algorithms in particular Random Forest (RF), Support Vector Machine (SVM), and Extra Gradient boosting Trees (XG Boost) were used in generating the models. The performance of the algorithms across the different tasks is reported in Table II-IV.

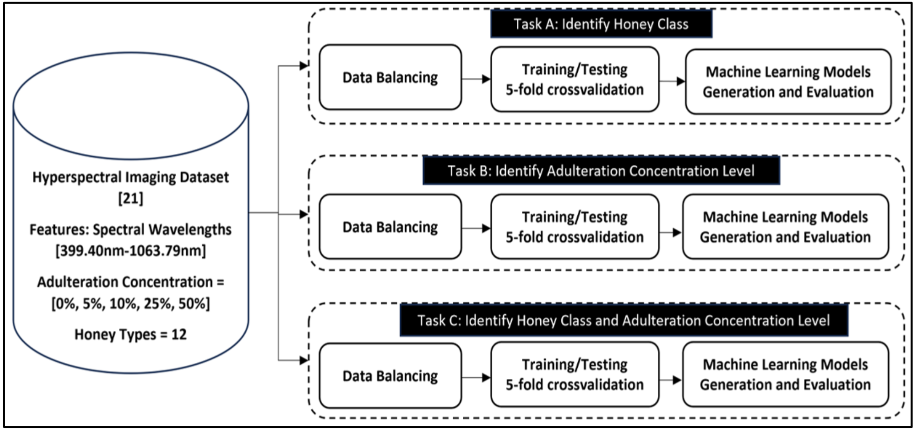

Hyperspectral Imaging Dataset [21]
Features: Spectral Wavelengths [399.40nm-1063.79nm]
Adulteration Concentration = [0%, 5%, 10%, 25%, 50%]
Honey Types = 12
Task A: Identify Honey Class
Data Balancing
Training/Testing 5-fold crossvalidation
Machine Learning Models Generation and Evaluation
Task B: Identify Adulteration Concentration Level
Data Balancing
Training/Testing 5-fold crossvalidation
Machine Learning Models Generation and Evaluation
Task C: Identify Honey Class and Adulteration Concentration Level
Data Balancing
Training/Testing 5-fold crossvalidation
Machine Learning Models Generation and Evaluation

Figure 2: Methodology and execution of various tasks in the current research study.

Result and Discussion:

Table 2-4 shows the performance of the ML models across three different tasks (Task A, Task B, and Task C) shown in Figure 2. In order to access the performance of the ML algorithm, various performance metrics are used. In the current study ML performance metrics namely accuracy, precision, recall, and f1-score are reported herewith.

In Table 2, for Task A, out of the three ML models, the RF model performed best with an accuracy of 99.89% while SVM scored an accuracy of 85.35%. In this task, each honey type whether pure or adulterated is categorized as one. The result indicates that the spectral parameters derived from hyperspectral imaging can identify the type of honey investigated.

Table 2: Performance of ML Models on task A, given a honey sample, identify the Type of Honey.

ML Model	Accuracy	Precision	Recall	F1 Score
RF	**0.99899**	0.999043	0.998991	0.999013
SVM	0.853535	0.864275	0.854573	0.851682
XG Boost	0.997306	0.997337	0.997351	0.997334

Table 3: Performance of ML Models on task B, given a honey sample, identify the concentration adulteration Level

ML Model	Accuracy	Precision	Recall	F1 Score
RF	0.996923	0.99691	0.996953	0.996926
SVM	0.522564	0.51252	0.526156	0.4714
XG Boost	0.995897	0.995878	0.995958	0.995908

While in Table 3, Task B, RF performed best with an accuracy of 99.69%. SVM was found to perform with an accuracy of 52.25%. In this task, the different types of honey were grouped based on their adulteration concentration level. The adulterant used in the dataset was sugar syrup. The task was particularly challenging as honey also contains natural sugar.

Table 4: Performance of ML Models on task C, given a Honey sample, identify both class and type of Honey.

ML Model	Accuracy	Precision	Recall	F1 Score
RF	0.998936	0.998981	0.998897	0.99893
SVM	0.818794	0.84086	0.821589	0.800515
XG Boost	0.992553	0.992895	0.992604	0.992638

In Table 4, Task C, RF performed with an accuracy of 99.92%. The identification of honey and adulteration levels via a single ML model was found to be better than two distinct models. This might be due to the reason honey types represent unique spectral signatures. Hence the change in signature on the spectral properties is reflected differently among different honey types.

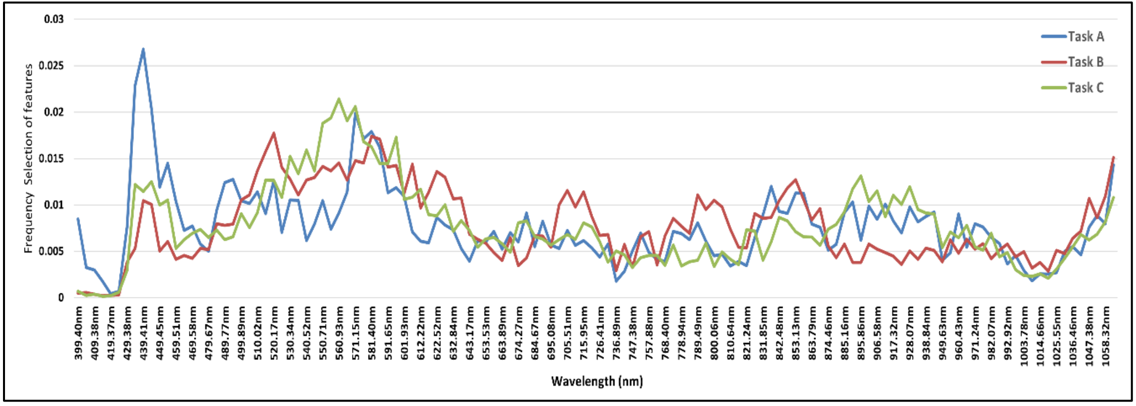


Figure 3: Feature significance graph of the RF models generated for Task A, Task B, and Task C respectively.

Figure 3 shows the feature significance graph obtained from the RF models of the three tasks A, B, and C. The feature importance shows that for Task A two identifiable peaks are obtained at a spectral range of 439.41nm and 566.04nm, for Task B around 520.17nm and 586nm while for Task C spectral range of 560nm and 596.79nm were found to be the important spectral features in these tasks. It's interesting to note that from Task A to Task C a shift in peak features is observed.

Table 5, shows the comparison of the ML models with the ones reported in literature [13] on the dataset. As the dataset was not balanced in the reported literature hence f1-score is low. Due to the balancing of the dataset, our study was able to improve on these metrics.

Table 5: Comparison of ML models with others reported in the literature

Task	Accuracy	Precision	Recall	F1 Score
Adulteration Concentration	0.996923	0.99691	0.996953	0.996926
Adulteration Concentration [18]	0.951	x	x	0.940

Conclusion:

The adulteration of honey has become a widespread practice aimed at increasing economic benefits, however, it has been shown to have detrimental effects on an individual's health. The current study explores the potential of machine learning algorithms in the accurate identification of honey adulteration on a recently publicly available dataset. SMOTE algorithm was used to balance the dataset. Random Forest, Support Vector Machine, and XGBoost algorithms were used to generate models whereby RF was found to perform better than the other two algorithms in the identification of the quality of honey. In comparison to the reported study (95%), our study produced an accuracy of 99.69% on the same dataset. This indicates the potential of ML algorithms in the accurate identification and quantification of honey adulteration.

Acknowledgment: We acknowledge the Department of ORIC at CECOS University of IT & Emerging Sciences for funding resources.

Author's Contribution: All authors have equally contributed to the paper.

Conflict of Interest: The authors have no conflicts of interest to declare.

Project Details: NA

References:

[1] A. Noviyanto, W. Abdullah, W. Yu, and Z. Salcic, "Research trends in optical spectrum for honey analysis," 2015 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2015, pp. 416–425, Feb. 2016, doi: 10.1109/APSIPA.2015.7415305.

[2] C. Cordella, I. Moussa, A. C. Martel, N. Sbirrazzouli, and L. Lizzani-Cuvelier, "Recent Developments in Food Characterization and Adulteration Detection: Technique-Oriented Perspectives," J. Agric. Food Chem., vol. 50, no. 7, pp. 1751–1764, Mar. 2002, doi: 10.1021/JF011096Z.

[3] P. Lindner, E. Bermann, and B. Gamarnik, "Characterization of Citrus Honey by Deuterium NMR," J. Agric. Food Chem., vol. 44, no. 1, pp. 139–140, 1996, doi: 10.1021/JF950359K.

[4] S. Giraudon, M. Danzart, and M. H. Merle, "Deuterium Nuclear Magnetic Resonance Spectroscopy and Stable Carbon Isotope Ratio Analysis/Mass Spectrometry of Certain Monofloral Honeys," J. AOAC Int., vol. 83, no. 6, pp. 1401–1409, Nov. 2000, doi: 10.1093/JAOAC/83.6.1401.

[5] S. T. Brookes, A. Barrie, and J. E. Davies, "A Rapid 13C/12C Test for Determination of Corn Syrups in Honey," J. AOAC Int., vol. 74, no. 4, pp. 627–629, Jul. 1991, doi: 10.1093/JAOAC/74.4.627.

[6] J. W. White, K. Winters, P. Martin, and A. Rossmann, "Stable Carbon Isotope Ratio Analysis of Honey: Validation of Internal Standard Procedure for Worldwide Application," J. AOAC Int., vol. 81, no. 3, pp. 610–619, May 1998, doi: 10.1093/JAOAC/81.3.610.

[7] S. Gok, M. Severcan, E. Goormaghtigh, I. Kandemir, and F. Severcan, "Differentiation of Anatolian honey samples from different botanical origins by ATR-FTIR spectroscopy using multivariate analysis," Food Chem., vol. 170, pp. 234–240, Mar. 2015, doi: 10.1016/J.FOODCHEM.2014.08.040.

[8] S. Sivakesava and J. Irudayaraj, "Detection of inverted beet sugar adulteration of honey by FTIR spectroscopy," J. Sci. Food Agric., vol. 81, no. 8, pp. 683–690, Jun. 2001, doi: 10.1002/JSFA.858.

[9] S. Sivakesave and J. Irudayaraj, "Prediction of Inverted Cane Sugar Adulteration of

Honey by Fourier Transform Infrared Spectroscopy," J. Food Sci., vol. 66, no. 7, pp. 972–978, Sep. 2001, doi: 10.1111/J.1365-2621.2001.TB08221.X.

[10] J. Irudayaraj, F. Xu, and J. Tewari, "Rapid Determination of Invert Cane Sugar Adulteration in Honey Using FTIR Spectroscopy and Multivariate Analysis," J. Food Sci., vol. 68, no. 6, pp. 2040–2045, Aug. 2003, doi: 10.1111/J.1365-2621.2003.TB07015.X.

[11] G. ElMasry and D. W. Sun, "Principles of Hyperspectral Imaging Technology," Hyperspectral Imaging Food Qual. Anal. Control, pp. 3–43, Jan. 2010, doi: 10.1016/B978-0-12-374753-2.10001-2.

[12] A. A. Gowen, C. P. O'Donnell, P. J. Cullen, G. Downey, and J. M. Frias, "Hyperspectral imaging – an emerging process analytical tool for food quality and safety control," Trends Food Sci. Technol., vol. 18, no. 12, pp. 590–598, Dec. 2007, doi: 10.1016/J.TIFS.2007.06.001.

[13] S. Minaei et al., "VIS/NIR imaging application for honey floral origin determination," Infrared Phys. Technol., vol. 86, pp. 218–225, Nov. 2017, doi: 10.1016/J.INFRARED.2017.09.001.

[14] A. Noviyanto and W. H. Abdulla, "Honey botanical origin classification using hyperspectral imaging and machine learning," J. Food Eng., vol. 265, p. 109684, Jan. 2020, doi: 10.1016/J.JFOODENG.2019.109684.

[15] A. Noviyanto and W. H. Abdulla, "Signifying the information carrying bands of hyperspectral imaging for honey botanical origin classification," J. Food Eng., vol. 292, p. 110281, Mar. 2021, doi: 10.1016/J.JFOODENG.2020.110281.

[16] T. Phillips and W. Abdulla, "Class Embodiment Autoencoder (CEAE) for classifying the botanical origins of honey," Int. Conf. Image Vis. Comput. New Zeal., vol. 2019-December, Dec. 2019, doi: 10.1109/IVCNZ48456.2019.8961004.

[17] A. Noviyanto and W. H. Abdulla, "Honey dataset standard using hyperspectral imaging for machine learning problems," 25th Eur. Signal Process. Conf. EUSIPCO 2017, vol. 2017-January, pp. 473–477, Oct. 2017, doi: 10.23919/EUSIPCO.2017.8081252.

[18] T. Phillips and W. Abdulla, "A new honey adulteration detection approach using hyperspectral imaging and machine learning," Eur. Food Res. Technol., vol. 249, no. 2, pp. 259–272, Feb. 2023, doi: 10.1007/S00217-022-04113-9/TABLES/6.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/JAIR.953.

Sherpa: Implementing a Hybrid Recommendation System for Next-Gen Tourist Experience

Inam Ullah Khan, Abdur Rahman, Muhammad Awais Khan, Dr. Madiha Sher, Dr. Yasir Saleem
Department of Computer Systems Engineering, University of Engineering and Technology Peshawar Pakistan

***Correspondence**: 20pwcse1862@uetpeshawar.edu.pk 20pwcse1878@uetpeshawar.edu.pk 20pwcse1871@uetpeshawar.edu.pk madiha@uetpeshawar.edu.pk yasirsaleem@uetpeshawar.edu.pk

Citation | Khan. I. U, Rahman. A, Khan. M. A, Sher. M, Saleem. Y, “Sherpa: Implementing a Hybrid Recommendation System for Next-Gen Tourist Experience”, IJIST, Special Issue. 2 pp 37-42, May 2024

Received | May 02, 2024, **Revised |** May 10, 2024, **Accepted |** May 15, 2024, **Published |** May 20, 2024.

In the digital era, Sherpa revolutionizes personalized tourism with an AI-driven recommendation system, fostering meaningful connections between travelers and local guides. This study explores Sherpa's integration of collaborative and content-based filtering—specifically, singular value decomposition (SVD) and cosine similarity—to tailor travel experiences uniquely. Our methodology includes a detailed examination of Sherpa's algorithm and its implementation within a cross-platform, MERN Stack-powered backend. We assess the system's efficacy in aligning recommendations with individual user preferences, based on quantitative user feedback and engagement metrics. Initial results demonstrate a significant improvement in personalized experience satisfaction. The paper concludes that Sherpa's innovative approach not only enhances the quality of travel recommendations but also sets a new standard for interactive and adaptive tourism platforms. Through continuous algorithmic refinement, Sherpa is positioned to lead a transformative shift in how travelers explore new destinations, offering not just journeys, but transformative experiences.

Keywords: Artificial Intelligence; Digital Tourism; Collaborative Filtering; Content-Based Filtering; Hybrid Recommender System; Singular Value Decomposition; Cosine Similarity Matrix.

Introduction:

Travel isn't just seeing sights anymore. People want deeper experiences when they go places. They don't just want planned schedules and surface interactions. Now, travelers aim to connect with locals. They hope to uncover hidden gems and make lasting memories. The old tourism ways don't satisfy these changing desires. Connectivity and curiosity drive travel today. Travelers yearn for meaningful encounters, not just observing from outside. They aspire to immerse themselves in the cultures they visit. Sherpa shows a change in travel. It pictures tourists acting as members of host towns, not just visitors. Sherpa aims to make guided tourism different. It connects tourists with local guides, creating a good partnership where exploration is a shared project.

Traditional tourism has limits - guidebooks and online sites offer insights but lack the personal touch and real-time flexibility that only locals provide. Sherpa recognizes this limitation. It aims to give visitors and guides more influence by fostering cross-cultural dialogue and offering customized memorable experiences. Sherpa's genesis lies in addressing tourism's shortcomings. It provides a venue where cultures connect, making each trip unique through personalization. Sherpa uses an AI matching system. This system matches tourists with guides. It looks at what tourists like. It also looks at guides' skills. The AI finds hidden patterns. It uses info from tourists and guides. The goal is great tourist trips. The system matches based on tourist wants.

The goal of offering suggestions suited to each person's particular tastes and interests remains a key issue for the travel industry, despite technological advancements. Sherpa tackles this challenge head-on with an innovative hybrid recommendation system. This approach blends content-based filtering using cosine similarity matrices with a seamless integration of collaborative filtering via singular value decomposition (SVD) models. By combining these methods, Sherpa provides travelers with immersive, personalized experiences that transcend the limitations of conventional tourism models. Simply, Sherpa isn't only a mobile app. It shows a new way to do guided tours. Sherpa means adventure, connecting, and real experiences. It will make travel better for visitors and locals. As Sherpa grows, it will keep changing how we travel. One personal tip at a time.

Related Work:

The impact of technology on guided tourism is significant. Early examples like Trip Mate, TripAdvisor, and Airbnb have developed services centered on personalized travel experiences. These platforms offer personalized suggestions, local experiences, and intelligent lodging options through AI and big data. However, they sometimes compromise traveler interests, as they cater to only a limited range of customer preferences.

On the other hand, Sherpa sets itself apart by using a hybrid recommender system that merges the advantages of collaborative filtering and content-based filtering. This combination enables Sherpa to offer personalized recommendations that cater to the distinct preferences of individual travelers. Lots of research has been done to improve the performance of basic Matrix Factorization used in Content-based Filtering. In [1], Zhang and colleagues introduced Weighted Non-negative Matrix Factorization (WNMF) as a method to enhance NMF (Non-negative Matrix Factorization). They utilized weights as an indicator matrix to represent the visibility of entries in the matrix R. In [2] Lee et al. gave the idea of Non-negative Matrix Factorization (NMF) to enforce non-negativity in U and V, which was proved to be useful in computer vision fields. In [3], Salakhutdinov et al. showed Probabilistic Matrix Factorization (PMF), which used Gaussian distribution to initialize U and V and applied a logistic function to limit the range of predicted to [0,1]. Koren et al. summarized this work in [4] and gave a generic framework for Matrix Factorization. Researchers also managed to incorporate information from other data sources. Zhang et al. used review sentiment analysis to construct virtual ratings for users who have not explicitly on the item [5]. Gu et al. proposed the Graph Weighted Nonnegative Matrix

Factorization (GWNMF) [6] to use user/item neighborhood graphs to preserve neighborhood information in user/item latent vectors [5][2]. utilized social network information under the assumption that friends share similar tastes and interests. In the realm of recommender systems, collaborative filtering approaches like matrix factorization and nearest-neighbor methods have received a lot of attention and have proven to be effective at recognizing similar people or products and capturing user preferences. Similar to this, item attributes and user profiles have been analyzed by content-based filtering techniques like cosine similarity and natural language processing (NLP) to produce personalized suggestions.

Methodology:

Sherpa's development process is methodical and iterative, starting with the design and implementation of its fundamental components and system architecture. The MERN (MongoDB, Express.js, React.js, Node.js) stack powers the program's backend, while a cross-platform mobile application frontend created with React Native powers the application's architecture. This architecture makes sure that various devices and operating systems work together seamlessly, giving users a consistent and easy experience. The cornerstone of Sherpa's personalized experience is its hybrid recommender system, which combines collaborative filtering with SVD and content-based filtering using cosine similarity matrices. Content-Based Filtering (Cosine Similarity Matrix):

Matrix factorization [4] is one of the most used approaches in recommender systems. Despite its efficiency, MF still suffers from sparsity problems, i.e., users who rate only a small portion of items could not get proper recommendations, and items with few ratings may not be recommended well. To cure the Sparsity problem, we have utilized the Novel approach of Cos MF from [7]. The formula for Matrix Factorization is given in eq. (1):

$$R_{i,j} = U_i V_j^t \qquad (1)$$

Where n is users and m is items. Each element of $R_{i,j}$ represents the rating of user i to item j (trip). Ui denotes the latent preference row vector of user i and Vj denotes the latent feature row vector of item j, and both have k dimensions (latent factors).

Cosine Similarity is calculated using eq. (2):

$$R_{i,j} = \frac{U_i V_j^t}{||U_i||.||V_j||} = \cos\,(U_i, V_j) \qquad (2)$$

Collaborative Filtering analyzes a user preferences database to predict additional products or services in which a user might be interested [8]. Collaborative filtering techniques analyze user-item interactions and user preferences to identify similar users or items, while content-based filtering techniques analyze item attributes and user profiles to generate recommendations that match users' preferences. Singular Value Decomposition is carried out using eq. (3):

$$A = U\,\mathcal{E}\,V^t \;(3)$$

Where A is the input matrix, U is the left singular matrix, Sigma is diagonal/eigenvalues and V is the right singular matrix. By leveraging the strengths of both approaches, Sherpa can deliver highly personalized recommendations that resonate with individual users.

Sherpa's hybrid recommender system is trained on a curated dataset sourced from TripAdvisor, one of the world's largest travel platforms [9]. This dataset contains a diverse range of user interactions, reviews, and item attributes, providing Sherpa with rich and comprehensive data to inform its recommendation process. The dataset is preprocessed and transformed into a suitable format for training the recommender system, ensuring that it captures the underlying patterns and relationships in the data.

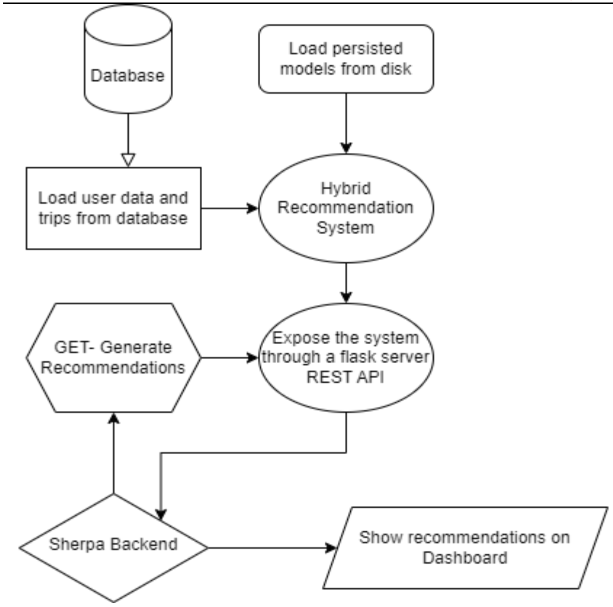

Database
Load persisted models from disk
Load user data and trips from database
Hybrid Recommendation System
GET- Generate Recommendations
Expose the system through a flask server REST API
Sherpa Backend
Show recommendations on Dashboard

Figure 1: Training the Hybrid Recommender system.

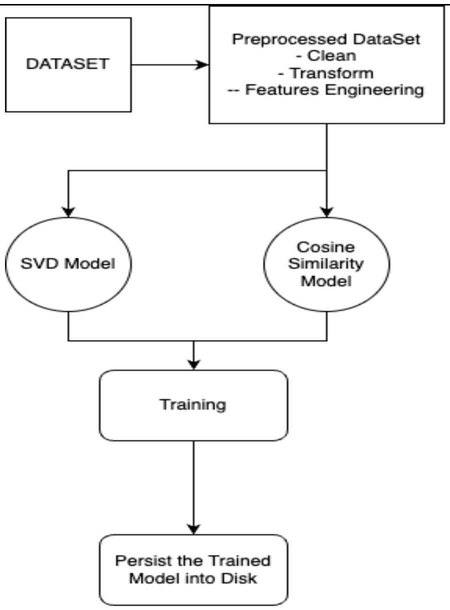


Figure 2: Detailed Process: Integration with Sherpa

Figure 1 outlines the methodology adopted to train the hybrid recommendation system using the procured Trip Advisor Dataset. First, we've carried out necessary cleaning and transformations as well as feature extraction on the dataset. This dataset is then divided into training and testing sets by a 70/30 margin. The training set is used to train the SVD and cosine similarity Models of course each model utilizes different features of the dataset, it is subsequently tested and performance metrics are calculated before persisting the model onto storage.

As Figure 2 demonstrates, the hybrid recommendation system is then integrated with the Sherpa Backend by exposing the get recommendation function through a Flask Server Rest API, which the backend calls/hits to generate recommendations for each user on the go.

Results:

As shown in the table the Hybrid Recommendation System managed to achieve a Root Mean Square Error (RMSE) of 0.8 and a Mean Absolute Error (MAE) of 0.64 both of which are indicative of good and relevant recommendations on the part of Collaborative Filtering. The content-based filtering has also proved to suggest relevant recommendations based on tags and descriptions relevant to the user's interactions with the app. The application back and front end both have been quality tested for user experience, robustness, and seamless performance. According to closed alpha test feedback, the hybrid recommendation system has streamlined trip finding for the user and the issue of cold starts for new users is almost non-existent, "recommendations were organic similar to the likes of Netflix or YouTube". We believe these results to be a good omen for the realization of our hybrid recommendation system and app "Sherpa".

Table 1: Performance Metrics of the Hybrid Recommendation System

Metric	Value
RMSE	0.8
MAE	0.64

Discussion:

The heart of Sherpa's personalized experience lies in its hybrid recommender system. The content-based component utilizes natural language processing to analyze tour descriptions and reviews, creating user and item profiles. The collaborative component leverages user-item interactions to understand and predict preferences. The integration of these methods allows

Sherpa to deliver highly relevant guide recommendations that resonate with individual user interests. The Hybrid Recommender Sherpa engine is trained on a curated dataset, ensuring diversity in user preferences and guide offerings. The iterative development of Sherpa incorporates user feedback at every stage, aligning with Agile methodologies. The repetitive cycle guarantees continuous polishing and adjusting of the application to satisfy user anticipations. The engine behind, fueled by Node.js alongside Express.js, manages data handling and API oversight, whereas the front part, shaped using React, provides an easy-to-use and swift interaction. The RESTful APIs developed in the stage of execution permit effective dialogue between the smartphone app and the server backend.

Our dataset often appeared as a sparse matrix, making it difficult to derive meaningful insights or predictions. To address this, we employed Singular Value Decomposition (SVD), which helped reduce the dimensionality of the dataset and extract underlying patterns. Additionally, we tackled the cold start problem by constructing cosine similarity matrices, enabling us to measure the similarity between users or items based on their features or preferences. By implementing these solutions, we were able to enhance the effectiveness and robustness of our recommendation system, ensuring better recommendations even in the face of sparse data and new user or item entries. The results from Sherpa's deployment indicate a successful integration of AI within a mobile tourism application.

Conclusion:

Utilizing SVD alongside the Matrix of Cosine Similarity for crafting uniquely tailored journey suggestions from data on TripAdvisor, Sherpa stands as a pioneer in the evolving era of navigated travel. Prospective endeavors could focus on advanced processing of natural language for an enriched comprehension of inclinations, integration of data instantaneously, and betterment through feedback from users. Sherpa will become the leader in personalized travel technology by including augmented reality previews, boosting sustainability, expanding suggestions to encompass more travel-related topics, and optimizing scalability. This ground-breaking method promises to revolutionize travel by providing unmatched customization and paving the way for further developments in the travel industry.

Acknowledgments:

Our profound appreciation goes out to the instructors, fellow students, and mentors who were instrumental in making the Sherpa project a reality. Their leadership, encouragement, and contagious excitement helped to transform this project into a worthwhile and prosperous endeavor. We extend a special thanks to Trip Advisor for the dataset that made the conception of our Hybrid Recommendation System possible.

References:

[1] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," Proceedings, vol. 2006, pp. 549–553, 2006, doi: 10.1137/1.9781611972764.58.

[2] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization".

[3] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization".

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer (Long. Beach. Calif)., vol. 42, no. 8, pp. 30–37, 2009, doi: 10.1109/MC.2009.263.

[5] W. Zhang, G. Ding, L. Chen, and C. Li, "Augmenting Chinese online video recommendations by using virtual ratings predicted by review sentiment classification," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 1143–1150, 2010, doi: 10.1109/ICDMW.2010.27.

[6] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," Proceedings, pp. 199–210, 2010, doi:

10.1137/1.9781611972801.18.

[7] H. Wen, G. Ding, C. Liu, and J. Wang, "Matrix Factorization Meets Cosine Similarity: Addressing Sparsity Problem in Collaborative Filtering Recommender System," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8709 LNCS, pp. 306–317, 2014, doi: 10.1007/978-3-319-11116-2_27.

[8] S. Zhang, W. Wang, J. Ford, F. Makedon, and J. Pearlman, "Using singular value decomposition approximation for collaborative filtering," Proc. - Seventh IEEE Int. Conf. E-Commerce Technol. CEC 2005, vol. 2005, pp. 257–265, 2005, doi: 10.1109/ICECT.2005.102.

[9] "TripAdvisor Dataset." Accessed: May 05, 2024. [Online]. Available: https://www.researchgate.net/publication/308968574_TripAdvisor_Dataset

Meta-Space: Pioneering Education in the Metaverse

Ibrahim Khalid[1], Umair Adnan[1], Hussnain Sajjad[1], Muhammad Abeer Irfan[1], Yaser Ali Shah[2], Abid Iqbal[1],

[1] Dept. of Computer Systems Engineering University of Engineering and Technology, Peshawar Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan.

Correspondence: 20pwcse1970@uetpeshawar.edu.pk, 20pwcse1960@uetpeshawar.edu.pk, 20pwcse1958@uetpeshawar.edu.pk, abeer.irfan@uetpeshawar.edu.pk, Yaser@cuiatk.edu.pk, abid.iqbal@uetpeshawar.edu.pk,

Citation | Khalid. I, Adnan. U, Sajjad. H, Irfan. M. A, Shah. Y. A, Iqbal. A, "Meta-Space: Pioneering Education in the Metaverse", IJIST, Special Issue pp 43-50, May 2024

Received| May 03, 2024, **Revised|** May 08, 2024, **Accepted|** May 14, 2024, **Published|** May 21, 2024.

In the evolving landscape of learning methodologies, technology has emerged as a catalyst, transforming the educational experience. This study delves into the realm of Virtual Reality (VR) and Augmented Reality (AR), collectively referred to as the "Metaverse," as a pivotal tool in education. By conducting systematic literature reviews, we investigate the potential, effectiveness, and associated pros and cons of employing the Metaverse for learning. Our findings affirm that the Metaverse proves to be a highly effective learning platform, enhancing engagement through lifelike avatars and bridging the gap between the real and virtual worlds. While this innovative approach facilitates visualizing materials and fosters interactive and interesting learning environments, challenges such as the cost of requisite devices remain. Despite limitations, the advantages of integrating the Metaverse into education are evident, necessitating ongoing development to amplify benefits and address existing constraints. This research contributes valuable insights to the ongoing discourse on leveraging Metaverse technologies for enriching educational practices.

Keywords: Metaverse; Virtual Classroom; Education; Communication.

Introduction:

Learning is a lifelong journey, an ever-present thread woven into the fabric of our lives, guiding us to new insights and knowledge, regardless of age. From our earliest days, the pursuit of knowledge has been a fundamental aspect of our existence, often shaped by formal education. However, the methods through which we learn vary in their ability to captivate our interest. Traditional learning materials, dominated by text with sparse illustrations, often present a challenge to making the learning experience truly enjoyable and engaging.

In the contemporary landscape, the rapid evolution of technology has revolutionized various domains, with education standing prominently affected [1]. Technological tools, once as simple as projectors in classrooms [2], have now transformed into virtual classrooms conducted via video conferences, especially accentuated by the global impact of the COVID-19 pandemic. Yet, the shift to virtual learning comes with its own set of limitations, particularly in fostering interactive student-teacher and peer interactions.

Amid the ongoing quest for optimal technological solutions to enhance education, Virtual Reality (VR) and Augmented Reality (AR) have emerged as promising contenders [3][4]. Virtual Reality enables users to immerse themselves in computer-simulated environments, as seen in applications like Google Earth's VR implementation. On the other hand, Augmented Reality seamlessly integrates digital elements into the real world, exemplified by social media filters and popular games like Pokémon Go. Together, VR and AR converge to form what is colloquially known as the "Metaverse", a virtual realm that melds the tangible and digital universes.

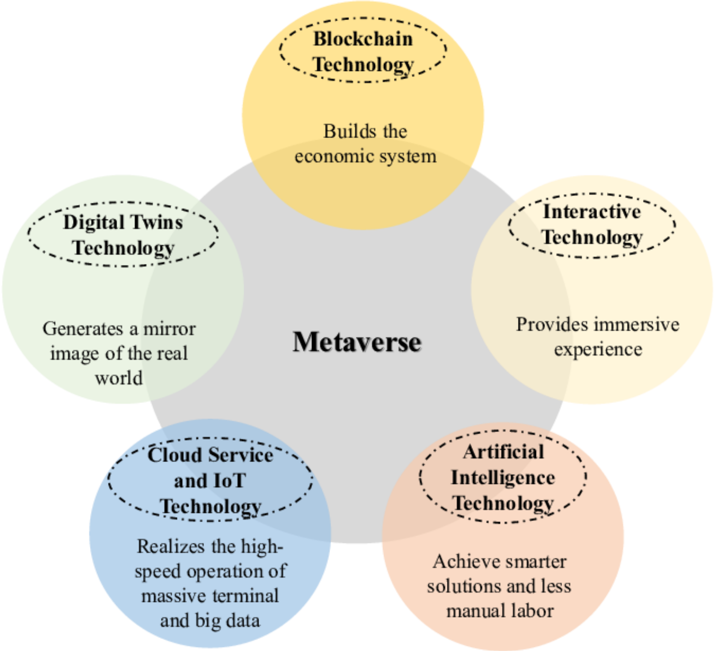

Blockchain Technology
Builds the economic system
Digital Twins Technology
Generates a mirror image of the real world
Interactive Technology
Provides immersive experience
Metaverse
Cloud Service and IoT Technology
Realizes the high-speed operation of massive terminal and big data
Artificial Intelligence Technology
Achieve smarter solutions and less manual labor

Figure 1: Metaverse-related technologies and their impact on the Meta verse [5]

In addition, the virtual classroom has broader applications in studying human behavior, skill training, and gaming. Neuropsychology benefits greatly from immersive virtual environments. The virtual classroom is revolutionizing education with its impact on accessibility, flexibility, and the learning experience. We examine the difficulties and opportunities for social good associated with the metaverse in this study as we delve into its complex domain. We look at how digital technologies might improve educational experiences and meet social requirements, especially in the setting of virtual classrooms. As part of our methodology, a metaverse classroom is meticulously built using state-of-the-art technologies and techniques to guarantee realism and usefulness. Our study strives to unearth insights that pave the path for revolutionary improvements in education and beyond as we negotiate the complexity of the metaverse landscape.

Challenges in the Educational Metaverse:

Challenges from Regulators in the Education Metaverse:

The government and other regulatory bodies have not given the education metaverse the attention it deserves, which is one of its main challenges. Through resource sharing and real-time interactive platforms, this virtual learning environment offers enormous potential to educate more people. But because the regulators weren't involved from the beginning, there aren't any clear guidelines or directives. On paper, the government's plans don't offer enough support for this new approach to education, which incorporates socializing, trading, and creative activities in addition to traditional classroom instruction. Encouraging the regulators to join the education metaverse is essential to its growth and development [6].

Challenges Faced by Designers in the Education Metaverse:

The education metaverse is a dynamic and easily navigable digital learning environment that you can utilize at any time and from any location thanks in large part to the work of metaverse designers. Currently, the technology mostly depends on AR/VR/MR devices, however, there is a small issue. The learning experience isn't as immersive as we'd like it to be because of these devices' limitations. It's like having a great gadget that's not quite ready for prime time. The metaverse's required gadgets are also somewhat expensive and difficult to transport. So, even if you have this amazing learning space, it's not always practical to use it. Furthermore, designers still haven't quite figured out how to best organize the many components of the metaverse and determine what each one should be used for in terms of teaching. Therefore, more work needs to be done to ensure that the education metaverse is truly excellent.

Challenges Faced by Users in the Education Metaverse:

When users step into the education metaverse, they rely on a network of shared resources and strong social connections to enhance their learning. But here's the catch – the ethical rules of this virtual world haven't been set up properly yet. This means users might face similar problems as they would in the real world. Because the metaverse is a big collection of social ties, users might be tempted to explore using less trustworthy resources and tricks.

Metaverse for Social Good:

Even though the metaverse is essentially a virtual world centered around human interaction, it has a notable positive influence on the real world. This impact is particularly evident in areas such as accessibility, diversity, equality, and humanity. In the following section, we highlight some notable applications that showcase how the metaverse contributes to social good [7].

Metaverse Applications for Enhanced Accessibility:

In today's globalized world, communication and collaboration between countries have become more frequent. However, the challenge of geographical distance remains a significant obstacle, leading to increased costs in various processes. The COVID-19 pandemic further exacerbated this issue, causing the suspension of many events due to preventive measures.

Enter the metaverse, a solution that enhances accessibility to meet diverse social needs. For instance, numerous events have seamlessly transitioned to virtual formats, thanks to the metaverse. A notable example is UC Berkeley hosting its graduation ceremony on Minecraft in 2020. Additionally, platforms like Fortnite host a plethora of virtual events, including concerts such as the one featuring Travis Scott. These instances underscore how the metaverse has seamlessly integrated into our daily lives, offering a cost-effective and secure means to fulfill our social needs.

Exploring Metaverse for Inclusive Experiences:

The limitations of the physical world, including factors like geography and language, make it challenging to integrate diverse elements in one place to cater to the needs of different individuals. Enter the metaverse, offering an expansive virtual realm with seamless scene transformations that can effectively achieve diversity. The metaverse provides a platform for a myriad of intriguing scenarios, breaking free from physical constraints. For instance, Animal Crossing organized a presidential campaign for Joe Biden, showcasing the diverse possibilities within the metaverse. Similarly, students at Stanford University exhibited their posters in Second Life. These examples, however, only scratch the surface as the metaverse hosts a plethora of activities spanning education, shopping, political campaigns, artwork, pets, haunted houses, and more. Consequently, the metaverse significantly fulfills the diversity requirements of our physical society.

Digital Twins:

Besides metaverse users, even things in our everyday world can also connect and interact with the virtual realm, appearing as digital twins in this digital space. Imagine them as identical virtual copies that mimic real-world objects. How does this work? Well, devices in the real world have their information collected through widespread sensing technologies. These technologies keep the virtual copies, or digital twins, updated to reflect the current status of their real counterparts. This connection between the physical and virtual worlds opens up exciting possibilities. It's like having a parallel version of the real world in the metaverse, allowing for a seamless exchange of information and actions between our physical surroundings and the virtual space.

Methodology:

This methodology presents a comprehensive guide to developing an interactive and immersive metaverse classroom using Blender, a powerful 3D modeling software. The process commences with meticulous planning of the classroom layout, encompassing crucial elements like avatars, furniture, walls, doors, windows, whiteboard, roof, and lighting. Designing the core structure in Blender involves creating walls, doors, windows, and roofs, ensuring accurate proportions and a realistic environment.

The methodology emphasizes applying appropriate texturing and lighting to achieve an authentic ambiance within the virtual classroom. Thorough testing and iterative refinement phases ensure optimal functionality and visual quality, promoting an immersive and rewarding learning experience. Upon successful completion, the virtual classroom project is saved in Blender and exported to incompatible formats, making it readily accessible and adaptable for various virtual reality platforms and Metaverse applications.

3D Modeling with Blender:

The very first step in our workflow, shown in Figure 2, is creating 3D assets for our metaverse. Before diving into Blender, we carefully planned the layout and design of the virtual classroom. Considering the essential elements, such as avatars, chairs, tables, walls, whiteboards, windows, doors, roofs, and lighting. We envisioned a modern and interactive learning space to ensure an engaging and immersive experience for students.

Designing Classroom:

With the design plan in mind, we opened Blender, the powerful 3D modeling software, created a new project, and set the appropriate dimensions for the virtual classroom scene. Using Blender's versatile tools, creating the core structure of the virtual classroom. Modeling the walls, doors, windows, and roof to give the classroom its basic form. We ensured that the proportions and scale were accurate to provide a realistic environment.

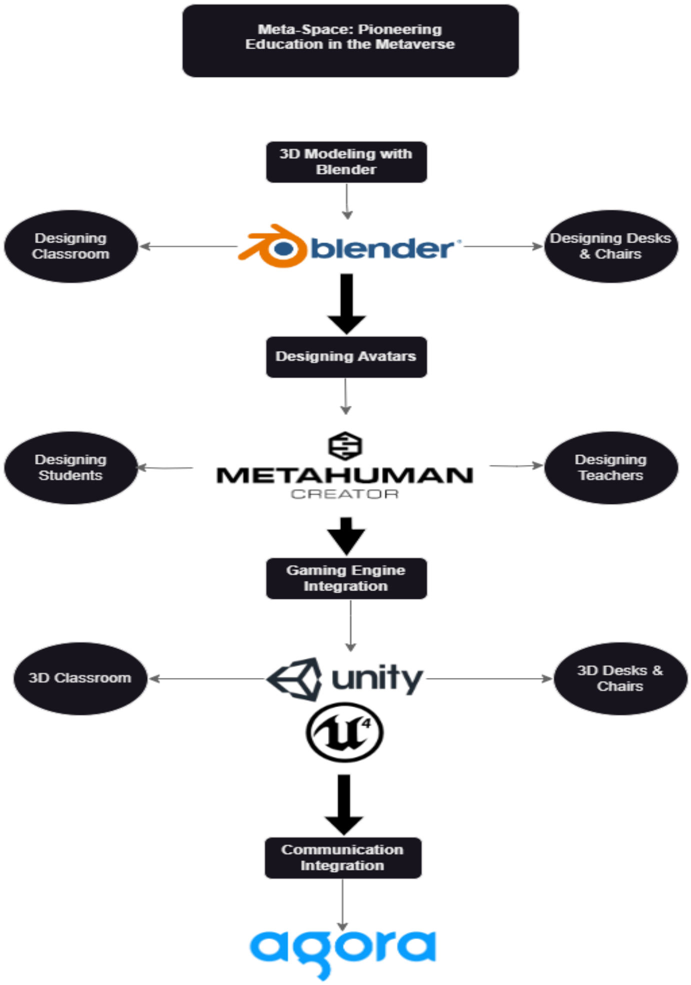


Figure 2: Meta-Space Flowchart.

Designing Desks and Chairs:

We focused on designing the furniture and props for the virtual classroom. Using Blender's modeling tools, we crafted chairs, tables, and a whiteboard that fit seamlessly into the classroom setting. Attention to detail, such as textures and colors, to make the objects visually appealing and true to real-life counterparts.

Designing Avatars:

Meta-Human Creator is a free, cloud-streamed tool you can use to create your digital humans in an intuitive, easy-to-learn environment. Using Meta Human Creator, you can customize your digital avatar's hairstyle, facial features, height, body proportions, and more [8]. In the Metaverse classroom, integrating avatars using Meta Humans enhances interactivity and engagement. These lifelike digital representations of users allow for real-time communication and collaboration. Avatars foster active participation in discussions, role-playing, and group activities, creating a personalized and immersive learning environment. With that in mind, we created avatars for teachers and students.

Gaming Engine Integration:

A gaming engine like Unity or Unreal Engine would be the perfect platform to use to bring our Meta-Space concept to life, merging our 3D classroom elements like chairs, desks, and avatars into a unified virtual setting. Using a gaming engine's capabilities enables dynamic interactions, smooth asset integration, and the deployment of several functionalities. We can give avatars the ability to move around the virtual classroom, interact with furniture like desks and chairs, and even participate in group activities by using scripting and programming within the gaming engine. These engines also provide capabilities for improving visual fidelity and performance, guaranteeing an engaging and immersive experience in our Meta-Space environment.

Communication Integration:

Agora SDK is an all-inclusive real-time communication solution that gives developers the infrastructure and resources they need to incorporate message, audio, video, and live streaming features into their projects. The Agora SDK's capability to provide 3D spatial audio is one of its most notable features. By mimicking how sound behaves in the real world, spatial audio improves the immersive experience of virtual worlds by enabling users to detect audio sources from various angles and distances. Agora uses spatial audio techniques and sophisticated audio algorithms in its SDK to do this. Agora SDK generates realistic audio environments that improve users' sense of presence and immersion when interacting with 3D spaces or virtual worlds. These environments are achieved by precisely placing audio sources within a virtual space and adjusting properties like volume, directionality, and distance attenuation. Applications where spatial awareness and realistic audio interactions are crucial for an engaging user experience, like virtual events, online gaming, remote collaboration, and virtual classrooms, will greatly benefit from these capabilities. We currently are working to implement Agora SDK in our project.

Results:

The results of the experimental process yielded insightful findings across various dimensions, shedding light on the efficacy and potential of the implemented metaverse classroom.

Classroom in Blender:

The classroom is designed with a capacity of 10-15 students. This number ensures that there isn't too much traffic on the server. The classroom has enough room for the teacher to move around during lectures and for students to not get too cramped in a space.

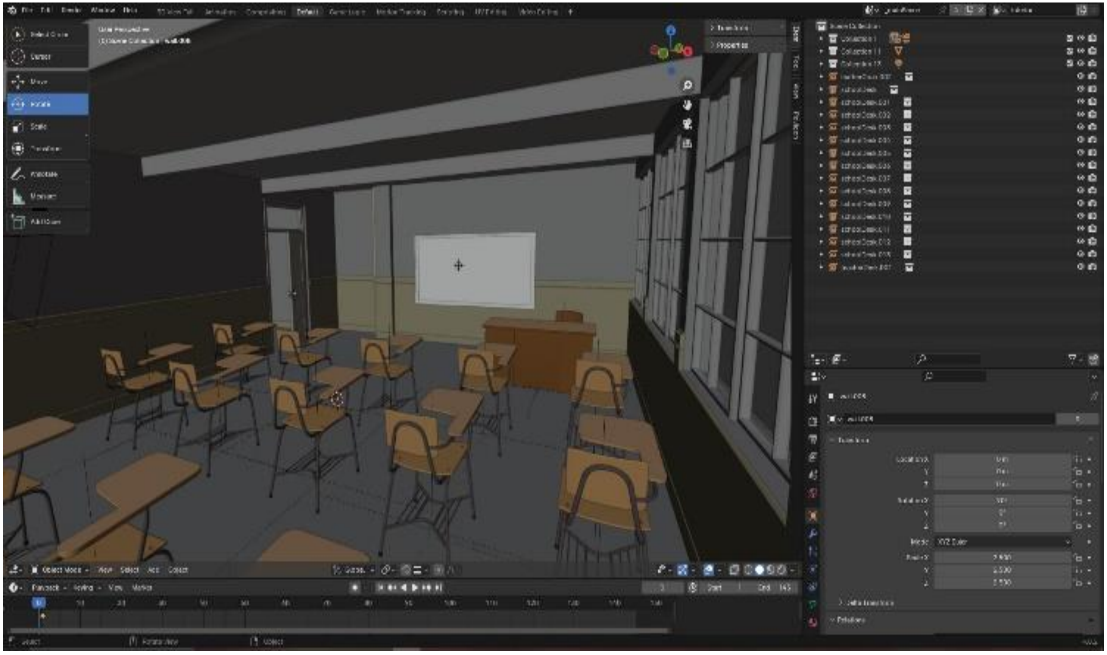


Figure 3: Classroom Model in Blender.

Furniture and Props:

For the furniture, we designed simple chairs and desks for the classroom to give a real-life-like feeling. The props include lights in your classroom and a whiteboard. The users can freely interact with the furniture and lights, bringing the virtual classroom to life. Figure 3 and Figure 4 show the whole classroom and the teacher's desk.

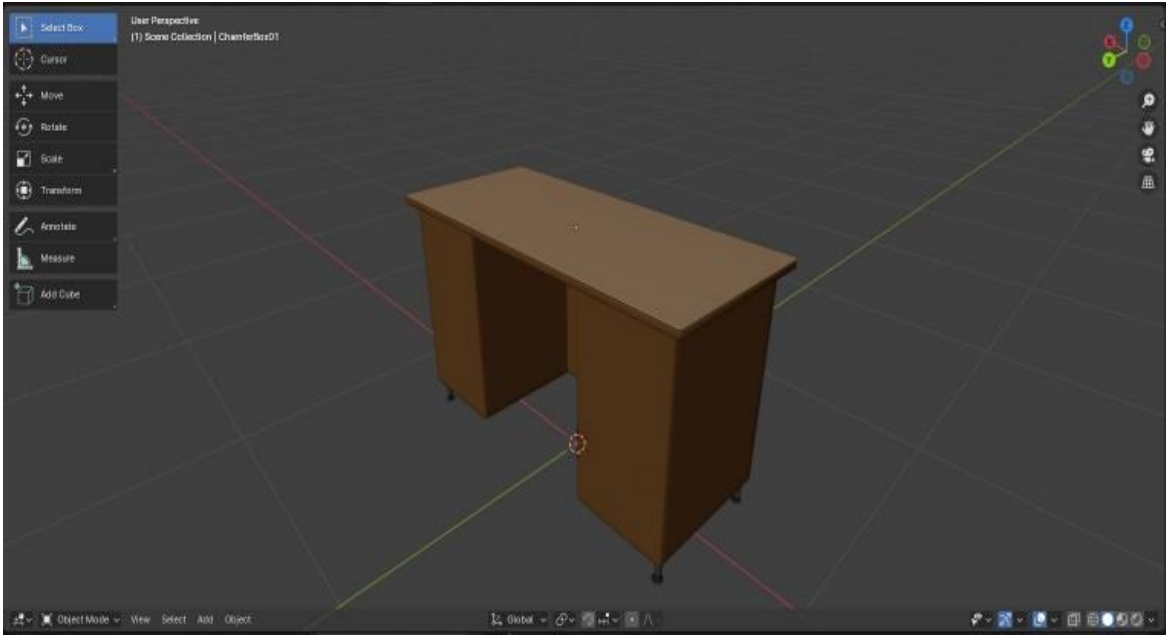
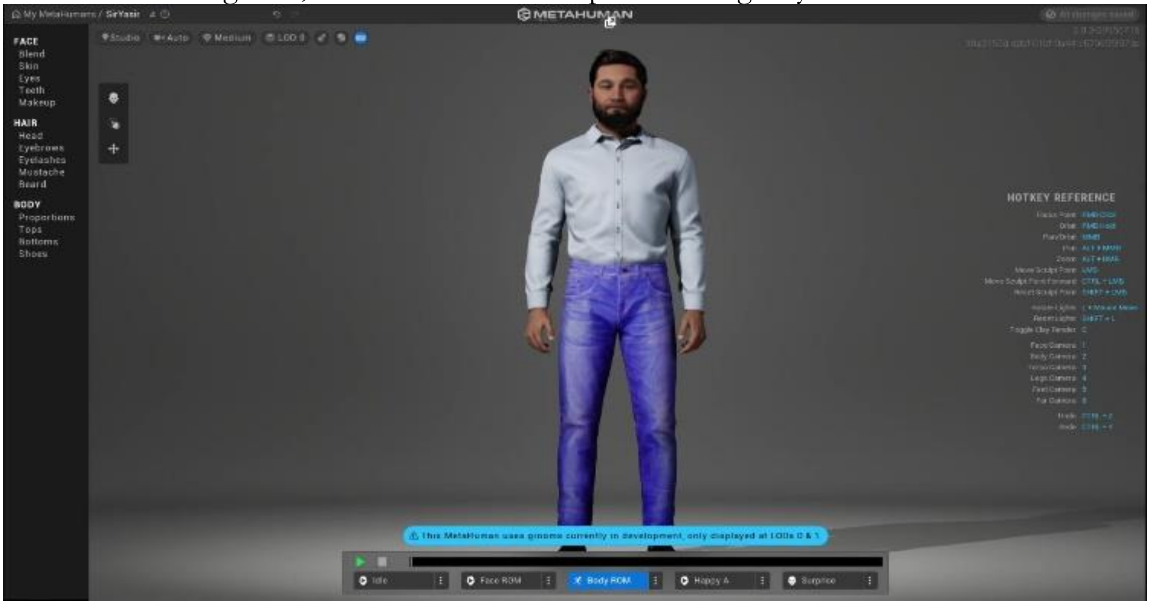


Figure 4: Modeling of Desk in Blender

Avatars for the Classroom:

In the Metaverse classroom, we are currently working on integrating avatars using Meta Hu men, which enhances interactivity and engagement. These lifelike digital representations of users allow for real-time communication and collaboration. Avatars foster active participation in discussions, role-playing, and group activities, creating a personalized and immersive learning environment. In Figure 4, we tried to create our professor digitally.



This MetaHuman uses grooms currently in development, only displayed at LODs 0 & 1

Figure 5: Sir Yasir Avatar in Meta-Human

Conclusion:

To sum up, the education metaverse has the amazing potential to completely transform the way we learn by removing obstacles based on geography and enabling access to education for everyone. We do, however, have certain obstacles. To control the expansion of the metaverse, regulators including the government, need to be more vigilant and establish explicit guidelines. To use accessible technology to create an immersive and useful learning experience, Designers have their work cut out, too, as they strive to make the learning experience immersive and practical with accessible technology. While social networks and shared resources are beneficial, users must exercise caution to avoid taking unnecessary risks in this rapidly changing digital environment. To guarantee that everyone involved has a positive and meaningful learning experience, teamwork, careful rules, and ongoing technological advancements are necessary to realize the full potential of the education metaverse.

References:

[1] P. C. N. R. Raja, “Impact of modern technology in education,” J. Appl. Adv. Res., vol. 3, no. 1, pp. 33–35, 2018.

[2] R. A. Liono, N. Amanda, A. Pratiwi, and A. A. S. Gunawan, “A Systematic Literature Review: Learning with Visual by The Help of Augmented Reality Helps Students Learn Better,” Procedia Comput. Sci., vol. 179, pp. 144–152, Jan. 2021, doi: 10.1016/J.PROCS.2020.12.019.

[3] V. C. C. Akshay, D. Visagaperumal, “Metaverse future of internet,” Int. J. Res. Publ. Rev., vol. 2, no. 8, pp. 386–392, 2021.

[4] Y. Sun and M. Gheisari, “Potentials of Virtual Social Spaces for Construction Education,” vol. 2, pp. 469–459, 2021, doi: 10.29007/sdsj.

[5] Z. Chen, J. Wu, W. Gan, and Z. Qi, “Metaverse Security and Privacy: An Overview,” Proc. - 2022 IEEE Int. Conf. Big Data, Big Data 2022, pp. 2950–2959, 2022, doi: 10.1109/BIGDATA55660.2022.10021112.

[6] T. Hao and H. Lailin, “Educational Metaverse Dilemmas and Solutions: A stakeholder-based perspective,” Proc. - 2022 12th Int. Conf. Inf. Technol. Med. Educ. ITME 2022, pp. 714–718, 2022, doi: 10.1109/ITME56794.2022.00150.

[7] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, “Metaverse for Social Good: A University Campus Prototype,” MM 2021 - Proc. 29th ACM Int. Conf. Multimed., pp. 153–161, Oct. 2021, doi: 10.1145/3474085.3479238.

[8] “Meta Human Creator Overview | MetaHuman Documentation | Epic Developer Community.” Accessed: May 04, 2024. [Online]. Available: https://dev.epicgames.com/documentation/en-us/metahuman/metahuman-creator

A Deep Learning Based Mobile Application for Wheat Disease Diagnosis

Sarmad Riaz[1] Raja Taimour[1], Mashab Ali Javed[2], Amaad Khalil[3] Yasir Saleem Afridi[3] Abid Iqbal[4]

[1]Department of CS & IT, University of Engineering & Technology, Peshawar 25000, Pakistan

[2]Department of Computer Systems Engineering, Sir Syed CASE Institute of Technology Islamabad Pakistan,

[3]Department of Computer Systems Engineering, University of Engineering & Technology, Peshawar 25000, Pakistan

[4]Department of Electrical Engineering Jalozai Campus, University of Engineering & Technology, Peshawar 25000, Pakistan

*Correspondence:sarmadriaz385@gmail.com,rajataimur794@gmail.com ,Jd.mashab@gmail.com.,amaadkhalil@uetpeshawar.edu.pk.,yasirsaleem@uetpeshawar.edu.pk ,abid.iqbal@uetpeshawar.edu.pk

Citation | Riaz. S, Taimour. R, Javed. M. A, Khalil. A, Afridi. Y. S, Iqbal. A, “A Deep Learning Based Mobile Application for Wheat Disease Diagnosis”, IJIST, Special Issue pp 51-62, May 2024

Received | May 03, 2024, **Revised |** May 11, 2024, **Accepted |** May 16, 2024, **Published |** May 21, 2024.

Wheat is one of the major staple crops in Pakistan, playing a crucial role in ensuring food security and contributing to the country's economy. The productivity and quality of wheat crops, however, are vulnerable to several illnesses. The ability to diagnose these diseases quickly and accurately is crucial for taking the appropriate preventative actions, limiting losses, and maintaining food security. In this research paper, we build and test a wheat disease detection system adapted to the conditions in Pakistan. The suggested method uses machine learning-based techniques along with image processing algorithms to automatically detect and categorize various wheat diseases based on their symptoms. High-resolution photos of healthy wheat plants and sick plants displaying different diseases were collected from different regions of Pakistan in order to construct an accurate and robust disease detection model. The dataset has been annotated by plant pathologists who provided true labels for use in evaluation and training. To achieve the best results in wheat disease diagnosis, many cutting-edge deep-learning architectures were investigated and optimized. These included Convolutional Neural Networks (CNNs) and Transfer Learning models. Multiple models’ effectiveness was evaluated using accuracy, precision, and recall, in a series of extensive trials.

Keywords: Wheat diseases, Convolutional Neural Networks (CNNs), Transfer Learning, Tensor Flow.

Introduction:

Wheat is a vital cereal crop in Pakistan, serving as a staple food for the nation's population and playing a significant role in its agricultural economy. However, the cultivation of wheat faces formidable challenges, with plant diseases being a primary concern. Various diseases, caused by fungal, bacterial, and viral pathogens, can severely affect wheat crops, leading to substantial yield losses and compromising food security in the country. Timely and accurate diagnosis of wheat diseases is crucial for implementing effective strategies for disease management [1]. Traditionally, disease identification has relied on manual visual inspection by experienced agronomists and plant pathologists. While effective, this process is time-consuming, and subjective, and may lead to misdiagnosis due to the similarity of symptoms among different diseases.

In recent years, technological advancements in the fields of machine learning, computer vision, and image processing have revolutionized the agricultural sector **Error! Reference source not found.**. These developments offer promising opportunities to automate disease detection and revolutionize the way we monitor and manage plant health. This paper aims to develop a wheat disease detection system tailored specifically to the conditions prevailing in Pakistan. By integrating cutting-edge machine learning algorithms and image processing techniques, the system will automatically identify and classify various wheat diseases based on visual symptoms exhibited by infected plants. To achieve this, a comprehensive dataset comprising high-resolution images of healthy and diseased wheat plants was collected from diverse regions in Pakistan.

In many regions of the world, the absence of infrastructure makes it difficult to quickly identify wheat infections, which pose a danger to food security. To that end, we plan to develop a mobile application for diagnosing agricultural diseases. We propose an intelligent and effective application that leverages Al computer vision and machine learning algorithms to identify agricultural diseases. Our dataset is the Plant-Village Dataset (New), which is part of the CNN family. The latest iteration of the plant-village dataset includes 10,000 photos for training and 2,500 for validation. A separate set of 33 test photos was used to evaluate the accuracy of the model.

Literature Review:

Researchers have made great strides in the field of agriculture recently. The detection of plant diseases has been accomplished using a variety of methods. Many studies reveal that they employ various algorithms; among these are Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) **Error! Reference source not found.**. Despite this, researchers are constantly on the lookout for new methods that are both more effective and more interesting to users.

Yusuke Kawasaki and Hiroyuki Uga in **Error! Reference source not found.** analyzed methods for spotting plant diseases using photographs of their leaves. They discussed several methods for removing the afflicted portion of the plant. They also looked at certain feature extraction and clustering techniques for identifying plant diseases and distinguishing between healthy and contaminated leaves. To ensure a successful harvest, accurate recognition and categorization of plant infections using image processing is essential. Methods for removing the highlights of a contaminated leaf, characterizing plant diseases, and several techniques for fragmenting the infected portion of the plant. Disease in plants can be characterized with ease using ANN methods including self-sorting highlight maps, back spread calculations, support vector machines, and so on. Based on these methods, a picture-handling strategy can be used to accurately identify and rank various plant diseases [5]. Results were evaluated using a dataset of 87k RGB photographs of healthy and defected plant leaves divided into groups of 38, out of which 25 were chosen for experimental purposes. This makes use of Al and cameras to detect objects. The proposed approach has a 93% success rate in identifying 20 distinct plant diseases across 5 different species together with back propagation neural network (BPNN) and other digital image processing techniques.

In **Error! Reference source not found.**, Monica Jhuria, Ashwani Kumar, and others discussed techniques for identifying plant illness in photographs of leaves. Two datasets are used in the implementation of support vector machine (SVM). In this case, training datasets are compared to their corresponding datasets stored photographs. After applying a filter, two photographs are compared to one another. After contrasting healthy and unhealthy regions, they arrived at a percentage fraction. An artificial neural network method has been applied for disease identification. They have a wide variety of algorithms for disease detection. Some examples of algorithms used in the artificial neural network method are backpropagation, support vector machine, and major component analysis **Error! Reference source not found.**. The proposed method has an accuracy of 91%.

With the help of deep learning and image identification, Ahmed, A. A., and Reddy, G. H. **Error! Reference source not found.** examined the technological possibility of automating disease detection. Using a publicly available dataset with 54,306 images of healthy and diseased plant leaves, a deep convolutional neural network has been trained to classify crop species based on their disease status into 38 categories, including 14 species of crops and 26 types of crop diseases. The model is accurate to within 1% on average. The average accuracy of random guessing on a dataset with 38 class labels is only 2.63%. Overall accuracy on the Plant Village dataset ranged from 85.53% (in case of Alex Net: Training from Scratch: Gray Scale: 80 – 20) to 99.34% (in case of Google Net: Transfer Learning: Color: 80 - 20), demonstrating the promising potential of the deep neural network architecture.

Methodology:

This section explains how to identify plant diseases using photographs of affected leaves. Extracting the image's attributes or valuable information from the image is the goal of image processing, a subfield of signal processing. It displays accurate results by evaluating multiple picture attributes to diagnose illnesses on plant leaves as shown in Figure 1. Crop diseases pose a significant risk to food security, yet their prompt detection continues to be challenging in numerous regions globally due to inadequate infrastructure. The aim is to develop a mobile application for diagnosing diseases that can be easily accessed by farmers. The implementation through a mobile application aims to enhance access for the average farmer. Our solution includes identifying potential causes of diseases and offering corresponding treatments. Utilizing mobile phone applications will assist farmers in improving their production levels and income.

Plant and crop diseases can be broken down into four categories: oomycetes, hyphomycetes, bacteria, and viruses. Visual examination of leaf color patterns and crown architecture remains the gold standard in conventional field scouting for crop diseases. Examining plant leaves for disease symptoms and diagnosing plant diseases based on experience requires a significant amount of time, effort, and expertise when done using the naked eye. In addition, the wide range of plants means that illnesses can manifest themselves in a wide range of ways across various crops, adding a layer of complication to the process of categorizing plant diseases. Meanwhile, a lot of research has been done using machine learning to categorize plant diseases. First, the background is removed, or the infected part is segmented using preprocessing techniques; second, distinguishing features are extracted for analysis; and third, classification or clustering algorithms are used for feature classification [9].

Most of the algorithms developed for previous machine learning techniques did not meet the needs of real-world applications, even though many novel algorithms have been established in this field. The agricultural sector that uses machine learning to improve crop yields is increasingly moving towards Deep Learning techniques and in particular CNNs. Because of its versatility in detection and classification, such as weed detection, crop pests' categorization tasks, or identification of crop illnesses, deep learning approaches are increasingly used in agriculture production. One advantage of using a Deep Learning model is that it eliminates the need for a segmentation operation when extracting features from a task. The object's retrieved features are successfully mined from the raw data.

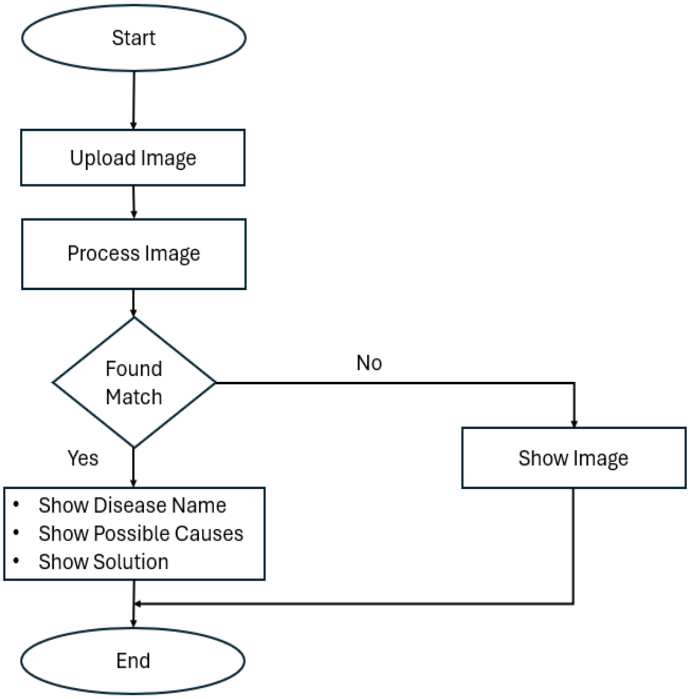


Figure 1: Flow chart of application

Convolutional Neural Network (CNN):

Major demands on CNN -based, when it comes to the classification of plant diseases, deep learning is unmatched in terms of both scale and variety of datasets **Error! Reference source not found.**. The majority of leaf disease classification systems use CNN. Other types of DL networks, such as deconvolution networks and fully convolutional networks (FCNs) are more commonly utilized for picture segmentation and medical diagnosis than for classifying diseases in plant leaves [11].

The image's local correlation is used by the convolutional layer in order to extract relevant features. The image's upper left corner is marked with a kernel. Multiplying the pixel values by their matching kernel values, summing the products, and finally adding the bias. The kernel is shifted by one pixel and the filtering process is repeated until the entire image has been processed. The pooling layer makes the model robust to translations, rotations, and scaling by randomly choosing features from the feature map of the higher layer. Maximum or average pooling is the most popular option. In maximum pooling, the input image is divided into numerous rectangular areas with the size of the filter determining which regions receive the maximum value. When regions are pooled together, the result is an average of all of them. In many implementations, convolutional layers follow a pooling layer and vice versa. For classification or detection tasks, the classifier integrates and converts multidimensional information into one-dimensional features at the fully connected layer, where each neuron is connected to the neuron above it **Error! Reference source not found.**.

VGG19:

VGG stands for Visual Geometry Group. The VGG network is specially crafted for tasks related to image classification. VGG19, a part of this network, is comprised of 19 layers, with 16 being convolutional layers and 3 fully connected layers. The convolutional layers are tasked with extracting features from input images, whereas the fully connected layers handle the classification of these features into various categories or classes.

Table 1: Dataset

Dataset	Images
Leaf rust	2000
Loose Smut	1800
Crown and Root Rot	1700
Healthy	2200

MobileNetV2:

MobileNetV2 is an attempt to design a convolutional neural network that can function well on mobile devices. It is predicated on a backward residual structure, with the bottleneck levels connecting via residual nodes. Lightweight depth-wise convolutions are used in the intermediate expansion layer to filter features and introduce non-linearity. MobileNetV2's overall architecture consists of a 32-filter fully convolutional first layer, followed by 19-filt]er residual bottleneck layers **Error! Reference source not found.**.

Plant-Village Dataset:

- The dataset as shown in Table 1 has been acquired from Hazara University Mansehra, KPK, Pakistan. It has 10k total images.
- It has 4 classes having diseased and healthy leaves.
- The dataset is divided into 80/20 ratio (8000/2000) for training and validation respectively.

Data Augmentation:

It's common knowledge that deep learning works best with a lot of information. Little data may not be sufficient for model training. For this purpose, data augmentation to create new examples for the training phase. Common methods of data enhancement include geometric modifications including mirroring, cropping, rotation, and translation as shown in Figure 2.

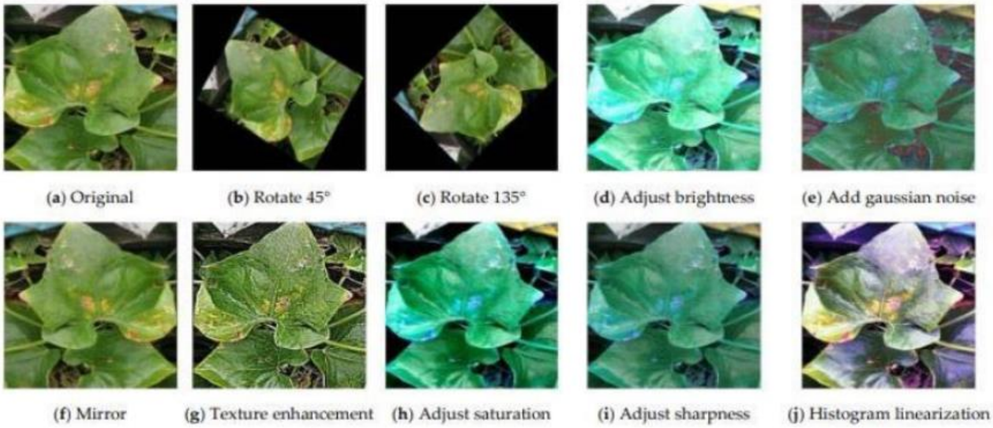


Figure 2: Data Augmentation

Data Generation:

In some cases, access to reliable information is limited. In this scenario, fake data for used in the training detection model. The usage of synthetic data creation has grown in the field of machine learning because of its inexpensive cost. Generative Adversarial Networks (GANs) methods can be used to generate fake data. To construct fictional instances from a dataset with the same properties as the original set, GAN uses a generative modeling technique called generative adversarial networks [12].

Model Selection:

Alex Net, VGG Net, GoogLe Net, Res Net, Mobile Net, and Efficient Net are only a few of the CNN-based classification models produced in DL-related research for use in classification tasks **Error! Reference source not found.**. After placing first in the ImageNet competition in 2015, Microsoft Lab unveiled the ResNet network. Using shortcut connections and residual blocks, the network was able to address the problem of gradient reduction. In 2016, Res Net networks gained more attention in the field of Deep Learning.

In 2017, Google's engineering teams unveiled the Mobile Net network for use in mobile gadgets and embedded software. In 2019, those same Google groups unveiled the Efficient Net network. The network used an easy compound coefficient to implement the strategy of scaling the depth, resolution, or width. When it comes to plant and crop diseases, Deep networks are not required for classification but are the ideal solution because of their high performance. Alex Net and VGG16 are thought to be suitable for the actual accuracy performance needed in agricultural production, in addition to Deep networks **Error! Reference source not found.**. This process of training and deployment can be divided into the following three steps. The first step is data preprocessing and preparation. The second step is Model building, training, and evaluation and the final step is model inference and deployment as shown in Figure 3.

Preparing the Data and Preprocessing

Deep Learning models prioritize data preparation and preprocessing data. Strong accurate and trained input data bounds to precise results. Original datasets require training, validation, and testing set processes, the general statistical percentages for such are 70:20:10, 80:10:10, and 60:20:20 [15]. In our model system, the model has been fed with public plant datasets by using the Kaggle platform, as it contains 87k images.

A good DL model architecture is required prior to training. Better accuracy and faster classification can be achieved with a well-designed model. CNNs, RNNs, and GANs are the three most common forms of DL networks nowadays. When it comes to the task of detecting and classifying plant diseases, CNN is by far the most used feature extraction network.

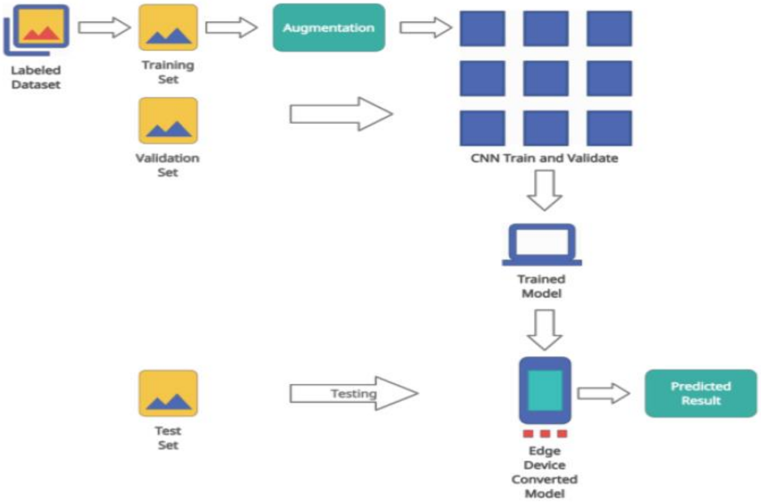


Figure 3: Steps in model training

Once the framework of the model has been built, hyperparameters can be adjusted for use in training and testing. The grid search technique can be used to iteratively explore several parameter configurations in pursuit of the optimal one. During neural networks, training data are stored in the first layer, and back-propagation is used to adjust the weight of each neuron based on whether the output matches the label. This cycle is repeated until a new skill can be taught using the available data. The model's efficacy was measured with metrics like accuracy, precision, recall, and F l score. These indexes can't be discussed in isolation; rather, they need to be introduced alongside the more general concept of a confusion matrix. In binary classification, the confusion matrix displays the expected yes/no answers **Error! Reference source not found.**.

Evaluation Measures

The classification models, involving both the detection and classification of plant diseases, were implemented with deep learning. The statistical evaluation classified the samples of images into the following statuses: True-Positive (TP), which determines the perfectly-identified image samples being infected, False-Positive (FP), which determines the wrong classified image samples being infected, True-Negative (TN), which determines the correct classified image samples being healthy and False-Positive (FP), which determines the wrong identified image samples being healthy.

$$\text{Accuracy} = TP+TN/TP+TN+FP+FN \qquad (1)$$

In plant disease evaluation, classification and accuracy are considered essential for the purpose. From equation 1 High value of accuracy and precision tends to be regarded as better for the performance. When the value of F1 is less, the trained model tends to perform much better. The capability of the trained model is applied to new data when the training and evaluation processes are finished.

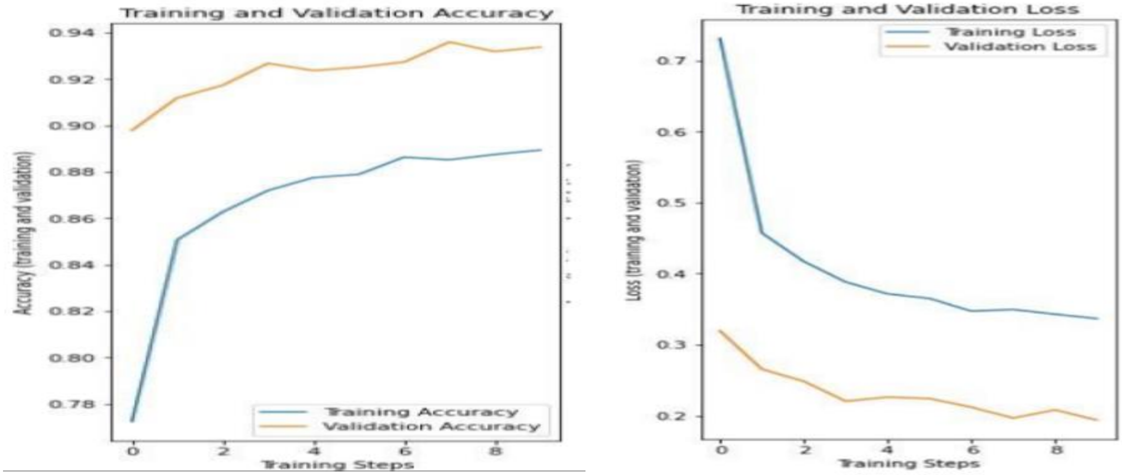


Figure 4: Model training and validation**.**

Transfer Learning:

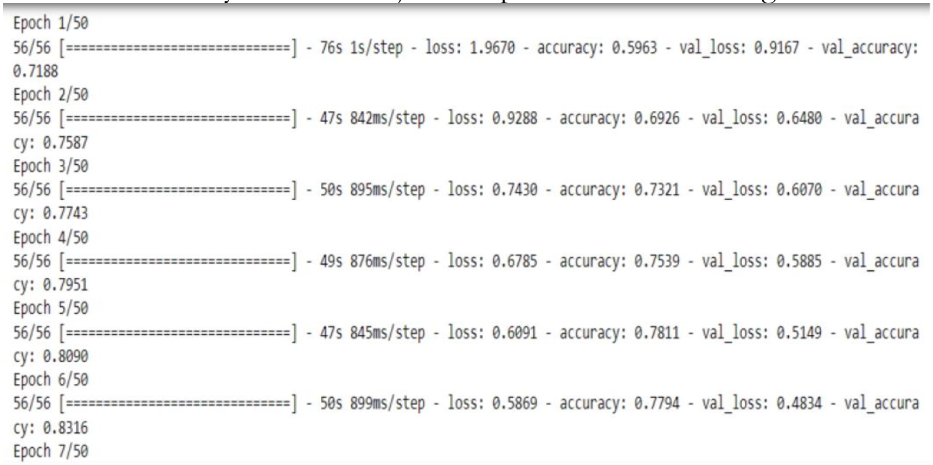
Transfer learning comes in the classification of the machine learning technique domain. The certain technique adopts the learning capabilities from the recent tasks to the proceeding tasks **Error! Reference source not found.**. With new databases, a few layers of pre-trained networks of the model are retrained by reducing the need for masses of datasets, inclining the model towards better performance. Research by Mukti et al reports the utilization of the transfer learning model by using Res Net 50, by implementing the approach of recognizing plant diseases, as it gives satisfactory results. The report contains a dataset of 87.867 image samples where 80% of the dataset is used for the training set and the remaining 20% is used for validating the set process. The report concludes with an accuracy of 99.80% from the model implemented practically.

Tensor Flow:

TensorFlow is an open-source machine-learning framework used by researchers and developers. It offers a rich ecosystem of tools and resources for creating and deploying ML apps. Users can choose the level of abstraction they need, with the Keras API simplifying model building. Eager execution allows for fast iteration and debugging. The Distribution Strategy API enables distributed training without modifying model design, making it suitable for large ML jobs. TensorFlow also supports creating complex models efficiently with features like the Keras Functional API. Supplementary libraries like Tensor Flow Probability and BERT can be used alongside Tensor Flow for various tasks.

Results:

The proposed system results are gathered from the trained model before its deployment and after its deployment in mobile-based applications. We used a technique called cross-validation, in which we divided their data into a training set and a validation set. The model is "trained" using the training set, while its performance is "validated" using the validation set. Figure 4 depicts loss and accuracy during training and validation, respectively. The validation accuracy is higher than the training accuracy. On the training dataset, the model's loss will nearly always be smaller than on the validation dataset. Thus, a discrepancy between the train and validation loss learning curves is to be anticipated. The void between these two ideals is known as the "generalization gap." A validation loss that is smaller than the training loss may also be used to detect it. It suggests the validation dataset may be more predictable by the model than the training dataset. Accuracy is used as a measure of style in the context of typography. Accuracy refers to the proportion of correctly predicted events that our version anticipated. The formal definition of precision is as follows: the proportion of correct predictions to total forecasts is the standard by which accuracy is measured. We see convergence in our model. We obtained a validation accuracy of 93%+ in just 10 epochs as shown in Figure 4.

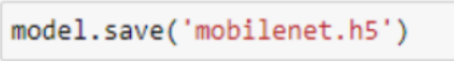


```
Epoch 1/50
56/56 [==============================] - 76s 1s/step - loss: 1.9670 - accuracy: 0.5963 - val_loss: 0.9167 - val_accuracy:
0.7188
Epoch 2/50
56/56 [==============================] - 47s 842ms/step - loss: 0.9288 - accuracy: 0.6926 - val_loss: 0.6480 - val_accura
cy: 0.7587
Epoch 3/50
56/56 [==============================] - 50s 895ms/step - loss: 0.7430 - accuracy: 0.7321 - val_loss: 0.6070 - val_accura
cy: 0.7743
Epoch 4/50
56/56 [==============================] - 49s 876ms/step - loss: 0.6785 - accuracy: 0.7539 - val_loss: 0.5885 - val_accura
cy: 0.7951
Epoch 5/50
56/56 [==============================] - 47s 845ms/step - loss: 0.6091 - accuracy: 0.7811 - val_loss: 0.5149 - val_accura
cy: 0.8090
Epoch 6/50
56/56 [==============================] - 50s 899ms/step - loss: 0.5869 - accuracy: 0.7794 - val_loss: 0.4834 - val_accura
cy: 0.8316
Epoch 7/50
```

Figure 4: Accuracy values:

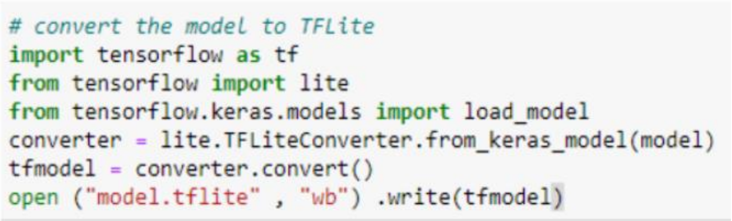
To make our model communicate with App we must convert it into the Tensor Flow lite version, which is made for mobile Versions **Error! Reference source not found.**. So, you can build or create a mobile app and make the app communicate with the model. The following steps are being done:

- The saved model was converted into TFLite.



```
model.save('mobilenet.h5')
```

- Model converted to TFLite which is then used to develop a mobile application.



```
# convert the model to TFLite
import tensorflow as tf
from tensorflow import lite
from tensorflow.keras.models import load_model
converter = lite.TFLiteConverter.from_keras_model(model)
tfmodel = converter.convert()
open ("model.tflite" , "wb") .write(tfmodel)
```

The prototype application uses a camera or device media to get an image of the crop as shown in Figure 5. Preview the images and send them to API, for disease detection are shown in figure 6 and figure 7. Results page showing detected disease as given in Figures 7 and 8 for diseases such as Root Rot and Leaf Rust. The healthy wheat plant results are given in Figure 9.

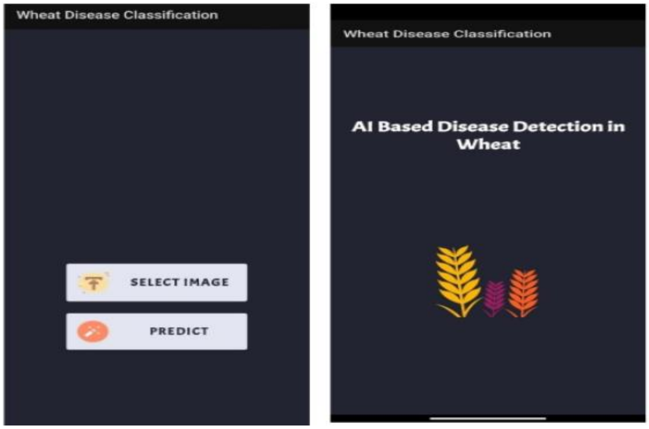
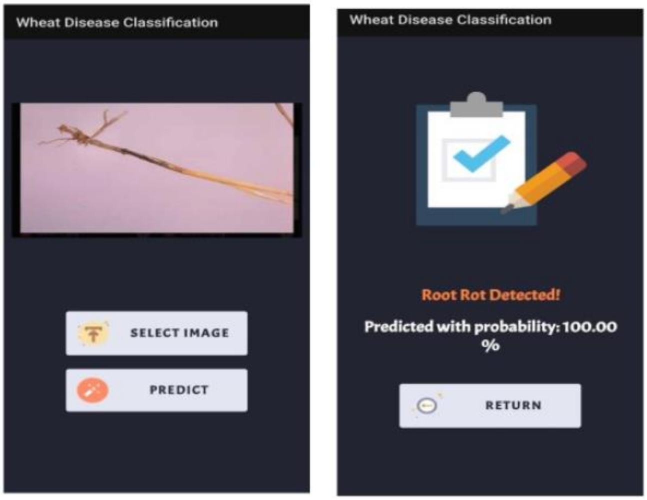

Wheat Disease Classification
SELECT IMAGE
PREDICT
Wheat Disease Classification
AI Based Disease Detection in Wheat

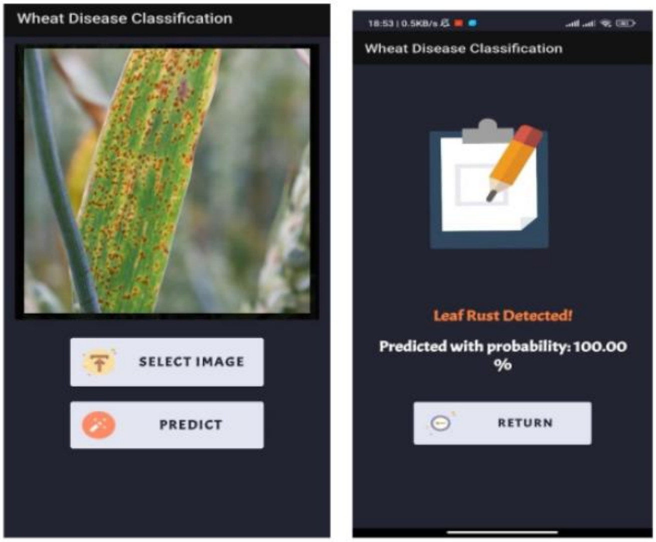
Figure 5: Prototype of mobile application



Root Rot Detected!

Predicted with probability: 100.00 %

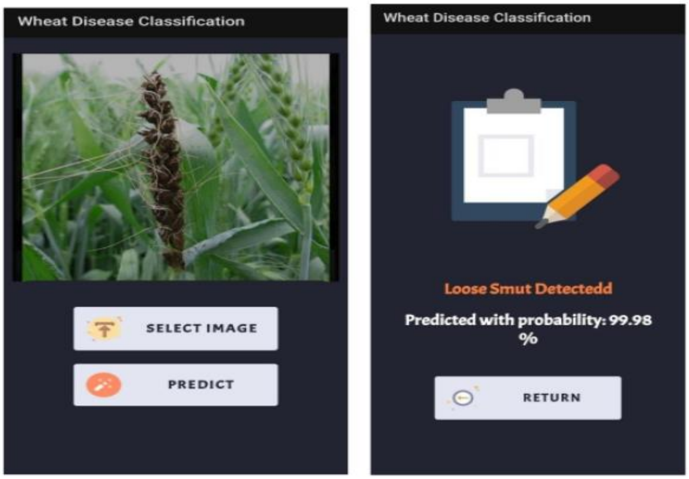
Figure 6: Root Rot disease



Leaf Rust Detected!

Predicted with probability: 100.00 %

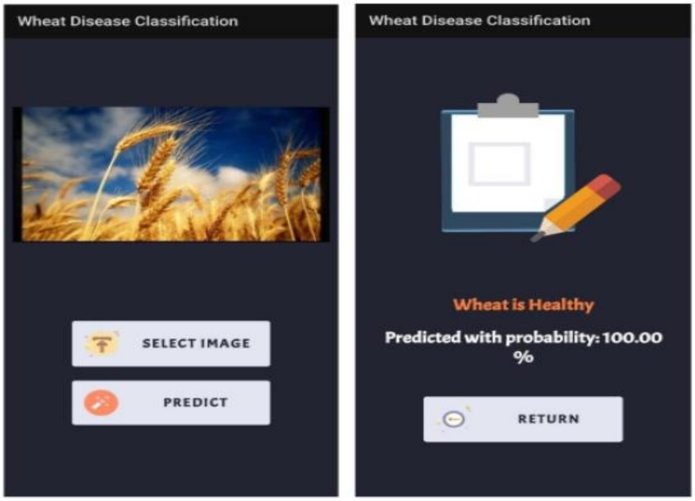
Figure 7: Leaf Rust disease



Loose Smut Detectedd

Predicted with probability: 99.98 %

Figure 8: Loose smut disease



Wheat is Healthy

Predicted with probability: 100.00 %

Figure 9: Healthy leaves

Conclusion:

Wheat diseases are serious hazards to food security and must be addressed immediately to prevent total crop failure. Many times, however, farmers are unable to tell the difference between diseases that present with identical symptoms. Because of this, fertilizer applications may be under- or over-applied. If diseases are misdiagnosed. Professionals in the field are required to identify wheat crop illnesses. Who can prevent the loss of the entire harvest? To mitigate this loss and better instruct farmers via video, in the proposed solution we apply Convolutional Neural Network (CNN) multi-layer ANN algorithms hereinafter referred to as Deep Learning Algorithms. The overarching goal was to enhance the efficacy of agricultural methods. Currently, machine vision-based plant disease and pest detection equipment are being widely used in agriculture, replacing the more laborious and time-consuming practice of identifying these problems by eye. We trained and validated our strategy using a transfer learning method. We also conducted an evaluation of the design using a dataset of 14 plant types and 38 class labels. We propose an intelligent and effective application software that leverages AI computer vision and machine learning algorithms to identify agricultural diseases. We have faith that this study's findings will be seen as supplementary to those already published, opening the door to important studies of transfer learning methodologies for plant and disease identification.

References:

[1] S. S. Hari, M. Sivakumar, P. Renuga, S. Karthikeyan, and S. Suriya, "Detection of Plant Disease by Leaf Image Using Convolutional Neural Network," Proc. - Int. Conf. Vis. Towar. Emerg. Trends Commun. Networking, ViTECoN 2019, Mar. 2019, doi: 10.1109/VITECON.2019.8899748.

[2] L. Ale, A. Sheta, L. Li, Y. Wang, and N. Zhang, "Deep learning based plant disease detection for smart agriculture," 2019 IEEE Globecom Work. GC Wkshps 2019 - Proc., Dec. 2019, doi: 10.1109/GCWKSHPS45667.2019.9024439.

[3] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," Front. Plant Sci., vol. 10, p. 941, Jul. 2019, doi: 10.3389/FPLS.2019.00941.

[4] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," Turkish J. Electr. Eng. Comput. Sci., vol. 27, no. 3, pp. 1636–1651, Jan. 2019, doi: 10.3906/elk-1809-181.

[5] K. Lin, L. Gong, Y. Huang, C. Liu, and J. Pan, "Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network," Front. Plant Sci., vol. 10, p. 422622, Mar. 2019, doi: 10.3389/FPLS.2019.00155/BIBTEX.

[6] M. Agarwal, S. Gupta, and K. K. Biswas, "A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant," Sustain. Comput. Informatics Syst., vol. 30, p. 100473, Jun. 2021, doi: 10.1016/J.SUSCOM.2020.100473.

[7] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic Study of Automated Diagnosis of Viral Plant Diseases Using Convolutional Neural Networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9475, pp. 638–645, 2015, doi: 10.1007/978-3-319-27863-6_59.

[8] A. A. Ahmed and G. Harshavardhan Reddy, "A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning," AgriEngineering 2021, Vol. 3, Pages 478-493, vol. 3, no. 3, pp. 478–493, Jul. 2021, doi: 10.3390/AGRIENGINEERING3030032.

[9] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," Comput. Electron. Agric., vol. 72, no. 1, pp. 1–13, Jun. 2010, doi: 10.1016/J.COMPAG.2010.02.007.

[10] G. Fenu and F. M. Malloci, "An application of machine learning technique in forecasting

crop disease," ACM Int. Conf. Proceeding Ser., pp. 76–82, Nov. 2019, doi: 10.1145/3372454.3372474.

[11] Y. Fang and R. P. Ramasamy, "Current and Prospective Methods for Plant Disease Detection," Biosensors, vol. 5, no. 3, p. 537, 2015, doi: 10.3390/BIOS5030537.

[12] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Front. Plant Sci., vol. 7, no. September, p. 1419, Sep. 2016, doi: 10.3389/FPLS.2016.01419/BIBTEX.

[13] V. Malathi and M. P. Gopinath, "RETRACTED ARTICLE: Classification of pest detection in paddy crop based on transfer learning approach," Acta Agric. Scand. Sect. B — Soil Plant Sci., vol. 71, no. 7, pp. 552–559, Oct. 2021, doi: 10.1080/09064710.2021.1874045.

[14] J.-F. Yeh, K.-M. Lin, C.-Y. Lin, and J.-C. Kang, "Intelligent Mango Fruit Grade Classification Using AlexNet-SPP With Mask R-CNN-Based Segmentation Algorithm," IEEE Trans. AgriFood Electron., vol. 1, no. 1, pp. 41–49, May 2023, doi: 10.1109/TAFE.2023.3267617.

[15] "Plant Leaf Disease Analysis using Image Processing Technique with Modified SVM-CS Classifier | PDF." Accessed: May 04, 2024. [Online]. Available: https://www.slideshare.net/slideshow/1490692238v5-2/74248022

[16] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," Conf. Adv. Signal Process. CASP 2016, pp. 175–179, Nov. 2016, doi: 10.1109/CASP.2016.7746160.

[17] "The Essential Guide To Learn TensorFlow Mobile and Tensorflow Lite | by Rinu Gour | Towards Data Science." Accessed: May 04, 2024. [Online]. Available: https://towardsdatascience.com/the-essential-guide-to-learn-tensorflow-mobile-and-tensorflow-lite-a70591687800

Optimized Coverage and Capacity Planning of Wi-Fi Network based on Radio Frequency Modeling & Propagation Simulation

Syed Junaid Ali Shah[1], Naveed Mufti[2], Toufeeq Ahmad[1]

[1]Telecommunication Engineering University of Engineering & Technology Mardan Pakistan

[2]Director Technical SNM Solutions Pvt. Ltd. Rawalpindi, Pakistan

*Correspondence:s.junaid52462@gmail.com,mufti.naveed@gmail.com, drtoufeeq@uetmardan.edu.pk

Citation | Shah. S. J. A, Mufti. N, Ahmad. T, "Optimized Coverage and Capacity Planning of Wi-Fi Network based on Radio Frequency Modeling & Propagation Simulation", IJIST, Special Issue. 2 pp 63-73, May 2024

Received | May 04, 2024, **Revised |** May 10, 2024, **Accepted |** May 15, 2024, **Published |** May 21, 2024.

Investigation for optimized coverage and capacity planning of Wi-Fi network is carried out in the testbed for the purpose of optimization in terms of Received Signal Strength Indicator (RSSI), Signal to Noise & Interference Ratio (SNIR), Interference + Noise (I+N), downlink/uplink data rate and user capacity. The plan is carried out by conducting a site prediction survey through Altair's Win Prop Software which is a Radio Frequency (RF) modeling and signal propagation simulation software, using the configuration of actual Wireless Local Area Network-Access Points (WLAN-APs). First, the map of the testbed with all respective material properties is drawn in Win Prop's Wall Manager (Wall Man) Tool as a 3-Dimentional (3D) model. Then that 3D model is implemented in Win Prop's Propagation Manager (ProMan) Tool where APs are deployed and wave propagation analysis as well as capacity planning is done. Results are analyzed for optimal signal strength, data rate, and user handling capacity. The results are validated by a smartphone-embedded software known as Cellular-Z. The average optimization increase in coverage, downlink & uplink data rates is 3.95 dB, 2.53 Mbps & 3.42 Mbps respectively.

Keywords: Capacity; Coverage; Heatmap; Optimization; Wi-Fi.

Introduction:

Wi-Fi means wireless fidelity. It is the technology of providing wireless signals for internet connectivity to the users within an area of deployment. Wi-Fi works on IEEE standards of 802.11b/g/n/a/ac/ax, mainly within 2.4GHz and 5.6GHz Industrial Scientific Medical (ISM) bands. The available bandwidth within each band is further divided into sub-bands/channels which are responsible for user handling capacity. The 2.4GHz band provides a low data rate but large coverage as compared to the 5.6GHz band and vice versa. The Wi-Fi is provided to the internet users through a Wireless Local Area Network Access Point (WLAN-AP). The antennas of this AP are nearly omnidirectional with some gain, radiating radio signals in maximum directions toward the users. The Received Signal Strength (RSS) of the Wi-Fi signals decreases as the user moves away from the WLAN-AP. This variable signal strength as the users move towards and away from the WLAN-AP results in a variable data rate [1].

To provide connectivity of the internet to a maximum number of users with an acceptable data rate, it is important to know the optimum location of WLAN-AP for providing coverage at a specific RSS level to the internet users, the number of users that can be handled by a WLAN-AP and sufficient data rate that is provided to each user at a specific time. Conventional ways of performing this task are practical deployment and modification of the location of WLAN-APs and measuring the RSS at different portions of the area of interest without prior simulation surveys. This approach is tedious as well as costly, having low accuracy and taking more time to complete. To address the above issues, software-based simulation surveys are performed before the actual deployment of the WiFi network.

In the background study, the same software is used for a similar type of survey but the study did not use the actual WLAN-AP configuration, assumed omnidirectional antennas and the same antenna pattern (single frequency) for all WLAN-APs throughout the simulation process which lacks the opportunity of handling the co-channel interference and did not determine the number of users that could be supported with a specific minimum guaranteed data rate i.e. that survey is limited to propagation with no capacity planning [2].

Literature Review:

S. Zvanovec, P. Pechac, and M. Klepal experimented to analyze the merits and demerits of two distinct methods for Wi-Fi Network Deployment Survey. One was the practical survey in which Wi-Fi transmitters were deployed in a testbed and Received Signal Strength Indicator (RSSI) measurements were taken on multiple points and the second method was a simulation of the signal propagation model in a software tool. The experimental data was simulated in MATLAB. Results from both methods were analyzed and the software simulation method was preferred for Wi-Fi Network Deployment [3].

T. Honda, M. Ikeda, and L. Barolli conducted experiments to optimize the coverage by correct placement of Wi-Fi APs through site surveys and network simulations for the solution of connectivity problems. Results indicated that the received power from APs was not uniform [4]. T. Witono and Y. Dicky did practical site measurements of RSS for the optimization of 12 Wi-Fi transmitters in an overlapping Wi-Fi environment. The key controllers of a single Wi-Fi transmitter were the direction of transmitting antennas, the combination of channels, and the transmit power, all of which were adjusted for the optimization of Wi-Fi deployment [5].

U. Mir, O. U. Sabir, H. Ullah, and A. U. Khan experimented to achieve coverage optimization through accurate placement of APs based on RSSI measurements in the testbed. Site simulation survey was conducted through Tamograph and real-time measurements were taken through SSIDer software and Air Magnet hardware (validation). The results from the simulation and measurements were analyzed for optimization along with the voltage variation effects on the RSSI measurements [6].

J. Tan, X. Fan, S. Wang, and Y. Ren collected RSS measurements from the inertial sensors of a smartphone by walking in the testbed of the Wi-Fi environment for the purpose of accurate Radio Map (Wi-Fi Fingerprint) construction. The RSS data was processed by the pedestrian dead-reckoning algorithm for the production of raw trajectories. Those trajectories were refined by the assembling of constraints collected at the landmarks, by the use of Factor Graph Optimization (FGO). Then k-Nearest Neibour (kNN) algorithm was applied for the validation and localization performance testing of the Radio Map. The Radio Map was practically implemented in a shopping mall and a mean error of 1.10m and maximum error of 2.25m was recorded for Wi-Fi transmitter locations which is an acceptable error for Radio Map [7]. A. Srivastava, R. Vatti, V. Deshpande, J. Patil and O. Nikte did practical measurement of RSSI for finding dead zones with less or no coverage of Wi-Fi signals in the area through Netspot Tool. Further, optimization techniques of Particle Swarm Optimization {PSO (responsible for optimal Wi-Fi transmitter placement)} and Repeater deployment were implemented to solve the problem of coverage in the dead zones [8].

Y. Tian, B. Huang, B. Jia, and L. Zhao, developed an algorithm for the accurate placement of Wi-Fi access points and Bluetooth beacons in a Wi-Fi / Bluetooth hybrid environment. The "heuristic differential evolution algorithm" (HDEA) is based on the "Cramer-Rao lower bound (CRLB)". The CRLB is considered a standard for the localization and coverage of Wi-Fi/Bluetooth signals. Further, the Motley-Keenan model is assembled in the algorithm instead of the ideal Log Distance Path Loss (LDPL) model for the analysis of the effects caused by obstacles in the indoor environment. Based on these contents, the algorithm is deployed in a software application that is used with Geo-Tools for the localization of Wi-Fi access points and Bluetooth beacons. Extensive simulations and experiments in the field were conducted to validate the efficiency of the algorithm [9].

N. A. M. Maung and W. Zaw, conducted experiments to compare and analyze the performance of two techniques of Wi-Fi indoor positioning in 2.4GHz and 5GHz frequency bands. The implemented techniques were the path loss model and RSS Fingerprint. Results show that the RSS-based indoor Wi-Fi positioning performed with better accuracy than the other technique because the path loss model takes a direct reading of RSS value (highly variable due to multipath and interference) and estimates the location which leads to positioning error [10].

M. R. Akram, A. H. Al-Nakkash, O. N. M. Salim, and A. A. S. AlAbdullah, developed a multi-objective algorithm for the optimization of Wi-Fi coverage, location, and number of access points by the use of MATLAB software. The algorithm works on the Binary Particle Swarm Optimization (BPSO) technique which takes predefined RSS values to estimate the optimization of the aforesaid objectives. The deployment of the algorithm resulted in 64.6% coverage and 7dBm on average received power optimization [11]. O. S. Naif and I. J. Mohammed experimented Binary Particle Swarm Optimization (BPSO) algorithm used with Wireless Insite (WI) simulation software, to optimize the coverage and interference parameters of a multi-floor Wi-Fi AP deployment. The WI takes RSS values, signal thresholds, and current AP deployment locations to process the optimization in conjunction with BPSO. Results depict that the proposed work outperforms the present Wi-Fi deployment in RSS (-11.5dBm), path loss (11.5dBm), interference (7.87%), and coverage/ optimal AP placement of 39.23% [12].

A. S. Haron, Z. Mansor, I. Ahmad, and S. M. M. Maharum did a simulation-based survey to optimize the location of presently deployed Wi-Fi transmitters in 2.4GHz and 5GHz bands in terms of signal strength to overcome the problem of connectivity in the Communication Technology Laboratories Area at University of Kuala Lumpur British Malaysian Institute. Hyper Works' Win Prop simulation software was used. First, the layout of the testbed was modeled in the Win Prop's WallMan Tool, having similar properties/sizes of materials from the map. Then omni-directional antenna patterns of 2.4GHz and 5GHz were modeled in Win Prop's A Man Tool. After that, both models from WallMan and A Man were implemented in Win Prop's Pro Man Tool where the signal propagation analysis was done for each antenna with the exact deployment location as was in real. The simulation was carried out in 2.4GHz and 5GHz bands separately for all antennas with modification in the location of WLAN-APs and results were compared. After that physical validation of the acquired results was done using In SSIDer by measuring the RSS and the optimum location for the Wi-Fi transmitters was determined [2].

S. Baua and S. Karuppuswami, presented the Machine Learning (ML) technique of Modified Extensible Lattice Sequence (MELS) which is a regression-based supervised learning algorithm used with the Global Response Search Method (GRSM) optimization routine to optimize the coverage by correct placement of transmitters and number of Wi-Fi APs for an office area. A dual-slot antenna is designed operating in 5.2GHz (S46 / S54 bands) and having 6dBi of gain, to represent a single Wi-Fi AP for the processing of radio optimization. The target of optimization is to reduce the number of APs, place them in inaccurate locations for wider coverage, and provide at least 13Mbps of data rate at a 15000 square foot area at the office location [13].

I. Bridova and M. Moravcik did predictive and passive surveys to overcome the problem of Wi-Fi connectivity at the Department of Information Networks, University of Zilina. The methodology includes model construction from a Map and the creation of predictive heat maps in Ekahau simulation software in terms of relocation of APs with respect to hotspots and unnecessary areas. During prediction, no objects (furniture, electronic devices) were considered. Then physical readings were collected in the form of a passive survey and results were analyzed for optimal coverage, Signal to Noise Ratio, and Throughput [14]. All studies have focused on coverage optimization with no capacity planning. This research work performed propagation (coverage optimization) as well as network simulations (capacity planning) which provided radiation patterns for all antennas at once in the testbed. As a result, the RSSI, Signal Noise & Interference Ratio (SNIR), and Interference + Noise (I+N) in the testbed are analyzed as a combined result of all antennas at once (received power of the network) as well as data rate (downlink/uplink) and user handling capacity calculations are done.

Methodology:

It consists of three parts i.e. Physical data collection, implementation of collected data as 3D model generation & Wi-Fi network simulations, and optimization analysis & validation of results. The First part is carried out by acquiring physical data of the testbed including a 2D map with construction objects' sizes and materials; a technical datasheet of the access point with supporting Wi-Fi technologies, antenna patterns with respective gain, transmit powers, receiver sensitivities, polarizations & operational bands; RSSI values and downlink/uplink data rates as real-time measurements of the present case acquired through Cellular-Z. In the second part, the collected data is practically implemented in Altair's Win Prop software in which WallMan is a 3D modeling construction Tool and Pro Man is a signal propagation simulation Tool. Map with all details of the floor plan, construction sizes, and material properties are designed for the area where the Wi-Fi network is optimized as a 3D model in the WallMan Tool.

Then, [1] Access Point's Wi-Fi technologies, antenna patterns with respective gain, transmit powers, receiver sensitivities, polarizations & operational bands; RSSI values, and downlink/uplink data rates are created and deployed within the 3D model created in the previous step through Win Prop's Pro Man Tool. Then, radio coverage and network capacity planning in the modeled environment are simulated for all Wi-Fi APs using Win Prop's Pro Man for the present case. The simulation process is repeated 73 times in pursuit of coverage and capacity optimization, comparing each simulated case with the present case. Results are acquired for the 2.4GHz band. In the third part, results are analyzed for optimal signal strength, data rate, and user handling capacity. The best case is selected and applied in the testbed. The results are validated by smartphone-embedded software (Cellular-Z).

Results:

The results in Table 1 & Table 2 show the difference in RSSI, Data Rate, SNIR, I+N, and Modulation & Coding Schemes (MCSs) in the testbed.

Table 1: Before Optimization of APs

Received Power			
Mean (dBm)	**Median (dBm)**	**Standard Deviation (dBm)**	**Maximum (dBm)**
-73.13	-73.58	12.07	-45.02
Downlink/ Uplink Data Rate (Mbps)	Downlink: Signal to Noise & Interference Ratio (dB)	Downlink: Interference + Noise (dBm)	The number of Modulation & Coding Schemes Operated
19.48 / 4.10	10.17	-53	3

Table 2: After Optimization of APs:

Received Power			
Mean (dBm)	**Median (dBm)**	**Standard Deviation (dBm)**	**Maximum (dBm)**
-71.71	-72.62	11.12	-43.25
Downlink / Uplink Data Rate (Mbps)	Downlink: Signal to Noise & Interference Ratio (dB)	Downlink: Interference + Noise (dBm)	The number of Modulation & Coding Schemes Operated
64.94 / 13.67	63.59	-65	8

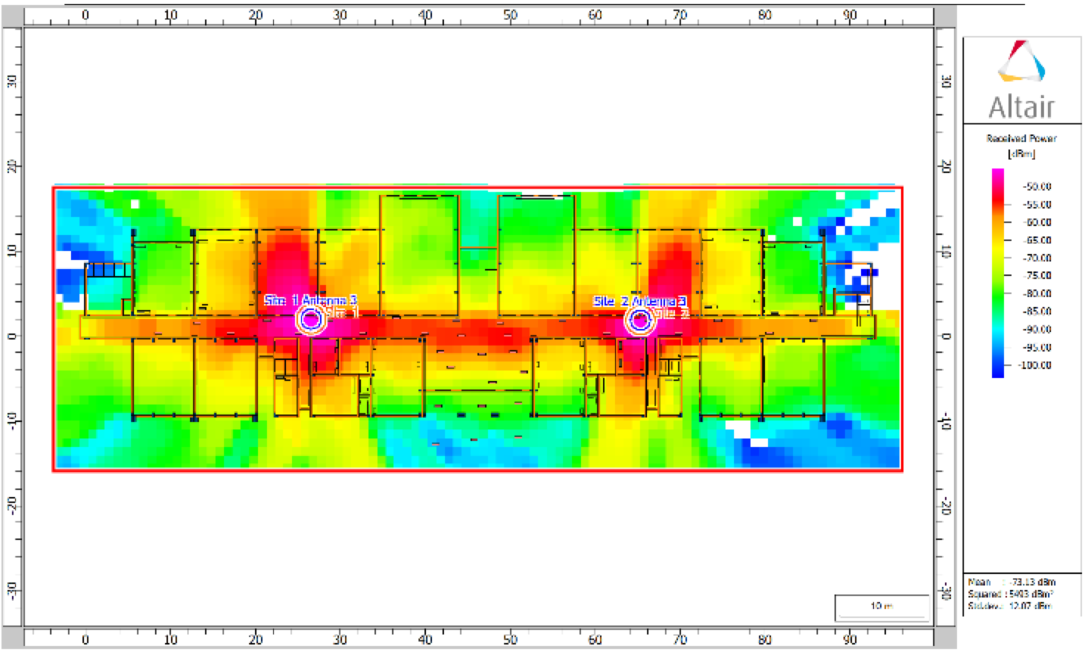


Figure 1: Simulation Values of RSSI Before Optimization

In Table 4, 19 location points are taken for comparison between real-time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 3 & Figure 4 respectively. The average optimization difference between non-optimized real measurements & optimized real measurements is 2.53 Mbps, the average optimization difference between non-optimized simulation predictions & optimized simulation predictions is 0.75 Mbps, and the average optimization difference between optimized simulation prediction & optimized real measurements is 4.17 Mbps.

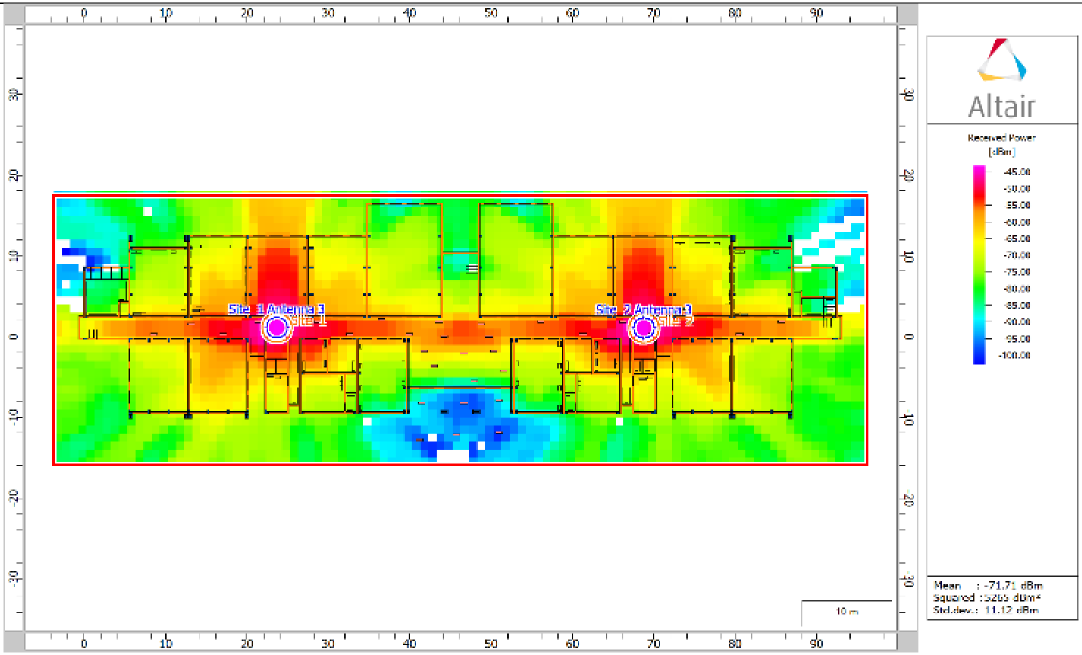


Figure 2: Simulation Values of RSSI After Optimization

In Table 3, 20 In-location points are taken for comparison between real-time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 1 & Figure 2 respectively. The average optimization difference between non-optimized real measurements & optimized real measurements is 3.95 dB, the average optimization difference between non-optimized simulation predictions & optimized simulation predictions is 0.75 dB, and the average optimization difference between optimized simulation predictions & optimized real measurements is 1.47 dB.

Table 3: Coverage Optimization Comparison

Locations (x, y)	Real-Time RSSI (dBm)		Simulation Predictions RSSI (dBm)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
19.5, -8.5	-71	-66	-74.48	-67.26
27.5, -8.5	-71	-68	-65.58	-72.22
31.5, -8.5	-74	-69	-69.93	-74.26
27.5, -5.5	-71	-68	-58.33	-67.01
9.5, -4.5	-76	-72	-71.23	-71.33
23.5, -1.5	-54	-51	-53.89	-50.93
27.5, -1.5	-53	-51	-49.91	-53.88
18.5, 0.5	-58	-53	-53.98	-52.09
23.5, 0.5	-42	-41	-47.77	-44.13
18.5, 1.5	-58	-53	-54.66	-52.09
23.5, 1.5	-42	-41	-47.23	-44.13
0.5, 3.5	-85	-80	-85.3	-78.98
4.5, 3.5	-84	-77	-80.15	-70.94
19.5, 3.5	-63	-56	-59.04	-55.91
26.5, 3.5	-51	-47	-49.84	-53.11
4.5, 6.5	-80	-79	-84.94	-80.64
39.5, 9.5	-78	-72	-77.37	-71.47
47.5, 9.5	-82	-77	-80.14	-82.73
35.5,15.5	-77	-75	-73.79	-80.62
42.5,15.5	-82	-77	-79.87	-78.65

Table I: Capacity Optimization Comparison {Per User Data Rate (Downlink)}

Locations (x, y)	Real-Time Data Rate (Mbps)		Simulation Predictions Data Rate (Mbps)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
2.7, 4.5	0.5	1	0.08	0.61
16.2, 7.5	4.3	5.6	0.08	0.93
24, 7.4	4.5	11.5	0.08	0.61
31, 7.6	2.6	2.6	0.08	0.93
39, 9.3	1.2	1.2	0.08	0.93
46, 6	1.4	4.9	0.08	0.81
9, 1	0.8	10.3	0.08	1.03
16.5, 1	1	5.8	0.08	1.03
23.6, 1	2.6	10.3	0.08	1.03
29.7, 1	1.3	6.3	0.08	1.03
36.7, 1	1.3	5.6	0.08	1.03
42, -3	1.4	1.2	0.08	0.61
45.8, -3	1.8	1	0.08	0.61
50.6, -3	2.4	1.6	0.08	0.61
9, -4.9	4.3	1.2	0.08	0.81
16.5, -4.9	11.8	4.5	0.08	0.81
28, -2.5	1.5	11.7	0.08	0.81
29, -7	1	5.3	0.08	0.93
36.5, -5	1.2	3.4	0.08	0.61

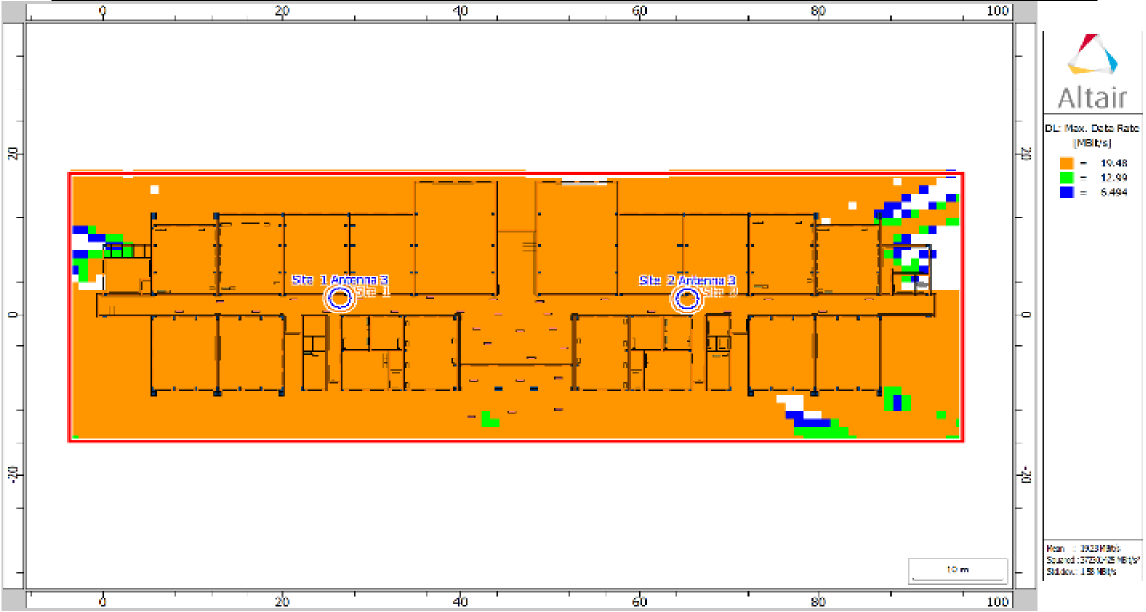


Figure 3: Downlink Data Rate Before Optimization

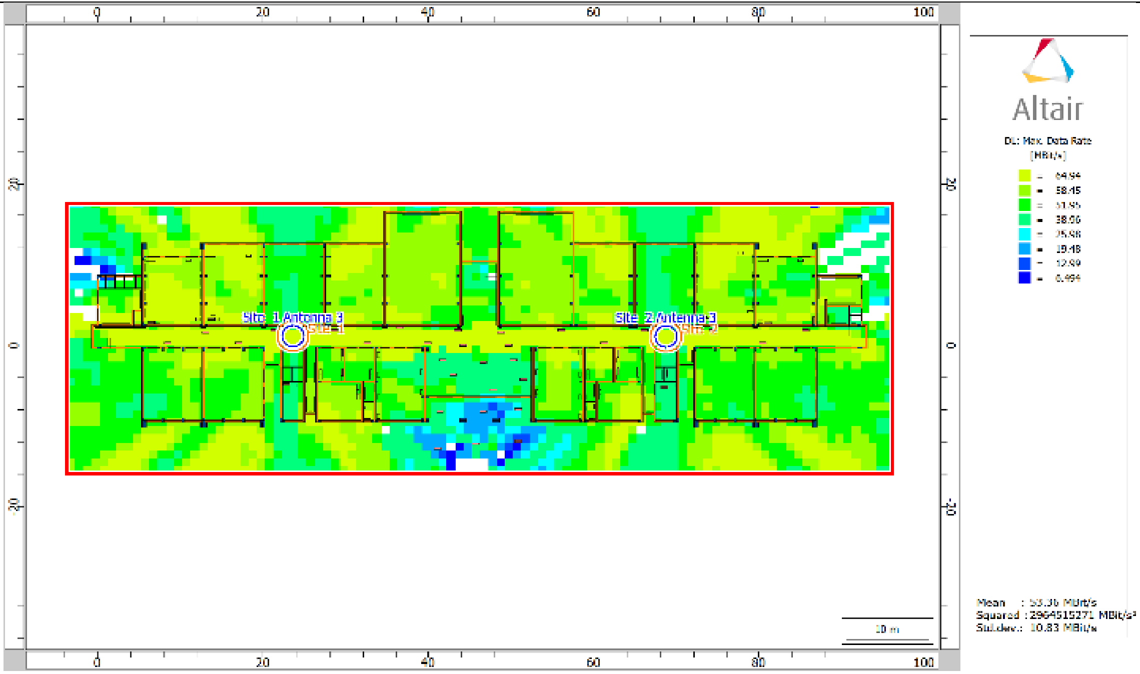


Figure 4: Downlink Data Rate After Optimization

In Table 5,19 location points are taken for comparison between real-time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 5 & Figure 6 respectively the average optimization difference between non-optimized real & optimized real measurements is 3.42 Mbps, the average optimization difference between non-optimized simulation & optimized simulation predictions is 0.16 Mbps and the average optimization difference between optimized simulation predictions & optimized real measurements is 7.55 Mbps.

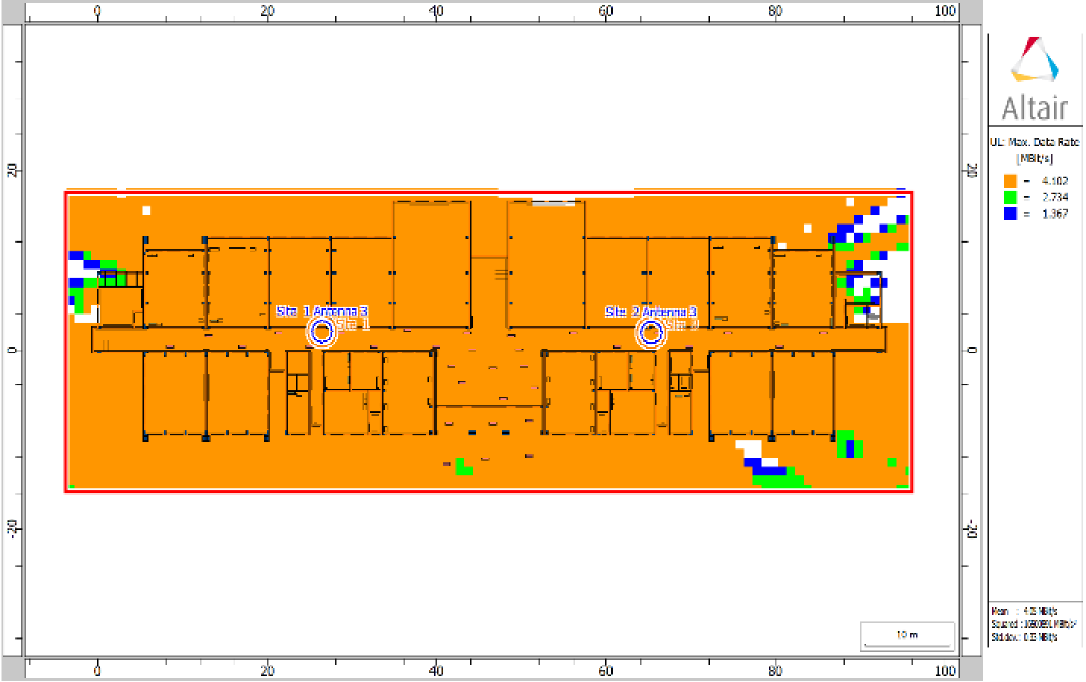


Figure 5: Uplink Data Rate Before Optimization

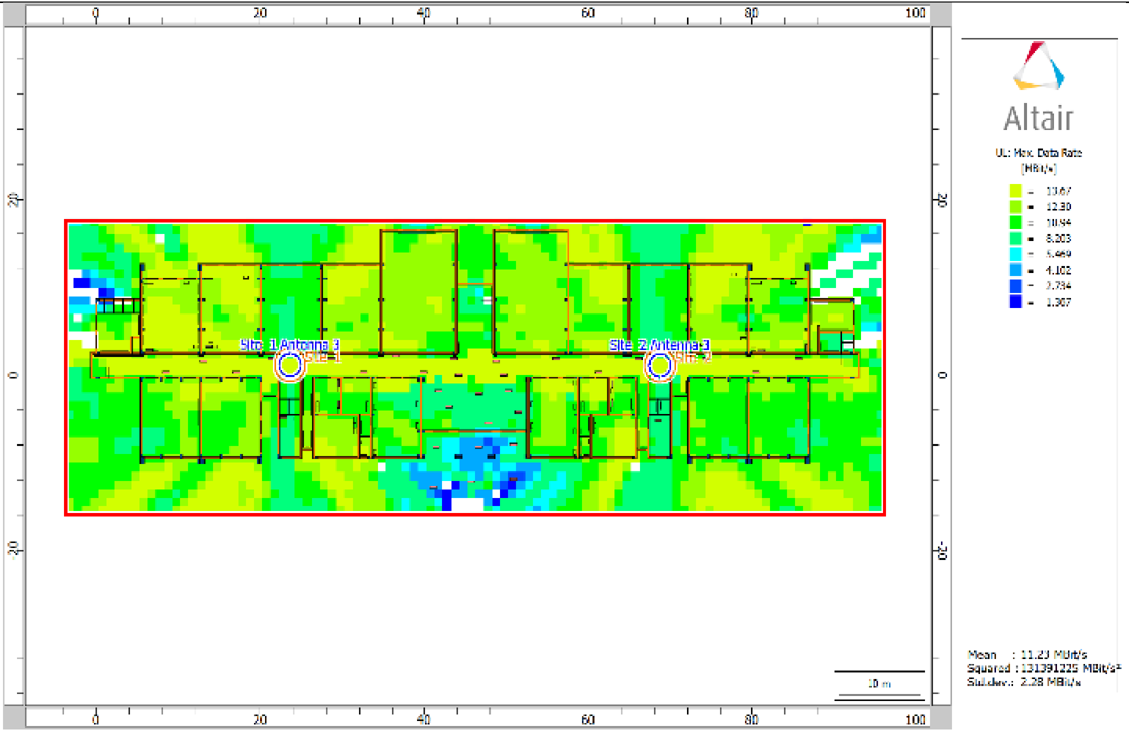


Figure 6: Uplink Data Rate After Optimization

Table 5. Capacity Optimization Comparison {Per User Data Rate (Uplink)}

Locations (x, y)	Real-Time Data Rate (Mbps)		Simulation Predictions Data Rate (Mbps)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
2.7, 4.5	0.5	0.2	0.02	0.13
16.2, 7.5	5	4.7	0.02	0.2
24, 7.4	7	12.9	0.02	0.13
31, 7.6	5.8	10.9	0.02	0.2
39, 9.3	1.1	9	0.02	0.2
46, 6	3	3	0.02	0.17
9, 1	6.4	10.3	0.02	0.22
16.5, 1	8	2.2	0.02	0.22
23.6, 1	9.4	11.8	0.02	0.22
29.7, 1	5.8	11.2	0.02	0.22
36.7, 1	4.5	8.4	0.02	0.22
42, -3	2.2	9.9	0.02	0.13
45.8, -3	2.2	6.1	0.02	0.13
50.6, -3	4.8	3.3	0.02	0.13
9, -4.9	1	6.8	0.02	0.17
16.5, -4.9	4.2	5.8	0.02	0.17
28, -2.5	4.4	11	0.02	0.17
29, -7	3.6	10.6	0.02	0.2
36.5, -5	2.9	8.7	0.02	0.13

The maximum logical user handling capacity of each AP is 254 users because they use Class C IPv4 addressing whose range is 2^8 = 256 addresses in which the first address is allocated to the AP itself and the last one is a subnet mask, so 256 – 2 = 254. This setup of IP addressing is assigned by the network administration and the research for optimization is carried out within these bounds. If a Class B address is assigned, then capacity planning should be done to that setup accordingly. Practically, the user handling capacity of an AP is further limited by the physical resources but as the number of users under specific MCS decreases, the per-user data rate increases and vice versa.

Conclusion:

This research has achieved coverage and capacity optimization in terms of RSSI, data rate, SNIR, I+N & user capacity in the already deployed Wi-Fi

network. The number of APs in the testbed is decreased from 7 to 2. The optimization is performed in Win Prop's Pro Man through simulations based on location and antenna polarization/ physical orientation variation of the APs in the testbed. Then the most optimum case with respect to coverage and capacity is selected within the dataset of all possible cases (73) and applied in a real-time environment. The real-time measurements are taken with respect to coverage and capacity (validation). In the calculation of the per-user data rate (simulation), the total number of users is divided by the total number of MCS operated in the testbed. In this case, every MCS serves an equal number of users. So, users within the same MCS get the same data rate in downlink and uplink. But in reality, more users come under the same MCS, less data rate each user will get, and vice versa.

The future work consists of Monte Carlo Simulation (location-dependent traffic analysis), Prediction analysis (Delay Spread, Angular Spread & Angular Means), Electromagnetic Compatibility (EMC) analysis, Consideration of Mobile Station properties (propagation & channel properties), MIMO analysis in uplink & downlink, designing accurate pattern of AP antennas in AMan, implementation and analysis of IEEE 802.11 a/ac/ax technologies for higher capacity and RSSI calculations for worst case scenario.

References:

[1] "Cisco WAP4410N Wireless-N Access Point - PoE/Advanced Security - Retirement Notification - Cisco." Accessed: May 04, 2024. [Online]. Available: https://www.cisco.com/c/en/us/obsolete/wireless/cisco-wap4410n-wireless-n-access-point-poe-advanced-security.html

[2] A. S. Haron, Z. Mansor, I. Ahmad, and S. M. M. Maharum, "The Performance of 2.4GHz and 5GHz Wi-Fi Router Placement for Signal Strength Optimization Using Altair WinProp," 2021 IEEE 7th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2021, pp. 25–29, Aug. 2021, doi: 10.1109/ICSIMA50015.2021.9526299.

[3] P. P. and M. K. S. ZVANOVEC, "Wireless LAN Networks Design: Site Survey or Propagation Modeling?," RADIOENGINEERING, vol. 12, 2003.

[4] T. Honda, M. Ikeda, and L. Barolli, "Performance analysis of user connectivity by optimizing placement of wireless access points," Proc. - 16th Int. Conf. Network-Based Inf. Syst. NBiS 2013, pp. 488–493, 2013, doi: 10.1109/NBIS.2013.81.

[5] T. Witono and Y. Dicky, "Optimization of WLAN deployment on classrooms environment using site survey," Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. ICTS 2017, vol. 2018-January, pp. 165–168, Jan. 2018, doi: 10.1109/ICTS.2017.8265664.

[6] H. U. and A. U. K. U. Mir, O. U. Sabir, "WLAN configuration and location optimization on RSSI basis," Univ. Eng. Technol. Mardan, 2017.

[7] J. Tan, X. Fan, S. Wang, and Y. Ren, "Optimization-Based Wi-Fi Radio Map Construction for Indoor Positioning Using Only Smart Phones," Sensors 2018, Vol. 18, Page 3095, vol. 18, no. 9, p. 3095, Sep. 2018, doi: 10.3390/S18093095.

[8] A. Srivastava, R. Vatti, V. Deshpande, J. Patil, and O. Nikte, "Coverage Improvement of IEEE 802.11n Based Campus Wide Wireless LANs," 2018 Int. Conf. Adv. Commun. Comput. Technol. ICACCT 2018, pp. 126–129, Nov. 2018, doi: 10.1109/ICACCT.2018.8529625.

[9] Y. Tian, B. Huang, B. Jia, and L. Zhao, "Optimizing AP and Beacon Placement in WiFi and BLE hybrid localization," J. Netw. Comput. Appl., vol. 164, p. 102673, Aug. 2020, doi: 10.1016/J.JNCA.2020.102673.

[10] N. A. M. Maung and W. Zaw, “Comparative Study of RSS-based Indoor Positioning Techniques on Two Different Wi-Fi Frequency Bands,” 17th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. ECTI-CON 2020, pp. 185–188, Jun. 2020, doi: 10.1109/ECTI-CON49241.2020.9158211.

[11] M. Rawaa Akram, A. H. Al-Nakkash, O. N. M. Salim, and A. A. S. Alabdullah, “Proposed APs Distribution Optimization Algorithm: Indoor Coverage Solution,” J. Phys. Conf. Ser., vol. 1804, no. 1, p. 012134, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012134.

[12] O. S. Naif and I. J. Mohammed, “Wireless Optimization Algorithm for Multi-floor AP deployment using binary particle swarm optimization (BPSO),” J. Phys. Conf. Ser., vol. 1963, no. 1, p. 012028, Jul. 2021, doi: 10.1088/1742-6596/1963/1/012028.

[13] S. Baua and S. Karuppuswami, “WiFi coverage planning and router position optimization using machine learning,” 2022 IEEE Int. Symp. Antennas Propag. Usn. Radio Sci. Meet. AP-S/URSI 2022 - Proc., pp. 689–690, 2022, doi: 10.1109/AP-S/USNC-URSI47032.2022.9887259.

[14] I. Bridova and M. Moravcik, “A System Approach in a WiFi Network Design,” Conf. Open Innov. Assoc. Fruct, vol. 2023-May, pp. 15–20, 2023, doi: 10.23919/FRUCT58615.2023.10142994.

Game Brains: NPCs Intelligence Using Neural Network Brains

Mosaddiq Billah[1], Aanoora Seher, Ahmad Bahar, Muniba Ashfaq, Abdullah Hamid
Computer Systems Engineer (UET Peshawar)
***Correspondence**:20pwcse1863@uetpeshawar.edu.pk

Citation | Billah. M, Seher. A, Bahar. A, Ashfaq. M, Hamid. A, "Game Brains: NPCs Intelligence Using Neural Network Brains", IJIST, Special Issue. pp 74-82, May 2024

Received| May 04, 2024, **Revised|** May 09, 2024, **Accepted|** May 16, 2024, **Published|** May 21, 2024.

This paper aims to develop the foundational knowledge about the Unity game development engine embedded with AI for the development of a hyper-casual game that has intelligent NPCs, which operate strategically in the environment. The targeted audience comes in the class of those who are pursuing their career in the niche of AI game development and enhancing the gaming experience for single-player game users. Using Unity Engine and Python, Curriculum learning and self-learning experiments were conducted to test the AI game. Moreover, in this paper, different reinforcement learning methods have been discussed, which have been implemented in the game that produces the optimal results for the behavior of NPCs. Hence, this paper tends to represent a glimpse into the future perspective of the gaming industry in hyper-casual gaming platforms.

Keywords: Unity; Reinforcement Learning; AI Games; NPCs, and Agents.

Introduction:

In traditional games, the behavior of NPCs (Non-Player Characters) follows a scripted, generic standard flow, leading them to perform actions in a predictable manner. Consequently, users disengage from such games after a while since they anticipate the patterns of the NPCs' movements. However, this paper tends to introduce AI to the platforms of hyper-casual games, since AI is mostly found in high-level games such as Red Dead Redemption 2. Our paper aims to develop a hypercausal game, which is a lightweight game with fewer mechanics that would have intelligent NPCs, using Unity Engine.

Unity is a cross-platform game engine, used for 2D and 3D games. It supports a variety of platforms such as desktops and mobiles. Furthermore, Unity is one of the most popular engines among the other game development engines due to its flexibility, efficiency, and convenience. The gaming engine comprises various tools that are convenient tools for modifying your project. The feature of real-time play mode with smart previews allows you to monitor the modifications instantly [1]. Unity engine supports the deployment of its projects on different operating systems such as Mac, Linux, and Windows, along with artist-friendly tools by allowing developers to develop efficient games [2]. Additionally, the navigation system in Unity enables NPCs to logically move around the environment of the game [3]. However, this feature only limits the movement of the NPCs or agents to make decisions in terms of movement around obstacles in the environment. On the scripting aspect of the Unity engine, it supports C#, which is an object-oriented programming language developed by Microsoft. There are two ways to design C# scripts in the Unity engine. The first one is object-oriented design, which is referred to as the traditional approach and is used by the majority of developers. The second is data-oriented design, which is also supported by Unity [4]. AI games have been on the rise since 1960 when games such as "SpaceWars" introduced AI in the field of game development. In contemporary times, games like Red Dead Redemption 2 and Grand Theft Auto 6 have been found to be popular due to their AI-trained NPCs.

The paper has been divided into four sections. Section 1 elaborates on the introduction, where the problem statement, literature review, recent advancements, and objectives of the research have been elaborated. Section 2 covers the methodology and experiments implemented for the research. Section 3 covers the Results and Discussion to present the result of the RL algorithm and discuss the objectives of the research. Section 4 covers the conclusion.

Research Paper Objectives:

- To study Unity and AI for the development of innovative AI hypercasual games.
- To examine the best optimal Reinforcement Learning methods for the agents.
- To outline the integration of AI with Unity

This paper tends to represent a glimpse into the future perspective of the gaming industry in hyper-casual gaming platforms.

Methodology:

The following methodology is adopted in this paper in which data is gathered from different sources, which relates to the domain of this paper. The collection of data for this work has been gathered from the following tools:

- Unity Engine
- C#
- Python (Anaconda Software)

Training Environment:

The environment was created for the hyper-casual game to test the reinforcement methods. The environment is based on a football game where two teams play against each other by scoring the ball into the located goals. The important feature used in the game is "ray cast" which collects data. Ray cast is used to detect the object from a distance by the agent to make intelligent decisions based on the values collected. The agent receives the reward after completing their desired objective of scoring a goal.

Reinforcement Learning Methods:

Multiple RL methods exist in this case for the purpose of training these agents, whereas the following RL methods were implemented in order to attain the aim of leading it to an AI game where the agents or NPCs operate intelligently. Furthermore, the training of the neural network was implemented parallel to enhance the training process.

The following methods were utilized for the agents:

- Proximal Policy Optimization (PPO)
- Random network distillation
- Behavior Cloning (BC)
- Curiosity-driven exploration by self-supervised prediction
- Generative Adversarial Imitation Learning (GAIL)

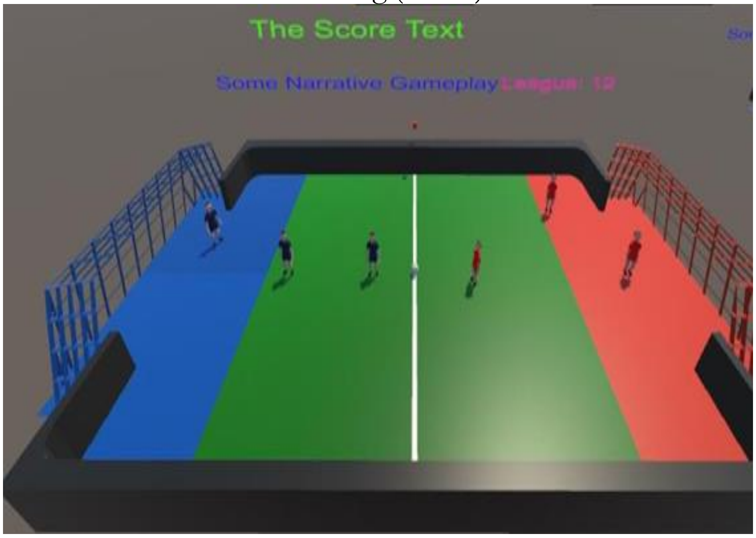


Figure 1: Training Environment (Soccer field)

Collecting Training Data:

Statistics were collected from the feature of ray-cast, which is a component in Unity that detects objects from a distance when it is implemented on an object. It retains information about an object through an invisible laser, which is visible in the editor mode of Unity Engine.

Agent's Behavior Parameters:

In this game, the NPCs or agents were assigned various ray-cast lasers for detecting various objects. Agent or NPCs, for observation space, have been assigned with 11 ray-casts forward and 3 ray-casts backward in order to collect data for the training model in order to make intelligent decisions through reinforcement learning.

Experiments:

- **Curriculum Learning:**

In this experiment, the AI agents are trained using a curriculum approach, where they start by learning from simpler scenarios and gradually progress to more complex ones. Initially, the AI agents may compete against each other in basic soccer scenarios, with the difficulty level increasing over time. In general, it is a player versus an AI agent.

- **Self-Learning:**

This experiment focuses on training AI agents to play soccer against each other. All players in the game are controlled by AI algorithms, which learn from their interactions and experiences within the simulated environment. In general, it is an AI agent versus an AI agent.

Result and Discussion:

In this section of the proposed paper, the research objectives have been discussed. Clear themes have been identified to assist in discussing the research goals effectively in order to attain the main aim of the proposed paper.

To Study Unity and AI for the Development of Innovative AI Hyper Casual Games:

Unity 3D offers several features, which has not been discussed yet. Unity supports asset tracking, scripting, processing, and physics which eases the development time for developers for their projects. According to a paper [5], the Unity engine supports 27 diverse platforms and devices in an environment that is user-friendly development. Regarding AI in the game industry, it is a broad topic. AI comprises machine learning and decision-making abilities specifically for the behavior of agents in the game. It is a significant aspect of contemporary game development. Developing an AI game can have an immersive impact on gameplay [6].

The first AI game, called the "Spacewar!", was developed at MIT in the early 1960s [6]. It was a multiplayer space combat genre game that had AI-controlled opponents. Nevertheless, AI games evolved over the years since then for the purpose of enhancing the gaming experience. During those days, PCs did not exist as large mainframes or computers were mostly found in institutions. In recent times, large gaming organizations have started to implement AI into their games such as Red Dead Redemption 2, which is a popular game. But it must be taken into notice that such games are mostly high-level and require specific consoles or upgraded PCs.

In this case, hypercasual games, such as Candy Crush game or Subway Surfers are known for their simple features and mechanics, convenient gameplay, and minimum design, enabling them to be accessible to a broad audience of the gaming genre. To convey AI games to a broad audience of such genres, hypercasual games are mostly played by people. By introducing AI to the platform of such games, it can uplift the gaming industry to a greater extent.

To Examine the Best Optimal Reinforcement Learning Methods for the Agents:

According to a paper, there are various training tools for the evaluation of reinforcement learning algorithms [1]. There is a reinforcement learning algorithm known as the NEAT or Neuro Evolution of Augmenting Topologies. NEAT has various applications in several domains, including game development. However, the shortcoming of NEAT, which is compared to the reinforcement learning methods implemented in this paper, is that it does not inherit a memory system. It focuses on evolving the structure and linkage of neural networks instead of having a mechanism for remembering information over the period. However, in this paper as mentioned above, the following reinforcement learning methods have been utilized for the training agents in the game environment:

Proximal Policy Optimization (PPO):

The default core method is the Proximal Policy Optimization (PPO) reinforcement algorithm, which was first presented by [7]. By limiting policy updates to a limited range of $[1-\varepsilon, 1+\varepsilon]$, PPO is renowned for its stable and user-friendly architecture. This prevents huge policy changes from gradient ascent, which can be damaging. Deviations of rt (θ) from 1 in policy changes are penalized by the objective function. Mathematically, the function of PPO is represented by;

$\mathrm{LCLIP}\ (\theta) = \hat{\mathrm{E}}\mathrm{t}\ [(rt\ (\theta)\hat{\mathrm{A}}\mathrm{t},\ ,\ clip(\mathrm{rt}\ (\theta), 1 -\varepsilon, 1+\varepsilon)\ \hat{\mathrm{A}}\mathrm{t}]$

- ε is a hyperparameter, (usually 0.1 or 0.2).
- Êt is the expectation over timesteps.
- rt is the probability ratio of new and old policies.
- Ât is the estimated advantage at time t.
- θ is the policy parameter.

The Proximal Policy Optimization (PPO) reinforcement algorithm is a stable and user-friendly method that is ideal for training agents in-game environments since it makes use of the clipped objective function.

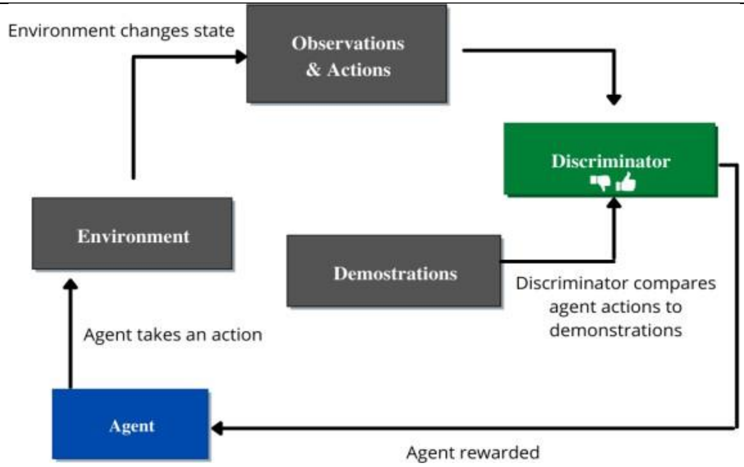


Figure 2: Generative Adversarial Imitation Learning

Generative Adversarial Imitation Learning (GAIL):

Among model-free imitation algorithms, Generative Adversarial Imitation Learning (GAIL) is unique in that it performs better than previous model-free techniques, particularly in complex contexts [8]. This approach depends more on learning from the environment and requires more contact with it than model-based approaches because it is model-free. The method is ad hoc action exploration to find out which activities lead to a policy shift in favor of the expert's policy.

With GAIL, agents can learn behavior without explicitly knowing the reinforcement signal by watching the expert's activities. The expert records and saves its actions in a demonstration file with associated states and actions. With this framework, behavior may be directly extracted from the given example.

Curiosity-Driven Exploration by Self-Supervised Prediction:

According to [9], the reports have observed that in real-world settings, rewards are frequently insufficient or nonexistent. In such situations, agents might be encouraged to explore their surroundings and learn new abilities that are necessary to accomplish their goals by using their curiosity as a motivator. Here, the agent's curiosity is defined as its ability to predict the results of its actions.

According to [9], the reports have explained that curiosity-driven exploration encourages agents to explore more effectively, lowering the requirement for significant contact with the environment to accomplish objectives. This method also allows the agent to forecast how its actions will turn out in scenarios that haven't been witnessed before. One notable benefit of curiosity-driven exploration is its capacity to motivate agents to discover new places in games.

Behavior Cloning (BC):

The need to accurately mimic human behavior in intelligent entities is addressed by behavior cloning, a concept Hussein [10] introduced. This method allows behavior-cloned agents to mimic human behavior in similar circumstances. Autonomous vehicles, helper robots, and computer-human communication are just a few of the domains in which it finds use. As a basic aspect of behavior cloning, imitation learning depends on insights from the environment and the agent's actions.

Imitation learning has intrinsic limitations, despite its usefulness. It is contingent upon the availability of expert demonstrations of a high caliber, and the agent's competence is limited to what people can demonstrate. Cloning behavior is very useful for agents that need to closely mimic the examples given. One disadvantage of such agents is that they are not able to improve when choosing what to do.

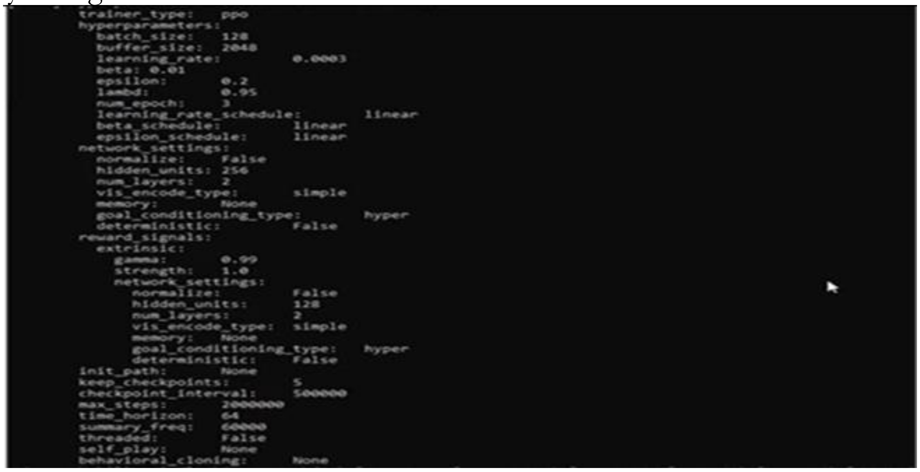
Behavior cloning is a useful method for giving agents human-like behavior, and it has applications in many different fields. Although it works well for faithful imitation, it has limitations in terms of the caliber of expert demonstrations and the incapacity of agents to perform at levels higher than those of humans.

Random Network Distillation (RND):

Burda [11] introduced Random Network Distillation, which modifies the reinforcement learning techniques by adding an exploration incentive. The bonus is ascertained by calculating the prediction error of a neural network (NN) feature acquired from observations using a fixed, randomly initialized NN. This strategy serves as a directed exploration tool with the primary goal of addressing the problem of scant rewards. Essentially, the agent is rewarded for discovering new things when interacting with the environment or for venturing into locations that haven't been explored before.

Setting itself apart from a lot of curiosity-based approaches, the RND methodology shows persistence in the face of obstacles like becoming stuck when subjected to random noise, like static on a TV screen. The method's dependence on an exploration bonus instead of an absolute prediction error accounts for this robustness. As a result, RND shows great promise as an algorithm, especially for agents whose job it is to navigate complicated environments where the observation data obtained is highly noisy.

In reinforcement learning, Random Network Distillation presents a fresh viewpoint on exploration and efficiently handles situations with plenty of noise and scant rewards. Because of its unique methodology, it is positioned as a viable algorithm for agents exploring complex and noisy scenarios. With the training of each algorithm, PPO gave a positive response with higher accuracy in terms of reinforcement learning, where the agents strategically operated throughout the game. Other algorithms gave a response, while PPO turned out to have a higher accuracy of 95% as compared to the other RL algorithms, which were less in terms of their accuracy and response by the agents in the environment.



```
  trainer_type:   ppo
  hyperparameters:
    batch_size:   128
    buffer_size:  2048
    learning_rate:        0.0003
    beta: 0.01
    epsilon:      0.2
    lambd:        0.95
    num_epoch:    3
    learning_rate_schedule:       linear
    beta_schedule:        linear
    epsilon_schedule:     linear
  network_settings:
    normalize:    False
    hidden_units: 256
    num_layers:   2
    vis_encode_type:      simple
    memory:       None
    goal_conditioning_type:       hyper
    deterministic:        False
  reward_signals:
    extrinsic:
      gamma:      0.99
      strength:   1.0
      network_settings:
        normalize:        False
        hidden_units:     128
        num_layers:       2
        vis_encode_type:  simple
        memory:   None
        goal_conditioning_type:   hyper
        deterministic:    False
  init_path:      None
  keep_checkpoints:       5
  checkpoint_interval:    500000
  max_steps:      2000000
  time_horizon:   64
  summary_freq:   60000
  threaded:       False
  self_play:      None
  behavioral_cloning:     None
```

Figure 3: Training Behavior of PPO algorithm in the Soccer Game using Anaconda Prompt integrated with the Unity Engine's file project

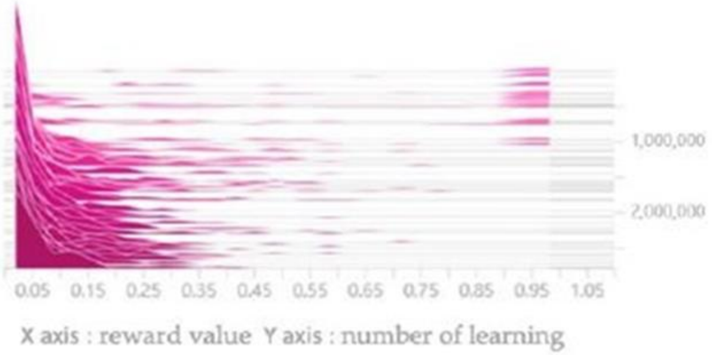


Figure 4: Cumulative-reward-value graph using the PPO algorithm

To examine the Integration of AI with Unity:

Unity cross-platform has the feature of supporting integration with other applications. Although, it has a library of AI that supports the Navigation Mesh, as discussed in the above section, which is for agents to move intelligently in the environment by avoiding obstacles. Unity contains the API that allows the integration of Python with the agents in the environment by assigning them the algorithm depending upon the purpose.

The Unity Machine Learning API package in Unity enables the developers to integrate Python with Unity Editor in order to train the behaviors of agents (See Figure Below) **[14]**. The API package contains the following three components. The components are agent, python API, and trainer. The agent is the character model in the game to which the model has been assigned. It communicates the training model through the Python API via a communicator in Unity. Hence, the agent utilizes the communicator in order to connect with the trainers through the Python API.

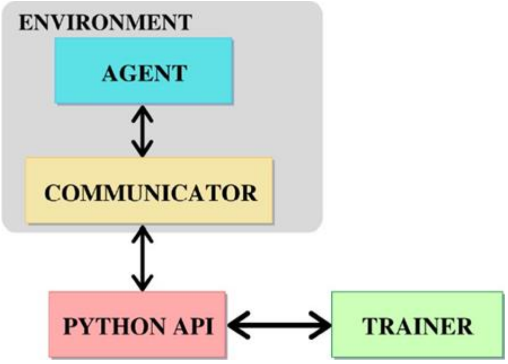


Figure 5: API Framework Model:

Conclusion:

To conclude, the paper seeks to enhance the gaming experience for users by using hyper-casual game platforms with intelligent NPCs or agents. With the discussed methodology and discussion, the algorithm utilized for the game is PPO due to its stability and policy improvement. Furthermore, it can be observed that introducing AI to hyper-casual games will not only uplift the game industry but encourage other independent developers to enter the domain of game development due to its modern perks of introducing AI with optimal reinforcement algorithms. Three objectives were formulated to attain these goals such as to study Unity and AI for the development of innovative AI hyper-casual games, to examine the best optimal Reinforcement Learning methods for the agents, and to outline the integration of AI with Unity. Associated with these objectives, three questions were also formulated for the same purpose of attaining these goals what are the benefits of AI games? To what extent AI games can shape the gaming industry? Which Reinforcement Learning methods are optimal for the game agents to operate intelligently in the game?

Hence, AI games are the future of the gaming industry and it has been rising since the 1960s with the development of the "Space Wars!" game to contemporary games like Red Dead Redemption 2. It will start appearing as a common feature in every game in the future.

Acknowledgement:

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have this all along the completion of our project. We want to convey our sincere gratitude and appreciation to our supervisor, Dr. Muniba Ashfaq, and our advisor, Engineer Abdullah Hamid, for the significant assistance and support during this effort. Finally, we want to thank our parents and professors for helping and motivating us with our projects and academic endeavors. We acknowledge our University, University of Engineering and Technology Peshawar, for providing us with a feasible environment in order to work on our project.

Author's Contribution:

Mosaddiq Billah:

- **Conceptualization:** Developed the initial idea for creating a soccer theme- based hypercasual game.
- **Writing:** Drafted the initial version of the project proposal and technical documentation outlining the game's architecture and development strategy.
- **Literature Review:** Conducted comprehensive review of literature on AI gaming and Unity development.

Aanoora Seher:

- **Data Collection:** Assisted in gathering relevant datasets and resources necessary for training and testing the AI algorithms.
- **Data Analysis:** Performed statistical analysis on game data to identify patterns, trends, and performance metrics for evaluating the AI's effectiveness.
- **Data Interpretation:** Analyzed experimental results to evaluate the AI's behavior, identify strengths and weaknesses, and propose improvements for enhancing gameplay.

Ahmad Bahar:

- **Unity Development:** Developed the initial prototype of the game and implemented the script logic for the agents and game objects.
- **Graphic Design:** Designed the User Interface of the game and the training environment for the agents in the game.

Muniba Ashfaq:

- **Supervision:** Supervised the entire research project, providing guidance and feedback on project milestones, research methodology, and documentation.

Abdullah Hamid:

- **Assistance:** Provided hands-on assistance throughout the project, offering technical support and troubleshooting for various aspects of game development.
- **Mentorship:** Offered mentorship by sharing knowledge of advanced concepts and features applicable to the game development process, guiding the team in implementing innovative ideas and best practices.

Conflict of Interest:

The authors declare that there is no conflict of interest regarding the publication of this paper. We have no financial or personal association with organizations or individuals that could influence the objectivity or interpretation of the project findings. This research was conducted with academic integrity and with the objective of achieving the primary goals of the research.

Project Details:

This research was conducted as part of the final year project for the Bachelor of Science in Computer Systems Engineering program at the University of Engineering and Technology Peshawar. The project was completed during the academic year of 2023-2024. The project did not incur any cost as the resources were utilized by the university, including software tools and guidance from faculty mentors. The completion date for the project was May 1st, 2024.

References:

[1] A. Juliani et al., "Unity: A General Platform for Intelligent Agents," Sep. 2018, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1809.02627v2

[2] A. Canossa, "Interview with Nicholas Francis and Thomas Hagen from Unity Technologies," Game Anal., pp. 137–142, 2013, doi: 10.1007/978-1-4471-4769-5_8.

[3] C. Becker-Asano, F. Ruzzoli, C. Hölscher, and B. Nebel, "A Multi-agent System based on Unity 4 for Virtual Perception and Wayfinding," Transp. Res. Procedia, vol. 2, pp. 452–455, Jan. 2014, doi: 10.1016/J.TRPRO.2014.09.059.

[4] T. N. Malete, K. Moruti, T. S. Thapelo, and R. S. Jamisola, "EEG-based Control of a 3D Game Using 14-channel Emotiv Epoc+," Proc. IEEE 2019 9th Int. Conf. Cybern. Intell. Syst. Robot. Autom. Mechatronics, CIS RAM 2019, pp. 463–468, Nov. 2019, doi: 10.1109/CIS-RAM47153.2019.9095807.

[5] "Using the Unity Game Engine to Develop SARGE: A Case Study." Accessed: May 06, 2024. [Online]. Available: https://www.researchgate.net/publication/265284198_Using_the_Unity_Game_Engine_to_Develop_SARGE_A_Case_Study

[6] J. Wexler, "Artificial Intelligence in Games: A look at the smarts behind Lionhead Studio's 'Black and White' and where it can and will go in the future".

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. K. Openai, "Proximal Policy Optimization Algorithms," Jul. 2017, Accessed: May 04, 2024. [Online]. Available: https://arxiv.org/abs/1707.06347v2

[8] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," Adv. Neural Inf. Process. Syst., pp. 4572–4580, Jun. 2016, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1606.03476v1

[9] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-Driven Exploration by Self-Supervised Prediction," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2017-July, pp. 488–489, Aug. 2017, doi: 10.1109/CVPRW.2017.70.

[10] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation Learning," ACM Comput. Surv., vol. 50, no. 2, Apr. 2017, doi: 10.1145/3054912.

[11] Y. Burda, H. Edwards, A. Storkey, and O. K. Openai, "Exploration by Random Network Distillation," 7th Int. Conf. Learn. Represent. ICLR 2019, Oct. 2018, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1810.12894v1

RESEARCH & INNOVATION DIVISION

IJIST

Machine Learning-Based Estimation of End Effector Position in Three-Dimension Robotic Workspace

Hamna Baig[1], Ejaz Ahmed[2], Ihtesham Jadoon[1],

[1]Department of Electrical and Computer Engineering COMSATS University Islamabad Attock Campus Attock, Pakistan.

[2]Department of Electrical Engineering Aero Space and Aviation, Air University Islamabad Kamra Campus Attock Pakistan)

*__Correspondence__: Ejaz Ahmed: 225458@aack.au.edu.pk

Citation | Baig. H, Ahmed. E, Jadoon. I, "Machine Learning-Based Estimation of End Effector Position in Three-Dimension Robotic Workspace", IJIST, Special Issue pp 83-92, May 2024

Received | May 04, 2024, **Revised |** May 12, 2024, **Accepted |** May 17, 2024, **Published |** May 21, 2024.

Introduction/Importance of Study: The Workspace is the area around the robot where a robot can freely move with possible input variations of different joint angles.

Novelty statement: Conventionally iterative simulation methods are used to find robotic workspace. Which are computationally slow and difficult to model. Our approach utilizes machine-learning algorithms to predict the workspace and position of an end effector.

Material and Method: Multiple Linear Regression (MLR), Decision-Tree Regression, and Artificial Neural Network (ANN) algorithms trained for prediction. The dataset, which is collected and used as train and test data, is further for the validation step.

Result and Discussion: By simulating the robot with the Denavit-Hartenberg (D-H) approach in MATLAB. The results findings show the accuracy of Machine learning algorithms specifically Artificial Neural Networks (ANN) perform better than conventional mathematical methods

Concluding Remarks: Artificial Neural Network (ANN) outperformed other machine learning methods.

Keywords: Robotics; Artificial Neural Network (ANN); Denavit-Hartenberg (D-H); Prediction and Machine Learning.

Acknowledgement:
Thank you to UET Peshawar for hosting the conference and considering this paper for conference ICTIS 2024.

Author's Contribution:
Ejaz Ahmed presented an idea and modeled the system also contributed to drafting the paper Hamna Baig made code for integrating robotics with Machine learning Ihtesham Jadoon works on paper drafting and results comparison.

Conflict of Interest:
No conflict of interest for publishing this manuscript in IJIST.

Project Details:
This work is done without any financial assistance from any source.

Introduction:

In many crucial cases, it is very difficult, if not impossible, to analytically predict the behavior of physical systems. The necessity to build a physical system prototype drives modeling, which in turn reveals strong motivations to investigate and analyze a system's operation. Modeling of robots is usually carried out through kinematics study. That deals with a model of the robot without any influence of force. The kinematics of a robot deals with the geometric and time-based properties under motion and in particular how various links of a robot move with respect to one another with variation of time. Which is the analytical way of explaining the relation between different joint variables. Kinematics modeling is divided into two types forward and inverse kinematics, The first one gives the position and orientation when joint angles and joint positions are known. While inverse kinematics deals with a set of complex equations computed by vector and analytical algebra to compute the joint angles once, the end effector position is given [1].

The robot is the heart of the automation industry used in manufacturing and assembly applications and many applications that require robots to move objects from one place to another using mechanically designed grippers these robots need to be precise in terms of placement of objects. To achieve the precise pick and place robot application it needs to be modeled with the least error. Intense research is carried out in the analytical modeling of robot systems, which are based on line and point transformation. Campos-Macías [2] proposed a geometric model to calculate and find the relation between input angles and output position unknown joint angles required for autonomous positioning of a robotic system. In [3] author Bayro-Corrochano uses a new algebraic method called quaternion for modeling different physical systems system. Popovic et al. [4] computed a method to model the upper extremity movement of the arm of a multi-leg moving robot inspired by an animal's movement. An analytical model-based approach to compute the kinematics of a humanoid robot was discussed in [5]. In [6] author presented an inverse kinematics model to calculate all the joint variables of a serial arm manipulator. Applications of machine learning in different robots are discussed in [7].

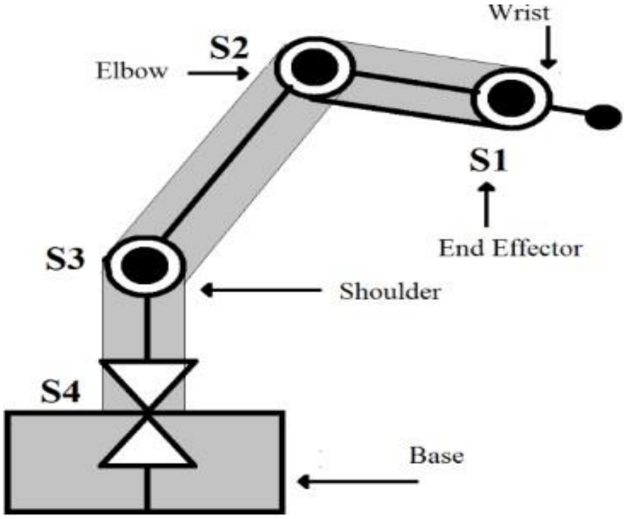


Figure 1: Kinematic Model Representation of 4-Degree-of-Freedom Robot.

Figure 1 shows the kinematic model's simplified view of the robotic arm in an inverted 'L' pose. The first joint S1 is used to move the arm claw to pick objects, and the joints S2 and S3 are the elbow and shoulder joint respectively to move the arm to the desired position. The S3 joint is the base joint to rotate the robot arm. The forward kinematic model of the 4 DOF Robot is presented in Section II Section III presents the discussion on forward kinematics using machine learning algorithms on MATLAB Section IV gives the result and Discussion of machine learning algorithms and Section V gives the final Conclusions.

The second last paragraph of the introduction section should explain the hierarchy/flow of research. The last paragraph should explain the objectives of the research and Novelty statement. This paper aims to focus on using machine-learning algorithms to estimate the end effector position in a three-dimensional robotic workspace. Specifically, the paper's objectives are as follows;

- **Dataset**: Generating dataset in MATLAB that contains values of joint angles (θ) and the corresponding end effector positions.
- **Model Training**: Using the dataset to train machine learning models MLR (Multiple Linear Regression), DTR (Decision-Tree Regression), and ANN (Artificial Neural Network) for estimating end effector position.
- **Prediction Analysis**: Using the values from the dataset to validate the trained models and access their RMSE (root Mean Squared Error) for prediction.

Model Comparison: Evaluate the trained MLR, DTR, and ANN models on the basis of computational time and RMSE

Material and Methods:

Forward Kinematics 4 Degree Robot:

The position and orientation of the end effector calculation of a robotic arm or mechanism based on the joint angles or displacement. The forward kinematics of a 4 Degree-of-Freedom robot usually has four rotational joints that can be calculated by imposing a sequence of transformations from the base frame to the end effector frame. A distinct coordinate system is introduced by every joint; the overall transformation of the end effector from the base can be calculated. This process can be implemented using a programming language such as Python and C++ in MATLAB in just a short code. The code will comprise parameters for each point for the Denavit-Hartenberg approach, resulting in transformation matrices and performing a multiplication process on matrices to calculate the overall transformation. Meanwhile, the implementation of the obtained transformation to the end effector's original location, the orientation, and the final position of the end effector in space can be calculated. This seemed to serve as the essential and indispensable tool for the robot's controlling and movement planning respectively.

Denavit-Hartenberg Parameters:

Denavit-Hartenberg parameters are employed commonly for the characterization of the robot’s structure. These parameters typically serve as the foundation for the conduction of robot kinematics and analysis. There are four DH parameters listed below in Table 1 to describe the orientation and position of a link. Each parameter for attaching reference frames to robots’ assembly link is linked to a specific convention. This standardization of coordinate frames across spatial linkages ensures consistency and facilitates analysis.

The Denavit-Hartenberg (D-H) decided to use the homogeneous transformation matrix instead. This matrix represents the end effector orientation and position of the robots with respect to the joint angles. Nonetheless, it does not specify the arm arrangement needed to reach this position. Figure 2 depicts the diagram of the link coordinate system which is created using DH parameters

Table 1: D.H Parameters Definition and symbols

Symbol	DH Parameters	Description	Symbol
$\boldsymbol{d_i}$	Joint Offset	Intersections Length of the joint axis to common normal	$\boldsymbol{d_i}$
$\boldsymbol{\theta_i}$	Joint Angles	Angle in between the normal plane to the joint axis and orthogonal projections	$\boldsymbol{\theta_i}$
L	Link Lengths	Axis to the common normal distance	L
$\boldsymbol{\alpha_{i-1}}$	Twist Angle	Orthogonal projections of joint axis on the normal plane to common plane Angle.	$\boldsymbol{\alpha_{i-1}}$

$$\boldsymbol{a_{i-1}} = \text{Translation } x_1 \; axis$$
$$\boldsymbol{a_{i-1}} = \text{Rotation around } x_i \; axis$$
$$\boldsymbol{d_i} = \text{Translation around } z_{i-1} \; axis$$
$$\boldsymbol{\theta_i} = \text{Rotation around } z_{i-1} \; axis$$

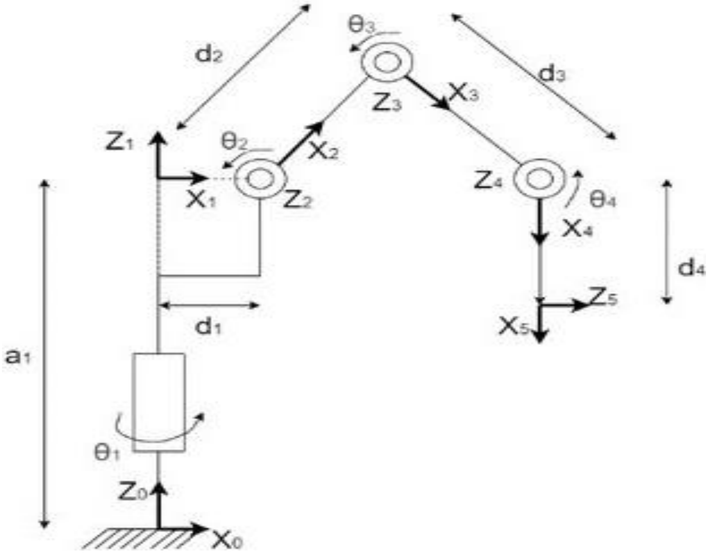


Figure 2: Denavit-Hartenberg Parameters Labeling of 4-Degree-of-Freedom Robot.

Denavit-Hartenberg (D-H) parameters are a collection of four parameters as depicted in Figure 2 for a 4 DOF robot that plays a crucial role in the robot kinematics. With the use of D-H representation, a systematic way to express relationships between consecutive links in a robotic serial manipulator is provided. Hence, this method provides a mathematical foundation, which is adaptable to a numerous robotic design. It also essentially defines the position and orientation of each link in relation to the other link that comes before this link [2].

Model of Forward Kinematics:

The 4-Degree-of-Freedom Forward kinematic model uses the Denavit-Hartenberg (DH) parameters in determining the end effector position and orientation based on joint angles for the robot. On an initial level, the DH parameters define each joint geometry including the link length, Joint angles, Joint offset, and Twist angles. With the help of these settings in place the transformation matrix for each joint explaining the transition between adjacent links as the robot moves is generated. A single transformation matrix indicating the total effect of each joint movement to the end effector frame from the base frame is generated via the multiplication of matrices. This transformation matrix provides a solution to the problems of forward kinematics by extracting the end effector orientation and location. The matrix for joint numbers 1 to ith can be calculated as shown in equation (1).

$$\boxed{A'_{i-1} = \begin{bmatrix} cos\,\theta & -sin\,\theta & 0 & 0 \\ sin\,\theta & cos\,\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\,\alpha & -sin\,\alpha & 0 \\ 0 & sin\,\alpha & cos\,\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$A'_{i-1} = \begin{bmatrix} cos\,\theta & -sin\,\theta . cos\,\alpha & sin\,\theta . sin\,\alpha & a. cos\,\alpha \\ sin\,\theta & cos\,\theta\,. cos\,\alpha & -cos\,\theta . sin\,\alpha & a. sin\,\theta \\ 0 & sin\,\alpha & cos\,\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

By multiplication of matrices as depicted in equation (2) gives the transformation matric.

$$^{0}T_4 = \, ^{0}A_1 \,. \, ^{1}A_2 \,. \, ^{2}A_3 . \, ^{3}A_4 \quad (2)$$

The model of the robot's forward kinematics was validated using MATLAB. A brief understanding of the kinematic behavior of the robot was obtained with the use of numerical analysis and also visualization in the MATLAB environment.

Table 2: D.H Parameters Description

Joint Angles θ	**Symbol**			
	a_{i-1}	d_i	a_{i-1}	θ_i
S1	0	0	0	0
S2	10	10	-90	0
S3	10	10	0	0
S4	10	10	0	0

For example, with a joint angle configuration of [S1 S2 S3 S4] for the values in Table 3, the transformation matrix is given below in Equation 3, and the visual representation is depicted in Figure 2

$$T = \begin{bmatrix} 1.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.4481 & -0.8940 & -13.3992 \\ -0.0000 & 0.8940 & -0.4481 & -7.9014 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \quad (3)$$

Results:

Problems related to machine learning can be classified into three main types; regression, classification, and clustering. In the context of predicting claw positions in mapping with non-linear input poses, the problem at hand falls under regression. An intelligent approach is employed, utilizing machine learning algorithms for obtaining forward kinematics solutions. This Section focuses on predicting the claw position of a 4 Degree-of-Freedom Robot using Multiple Linear Regression (MLR), Artificial Neural Network (ANN), and Decision-Tree Regression Techniques. The algorithms of machine learning are trained and implemented in the MATLAB environment. The performance of these three models is being evaluated on the root mean squared error and R squared value basis. The predicted result is then compared with the actual value of the claw position. The result is basically the end effector position on the basis of joint angles.

Dataset For Training Machine Learning Model:

The dataset is generated through the code in MATLAB through initializing arrays to store workspace points and link points. The code iterates for all the combinations of joint angles within the specified limits of theta range (θ) for calculating forward kinematics using DH parameters. For every combination, the end effector position (x, y, z) is calculated and both workspace and the link point are stored. Finally, the dataset is populated with joint angles (θ) and the corresponding end effector positions. Figure 3 shows the robot workspace with links and Table 3 shows the dataset with values of joint angles and the end effector position.

Table 3: Data set Values

	$\mathbf{S1(\theta_1)}$	$\mathbf{S2(\theta_2)}$	$\mathbf{S3(\theta_3)}$	$\mathbf{S4(\theta_4)}$	**X**	**Y**	**Z**
1	-1.5708	-1.5708	-1.5708	-1.5708	-0.6031	-1.5708	-1.5708
2	-1.5708	-1.5708	-1.5708	-1.2217	-2.3396	-1.5708	-1.5708
3	-1.5708	-1.5708	-1.5708	0.8727	-5.0000	-1.5708	-1.5708
4	-1.5708	-1.5708	-1.5708	-0.5236	-8.2635	-1.5708	-1.5708
5	-1.5708	-1.5708	-1.5708	-0.1745	-1.7363	-1.5708	-1.5708
6	-1.5708	-1.5708	-1.5708	-0.1745	-5.0000	-1.5708	-1.5708
7	-1.5708	-1.5708	-1.5708	0.5236	-7.6604	-1.5708	-1.5708
8	-1.5708	-1.5708	-1.5708	0.8727	-193969	-1.5708	-1.5708
10000	-2.3701	-3.3498	-1.2217	-3.7892	-3.8495	-4.6590	4.8908

Columns 1 to column 4 show the values of joint angles (S1, S2, S3, S4), and columns 5, 6, and 7 show the corresponding end effector position variable values (x, y, z). The dataset as shown in Table 3 has 10,000 values. This dataset is used later for the training of machine learning algorithms.

Machine Learning Algorithms:

For determining the forward kinematics the machine learning algorithms used are Multiple Linear Regression (MLR), Artificial Neural Network (ANN), and Decision-Tree (DT) Regression techniques. The following machine learning algorithms are trained and implemented in Matlab

Multiple Linear Regression (MLR):

The Multiple Linear Regression is defined as a straightforward regression technique for employing multiple variables in predicting the response or output variable. A connection between each joint angle and the end effector position variables is used in this technique. The equation general form for Multiple Linear Regression of multiple independent variables and single dependent variables is expressed as follows in equation (4).

$$y_{\,i} = b_{0i} + b_{1i}.x_{1i} + \cdots + b_{4i}.x_{4i} \qquad (4)$$

Here for our work the "yi" represents the estimated end effector position of the 4 DOF robotic claw and x1 to x4 denotes the four joint angles (θ) of the robot S1, S2, S3, and S4. The first joint S1 is of arm claw, and the joints S2 and S3 are the elbow and shoulder joints respectively to move the arm to the desired position. The S3 joint is the base joint to rotate the robot arm.

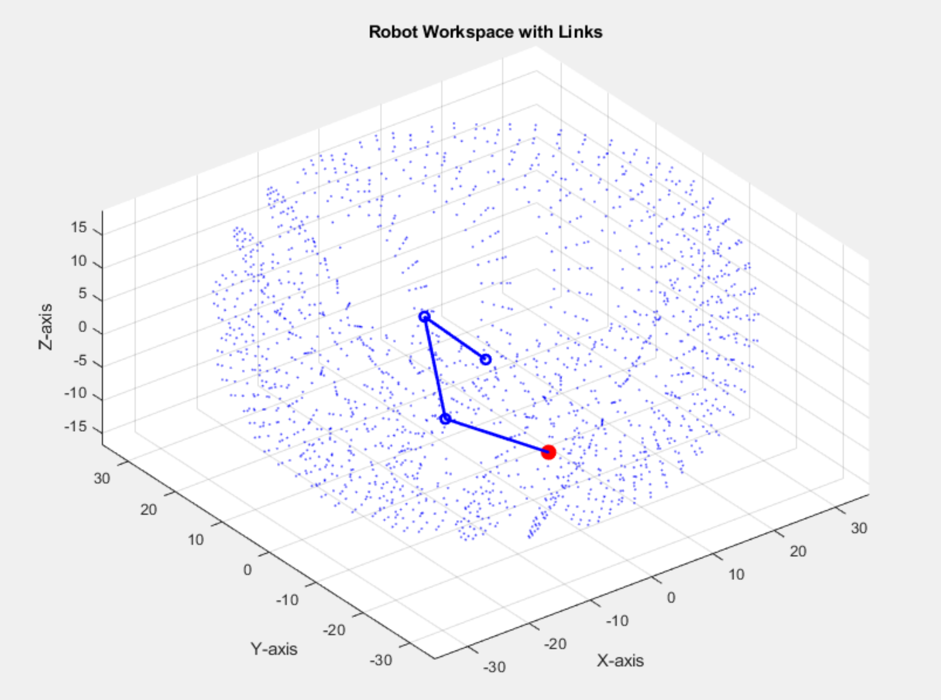


Figure 3: Workspace with Links demonstration on MATLAB.

Multiple Linear Regression (MLR):

The machine-learning algorithm is being used in predicting the target variable value by imposing the learning of simple decision rules inferred from the data features. This technique works by recursively partitioning the feature values (parent/root node) and then fitting a simple model specifically a constant value within each subset (Decision Node). Figure (4) illustrates the simple representation of Decision Tree Regression in general for this work with root node, Decision node, and leaf nodes. The decision tree regression predicts the continuous target variable by taking the average of the target values of all the training instances within each leaf node.

To illustrate how decision tree regression can be used to predict the end effector position (x, y, z) of a 4 DOF robot with 4 joint angles as input, we will train the Decision Tree Regression Model where the input features will be the joint angles $((\theta_1, \theta_2, \theta_3, \theta_4)$ and the output variable is the end effector position (x, y, z). The Decision Tree algorithm will learn to predict the position of the end effector based on each joint angle. After that the model performance will be evaluated on the root mean squared error (MSE) and R2 (R-squared error) basis. Hence in the work, the trained model can be used for the prediction. This prediction capability can be used in various robotics applications [8].

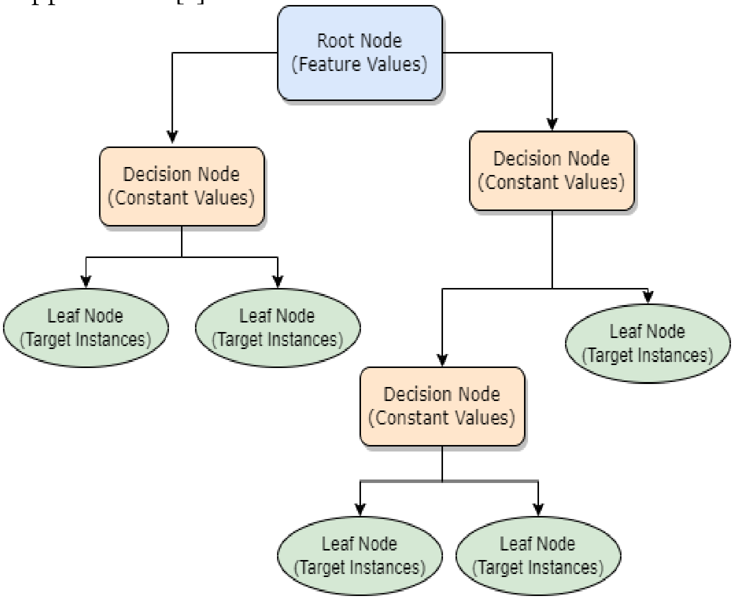


Figure 4: Decision Tree Algorithm 4-Degree-of-Freedom Robot.

Artificial Neural Network (ANN)

The ANN can be used to predict the end effector position of a 4-degree-of-freedom (DOF) robot based on 4 joint angles. We train the neural network using the training data. During the training of an ANN, the network adjusts its biases and weight iteratively to minimize the predicted end effector position and the actual positions in the training data. This is typically done by using the Gradient Descent algorithm on the backend of all the ML algorithms. Once the model is trained, we will evaluate its performance on the ground basis of RMSE (Root Mean Squared Error). The capability of ANN for capturing the complex and non-linear relationship between the inputs and the outputs makes it suitable for the prediction of the end effector position of a 4 DOF robot based on joint angles. Just by adjusting the architecture and training parameters of the neural network, the performance for specific predictions can be optimized. The ANN technique is widely used in robots. Figure 5 shows the artificial neural network topology used in this work.

During the ANN training, numerous parameters are adjusted, including hidden layer counts, neuron quality within each hidden layer, and the activation function choice applied at both the hidden and outer layers. The activation functions like sigmoid, tanh, Linear, and the (ReLU) Rectified Linear Unit are employed during the training duration. Moreover, optimization methods including the Stochastic Gradient Descent (SGD) and (Adam) adaptive moment estimation are utilized for refining the weights during the duration of training. The keen and careful selection of all these parameters leads to notable enhancement in prediction accuracy. The of epochs is set at 1,000 as a maximum number, as empirical evidence shows that the accuracy, as measured by metrics like (R2) R-squared and (RMSE) root mean square error, does not notably improve beyond this threshold. The result of training ANN in the MATLAB environment for our work is depicted in Table 4.

Table 4: ANN Model on MATLAB Platform

Data Division	Random
Performance	Mean Squared Error
Epoch	1000
Computational Time	0:00:13
Performance	0.000100
Gradient	0.266

Comparison and Discussion:

The dataset is compiled comprising of 10,000 entries for both inputs (joint angle values) and outputs (end predictor values) which were subsequently utilized for training of the Machine Learning Algorithms. The end effector position actual value is calculated through an analytical approach using the forward kinematics with DH parameters techniques. The predicted value is obtained from the machine learning algorithms implemented in the MATLAB environment. The performance of these algorithms is accessed by measuring the variance between the predicted and the actual value. However, the model's evaluation may vary depending on the random selection of samples within the training set, potentially resulting in either underestimation or overestimation.

The performance of every algorithm is evaluated based on RMSE. Table 4 shows the values of the end effector determined through (MLR) Multiple Linear Regression, (ANN) Artificial Neural Network (ANN) and Decision-Tree (DT) Regression. The actual value is also written. The values are found for the joint angle values. The visual representation of the actual and predicted end-effector value is shown in Figure (5). The values given for actual and predicted end effector position are calculated from joint angles given in equation 5.

$$[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4] = \left[0, \quad -\frac{pi}{2}, \quad -\frac{pi}{2}, \quad 0\right] \qquad (5)$$

The dataset is compiled comprising of 10,000 entries for both inputs (joint angle values) and outputs (end predictor values) which were subsequently utilized for the training of the Machine Learning Algorithms. The actual value of the end effector position is calculated through an analytical approach using the forward kinematics with DH parameters techniques.

Table 5: Actual And predicted positions

End Effector Position	X	Y	Z
Actual Value	26.841406	-10.000000	18.918461
Multiple Linear Regression (MLR)	17.735263	-3.969662	17.148879
Decision Tree Regression	28.214849	-12.180011	18.782643
Artificial Neural Network	26.871022	-11.481421	18.872894

The predicted value is obtained from the machine learning algorithms implemented Here's a table summarizing the (R2) R-squared and (RMSE) Root Mean Squared Error values for each machine learning algorithm.

Table 6: Estimated Error

	Multivariable	Decision Tree	Artificial Neural
	Linear Regression	Regression	Network
RMSE(X)	9.105143	5.374741	0.030384
RMSE(Y)	6.030947	5.399014	1.481579
RMSE(Z)	0.000000	0.000000	0.045567
R2 (X)	0.628651	0.868257	0.999999
R2 (Y)	0.805418	0.840723	0.987424
R2 (Z)	1.000000	1.000000	0.999938

Table 5 shows the (RMSE) Root mean squared error and (R2) R-squared error for the (MLR) Multiple Linear Regression, (ANN) Artificial Neural Network, and Decision-Tree (DT) regression, which is found on MATLAB because of each algorithm. For Multiple Linear

Regression (MLR) RMSE values are relatively high indicating a noticeable deviation between the predicted and actual values. R2 values suggest a moderate to good fit for the X and Y coordinates, but an excellent fit for the Z coordinate. Thus, MLR did not satisfy the demand and lags in capturing the complex relationships between variables. Hence, it had a higher error as compared to the other two algorithms used in this study.

For the Artificial Neural Network, the root mean squared error is extremely low as mentioned earlier in Table 5. So, there is a very small deviation between the actual and predicted value which can be easily observed in Figure 5 in comparison with MLR and Decision-Tree Regression. R2 values are close to one suggesting an excellent fit for all coordinates. Hence, it proves that for this work ANN excels in capturing the complex patterns and non-linear relationships, resulting in better performance compared to other algorithms to predict end effector position of 4 DOF on the basis of joint angles.

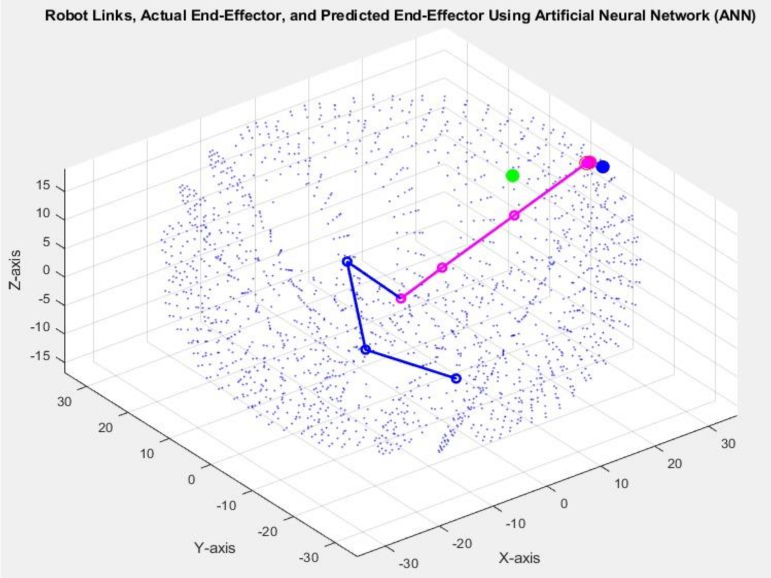

Robot Links, Actual End-Effector, and Predicted End-Effector Using Artificial Neural Network (ANN)

Figure 5: Graphical Comparison of different algorithms.

In summary, the decision tree regression and artificial neural network models outperform multivariable linear regression in accurately predicting the end effector position. The decision tree regression model performs exceptionally well, achieving zero error for the Z coordinate. The artificial neural network model demonstrates outstanding performance with negligible errors across all coordinates, displaying its effectiveness in handling complex data patterns for the work

Conclusion:

This study explores the machine learning algorithms to predict the operational area of the 4-Degree-of-Freedom robot end effector position based on joint angles. The Forward Kinematics While Multiple Linear Regression shows moderate performance, Decision Tree Regression excels with lower errors. However, Artificial Neural Network emerges as the top performer, showcasing remarkable accuracy in predicting end effector positions. These findings highlight the potential of machine learning in enhancing robotics autonomy and task planning.

References:

[1] "Forward and Inverse Kinematics Solution of A 3-DOF Articulated Robotic Manipulator Using Artificial Neural Network | Sharkawy | International Journal of Robotics and Control Systems." Accessed: May 05, 2024. [Online]. Available: https://pubs2.ascee.org/index.php/IJRCS/article/view/1017

[2] O. Carbajal-Espinosa, L. Campos-Macías, and M. Díaz-Rodriguez, "FIKA: A Conformal Geometric Algebra Approach to a Fast Inverse Kinematics Algorithm for an Anthropomorphic Robotic Arm," Machines, vol. 12, no. 1, p. 78, Jan. 2024, doi:

10.3390/MACHINES12010078/S1.

[3] E. Bayro-Corrochano, “A Survey on Quaternion Algebra and Geometric Algebra Applications in Engineering and Computer Science 1995-2020,” IEEE Access, vol. 9, pp. 104326–104355, 2021, doi: 10.1109/ACCESS.2021.3097756.

[4] J. Denavit and R. S. Hartenberg, “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,” J. Appl. Mech., vol. 22, no. 2, pp. 215–221, Jun. 1955, doi: 10.1115/1.4011045.

[5] I. Virgala, M. Kelemen, M. Varga, and P. Kuryło, “Analyzing, Modeling and Simulation of Humanoid Robot Hand Motion,” Procedia Eng., vol. 96, pp. 489–499, Jan. 2014, doi: 10.1016/J.PROENG.2014.12.121.

[6] R. Gao, “Inverse kinematics solution of Robotics based on neural network algorithms,” J. Ambient Intell. Humaniz. Comput., vol. 11, no. 12, pp. 6199–6209, Dec. 2020, doi: 10.1007/S12652-020-01815-4/METRICS.

[7] M. Soori, B. Arezoo, and R. Dastres, “Artificial intelligence, machine learning and deep learning in advanced robotics, a review,” Cogn. Robot., vol. 3, pp. 54–70, Jan. 2023, doi: 10.1016/J.COGR.2023.04.001.

[8] N. Kovincic, H. Gattringer, A. Müller, and M. Brandstötter, “A Boosted Decision Tree Approach for a Safe Human-Robot Collaboration in Quasi-static Impact Situations,” Mech. Mach. Sci., vol. 84, pp. 235–244, 2020, doi: 10.1007/978-3-030-48989-2_26.

Low-Cost Smart Metering Using Deep Learning

Farhan Khan[1], Sarmad Rafique[2], Gul Muhammad Khan[1],

[1]Department of Electrical Engineering University of Engineering and Technology Peshawar, Pakistan

[2]Department of Computer Systems Engineering University of Engineering and Technology Peshawar, Pakistan,

*Correspondence: farhankhan@uetpeshawar.edu.pk, sarmadrafiq.ncai@uetpeshawar.edu.pk, gk502@uetpeshawar.edu.pk

Citation | Khan. F, Rafique. S, Khan. G. M, "Low-Cost Smart Metering Using Deep Learning", IJIST, Special Issue pp 93-104, May 2024

Received | May 04, 2024, **Revised |** May 11, 2024, **Accepted |** May 17, 2024, **Published |** May 22, 2024.

Utility services like electricity, water, and gas are essential for modern living, and their demand has been rising worldwide. However, traditional manual meter reading is a standard procedure for billing purposes. This is not only labor and time-intensive but also prone to mistakes, which results in incorrect billing and revenue losses. In the era of advanced AI, leveraging cutting-edge technology to automate meter readings has become increasingly viable. However, Existing AI-based meter reading systems have limitations in detecting and recognizing meters from a distance. This research addresses these problems by presenting a novel system that utilizes the YOLOv8 model to detect meter screens from a distance. In addition, the system uses a fine-tuned Paddle OCR to recognize meter readings. A Novel dataset curated for the meter screen detection, recognition, and end-to-end OCR tasks related to electricity, gas, and water utility meters has been presented, containing up to 8,044 images. The proposed system was trained and extensively tested on the proposed dataset to gauge its performance. The system achieved an exceptional mean Average Precision (mAP) of 0.995 for both analog and digital meters on the detection task; furthermore, the system achieved an accuracy of 96.92% in the recognition task, which is 70% better than the accuracy of Pre-trained Paddle OCR. Moreover, an all-encompassing evaluation that combines detection and recognition using Paddle OCR and YOLOv8, i.e., the end-to-end OCR task, achieved an accuracy of 97.8%. Lastly, the system achieved an inference speed of up to 6 frames per second, guaranteeing real-time effectiveness.

Keywords: Yolo-v8; Paddle OCR; Meter Detection; Automatic Meter Recognition; Low-cost Smart Metering.

Introduction:

Accurate meter reading is an essential component of the utility industry. However, these utility sectors still rely on traditional manual meter reading, which is time-consuming, labor-intensive, and prone to errors, resulting in huge financial losses. According to the estimation of the World Bank, electricity distribution companies lose 96$ billion in revenue yearly due to billing errors [1]. American electric utility companies experienced an estimated 1–10$ billion USD loss due to billing errors which is 0.5% to 3.5% of its annual GDP [2]. The only energy provider in Peninsular Malaysia, Tenaga Nasional Berhad (TNB) [3], claimed revenue losses of up to 229$ million annually in 2004 due to billing errors. Even though photo billing has become a popular option nowadays, the manual meter reading remains the same. Each month, an employee of the service company goes to each house to take a picture of the meter and manually enter the billing data, which is time-consuming and error-prone [4].

To tackle these issues, Smart meters [5] are introduced for Automatic Meter Reading (AMR); the goal is to automatically record and invoice the reading of gas, water, and electricity. Even though smart meters have been adopted quickly, the standard procedure of manual meter reading in many places, particularly in developing countries, remains the same. Pakistan's energy industry, WAPDA, relies on manual meter readings, which has led to several losses. For utility providers, human errors like misreading or incorrectly recording meter readings can result in improper invoicing and revenue losses. These losses make the power industry less financially viable and may limit its capacity to invest in new and improved infrastructure. Similarly, the utility sectors like Water and Gas have also faced a lot of losses due to manual meter readings. Given the difficulties associated with manual reading processes and the gradual substitution of smart meters for traditional ones [6], [7], there is an increasing demand for image-based methods for text recognition to automate the meter reading process, minimize human errors, and lessen the requirement for substantial human resources [8]. Artificial intelligence (AI) [9], a promising technology, can address the difficulties of manual meter reading in utility industries. The process of reading meters could be revolutionized by implementing AI-based metering technologies. AI technology can deliver precise and timely data, it can automate meter reading which improves billing accuracy and lowers operational costs. AI-based technology, such as object detection [10], can be used to perform meter reading detection using models like Yolo [11], SSD [12], and FAST-RCNN [13]. Additionally, Optical Character Recognition (OCR) technology like Easy OCR [14], Keras OCR [15], and Tesseract OCR [16] can be used for meter reading recognition. Electric, water, and gas utilities stand to gain significantly from using AI and Computer Vision Automatic Meter Reading (AMR) technology in photo billing, making its adoption essential for the utility sector. While AI-based meter reading solutions exist, they have limitations in detecting and recognizing meters from a distance. This study presents a novel approach based on deep learning and advanced computer vision. To address the difficulties associated with detecting and recognizing meters from a distance. The suggested system is thoroughly trained on a variety of meter images taken from a variety of meter models both analog and digital, installed on electricity, gas, and water supplies to improve its detection and recognition performance for long distances. The intended result is a significant improvement in meter reading efficiency, accuracy, and dependability for utility companies across the globe. Specifically designed to operate in real-time from a distance, it utilizes the YOLO-v8 [17] algorithm, which has been trained on a novel custom dataset to provide the best possible detection for analog and digital meters. A novel dataset was also created just for recognition to improve the performance of Paddle OCR [18]. This two-pronged strategy achieves great results.

Literature Review:

The incorporation of AI in automated meter reading (AMR) technology for utility photo billing has recently been possible because of the development of strong AI models. Utility businesses can achieve increased accuracy in meter reading and billing procedures by utilizing AI algorithms, such as deep learning and computer vision techniques, eliminating human errors. However, AMR has several challenges, like image blur, rotated digits, light reflections, and poor image quality. To overcome these drawbacks, Muhammad Waqar et al. [4] proposed an automated method for extracting and identifying numbers from electric meters that uses Faster R-CNN. Using a dataset from Pakistani electrical providers, the model training achieved a promising result, outperforming Single Shot Detector (SSD), Google Vision API, and conventional techniques. Chun-Ming Tsai et al. [6] introduced a digital region detection system for electricity meters, which achieves a higher accuracy of 99% by implementing the SSD deep learning model. Their methodology involves optimizing the SSD model through training on a dataset of 777-meter pictures. Despite this achievement, one of the limitations is that more real-world tests are required for reliable validation. Convolutional neural networks (CNNs) have also shown great potential in solving the difficult automatic meter reading (AMR) task. Chunshan Li et al. [7] proposed a lightweight spliced convolution network for smart water meter reading that substantially reduces computing load and model space while increasing running time. The system's ability to handle data in real time when deployed on a distributed cloud platform validates its accuracy and suitability for industrial use. Rayson Laroca et al. [8] contributed a two-stage method for automatic meter reading (AMR) that uses three CNN-based algorithms (CR-NET, multitask learning, and CRNN) for recognition and Fast-YOLO for detection. With a recognition accuracy of 94.13%, the CR-NET model outperforms both multitask and CRNN models. The study also presents the UFPR-AMR dataset containing 2000 annotated images for meter screen detection. Abdullah Azeem et al. [19] proposed a MaskRCNN (AMR) approach for Detection, Recognition, and Digit Segmentation. The proposed method was assessed on the UFPR-AMR dataset. The suggested method outperforms existing approaches in terms of F-measure and detection accuracy, achieving a prediction rate of 99.82% for counters. An efficient technique for automatic meter reading (AMR) in real-world settings is put forth by Rayson Larcoa et al. [20]. Their method, including corner detection and counter classification, achieved a 34% reduction in reading errors. They also introduced Copel-AMR, a publicly available dataset with 12,500 images of meters; their approach surpassed ten baseline models regarding precision and recognition rate. With 30.64% parameter reduction, Sichao Zhuo [21] presents DAMP-YOLO, a lightweight network for meter reading, by combining DCB, ATA, MDA, and NP with YOLOv8. The model achieves 88.82% mAP50:95, able to recognize objects in real-time on the Jetson TX1. Additionally, Wenwei Lin [22] presents a deep-learning approach for restoring blurry images and recognizing LED digital meters. Polygon-YOLOv5 was used to extract the meter region, and YOLOv5s and CRNN models were employed to recognize the meter readings, achieving 98% accuracy with a 1% missing rate. A sophisticated method for automatic water meter reading was built by Mith Lewis W. Concio et al. [23] using deep learning in a cloud database and mobile app with U-Net binary segmentation for counter detection and Faster RCNN for counter recognition; the pipeline achieves 91.5% accuracy on foreign meters but struggles with 75% accuracy on local meters in the Philippines. Rafaela Carvalho et al. [24] presented a deep-learning model for flow meters and universal controllers as a means of automating manual meter readings. The method consists of screen detection, perspective correction, text detection, template matching, and text recognition. The full pipeline on a taken image takes approximately 1500 milliseconds to complete, whereas screen detection usually takes less than 250 milliseconds. Using YOLO v3 for text extraction and recognition, Muhammad Imran et al. [25] created an automated system for reading electrical energy meters that achieved a 77% precision and 98% recall on a dataset of 10,000-metre images. A lightweight DNN solution for automatic meter reading was introduced by Akshay Kumar Sharma et al. [26], and it outperformed traditional CNN models with 96% accuracy. Although the system contains an Android application for real-time storage and extracts the region of interest, it lacks advanced analysis capabilities and relies on OpenCV for identification. Deyuan Liu [27] combines

YOLOv5s with an enhanced k-means algorithm to detect reflecting places in pointer meters for inspection robots. The solution contains a novel robot pose control mechanism for effectively eliminating reflective surfaces, and it shows applicability in complicated situations with a remarkable accuracy of 80.9%. To automate the collection of water meter data in Morocco Ayman Naim et al. [28] developed an AI system that included a Recognition System built on a Convolutional Neural Network (CNN) model. With 140,000 high-quality digital meter photographs as its training dataset, the CNN model scored an astounding 98.70% accuracy during training.

In this work, we adopt a structured approach to address the problem at hand. Section III outlines our methodology, including details on the employed dataset, the Models used, and our proposed framework. Section IV expounded the experimental setup, covering data pre-processing, network training, and evaluation metrics. Section V delves into the results and discussions regarding the performance of our proposed system. Finally, conclusions are drawn in Section V, encapsulating the findings, contributions, and future directions of this work.

Methodology:

Employed Dataset:

Our process began with the acquisition of a diversified utility meter image dataset to successfully address the detection, recognition, and end-to-end OCR tasks. Therefore, a comprehensive training dataset including 3,905 pictures was produced by incorporating datasets from reliable sources, including the UFPR AMR dataset [8], Water Meters dataset [29], YUVA EB dataset [30], and Gas Meter dataset [31] and around 241 new images were added. The data set's high quality and diversity make it easier to create more sophisticated algorithms and models for detecting and recognizing meter readings. Random samples from the data set are shown in Figure 1 and Figure 2. Furthermore, a separate novel dataset of 3,154 images was gathered and labeled appropriately for optical character recognition by cropping the meter screen regions from the detection dataset. This dataset is unique, as such, a comprehensive dataset is not available elsewhere. Lastly, the end-to-end dataset contained 985 images and was taken as a subset of the detection dataset.

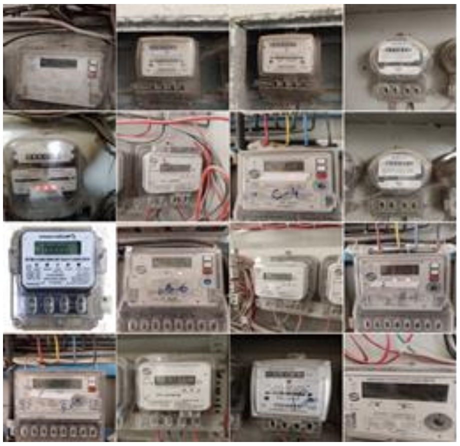


Figure 1: Detection Dataset

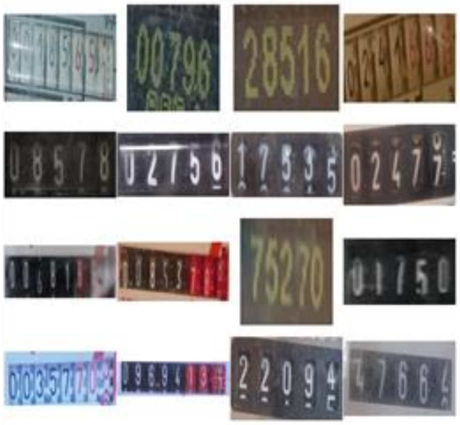


Figure 2: Recognition Dataset

YOLO-V8:

The YOLOv8 [17] model is a real-time, one-stage detection system built on Convolutional Neural Networks (CNN), and it is an improvement over the YOLO (You Only Look Once) series. Acknowledged for its effectiveness in fusing features and providing accurate detection outcomes in a lightweight design, YOLOv8 brings new features and enhancements over its predecessors. YOLOv8's anchor-free design, which deviates from conventional anchor-based methods, speeds up non-maximum suppression and improves overall detection efficiency. Designed to meet a variety of research requirements, YOLOv8 offers five different scale models (n, s, m, l, x). Three essential modules make up the network architecture, as shown in Figure 3.

The Head, Neck, and Backbone modules handle prediction output, multi-feature fusion, and feature extraction, respectively. The Backbone module includes the C2F structure and uses the Spatial Pyramid Pooling Fusion (SPPF) to improve gradient flow information while keeping a lightweight profile. To improve model generalization and resilience, the Head module provides a Decoupled Head structure, which extracts target location and category information independently. The Neck module uses a PAN (Path Aggregation Network) and FPN (Feature Pyramid Network) technique for feature fusion. Thus, YoloV8 is the current state-of-the-art in object detection.

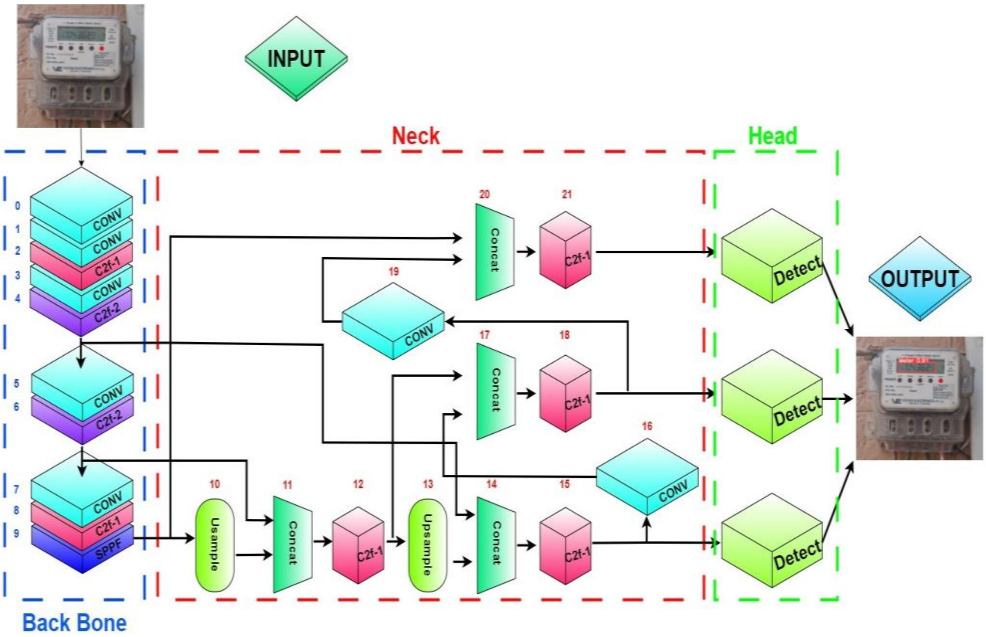


Figure 3: Yolo-v8 Object Detection Architecture

Paddle OCR:

Baidu's Paddle OCR [18] is a powerful OCR model that works with over 25 languages, has pre-trained models, and is very good at recognizing text that is lengthy, vertical, and has digits. It was created by Paddle and uses deep learning to extract text quickly and accurately. PP-OCRv3, the most recent release, offers independent usability for recognition, classification, and detection. In PP-OCRv3, several optimization techniques are added to increase the recognition model's effectiveness and precision. To achieve improved performance, Transformer-based SVTR and CNN-based PP-LCNet are combined in the lightweight text recognition network known as SVTR LCNet, improving prediction speed by 20% without appreciably sacrificing accuracy. The Attention module is used in the GTC method to provide guided CTC training, which enhances accuracy. Text Con Aug is a data augmentation approach used to improve contextual information variety for better model performance. Thus, Paddle OCR is a very diverse model for character recognition.

Proposed Framework:

The detection dataset was used to train various versions of the YOLO models for better bench-marking among which the YOLO V8 model produced the best results for detection. Additionally, Paddle OCR was trained and fine-tuned specifically for Recognition. The validation set was used to thoroughly validate the model's performance. After training, the detection and recognition models are incorporated into a unified workflow. The first step in the method is to take a picture of the meter display or screen, which will used as input into the model. Preprocessing for detection is then carried out. The model evaluates the confidence level after determining the location of the meter screen. The image is sent back to the detection pre-processing stage if the confidence value is less than 0.5. On the other hand, the Detected region is cropped and sent for character recognition preprocessing if the confidence level is higher than 0.5. The intended outcome is attained if the cropped region is Recognized and the model's confidence level is higher than 0.5. If the model's confidence level is less than 0.5, the picture is returned to the detection phase until the model's confidence level rises over 0.5. The Block diagram of the overall working of the system is shown in Figure 4.

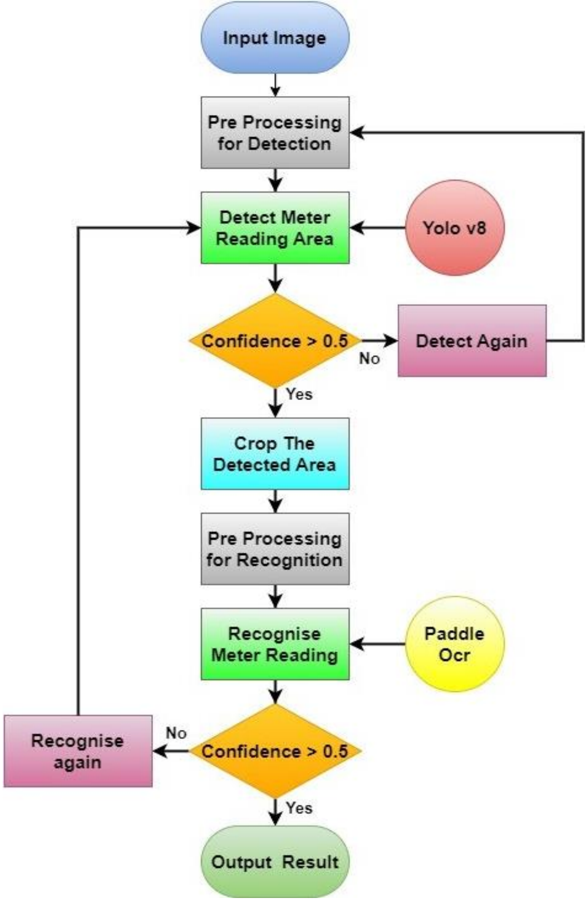


Figure 4: Block diagram of Proposed Framework

Experimental Setup:

Data Pre-Processing:

The Detection dataset was further split into training and validation sets using a split ratio of 70% for training and 30% for validation. Thorough pre-processing procedures, including the elimination of redundant, superfluous, and noisy images, were carried out before the model was trained, and bounding box annotation was used in the training of the detection model on the dataset. To improve the dataset balance and diversity, methods such as image scaling, standardization, and augmentation were also used. The recognition dataset was divided into training and validation sets using 80% and 20% split ratios, respectively. The end-to-end dataset was used to evaluate the overall performance of the proposed system and thus was not split into train and validation ratios. The Bounding box annotation was performed on the detection dataset using the Libeling tool, and the annotation files were saved in Txt format.

Networks Training:

The proposed system in this study was trained and evaluated on a computer running on the Windows 10 OS with Intel Core i7-10700 CPU @ 2.90 GHz, NVIDIA GeForce RTX 3060 12 GB, 16 GB Ram, and the programming language used was Python 3.7 with the PyTorch framework. The training parameters for the detection and recognition are summarized in Table 1.

Table 1: Training Parameters

	Parameters for Detection and Recognition Models			
No.	**Detection Parameters**	**Details**	**Recognition Parameters**	**Details**
1	Picture size	640 x 640	Picture size	48 x 320
2	Epochs	300	Epochs	500
3	Batch size	16	Batch size	128
4	Optimizer	SGD	Optimizer	Adam
5	Learning rate	0.01	Learning rate	0.001
6	Workers	8	Workers	4
7	Patience	40		

Evaluation Metrics:

To provide a thorough assessment of the proposed system, relevant evaluation metrics were used for each task. F1 score, precision, recall, mAP50, and mAP50-90 were used to measure detection performance; these metrics provide an extensive assessment of the system's efficiency in object detection. For the end-to-end OCR and recognition tasks, metrics like Character Error Rate, Recognition accuracy, and Character Accuracy were used to gauge their performance. CER is more suited for single-word recognition tasks, such as meter readings than Word Error Rate, which is used for sentences. Because the recognition task is unpredictable and the recognized output may contain extra or missing data, we refrained from utilizing precision, recall, or F1 score.

- **F1- Score:**

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{1}$$

- **Precision:**

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

- **Recall:**

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

- **mAP50:**

$$mAP50 = \frac{1}{|X|} \sum_{i=1}^{|X|} AvgP(z_i) \tag{4}$$

Where:

- $|X|$: Number of queries in the dataset.
- z_i: i-th query in the dataset.
- AvgP(z_i): Average precision for the i-th query, using the first 50 items in the ranked list.

- **mAP50-90:**

$$mAP50 - 90 = \frac{1}{|Q|} \sum_{i=1}^{|Q|} AvgP(q_i) \tag{5}$$

Where:

- $|Q|$: is the total number of queries in the dataset.
- q_i: represents the *i*-th query in the dataset.
- AvgP(q_i): Average precision for the *i*-th query, using only the top 50 to 90 ranked items.

- **Character Error Rate:**

$$CER = \frac{S + D + I}{C} * 100 \qquad (6)$$

Where:

- *S*: Substitutions
- *D*: Deletions
- *I*: Insertions
- *C*: Total characters in reference transcription

- **Recognition Accuracy:**

$$A = \frac{N}{T} * 100 \qquad (7)$$

Where:

- *N*: Correctly recognized meter readings
- *T*: Total meter display readings

- **Character Accuracy:**

$$CA = \frac{CC}{TC} * 100 \qquad (8)$$

Where:

- *CC*: Number of Correct Character
- *TC*: Total Characters

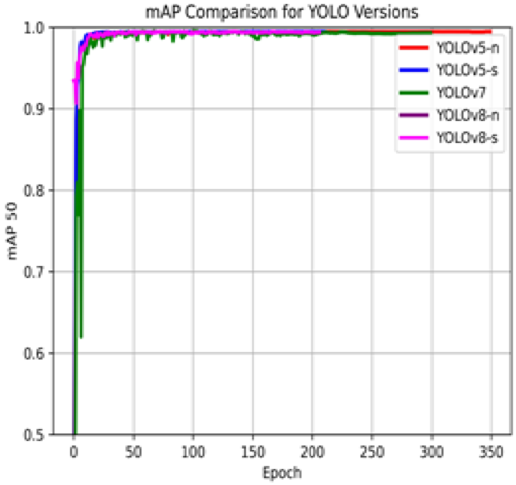


Figure 5: mAP0.5 Training curves

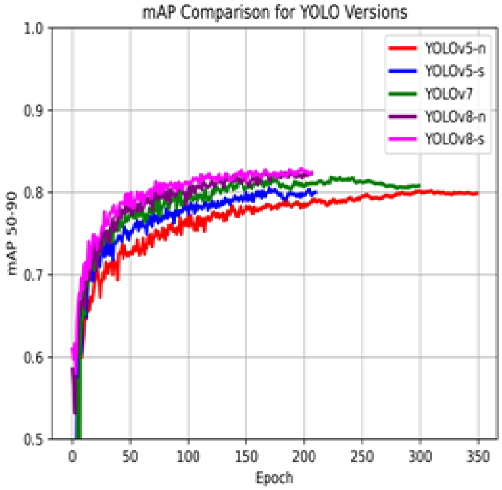


Figure 6: mAP0.5-0.9 Training curves

Results and Discussion:

Using the proposed dataset, the training results of several detection models produced impressive results, as shown in Table 2. Impressively, our system obtained an F1-Score of 99.3%. While every model performed well, YOLOv8 Nano was particularly noteworthy as it produced the greatest results in terms of mAP@50 achieving 0.995, and mAP@50-90 achieving 0.826, mAP50 and mAP50-90 plots of all training models are shown in Figure 5 and Figure 6. At 99.4% and 99.3%, respectively, the meter detection precision and recall hit their peak, and no further optimization was possible. Turning to recognition models, we carefully examined the most famous OCR models like Karas OCR and Paddle OCR. By testing these two OCR models, using a subset of 976 cropped meter screen images from the end-to-end dataset. Paddle OCR was the clear winner, with better performance than its competitor, as shown in Table 3. Therefore, Paddle OCR was fine-tuned on the recognition dataset, and an accuracy of 99.21% was achieved.

When this fine-tuned model was tested using the previous subset dataset, an accuracy of 96.92% with a CER of 0.0054 was achieved.

Table 2: Training results of the Detection Models

	Performance Evaluation Metrics for Detection Task				
Model	**F1 Score**	**Precision**	**Recall**	**mAP-50**	**mAP50-90**
YOLOv5-n	99.0%	99.2%	98.9%	**0.995**	0.801
YOLOv5-s	99.2%	99.2%	**99.3%**	**0.995**	0.805
YOLOv7	**99.3%**	99.5%	**99.3%**	0.994	0.817
YOLOv8-n	**99.3%**	99.4%	**99.3%**	**0.995**	**0.826**
YOLOv8-s	**99.3%**	**99.6%**	99.2%	0.994	0.825

Table 3: Bench-marking of Recognition Models

	Performance Evaluation Metrics for Recognition Task			
Model	**Accuracy**	**CA**	**CER**	**Training Accuracy**
Kera's OCR Pre-trained	1.536%	6.663%	0.8759	Nil
Paddle OCR Pre-trained	19.87%	39.79%	0.5042	Nil
Paddle OCR Fine-tuned	**96.92%**	**99.11%**	**0.0054**	**99.21%**

A. Proposed System Performance:

The fine-tuned YOLOv8 Nano was chosen due to its high inference speed and better results and was combined with Paddle OCR to form the proposed framework which was tested on the end-to-end dataset comprising 987 images, an overall accuracy of 97.8% was achieved encapsulating both detection and recognition performance. The results are visualized in Figure 7. The inference speed of the proposed framework was around 6 frames per second, showcasing real-time performance. This amalgamation of detection and recognition models showcases a promising avenue for bolstering the accuracy and efficiency of meter detection and recognition tasks.

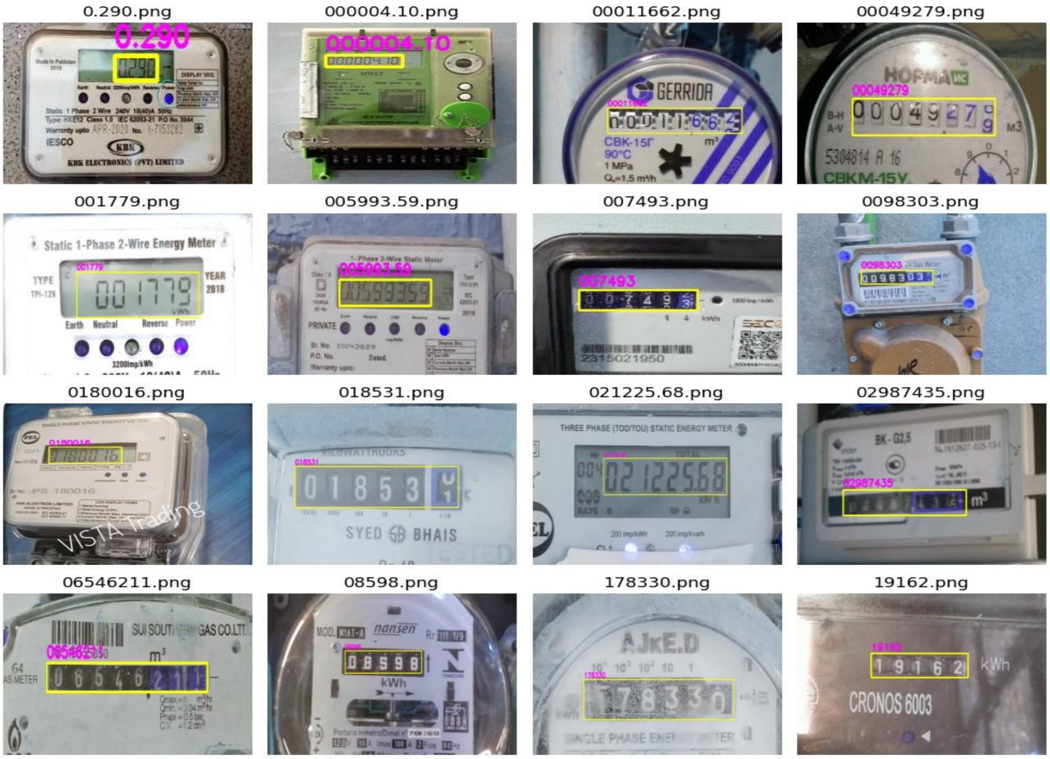


Figure 7: End-to-End OCR Model Results Visualization

Conclusion:

Our primary objective in this research has been to tackle the complex problem of detecting and recognizing both digital and analog meters from a distance. We observed the core limits of current AI systems and presented a groundbreaking solution by utilizing and combining the advanced features of YOLOv8 for meter screen detection and Paddle OCR for digit recognition for an end-to-end OCR system; our study attempted to close this gap and produced an excellent mean Average Precision (mAP) of 0.995 and an F1 score of 99.3%. Furthermore, the recognition performance of the system, powered by 99.21% accuracy Paddle OCR, highlights how effective our suggested method is in addressing the drawbacks of existing systems. We hope to further the progress of meter reading technology by releasing publicly accessible datasets that are expressly intended for detection, recognition, and end-to-end AMR tasks. We aimed to contribute not only to the advancements in meter reading technology but also to provide a benchmark for the research community to evaluate and build upon. Our system's dependability and efficiency are confirmed by the extensive testing on the proposed datasets, which includes a sizable dataset of 8044. There is room for our system to be expanded in the future. Using forecasting models in conjunction with past consumption data is one approach that is worth investigating. This calculated addition might improve meter reading validation, providing a more thorough and precise result.

Acknowledgment:

We express our sincere gratitude to the National Center of Artificial Intelligence at the University of Engineering and Technology, Peshawar, for providing the computational resources for this research.

References:

[1] "India's power sector. (2012, August 9). World Bank", [Online]. Available: https://www.worldbank.org/en/news/feature/2010/04/19/india-powersector

[2] T. B. Smith, "Electricity theft: a comparative analysis," Energy Policy, vol. 32, no. 18, pp. 2067–2076, Dec. 2004, doi: 10.1016/S0301-4215(03)00182-4.

[3] "annual report - Tenaga Nasional Berhad." Accessed: May 06, 2024. [Online]. Available: https://www.yumpu.com/en/document/view/51051742/annual-report-tenaga-nasional-berhad

[4] M. Waqar, M. A. Waris, E. Rashid, N. Nida, S. Nawaz, and M. H. Yousaf, "Meter Digit Recognition Via Faster R-CNN," 2019 Int. Conf. Robot. Autom. Ind. ICRAI 2019, Oct. 2019, doi: 10.1109/ICRAI47710.2019.8967357.

[5] A. Cooper, "Electric Company Smart Meter Deployments: Foundation for A Smart Grid," 2021.

[6] C. M. Tsai, T. D. Shou, S. C. Chen, and J. W. Hsieh, "Use SSD to Detect the Digital Region in Electricity Meter," Proc. - Int. Conf. Mach. Learn. Cybern., vol. 2019-July, Jul. 2019, doi: 10.1109/ICMLC48188.2019.8949195.

[7] Li. Chunshan, Yukun Su, Rui Yuan, Dianhui Chu, and Jinhui Zhu. "Light-Weight Spliced Convolution Network-Based Automatic Water Meter Reading in Smart City", [Online]. Available: https://ieeexplore.ieee.org/document/8917620

[8] R. Laroca, Victor Barroso, Matheus A. Diniz, Gabriel R. Gonçalves, William R. Schwartz, and David Menotti. "Convolutional neural networks for automatic meter reading", [Online]. Available: https://www.spiedigitallibrary.org/journals/journal-of-electronic-imaging/volume-28/issue-1/013023/Convolutional-neural-networks-for-automatic-meter-reading/10.1117/1.JEI.28.1.013023.short

[9] C. Zhang and Y. Lu, “Study on artificial intelligence: The state of the art and future prospects,” J. Ind. Inf. Integr., vol. 23, p. 100224, Sep. 2021, doi: 10.1016/J.JII.2021.100224.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[11] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of Yolo Algorithm Developments,” Procedia Comput. Sci., vol. 199, pp. 1066–1073, Jan. 2022, doi: 10.1016/J.PROCS.2022.01.135.

[12] A. C. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, “SSD: Single Shot MultiBox Detector”, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2

[13] R. Girshick, “Fast R-CNN,” Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[14] “Repository of GitHub - JaidedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc in May 2024.” Accessed: May 06, 2024. [Online]. Available: https://gitpiper.com/resources/python/computervision/JaidedAI-EasyOCR

[15] “keras-ocr — keras_ocr documentation.” Accessed: May 06, 2024. [Online]. Available: https://keras-ocr.readthedocs.io/en/latest/

[16] “GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository).” Accessed: May 06, 2024. [Online]. Available: https://github.com/tesseract-ocr/tesseract

[17] “GitHub - ultralytics/ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite.” Accessed: May 06, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[18] “GitHub - PaddlePaddle/PaddleOCR: Awesome multilingual OCR toolkits based on PaddlePaddle (practical ultra lightweight OCR system, support 80+ languages recognition, provide data annotation and synthesis tools, support training and deployment among server, mobile, embedded and IoT devices).” Accessed: May 06, 2024. [Online]. Available: https://github.com/PaddlePaddle/PaddleOCR

[19] A. Azeem, W. Riaz, A. Siddique, and U. A. K. Saifullah, “A Robust Automatic Meter Reading System based on Mask-RCNN,” Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020, pp. 209–213, Aug. 2020, doi: 10.1109/AEECA49918.2020.9213531.

[20] R. Laroca, A. B. Araujo, L. A. Zanlorensi, E. C. de Almeida, and D. Menotti, “Towards Image-based Automatic Meter Reading in Unconstrained Scenarios: A Robust and Efficient Approach,” IEEE Access, vol. 9, pp. 67569–67584, Sep. 2020, doi: 10.1109/ACCESS.2021.3077415.

[21] S. Zhuo, Xiaoming Zhang, Ziyi Chen, Wei Wei, Fang Wang, Q. Li and Y. Guan, “DAMP-YOLO: A Lightweight Network Based on Deformable Features and Aggregation for Meter Reading Recognition”, [Online]. Available: https://www.mdpi.com/2076-3417/13/20/11493

[22] W. Lin, Z. Zhao, J. Tao, C. Lian, and C. Zhang, "Research on Digital Meter Reading Method of Inspection Robot Based on Deep Learning," Appl. Sci. 2023, Vol. 13, Page 7146, vol. 13, no. 12, p. 7146, Jun. 2023, doi: 10.3390/APP13127146.

[23] M. L. W. Concio, F. S. Bernardo, J. M. Opulencia, G. L. Ortiz, and J. R. I. Pedrasa, "Automated Water Meter Reading Through Image Recognition," IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2022-November, 2022, doi: 10.1109/TENCON55691.2022.9977678.

[24] R. Carvalho, J. Melo, R. Graça, G. Santos, and M. J. M. Vasconcelos, "Deep Learning-Powered System for Real-Time Digital Meter Reading on Edge Devices," Appl. Sci. 2023, Vol. 13, Page 2315, vol. 13, no. 4, p. 2315, Feb. 2023, doi: 10.3390/APP13042315.

[25] M. Imran, H. Anwar, M. Tufail, A. Khan, M. Khan, and D. A. Ramli, "Image-Based Automatic Energy Meter Reading Using Deep Learning," Comput. Mater. Contin., vol. 74, no. 1, pp. 203–216, 2023, doi: 10.32604/CMC.2023.029834.

[26] A. K. Sharma and K. K. Kim, "Lightweight CNN based Meter Digit Recognition," J. Sens. Sci. Technol., vol. 30, no. 1, pp. 15–19, Jan. 2021, doi: 10.46670/JSST.2021.30.1.15.

[27] D. Liu, C. Deng, H. Zhang, J. Li, and B. Shi, "Adaptive Reflection Detection and Control Strategy of Pointer Meters Based on YOLOv5s," Sensors 2023, Vol. 23, Page 2562, vol. 23, no. 5, p. 2562, Feb. 2023, doi: 10.3390/S23052562.

[28] A. Naim, A. Aaroud, K. Akodadi, and C. El Hachimi, "A fully AI-based system to automate water meter data collection in Morocco country," Array, vol. 10, Jul. 2021, doi: 10.1016/j.array.2021.100056.

[29] "Water Meters Dataset, 1244 Photos & Masks." Accessed: May 06, 2024. [Online]. Available: https://www.kaggle.com/datasets/tapakah68/yandextoloka-water-meters-dataset.

[30] K. Kanagarathinam and K. Sekar, "Text detection and recognition in raw image dataset of seven segment digital energy meter display," Energy Reports, vol. 5, pp. 842–852, Nov. 2019, doi: 10.1016/J.EGYR.2019.07.004.

[31] A. Iqbal, A. Basit, I. Ali, J. Babar, and I. Ullah, "Automated Meter Reading Detection Using Inception with Single Shot Multi-Box Detector," Intell. Autom. Soft Comput., vol. 27, no. 2, pp. 299–309, Jan. 2021, doi: 10.32604/IASC.2021.014250.

Smart Fire Safety: Real-Time Segmentation and Alerts Using Deep Learning

Farhan Khan[1], Sarmad Rafique[2], Salman Khan[1], Laiq Hasan[2],

[1]Department of Electrical Engineering University of Engineering and Technology Peshawar, Pakistan,

[2]Department of Computer Systems Engineering University of Engineering and Technology Peshawar, Pakistan,

*Correspondence: farhankhan@uetpeshawar.edu.pk ,sarmadrafiq.ncai@uetpeshawar.edu.pk ,engrsalmankhan@uetpeshawar.edu.pk ,laiqhasan@gmail.com ,

Citation | Khan. F, Rafique. S, Khan. S, Hasan. L., "Smart Fire Safety: Real-Time Segmentation and Alerts Using Deep Learning", IJIST, Special Issue pp 105-115, May 2024

Received| May 04, 2024, **Revised|** May 10, 2024, **Accepted|** May 18, 2024, **Published|** May 22, 2024.

Fires are the major causes of property damage, injuries, and death worldwide. The ability to avoid or reduce the effects of fires depends on their early identification. The accuracy and responsiveness of conventional fire detection systems, such as smoke detectors and heat sensors, are constrained. Computer vision-based fire and smoke detection systems have been suggested as a replacement for conventional systems in recent years. To tackle the challenges a robust real-time framework has been proposed, whereby, images are taken from cameras and using a custom train YOLOv8 object segmentation model smoke and fires are localized in the image which are then fed to an expert system for alert generation. The expert system makes decisions on the fire status based on its size and growth across multiple frames. Furthermore, A new dataset was meticulously curated and annotated for the segmentation task, to assess the efficacy of the proposed system, comprehensive benchmarking was conducted on the proposed dataset using a suite of benchmarks. The proposed system achieved an mAP score of 74.9% on the benchmark dataset. Furthermore, it was observed that employing segmentation for localization as opposed to detection, resulted in system accuracy improvement. The system can immediately identify fires and smoke and send accurate alerts to emergency services.

Keywords: Yolo-v8; Instance Segmentation; Fire and Smoke detection; Fire size; Fire spread; Emergency alert message; Arduino Uno.

Introduction:

Fire has played a major role in the advancement of human society, but uncontrolled fires can lead to a significant loss of human life and property, so it is essential to prevent such types of fires because they can be widespread and result in huge losses. The National Fire Protection Association (NFPA) estimates that over 350,000 house-structure fires nationwide require the help of fire departments annually, with direct damages estimated to be around $7 billion. In addition, there are 12,300 civilian fire injuries and about 2,500 civilian fire fatalities per year [1]. In Pakistan, fire has caused hundreds of casualties and infrastructural damage totaling billions of rupees. One of Pakistan's most damaging fire disasters is thought to be the Baldia Town fire incident. However, the relevant government officials have not drawn any lessons from the situation [2]. Recently, a terrible fire in Lahore's renowned Hafeez Center destroyed hundreds of shops and caused traders to suffer severe losses [3]. A strong communication network is essential for the efficient operation of firefighting services. Today, many buildings have built-in smoke detectors and fire alarm systems as the most common fire detection method. These systems are activated when smoke from a fire rises and triggers sensors, usually located in the ceilings of the buildings. These sensors then activate the fire alarm and fire suppression systems however in 2018, 38% of fire alarms failed to sound when there was a fire, and 45% of these incidences were due to improper system positioning, according to the Home Office of the United Kingdom [4]. While this method is generally effective, there can be a delay between the smoke rising and hitting the sensor, which can allow the fire to spread quickly [5]. This delay should be minimized to prevent fires from getting out of control so traditional fire detection methods, including smoke detectors and heat sensors, may not provide the level of accuracy and speed necessary to quickly and effectively detect and respond to fires. In light of this, there has been a recent surge in the development of Deep Learning and computer vision-based fire and smoke detection systems as an alternative approach. These systems utilize cameras to gather visual data about the environment and apply machine learning techniques to analyze and detect fires and smoke. These Deep-learning techniques have been a major asset in the extraction of relevant features that best represent fires. Such methods have been applied to a wide range of fields, including image classification, autonomous vehicles, speech recognition, pedestrian detection, facial recognition, and cancer detection among others, showcasing their effectiveness in detecting and segmenting various object classes [6]. In this research, we propose a real-time fire and smoke segmentation detection system that uses the YOLO-v8 [7] object detection algorithm and an emergency alert and danger level warning system. The proposed system is designed to detect fires and smoke in real time, providing accurate and timely alerts to emergency services. The YOLO-v8 algorithm is trained on a custom-segmented dataset of fires and smoke to identify these phenomena in real-world environments. Additionally, the system employs color and texture-based features for segmenting and identifying the fire and smoke regions in the image. The system also includes a danger level warning and emergency alert that utilizes the size, location, and intensity of the fire or smoke to determine the level of emergency. The system can communicate the alerts to emergency services through email and push notifications. The results of the experiments demonstrate that the proposed system can detect fires and smoke in real time with high accuracy and provide accurate and timely emergency alerts. The system also can segment the fire and smoke regions in the image, providing more detailed information about the emergency. We believe that the proposed system has the potential to significantly reduce the impact of fires on people and property. It serves as a valuable tool for emergency response and building safety.

Literature Review:

Advances in AI, machine learning, and deep learning have fueled the widespread use of these technologies in fire and smoke detection. Souidene Mseddi et al. [8] proposed a fire detection model, combining YOLOv5 and U-net, achieving 99.6% accuracy. Ge Zhang et al.[9]

enhanced YOLOv5 with a Swin transformation, improving feature fusion and achieving a 0.7% accuracy boost. Chen et al.[10] introduced a mixed Gaussian algorithm and YOLOv5-based smoke detection with 94.7% accuracy and 66.7 FPS speed. Sun et al.[11] addressed instance segmentation drawbacks with a semi-supervised technique and a lightweight SOLOv2 network, improving accuracy. Solorzano et al. [12] researched gas sensors for fire detection, demonstrating persistent predictive calibration models. Feiniu et al. [13] proposed a smoke segmentation model using CNN with VGG16 architecture. Jiao et al. [14] utilized Yolo v3 for UAV-based forest fire detection, achieving 83% accuracy at 3.2 frames per second. Hao Xu et al. [15] Innovative algorithm which was built using YOLOv5n. A SepViT Block was used to replace the model's final layer to strengthen the connection between the backbone network and the global information. A self-designed Light-BiFPN was also used to strengthen and lighten the network, reduce information loss, and improve accuracy and training convergence speed. Lastly, the Mish activation function was employed. For real-time fire detection on mobile devices, the Light-YOLOv5 significantly decreases the number of parameters and computations while increasing detection accuracy. Bhanumathi. M et al. [16] proposed image-based techniques for fire detection using surveillance cameras. They employ a background subtraction technique to detect fire using an RGB color pattern and motion detection technology. The technique seeks to spot fire quickly to protect people and property from its danger. The proposed system makes use of CCTV cameras to detect environmental changes brought on by a fire. Fatma M. Talaat et al.[17] proposed a Smart Fire Detection System (SFDS), which is based on the YOLOv8 algorithm, is a transformative approach to fire detection in smart cities achieving a precision of 97.1%. One disadvantage in this is the possibility of including unnecessary items within the bounding box, making it difficult to identify the seriousness of a detected incident. This ambiguity might cause difficulties in discerning between circumstances that require immediate attention due to risk and those that may be less critical. SN Saydirasulovich et al.[18] presents an improved YOLOv8 model for UAV-based wildfire smoke detection, which addresses obstacles such as sluggish recognition and accuracy issues. The model includes Wise-IoU v3, Ghost Shuffle Convolution, and the BiFormer attention mechanism, resulting in a 3.3% improvement in average precision. Despite its success, the model is highly sensitive to atmospheric conditions, which can lead to false positives. Wahyono et al.[19] analyses Faster RCNN, Yolov4, and Yolov5 for fire detection in surveillance footage using datasets including FireNet, VisiFire, and FireSense. Yolov5 scores the best on the FireSense dataset, whereas Yolov4 excels with the highest TPR (84.62%) on VisiFire. However, including undesired background implies that more factors should be investigated in the future for better accuracy, particularly taking into account real-world camera constraints. Leibiao Hu et al.[20] proposed a novel YOLOv8 algorithm FSD-YOLOv8, trained on the FASDD dataset. Specifically designed for accurate flame and smoke detection. It incorporates a de-hazing stage for improved precision and uses dilated convolutions to extract complex features. FSDYOLOv8 works better than traditional techniques. however, its accuracy is reduced in dynamic lighting and continuously smoke-filled environments. De Venancio et al.[21] presented a low-power automatic fire detection system utilizing YOLOv4. The suggested method reduces computational costs and memory consumption by up to an astounding 83.86%, respectively. The model achieved a f1 score of 72% and a map of 73.9% using the D-Fire dataset during training. X Xie et al.[22] Propose a YOLOv5-based real-time flame detection system for firefighting drones. A coordinate attention method, enhanced small-target recognition, and a novel loss function were introduced in the model to enhance the performance of the model. The experimental findings reveal an average precision of 96.6%, which is 5.4% higher than the original. The lightweight structure of the algorithm makes it suitable for firefighting drones and allows for fast identification. how every its accuracy reduces in detection at night, indicating potential directions for further study. Chayma Bahhar et al. [23] suggests a novel method for detecting forest fires in real-time by combining a classifier to increase precision and using an ensemble of two YOLO architectures (Yolov5s and Yolov5l). The FLAME dataset was used to train the model. The model performs better than the others, as evidenced by the trial results, which show 0.87% recall and 0.8% precision. The model is more resilient in a variety of forest fire conditions thanks to the ensemble architecture which reduces false positives.

In summary, this work uses an organized approach. Our technique is described in Section III; the experimental setting is described in Section IV; the dataset used and evaluation measures are covered in Section V. Section VI presents the findings and debates, while Section VII wraps up this work.

Methodology:

The proposed system attempts to improve building fire safety by installing a fire and smoke detection and alert system. To assess visual data from cameras mounted in the building, the system used the most recent fire detection algorithms. The YOLO-v8 segmentation detection algorithm is the central element of the suggested system which is used to examine visual data from the cameras. The alarm is activated by Arduino setup when fire or smoke is successfully detected in real-time through cameras. The alarm's activation notifies the building's residents of the fire emergency and begins the water spray, both of which aid in containing the fire's progress. The system also had a smoke sensor coupled to the Arduino setup. If there is smoke in the building, the sensor will detect it, which will cause the buzzer to sound and the water spray to begin. As a result of the smoke sensor's integration, the system has a backup method of spotting fires. A danger level warning system that gives real-time updates on the severity of the fire emergency is another component of the planned system. Building occupants and emergency services can react to the fire emergency in an informed way thanks to the warning system, which is based on the size of the fire, the spreading of the fire, and the 1-minute time if the fire is still detected by the YoloV8 model. The proposed system sends emergency notifications to emergency services and building inhabitants via email. The notification gives emergency services vital details about the location and severity of the fire, enabling them to react to the emergency swiftly and effectively. It is crucial to remember that the effectiveness and accuracy of the suggested approach depend on the camera's accurate calibration. The whole methodology is shown through a block diagram in Figure 1.

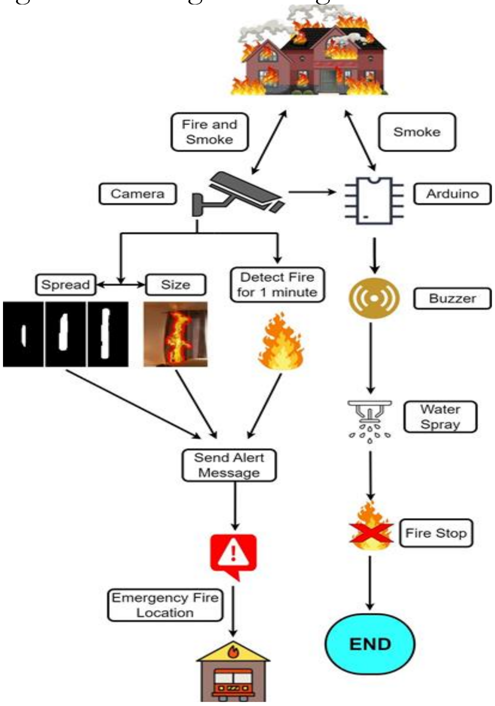


Figure 1: Methodology through block diagram

YOLO-v8 Instance Segmentation:

YOLOv8-seg (You Only Look Once version 8) is a cutting-edge object detection system that utilizes deep learning to achieve real-time results. It uses a unique network architecture to perform instance segmentation. The structure of yolo-v8 instance segmentation can be seen in Figure 2. YOLOv8-Seg is a significant improvement in the YOLO family, designed primarily for segmentation jobs. This model differs from its predecessors with an improved backbone structure that includes a 3 x 3 convolution, a C2f module, and an SPPF module. The C2f module replaces the basic 6 x 6 convolution, resulting in a lightweight architecture and improved gradient flow with skip connections and split operations. Notably, YOLOv8-Seg deviates from the traditional C3 module by incorporating improved cross-stage partial networks (CSP) for better residual connections. The head module demonstrates complicated feature fusion algorithms that include PANet and FPN, with major performance enhancements applied.

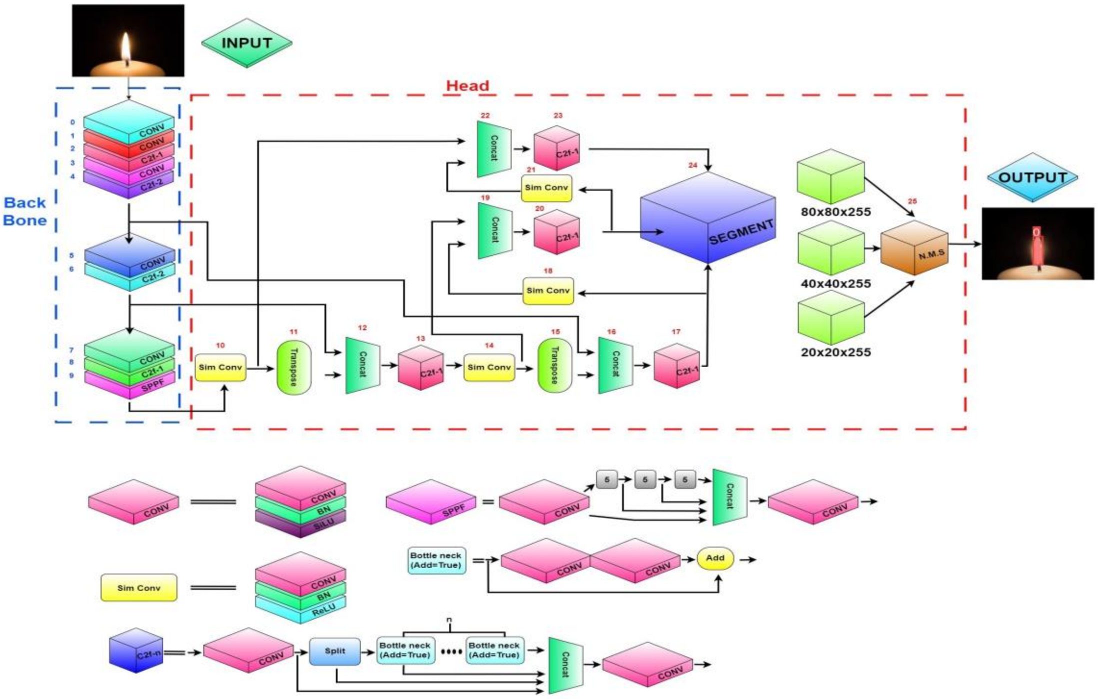


Figure 2: Represents all the layers present in the Yolo-v8 instance segmentation Architecture

Experimental Setup:

Experimental Environment:

The Experiment was conducted using the Google Colab platform.

Training:

The training parameters for the fire and smoke detection model are shown in Table 1. The fire and smoke data set used in the study was divided into a training set and a validation set, with the proportion being 80:20.

Table 1: Training Parameter.

Parameter	Details
Picture size	640 x 640
Epochs	300
Batch size	16
Optimizer	SGD
Learning rate	0.01
Early stopping Patience	40
Multi-scale	50%
Momentum	0.937

System Deployment:

The Arduino Setup used in this paper is shown in Figure 3. Arduino UNO will receive information from both the camera and the MQ2 Gas Sensor. When fire or smoke is detected by the camera or smoke is sensed by the MQ2 sensor, Arduino will turn on the red light and buzzer.

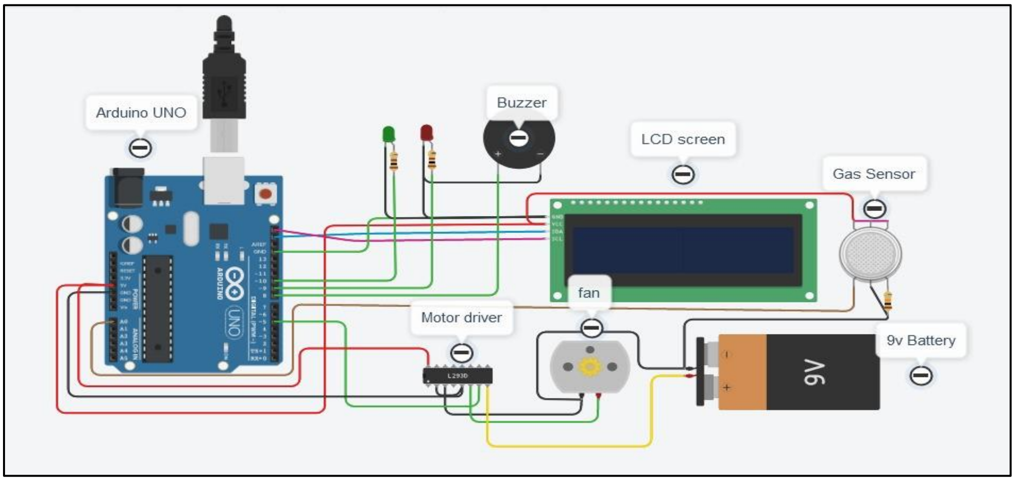


Figure 3: Represent the Experimental Setup

The primary components that have been used in this setup are i. Arduino Uno ii. MQ2 Gas Sensor iii. I2C LCD Display Driver iv. 16x2 LCD display
v. Passive Piezo Buzzer vi. Light-Emitting Diodes

Employed Dataset and Evaluation Metrics:

Dataset:

In this paper, we proposed a novel data set for fire and smoke gathered from images and videos. The data set contains 892 pictures in total, divided into three groups fire, fire-smoke, and smoke. A wide variety of locations, including both indoor and outdoor scenes were represented in the data set. The instance segmentation task was performed on the data set using the Label-Me tool and the annotation files were saved in JSON format. To make the data set compatible with the Yolov8 framework, we wrote a code to convert the JSON format to a text format. This data set can be utilized for a variety of tasks, such as object detection, semantic segmentation, and fire and smoke instance segmentation, thanks to its accurate annotations and wide set of photos depicting various situations. The data-set high quality and diversity made it easier to create more sophisticated algorithms and models for analyzing fire and smoke. Random samples from the data set are shown in Figure 4.

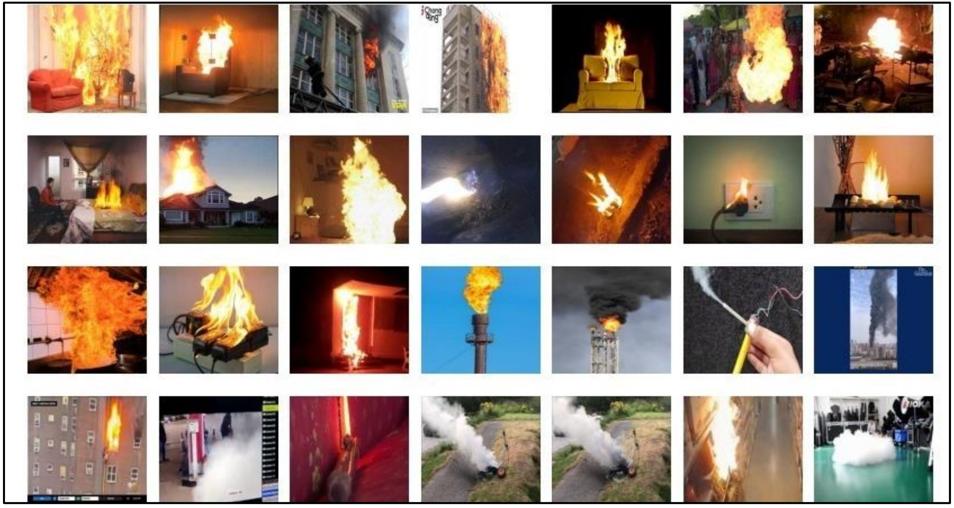


Figure 4: Illustrate fire and smoke images from over data-set

Evaluation Metrics:

We assess the effectiveness of over model through:

- **F1-Score:**

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{1}$$

- **Precision:**

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

- **Recall:**

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

- **mAP50:**

$$mAP50 = \frac{1}{|X|}\sum_{i=1}^{|X|} AvgP(z_i) \tag{4}$$

Where:

- $|X|$: Number of queries in the dataset.
- z_i: i-th query in the dataset.
- AvgP(z_i): Average precision for the i-th query, using the first 50 items in the ranked list.

- **mAP50-90:**

$$mAP50 - 90 = \frac{1}{|Q|}\sum_{i=1}^{|Q|} AvgP(q_i) \tag{5}$$

Where:

- $|Q|$: is the total number of queries in the dataset.
- q_i: represents the *i*-th query in the dataset.
- AvgP(q_i): Average precision for the *i*-th query, using only the top 50 to 90 ranked items.

Results and Discussion:

Table 2: Results of the Bounding Boxes

	Performance Evaluation Metrics				
Model	**F1 score**	**Precision**	**Recall**	**mAP-50**	**mAP50-90**
YOLOv5n	73.4%	82.3%	66.3%	72.0%	44.8%
YOLOv5-s	76.5%	**90.2%**	66.5%	74.8%	51.1%
YOLOv5-m	75.4%	85.5%	67.5%	74.0%	51.2%
YOLOv5-l	76.0%	84.5%	69.1%	75.6%	54.1%
YOLOv7	76.9%	86.0%	69.7%	**77.9%**	55.3%
YOLOv8-n	73.4%	85.0%	64.6%	72.5%	49.3%
YOLOv8-s	73.7%	86.0%	64.6%	72.4%	50.8%
YOLOv8-m	76.2%	86.1%	68.4%	76.5%	**56.1%**
YOLOv8-l	**77.4%**	86.4%	**70.2%**	76.8%	55.8%

The training outcomes of different models on the proposed dataset of fires and smoke are shown in Tables 2 and 3. The training's outcomes revealed that the system achieved an f1-Score of 77.4% for Boxes and 75.7% for Masks. Every model performed well but YOLOv8 Large was particularly noteworthy as it produced the greatest results for masks in terms of mAP50 and map50-90 achieving 74.9% and 51.0% respectively. Figure 5 shows the model's performance on some test images. The results of map50 for Masks of all models are shown in Figure 6. The precision and recall results for boxes and masks were 86.4%, 70.2%, 85.7%, and 67.9% respectively. The fine-tuned YOLOv8 large model was incorporated into the system due to its better results in detecting fire and smoke masks. The real-time performance of the system can be seen in Figures 7, and 8, showing an excellent result in detecting fire and smoke. The emergency alert system was successfully implemented in the model and can send emergency messages if any of the three danger level conditions are satisfied as shown in Figure 9.

Table 3: Results of the Masks

	Performance Evaluation Metrics				
Model	**F1 score**	**Precision**	**Recall**	**mAP-50**	**mAP50-90**
YOLOv5-n	70.8%	79.4%	63.9%	68.7%	39.8%
YOLOv5-s	74.5%	**87.9%**	64.7%	72.5%	43.6%
YOLOv5-m	73.9%	83.9%	66.1%	72.2%	44.3%
YOLOv5-l	74.3%	87.3%	64.8%	72.7%	44.4%
YOLOv7	74.7%	86.9%	65.6%	70.7%	41.7%
YOLOv8-n	73.3%	84.9%	64.5%	72.6%	47.5%
YOLOv8-s	73.5%	85.7%	64.4%	72.3%	48.1%
YOLOv8-m	75.0%	84.7%	67.4%	**74.9%**	48.9%
YOLOv8-l	**75.7%**	85.7%	**67.9%**	**74.9%**	**51.0%**

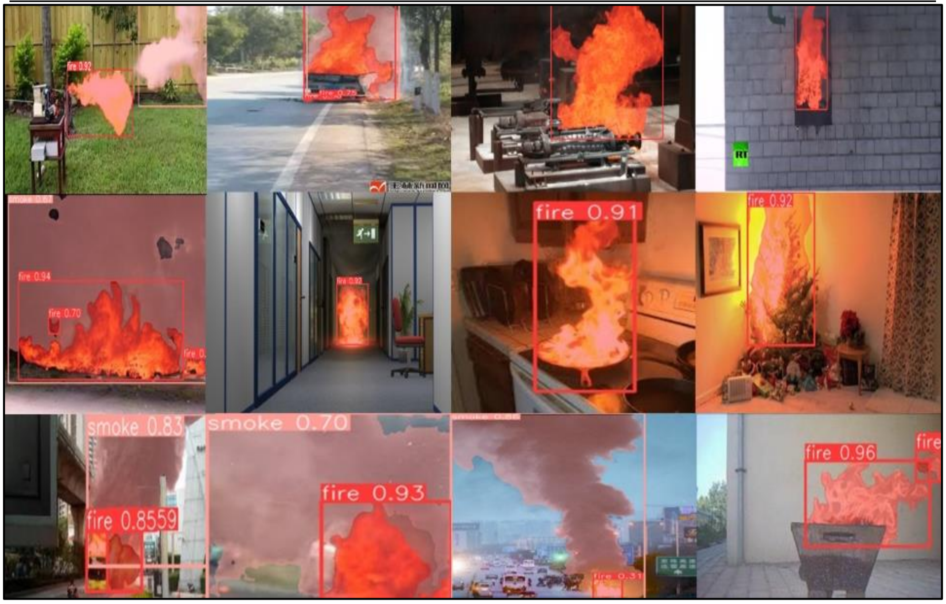


Figure 5: Results of the trained model on some test images

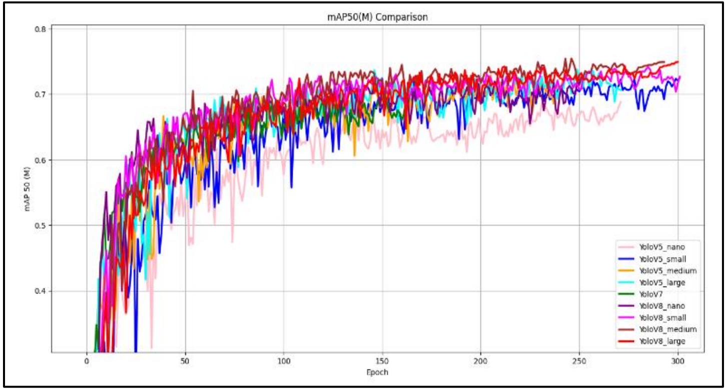


Figure 6: Model training curves of mAP0.5 metric for segmentation masks

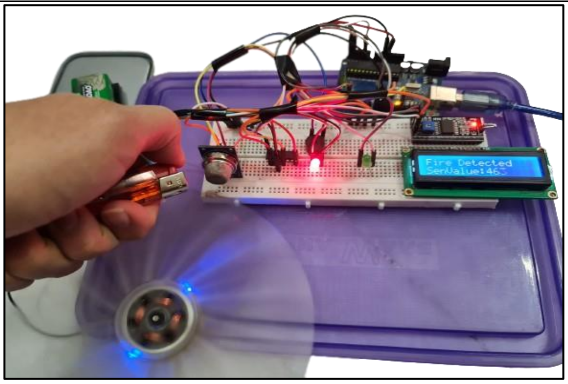


Figure 7: Show that when the smoke sensor detects smoke it starts the alarm and water spray

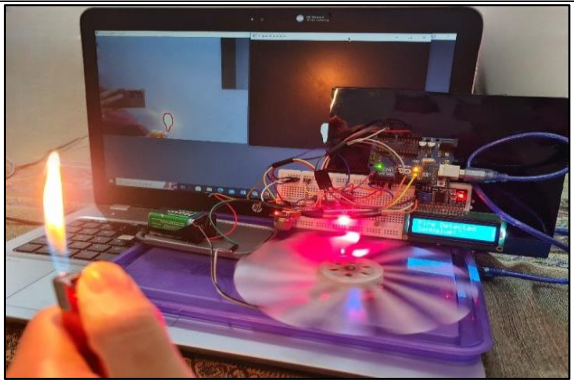
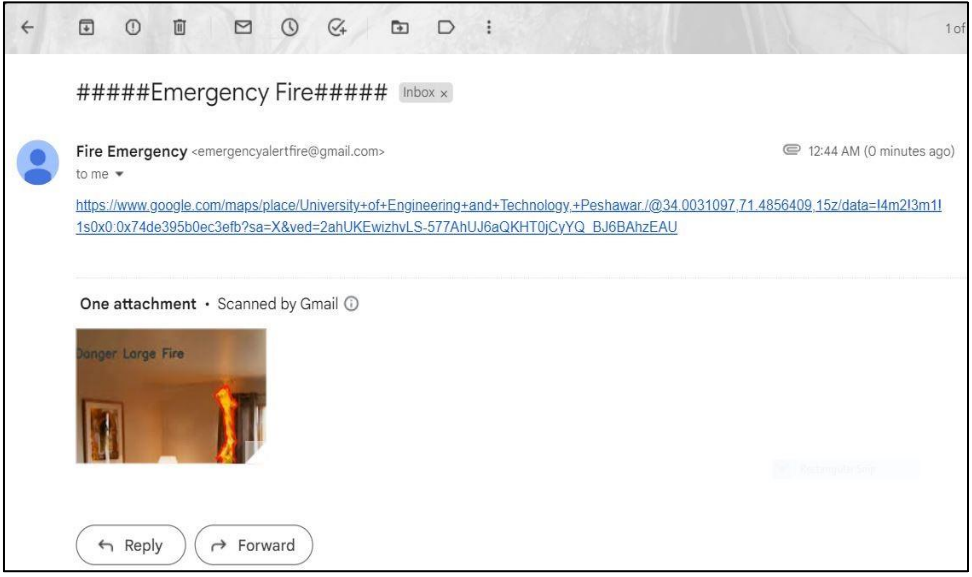


Figure 8: Show that when the camera detects fire and smoke it starts the alarm and water spray



#####Emergency Fire#####

Fire Emergency <emergencyalertfire@gmail.com>
to me
12:44 AM (0 minutes ago)

https://www.google.com/maps/place/University+of+Engineering+and+Technology,+Peshawar,/@34.0031097,71.4856409,15z/data=!4m2!3m1!1s0x0:0x74de395b0ec3efb?sa=X&ved=2ahUKEwizhvLS-577AhUJ6aQKHT0jCyYQ_BJ6BAhzEAU

One attachment • Scanned by Gmail

Danger Large Fire

Figure 9: Show that when any danger level condition is satisfied an emergency alert message is sent

Conclusion:

To ensure public and building safety we proposed and implemented a system that uses the YOLO-v8 object segmentation detection technique for real-time fire and smoke detection. The system was created to instantly detect fire and smoke and send precise, timely alerts to emergency services and building occupants. The system was fitted with a hazard level warning system that was based on the size of the fire, spreading of fire, and 1 minute perennially detection of fire. The YOLO-v8 algorithm was trained on a novel instance-segmented data set. The trained algorithm achieves f1-Score of 75.7% and mAP50 of 74.9% for the Masks. The study's findings demonstrated that the suggested system could accurately and promptly provide emergency notifications while detecting fires and smoke in real time with high precision. The suggested system showed excellent potential in lessening the damage caused by fires to people and property, and it might be an important aid in emergency response and building safety. The novel instance segmented data set, the hazard level warning system, and the emergency alert system are the main contributions to this research. In the future, the system performance can be further enhanced by training the YOLO-v8 algorithm on a larger and more diverse data set. The system for alerting of danger levels and emergency alerts can be further improved by including more complex algorithms to forecast how the fire or smoke will develop as well as interacting with emergency response systems.

References:

[1] "Home Structure Fires report NFPA," 2021, [Online]. Available: https://www.nfpa.org/Newsand-Research/Data-research-and-tools/Building-and-Life-Safety/HomeStructure-Fires

[2] "Fire Hazards and Firefighting in Pakistan - Webinar Report - IIPS." Accessed: May 08, 2024. [Online]. Available: https://iips.com.pk/fire-hazards-and-firefighting-in-pakistan/

[3] "500 shops gutted in Lahore's Hafeez Centre fire." Accessed: May 08, 2024. [Online]. Available: https://www.thenews.com.pk/print/731453-500shops-gutted-in-lahore-s-hafeez-centre-fire

[4] "Fire and rescue incident statistics: England, year ending March 2020 - GOV.UK." Accessed: May 08, 2024. [Online]. Available: https://www.gov.uk/government/statistics/fire-and-rescue-incident-statistics-england-year-ending-march-2020

[5] "Smoke Detectors." Accessed: May 08, 2024. [Online]. Available: https://www.explainthatstuff.com/smokedetector.html

[6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[7] "Jocher, G., Chaurasia, A. and Qiu, J. (2023) YOLO by Ultralytics. - References - Scientific Research Publishing." Accessed: May 08, 2024. [Online]. Available: https://scirp.org/reference/referencespapers?referenceid=3532980

[8] W. S. Mseddi, R. Ghali, M. Jmal, and R. Attia, "Fire Detection and Segmentation using YOLOv5 and U-NET," *Eur. Signal Process. Conf.*, vol. 2021-August, pp. 741–745, 2021, doi: 10.23919/EUSIPCO54536.2021.9616026.

[9] S. G. Zhang, F. Zhang, Y. Ding, and Y. Li, "Swin-YOLOv5: Research and Application of Fire and Smoke Detection Algorithm Based on YOLOv5," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–8, Jun. 2022, doi: 10.1155/2022/6081680.

[10] X. Chen, Y. Xue, Y. Zhu, and R. Ma, "A novel smoke detection algorithm based on improved mixed Gaussian and YOLOv5 for textile workshop environments," *IET Image Process.*, vol. 17, no. 7, pp. 1991–2004, May 2023, doi: 10.1049/IPR2.12719.

[11] G. Sun, Y. Wen, and Y. Li, "Instance segmentation using semi-supervised learning for fire recognition," *Heliyon*, vol. 8, no. 12, Dec. 2022, doi: 10.1016/j.heliyon.2022.e12375.

[12] A. Solórzano *et al.*, "Early fire detection based on gas sensor arrays: Multivariate calibration and validation," *Sensors Actuators B Chem.*, vol. 352, p. 130961, Feb. 2022, doi: 10.1016/J.SNB.2021.130961.

[13] F. Yuan, L. Zhang, X. Xia, B. Wan, Q. Huang, and X. Li, "Deep smoke segmentation," *Neurocomputing*, vol. 357, pp. 248–260, Sep. 2019, doi: 10.1016/J.NEUCOM.2019.05.011.

[14] Z. Jiao *et al.*, "A Deep learning based forest fire detection approach using uav and yolov3," *1st Int. Conf. Ind. Artif. Intell. IAI 2019*, Jul. 2019, doi: 10.1109/ICIAI.2019.8850815.

[15] H. Xu, B. Li, and F. Zhong, "Light-YOLOv5: A Lightweight Algorithm for Improved YOLOv5 in Complex Fire Scenarios," *Appl. Sci. 2022, Vol. 12, Page 12312*, vol. 12, no. 23, p. 12312, Dec. 2022, doi: 10.3390/APP122312312.

[16] et. al. Bhanumathi. M, "Fire Detection and Alarm Using Gaussian Blur Background Subtraction Technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 929–934, Apr. 2021, Accessed: May 08, 2024. [Online]. Available: https://turcomat.org/index.php/turkbilmat/article/view/1103

[17] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023,

doi: 10.1007/S00521-023-08809-1/FIGURES/9.

[18] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y. I. Cho, "An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images," *Sensors 2023, Vol. 23, Page 8374*, vol. 23, no. 20, p. 8374, Oct. 2023, doi: 10.3390/S23208374.

[19] W. Wahyono, A. Harjoko, A. Dharmawan, G. Kosala, and P. Y. Pranata, "A Comparison of Deep Learning Methods for Vision-based Fire Detection in Surveillance System," *ACM Int. Conf. Proceeding Ser.*, pp. 1–6, Dec. 2021, doi: 10.1145/3508072.3508074.

[20] L. Hu, C. Lu, X. L. Li, Y. Zhu, Y. Lu, and S. Krishnamoorthy, "An enhanced YOLOv8 for flame and smoke detection with dilated convolution and image dehazing," *https://doi.org/10.1117/12.3012472*, vol. 12970, pp. 604–608, Dec. 2023, doi: 10.1117/12.3012472.

[21] P. V. A. B. de Venâncio, A. C. Lisboa, and A. V. Barbosa, "An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices," *Neural Comput. Appl.*, vol. 34, no. 18, pp. 15349–15368, Sep. 2022, doi: 10.1007/S00521-022-07467-Z/METRICS.

[22] X. Xie, K. Chen, Y. Guo, B. Tan, L. Chen, and M. Huang, "A Flame-Detection Algorithm Using the Improved YOLOv5," *Fire 2023, Vol. 6, Page 313*, vol. 6, no. 8, p. 313, Aug. 2023, doi: 10.3390/FIRE6080313.

[23] C. Bahhar *et al.*, "Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN," *Electron. 2023, Vol. 12, Page 228*, vol. 12, no. 1, p. 228, Jan. 2023, doi: 10.3390/ELECTRONICS12010228.

Hybrid Approach to Solve Thermal Power Plants Fuel Cost Optimization Using Ant Lion Optimizer with Newton-Based Local Search Technique

Ejaz Ahmed *[1,2], Shahbaz Khan[1], Abdul Wadood[1], Husan Ali[1], Babar Sattar Khan[2]
[1]Department of Electrical Engineering Aero Space and Aviation, Air University Islamabad Kamra Campus Attock, Pakistan
[2]Department of Electrical and Computer Engineering COMSATS University Islamabad Attock Campus Attock, Pakistan.,
Correspondence: Ejaz Ahmed. 225258@aack.au.edu.pk

Citation | Ahmed. E, Khan. S, Wadood. A, Ali. H, Khan. B. S, "Hybrid Approach to Solve Thermal Power Plants Fuel Cost Optimization Using Ant Lion Optimizer with Newton-Based Local Search Technique", IJIST, Special Issue pp 116-124, May 2024
Received | May 04, 2024, **Revised |** May 13, 2024, **Accepted |** May 17, 2024, **Published |** May 22, 2024.

Introduction/Importance of Study: The optimization of the power system is a complicated problem that is extremely non-convex, nonlinear, and important for reducing the cost of production.
Novelty Statement: Despite the fact that several metaheuristic algorithms are proposed for solving power system optimization problems, the strength of hybridized global search-based techniques has not commonly been applied to power system optimization.
Material and Method: Deterministic power system optimization strategies are unable to yield global optimal outcomes because of the entrapment in local optimum zones. Stochastic approaches like those in which Ant-Lion Optimizer is used and hybridization algorithms with local search methods SQP, IPA, and active set give better results.
Result and Discussion: Hybridized global search-based techniques have been successfully applied to power system optimization with economic load dispatch in particular. Results from findings hybridized-ALO outperforms modern optimization methods.
Concluding Remarks: Results from findings show 3 and 13 generator systems that hybridized-ALO outperforms modern optimization methods.

Keywords: Economic Load Dispatch (ELD); Ant Lion Optimization (ALO); Valve point loading (VPLE), Fuel cost, and Objective function.

Introduction:

Due to the rising cost of producing electricity and the depletion of fossil fuels utilized in thermal power generating units, optimal Economic Load Dispatch (ELD) has gained significant attention in today's modern power system. The primary goal of the ELD problem is to distribute thermal power generating units' active power generation output as efficiently as possible while accounting for power system operational restrictions. The overall energy capabilities of electrical energy generation grow through the reduction of the generation cost and the enhancement of system reliability through optimum active power allocation. Because there are so many real-world power system scenarios, the research community is more interested in taking a realistic approach to solving the conventional ELD problem [1]. Due to the multilevel steam-generating valves connected to contemporary steam-powered thermal power generating units, also known as the valve point loading effect (VPLE), the input-output fuel cost generation curve is essentially non-differentiable, non-convex, and non-linear. These steam valves open systematically, which causes ripples in the fuel-cost characteristics curve. Conventional fossil fuel-based thermal power plants emit a variety of harmful gases (SOx, NOx, and COx) into the atmosphere, contributing to environmental pollution and global warming [2]. Some of the generating unit shaft's bearings are subject to physical constraints known as prohibited operating zones (POZs) because of enhanced vibrations in specific areas along the rotating axis of the shaft. In recent years, there has been a lot of focus on renewable energy generation sources while optimal power flow is taken into account for transporting electrical power over long distances to avoid power losses, optimal generation allocation and sizing are important factors in power system optimization for electrical power generation to lower fuel generation cost. Due to power losses from long-distance transmission lines and poor voltage regulation from a heavily loaded network, the efficiency of the power system effectively decreases. On the other hand, by lowering generation costs and power losses, taking into account the integration of renewable energy sources and their ideal placement inside conventional power systems can improve system reliability. Appropriate planning is required for the integration of renewable energy into conventional systems in order to prevent operational issues that could compromise system performance and dependability. In order for the power system to run efficiently and affordably, a number of generating units made up of thermal units renewable energy sources should be managed optimally considering practical constraints of real-world power system [3]. A genetic algorithm was applied in [4] for economic load dispatch to reduce the fuel cost. With the incorporation of emissions, the cost of production is increased [5] using hybrid PSO with SQP to solve the economic emission problem. The main aim of our work is to reduce the cost of production for a three- and thirteen-unit system incorporating hybrid methods.

Material and Methods:

Mathematical Model Fuel Cost equation:

The economic load dispatch problem is presented by a quadratic equation. The values of cost coefficients can be taken from [6] fuel cost is related to power as:

$$\boldsymbol{F(P) = \sum\left(up^2 + vP + w\right)} \qquad (1)$$

In equation 1, *F* (p) presents the total generation cost in $/hr whereas u, v, and w, are fuel cost equations. Includes loading on generators on value openings. While second equation 2 models the cost function with value point loading cost multiplied with sin function.

$$f_{WV}(P) = \sum_{j=1}^{ng}\left(u_j p_j^2 + v_j P_j + w_j + \left|x_j \times \sin(y_j \times (P_j^{\min} - P_j))\right|\right) \qquad (2)$$

In equation 2, F_{wv}(p) presents total generation cost with value point loading effect in $/hr whereas u, v, and w, are fuel cost equations x and y are value point coefficients which are non-linearity associated with loading and operating of governor system generators on value openings.

ANT LION Optimizer:

ALO design was inspired by the hunting style of Ant- lions. Ant lions originated from the Myrmeleontid family. Ant-lions life span in adulthood lasts only for 3 to 5 weeks out of the 3 years of age in which they spend the rest of their life it reproduces their offspring. Ant lions are known for their different style of preying on ant insects. Ant-lions dig out special cone (v) shaped traps in mud to hunt the ants [7]. The edges of the cone are sharp Ant-lions try to catch the prey within trap range and place itself in the middle of the cone under sand. Ants tend to slip at sharp edges while ant lions through the sand grains at the edge of the cone. When an ant tries to leave the trap during random movement, the Ant-lions trap size depends on the hunger. The ants move randomly in the trap and are shown as

$$x(T)=\left[0, cummsum\left(2S\left(T^{1}\right)-1\right), \cdots cummsum\left(2S\left(T^{N}\right)-1\right)\right] (3)$$

- In the above equation,
- x(T) denotes the moment of ants
- N shows the total iteration number.
- T is for step of walk-in random way.
- S is a random weight ranging from 0 to 1.

Initializing Position Matrix of Ants:

Random position matrix of ants is generated denoted by ANT^{pos} Each ant will move in different dimensions d and are equal to the number of generators for the ELD problem [5] also uses matrix, Position matrix of ants shown below

$$ANT^{pos}=\begin{pmatrix} a^{11} & a^{12} & a^{13}\cdots\cdots a^{1d} \\ a^{21} & a^{22} & a^{23}\cdots\cdots a^{2d} \\ \vdots & \vdots & \vdots \\ a^{n1} & a^{n2} & a^{n3}\cdots\cdots a^{nd} \end{pmatrix}; \qquad (4)$$

Fitness Value Calculation for Ants:

Each ant is passed through the required objective function that will return the fitness value of each ant saved in the column vector denoted by OA.

$$OA=\begin{pmatrix} oa^{11} \\ oa^{21} \\ \vdots \\ oa^{n1} \end{pmatrix}; \qquad (5)$$

Fitness Value Calculation for Ant-Lions:

Each ant is passed through the required objective function that will return the fitness value of each and saved in a column vector denoted by OAL

$$OAL=\begin{pmatrix} oal^{11} \\ oal^{21} \\ \vdots \\ oal^{n1} \end{pmatrix}; \qquad (6)$$

Random Walk of Ants:

During the optimization process, each updates its position by adopting a random walk in a random direction. equation 1 cannot be directly adopted to update its position. The randomness of the walk is normalized within range with a specific constant.

Trapping in Ant-Lions Pits:

Random walk of ants is affected by ant lion. Ants random walk in hypersphere defined by vectors e and d

$$e_d^t = Ant - lion_j^t + e^t \qquad (7)$$

$$d_d^t = Ant - lion_j^t + d^t \qquad (8)$$

e^t is minimal of variable d at t-th number iteration and d^t shows vector presenting maxima of a variable at t-th number iteration.

Building Trap:

Each ant is trapped by a single ant lion whose Selection is on fitness during optimization.

Sliding Ant Toward Ant-Lion and Catching Pray:

Once an ant comes in the range of the trap ant lions through sand to detract and slips ants toward the center while ant tries to escape.

Elitism:

Elitism is the solution, which is best at any instant of optimizer running. Best ant lions are considered elite so every ant moves randomly around that ant lion.

Results:

The first case is applied to a 3-unit system having a power demand of 850 MW Economic load dispatch with Value point loading shows the total cost is 8234.07174 ($/hr.) In the case of three Units-based ELD test systems, Optimizer was set with an initial setting of 15000 search agents. Each case is run on 20 independent trials and the best results are shown in Figure 1. Complete simulation results are shown in Table 1.

Table 1: Optimized values of power and fuel price for Three Units using A.L.O (Pd =850 M Watt)

Unit	With V_P_L_EE
	Best
Unit 1 (MW)	300.26687
Unit 2 (MW)	399.99999
Unit 3 (MW)	148.73312
Fuel Cost ($/hr.)	8234.07174
Total Power TP (MW)	850

Each generator has to produce specific power depending on coefficients with ALO optima alone results are shown in the convergence curve it can be seen that after 100 iterations results are very close also test function is plotted.

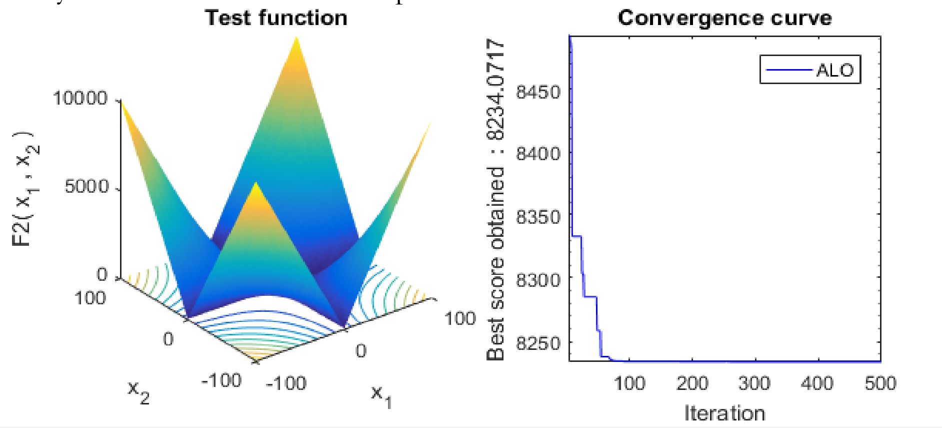


Figure 1: Convergence curve of A.L.O for three units

The second case is applied to 13 13-unit system having a power demand of 1800 MW Economic load dispatch with Value point loading shows the total cost is 17934.3211 ($/hr.)In the case of the 13-unit ELD test system, Optimizer was set with an initial setting of 15000 search agents and three dimensions equaling to number of generators. Each case is run on 20 independent trials and the best results are shown in Figure 2. 1000 iterations were set but can be seen that after 100 iterations solution starts to converge.

Table 2: Optimized values of power and fuel price for Thirteen Units using A.L.O (Pd =1800 M Watt)

Units	Without V_P_L_EE
	Best
Unit 1 (MW)	548.3007
Unit 2 (MW)	260.8828
Unit 3 (MW)	235.9395
Unit 4 (MW)	90.0035
Unit 5 (MW)	101.4202
Unit 6 (MW)	98.2217
Unit 7 (MW)	99.5210
Unit 8 (MW)	88.5453
Unit 9 (MW)	87.1620
Unit 10 (MW)	38.0000
Unit 11 (MW)	41.0032
Unit 12 (MW)	56.0000
Unit 13 (MW)	56.0001
Total Power (MW)	1800.0000
fc ($/hr.)	**17934.3211**

Complete simulation results are presented in Table 2 Moreover, the fine-tuning is carried out by Hybridizing procedures.

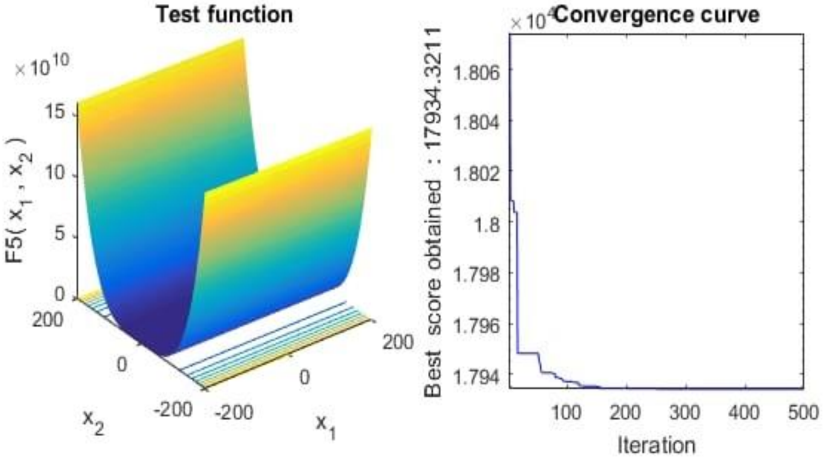


Figure 2: Convergence curve of A.L.O for Thirteen units

Comparison and Discussion:

Hybridizing procedures ALO-SQP, ALO-ASA, and ALO-IPA by taking the best results of ALO as a starting point and continuing with the refined values. Moreover, the results are summarized in Table 3 for each scenario. For every three approaches compared, one may observe that the results of ALO-IPA are better in terms of convergence and accuracy while ALO-SQP gave better outcomes as tabulated in Table 3 considering the ELD problem with VPLE.

Each generator has to produce specific power depending on coefficients with ALO optima alone results are shown in the convergence curve it can be seen that after 100 iterations results are very close also test function is plotted. The active set finds equality constraints in inequality constraints in consideration and uses a small delta after initial point gain throughout global search in this case ALO it can be seen that after 100 iterations results are very close also test function is plotted results are shown in Figure 4.

Table 3: Hybridized with a.l.o 13 bus (pd =1800 M Watt)

Unit	With V_P_L_EE		
	ALO-SQP	**ALO-Active Set**	**ALO-IPA**
Unit 1 (MW)	539.5587	538.6576	538.5587
Unit 2 (MW)	74.7998	79.2048	88.1070
Unit 3 (MW)	299.1993	300.1728	299.1993
Unit 4 (MW)	59.0000	60.0000	60.0004
Unit 5 (MW)	159.7331	180.0000	159.7332
Unit 6 (MW)	109.8666	109.8846	108.8666
Unit 7 (MW)	109.8666	109.8666	110.8666
Unit 8 (MW)	60.0000	60.0000	60.0004
Unit 9 (MW)	109.8666	60.0000	109.8666
Unit 10 (MW)	40.0000	40.0001	40.0005
Unit 11 (MW)	90.7094	114.8132	77.4000
Unit 12 (MW)	92.3999	92.4003	92.4002
Unit 13 (MW)	55.0000	55.0000	55.0005
Power Total	1800.0000	1800.0000	1800.0000
Fuel Cost	**18118.1679**	**18558.4847**	**18122.5229**

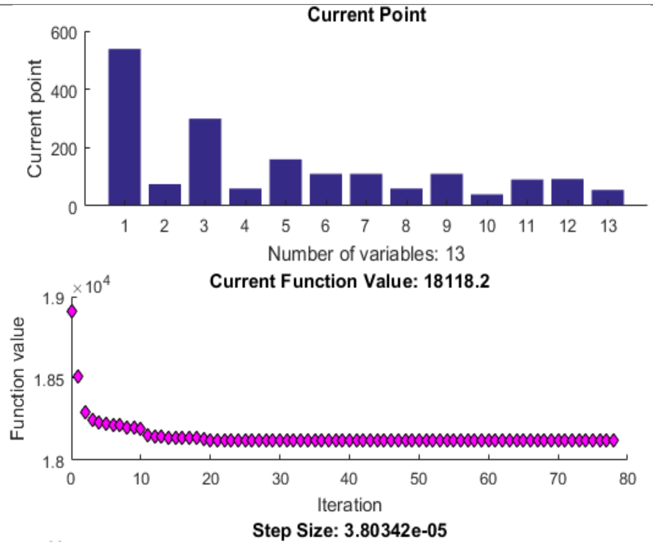


Figure 3: Simulated illustration of integrated ALO-SQP for 13 units with V-P-L-E

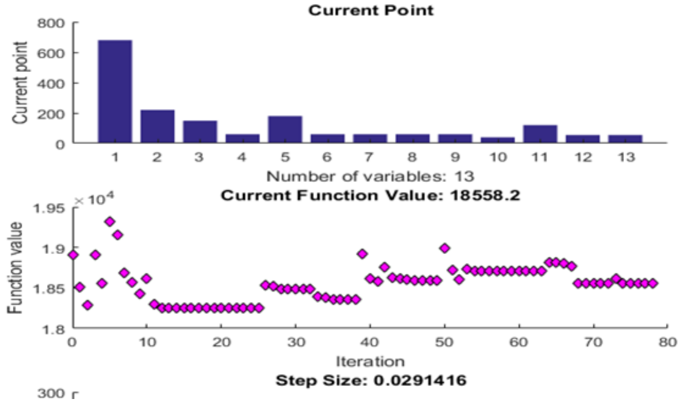


Figure 4: Simulated illustration of integrated ALO-Active set for 13 units with V-P-L-E

IPA approach combines the best features of perturbation analysis and sequence quadratic programming in a unified framework it can be seen that after 100 iterations results are very close also the test function is plotted.

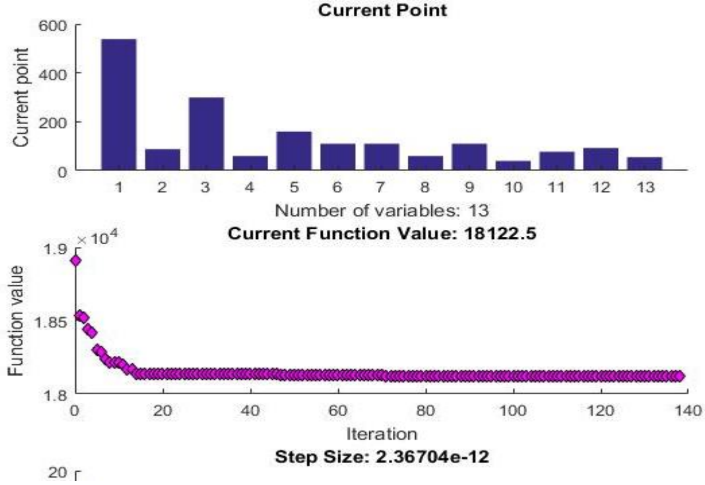


Figure 5: Simulated illustration of integrated ALO-IPA for a thirteen-unit system with V-P-L-E

The effectiveness of ALO is compared with the latest four techniques for the case of 13 units having a power demand of 1800MW. These methods include a teaching-learning optimizer [8], Harmony search optimizer [9], Quazi oppositional inertial weight [10], and Novel Heuristic optimizer [11] while comparison on fuel cost for the same operating constraints the results are shown in Table 4.

Table 4: Comparison with state of art methods

	TLBO [8]	H-S [9]	GPSO [10]	MPSO [11]	ALO
Unit 1 (MW)	364.9	628.3	628.3	628.2	548.30
Unit 2 (MW)	277.9	149.5	224.3	149.6	260.8
Unit 3 (MW)	217.4	222.7	148.7	222.7	235.9
Unit 4 (MW)	95.22	109.8	60	109.8	90
Unit 5 (MW)	106.6	60	109.8	60	101.4
Unit 6 (MW)	123.5	109.8	109.6	109.8	98.2
Unit 7 (MW)	112.5	109.8	60	109.8	99.5
Unit 8 (MW)	144.2	109.8	159.7	109.7	88.5
Unit 9 (MW)	126.7	109.6	109.5	109.8	87.1
Unit10 (MW)	60.23	40	40	40	38
Unit 11 (MW)	48.47	40	40	40	41
Unit 12 (MW)	91.36	55	55	55	56
Unit 13 (MW)	81.23	55	55	55	56
P total	1800	1800	1800	1800	1800
Cost ($/hr.)	**18141.2**	**17963.8**	**17978.6**	**17962.7**	**17934.3**

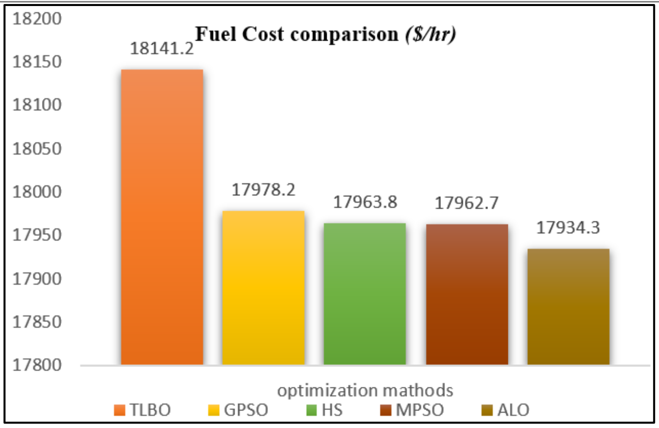


Figure 6: Fuel cost comparison chart

As seen from the chart below teaching learning optimizer [8] has the highest fuel cost of 18141.2 ($/hr), the Harmony search optimizer [9] is second, the Quazi oppositional inertial weight [10] is third and the Novel Heuristic optimizer [11] cost about 17962.7($/hr) while ALO performs best among all with least cost.

Conclusion:

This study explores the application of ALO with local search methods for economic load and emission dispatch, results of 3 generator systems show the strength of ALO also ALO is compared with four state-of-the-art optimizers in terms of fuel cost for 13 generating units' performance as best. With the optimum allocation of these generators, not only cost is reduced but emissions are reduced Furthermore, this work can be extended by applying it to more big generating units and also problems related to reactive power compensation and generator scheduling for day-ahead forecast. Also, other factors like the integration of wind and solar can be included in future work.

Acknowledgement:

Thank you to UET Peshawar for hosting the conference and considering this paper for the conference ICTIS 2024.

Author's Contribution:

Babbar Sattar Khan presented the idea and modeled the system, Ejaz Ahmed drafted and made the code, Abdul Wadood included the hybridization portion Shahbaz Khan helped in the comparison portion and Husan Ali helped in drafting and modeling.

Conflict of Interest:

No conflict of interest for publishing this manuscript in IJIST.

Project Details:

This work is done without any financial assistance from any source.

References:

[1] A. Pradeep and C. Sreekumar, "Economic Load Dispatch augmented with Environmental Considerations," Proc. - 2nd Int. Conf. Next Gener. Intell. Syst. ICNGIS 2022, 2022, doi: 10.1109/ICNGIS54955.2022.10079786.

[2] S. K. Goyal, N. Kanwar, J. Singh, M. Shrivastava, A. Saraswat, and O. P. Mahela, "Economic Load Dispatch with Emission and Line Constraints using Biogeography Based Optimization Technique," Proc. Int. Conf. Intell. Eng. Manag. ICIEM 2020, pp. 471–476, Jun. 2020, doi: 10.1109/ICIEM48762.2020.9160266.

[3] Nan Li, C. Uckun, E. Constantinescu, J. Birge, K. Hedman, and A. Botterud, "Flexible operation of batteries in power system scheduling with renewable energy," pp. 1–1, Nov. 2016, doi: 10.1109/PESGM.2016.7741730.

[4] C. L. Chiang, "Genetic-based algorithm for power economic load dispatch," IET Gener. Transm. Distrib., vol. 1, no. 2, pp. 261–269, 2007, doi: 10.1049/IET-GTD:20060130.

[5] A. M. Elaiw, X. Xia, and A. M. Shehata, "Hybrid DE-SQP and hybrid PSO-SQP methods for solving dynamic economic emission dispatch problem with valve-point effects," Electr. Power Syst. Res., vol. 103, pp. 192–200, Oct. 2013, doi: 10.1016/J.EPSR.2013.05.015.

[6] A. B. S. Serapião and A. B. S. Serapião, "Cuckoo Search for Solving Economic Dispatch Load Problem," Intell. Control Autom., vol. 4, no. 4, pp. 385–390, Nov. 2013, doi: 10.4236/ICA.2013.44046.

[7] S. Mirjalili, "The Ant Lion Optimizer," Adv. Eng. Softw., vol. 83, pp. 80–98, May 2015, doi: 10.1016/J.ADVENGSOFT.2015.01.010.

[8] S. Banerjee, D. Maity, and C. K. Chanda, "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect," Int. J. Electr. Power Energy Syst., vol. 73, pp. 456–464, Dec. 2015, doi: 10.1016/J.IJEPES.2015.05.036.

[9] V. R. Pandi, B. K. Panigrahi, A. Mohapatra, and M. K. Mallick, "Economic load dispatch solution by improved harmony search with wavelet mutation," Int. J. Comput. Sci. Eng., vol. 6, no. 1/2, p. 122, 2011, doi: 10.1504/IJCSE.2011.041220.

[10] U. A. Salaria, M. I. Menhas, and S. Manzoor, "Quasi oppositional population based global particle swarm optimizer with inertial weights (qpgpso-w) for solving economic load dispatch problem," IEEE Access, vol. 9, pp. 134081–134095, 2021, doi: 10.1109/ACCESS.2021.3116066.

[11] I. Hernando-Gil et al., "Novel Heuristic Optimization Technique to Solve Economic Load Dispatch and Economic Emission Load Dispatch Problems," Electron. 2023, Vol. 12, Page 2921, vol. 12, no. 13, p. 2921, Jul. 2023, doi: 10.3390/ELECTRONICS12132921.

AI-Driven Prediction of Electricity Production and Consumption in Micro-Hydropower Plant

Osman Safi[1], Gul Muhammad Khan[2], Gul Rukh Khattak[2]

[1] Electrical Engineering Department, University of Engineering and Technology Peshawar

[2] National Center of AI, University of Engineering and Technology Peshawar

***Correspondence:** Osman Safi, osman.eep@uetpeshawar.edu.pk

Citation | Safi. O, Khan. G. M, Khattak. G. R, "AI-Driven Prediction of Electricity Production and Consumption in Micro-Hydropower Plant", IJIST, Special Issue pp 125-133, May 2024

Received | May 05, 2024, **Revised |** May 11, 2024, **Accepted |** May 19, 2024, **Published |** May 22, 2024.

Micro hydropower plants must effectively manage demand response to preserve operational firmness and prevent system breakdowns. This research focuses on accomplishing a fine balance while predicting consumption and production, which is significant for upholding system integrity. The study delves into predictive modeling methods to forecast patterns in the production and consumption of electricity over an array of time horizons. We adopted a custom sliding window mechanism, in which actual and predicted values are used to predict the next hour of electricity. We set a baseline to resolve this and examined various algorithms, focusing on RNN-LSTM and CGP-LSTM. The CGP-LSTM forecasting output sequences with different time horizons precisely outperform the RNN-LSTM. The dataset utilized is downloaded from the Kaggle website. 50% of the data is used to train the models, and the rest is used to test the models. This work deals with the complex fluctuations in the demand response system and provides electricity production and consumption predictions. CGP-LSTM model gave a training MAPE of 6.67 (Accuracy of 93.33%) and a testing MAPE of 6.68 (accuracy of 93.32%) for the next three hours; on the other hand, LSTM gave a training MAPE of 6.53 (accuracy of 93.47%) and testing MAPE of 7.46 (accuracy of 92.54%) for the next three hours. The results offer a base for further developments and improvements in the field, drawing attention to more effective and reliable energy management capabilities in micro hydropower plants.

Keywords: Artificial Intelligence; Micro-Hydropower Plant; Time Series Forecasting; LSTM; CGP; Hourly Electricity Prediction

Acknowledgment

Wholeheartedly, I want to thank Almighty Allah for giving me the strength to conduct research. I would like to thank my supervisor, Dr. Gul Muhammad Khan, for his consistent help, instructions, and required mentorship throughout my research task. His knowledge, dedication, and patience have significantly influenced the quality of this work. I am especially grateful to Ms. Gul Rukh Khattak, my assistant supervisor, for her insightful comments, suggestions, and rectifications that enhance the depth of this work. Her assistance and insightful reviews were effective.

Author's Contribution

I (Osman Safi) am the sole writer of this article, I played a key role in each part of the research, from formulation to completion. I have collected the data, developed the models, conducted experiments, and drafted the entire manuscript. while I obtained useful guidance and methodological guidance from my supervisor, Dr Gul Muhammad Khan. He brought novelty to this research by introducing the custom sliding window approach which is adopted in this research. My assistant supervisor Ms Gul Rukh Khattak has also provided feedback on my work, her input greatly refined my work.

Conflict of Interest

The authors declare that there are no conflicts in the publication of this article in IJIST.

Introduction:

The power production capacity of micro-hydropower stations is significantly affected by seasonal changes in water inflow, leading to electricity shortages and inconstant electricity supply. The Jungle-Inn Micro-Hydro Power Plant located at Swat, Kalam region is an example of such a case, where lower water intake during the winter time results in lower power output and persistent power production issues. To tackle this problem, the present load management exercises at Jungle-Inn MHP involve the temporary disconnection of one of the three-phase connections in case of decreased water inflow. And in case of excessive production, they switch on the water heaters to manage extra energy. However, these local practices prove neither safe nor optimal for stable and efficient power distribution systems. Voltage unevenness can result from a failure to forecast consumption changes or excess production, which could cause harm to the electrical infrastructure. The Jungle-Inn micro hydropower station case shows the issues encountered with handling surplus energy, where traditional approaches such as switching on heaters or sometimes diversion of water flow are insufficient. Precise prediction of production and consumption will allow more reliable and efficient load management practices.

Precise electrical consumption forecasting models are requirements because of certain driving motives, the most obvious and serious being climate change. With information being published, carbon dioxide emissions are one of the prime reasons for climate change [1]. The significance of electrical power in everyday life means that forecasting its consumption is increasing in importance. Because of their universal application, a range of articles, study papers, blogs, and videos are accessible. Referring to Weron's [2] prediction techniques, he examines several methods to handle the electrical energy forecast issue, including reduced form, statistical, and artificial intelligence, ML methodologies. It has been observed that Machine learning models frequently surpass many traditional approaches. It can still be split into different computational methods (ML models); one uses deep learning models based on neural networks to explore time-variant data, and the other contains time series models focused on regression techniques [3]. The auto-regressive moving average is one of the regression techniques (ARMA) [4], and the moving average model that is integrated auto-regressive (ARIMA) [5] such models needs to have highly reliable data [3] which might not always be attainable.

Electric company's planning operations rely on accurate models for forecasting electric power consumption. An electric company may use consumption forecasting to assist in making important choices about the production and consumption of electricity, load switching, and industry development. Accurately forecasting consumption requirements is an electric power utility's main task. Energy is considered fundamental to the modern world and a core aspect of economic sustainability. A renewable energy resource supply is essential for economic growth. Most renewable energy sources, including wind, solar radiation, geothermal heat, hydropower, etc., are long-term sustainable. For instance, the hydroelectric turbine systems of large-scale traditional hydroelectric stations, or dams, with water reservoirs offer varying electricity production in response to variations in energy consumption. Atmospheric factors like precipitation and temperature influence small and micro hydropower plants' ability to generate energy. Due to the previously mentioned, the energy produced by these systems varies and must be predicted [6]. Lately, the usefulness of artificial intelligence techniques has overtaken that of traditional approaches, in particular in the domain of electricity consumption forecasting. Notably, ANNs have acquired significant prevalence and have been extensively used in this field [7][8].

As reported by Weron [2], Several prediction methodologies, including reduced-form, statistical, and computational intelligence methods like Machine Learning (ML), have been explored to address the electrical power forecasting challenges. ML models have performed better than conventional approaches in different circumstances. This finding is by the results drawn by Pallabi Paik et al.'s research [9], Which concentrates on stock price prediction;

nevertheless, the research context is separate, and the resemblances between the data trend, data types, and setup in stock price prediction and electricity energy prediction suggest similar methods. Both domains concentrate on time series as the key element. The survey by Pallabi Paik et al. reveals that data-capturing technologies oftentimes outperform traditional techniques in multiple cases. These findings highlight the ability of machine learning approaches, including data mining, to offer more precise and firm predictions for complicated time series data like electricity energy consumption, outperforming the capabilities of conventional methods.

Deep learning models have shown better performance while operating on sequential data that show fickleness and volatility compared to traditional regression approaches. Notably, real-world data is frequently subject to dynamic alterations and instability. By means of experience-based data, research states the performance of artificial neural network models, namely the Long Short-Term Memory model, surpasses regression methods in such schemes. [3][10][11], These results highlight the importance of deploying deep learning approaches, such as LSTM, to achieve more precise and reliable forecasting when dealing with diverse and non-stationary time series data. The approach used in this study is implemented using a typical machine learning project workflow [12] as depicted below (Figure 1).

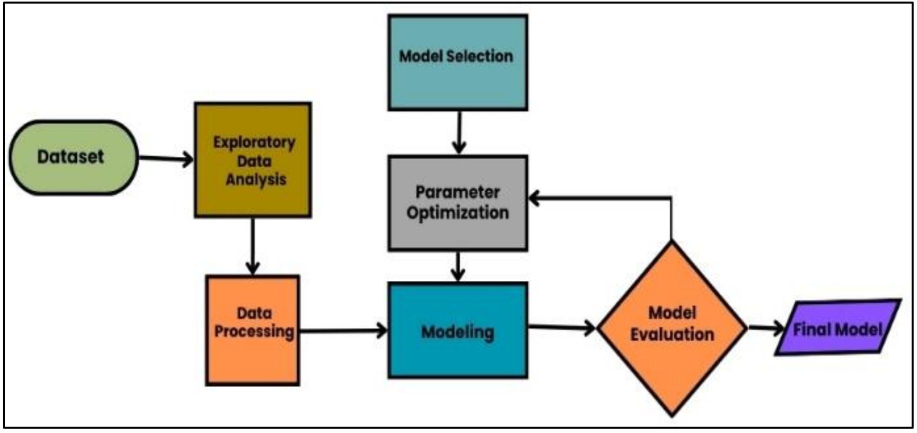
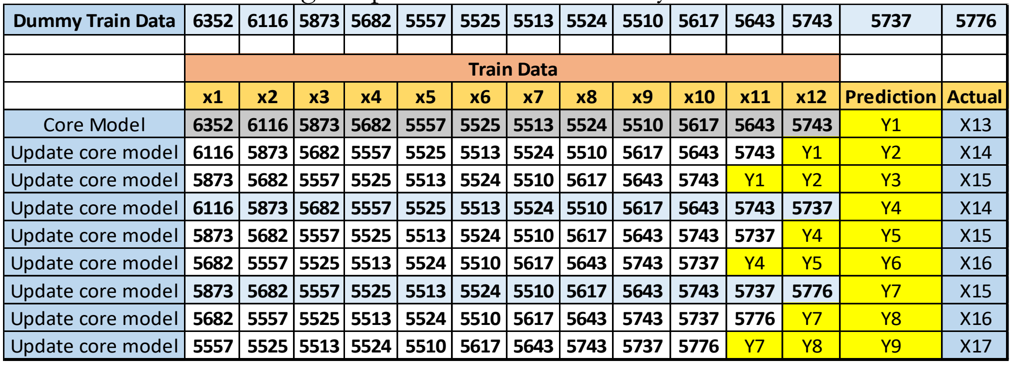


Figure 1: Research methodology adopted for this project [12]

The objectives of this research are to establish predictive models to precisely predict both electricity consumption and production for micro hydropower stations, enabling proactive management strategies to prevent unexpected alterations that could cause the system to fail. To continue to improve the effectiveness of load management strategies, it is vital to focus on the necessity of accurate predictions in light of rising electricity consumption and shifting dynamics in the environment surrounding the production of electricity.



Dummy Train Data	6352	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776
	Train Data													
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	Prediction	Actual
Core Model	6352	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	Y1	X13
Update core model	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	Y1	Y2	X14
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	Y1	Y2	Y3	X15
Update core model	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	X14
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	Y5	X15
Update core model	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	Y5	Y6	X16
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	X15
Update core model	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	Y8	X16
Update core model	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	Y8	Y9	X17

Figure 2: Sliding window mechanism used in this project

We used a novel algorithm CGP-LSTM and adopted a custom sliding window mechanism, our approach entails leveraging the preceding 12 hours of electricity consumption to predict the subsequent 3 hours. We used actual and predicted values in the input sequence. This iterative process involves predicting one hour at a time, with each predicted value being appended to the input sequence for the subsequent prediction. Subsequently, upon completing a prediction cycle, the window of observed values is shifted by unit size, and the process is reiterated until the model is sufficiently trained.

Material and Methods:

The study's methodology follows a stepwise approach, using artificial intelligence algorithms to forecast the consumption and production of micro hydropower plants. The initial phase of research included finding a dataset that has the hourly-based consumption and production of electricity historical data. First, I went to Dare Noor an MHP located in Nangarhar province, Afghanistan the data that I acquired for Dare Noor was insufficient to conduct a successful training. The data for Jungle-Inn was not available at first so I looked it up on the Internet Finally, I downloaded a dataset named "Hourly Electricity Consumption and Production" [13] from the Kaggle website. The dataset has hourly time series of electricity consumption and production data in Romania spanning over four years. All values are in Mega Watts.

After finding the dataset, the next phase involved data pattern inspection and evaluation. By assessing these patterns, we can discover crucial insights about power consumption and production in different conditions. This analytical process enabled us to identify anomalies, trends, and potential fields of improvement. With the understanding obtained from the analysis of the data, we moved on to the development phase. Here, we have utilized different algorithms DNN, CNN, RNN, RNN-LSTM, and CGP-LSTM designed for optimal predictions. The algorithms are destined to forecast the consumption and production of electricity on the basis of historical data. The established models such as DNN, CNN, and RNN are trained, validated, and tested. The dataset is divided into three parts, which are 70% of the data selected for training, 20% of the data chosen for validation, and 10% of the data selected for the testing of the models. The training phase uses the data to train the model on different scenarios and predictable responses. After the model is trained, then it is validated against a separate set of samples from the dataset to ensure its generalization.

We additionally explore autoregressive methods utilizing RNN-LSTM and CGP-LSTM. The models are trained and tested using 50-50 data from the dataset. For the execution of the autoregressive approach, the model input includes both observed and predicted values, to predict the second and third-hours' electricity. However, to predict the first hour the models only used observed values from the dataset achieved by a custom sliding window approach. Eventually, this approach is favored to provide a permanent solution to the issues of load management during times of low water intake and excessive energy production thus enhancing the reliability and efficiency of the micro hydropower station.

Result and Discussion:

The primary goal is to predict power production and consumption for the next three hours. To achieve this a sliding window approach is used, where we use observed and predicted values as an input for prediction. This approach is practical for real-world applications, especially micro-hydro power stations. The dataset that was utilized for this project spans four years, from 2019 to 2023, and has three columns: "Date Time," "Consumption," and "Production" in MWh. We use different types of models for this task. Initially, we start with simple models to establish a baseline. Then, we explore more models, including Convolutional, DNN, and Recurrent Neural Networks. These models make all their predictions in a single shot (all 24 hours prediction at a single shot), unlike a sliding window approach where we predict one hour at a time and then we make the predicted hour part of an input, the input then have observed and predicted values to predict the next hour. In the final phase, we introduce an approach using a custom sliding window technique with LSTM and the novel algorithm called CGP-LSTM. We use the MAE and MAPE to assess the effectiveness of both forecasting models.

baseline model, linear model, dense model, CNN model, and RNN models.

RNN-LSTM:

In the experiments to train the LSTM model, we have used standardization as a normalization technique, we used 50% of the data to train the model and 50% to test the model on it. We used 12 nodes with the ReLU activation function. Here's a summary of the steps in the provided code:

- Import necessary libraries including NumPy, Pandas, and TensorFlow for building and training a neural network model.
- Define a function split-sequence (sequence, n_steps) to split a univariate time series sequence into input-output pairs with a specified number of time steps (n_steps).
- Set the number of time steps (n_steps) for sequence splitting and define a neural network model (core model) using Tensor Flow's Kera's API.
- Compile the core model with the Adam optimizer and mean squared error loss.
- Define column names for the result Data Frames for both training and testing.
- Create empty Data Frames Result Train and Result Test to store the training and testing results, respectively.
- Define a function Model Train (sequence, model) to train the neural network model on a given sequence. This function fits the model to the data, makes predictions, and appends the predicted value to the sequence.
- Define a function seq_train (raw) to perform training iteratively. It calls Model Train for training the model on subsequences of the training data, appends the results to Result Train, and updates the input sequence for the next iteration.
- Set the number of iterations and the raw data length initial values based on the length.
- Iterate through the training data, extracting subsequences and applying the seq_train function.
- Extract test data from the remaining portion of the raw data.

Figures 3 and 4 display the model's training and testing curves for 7000 data points.

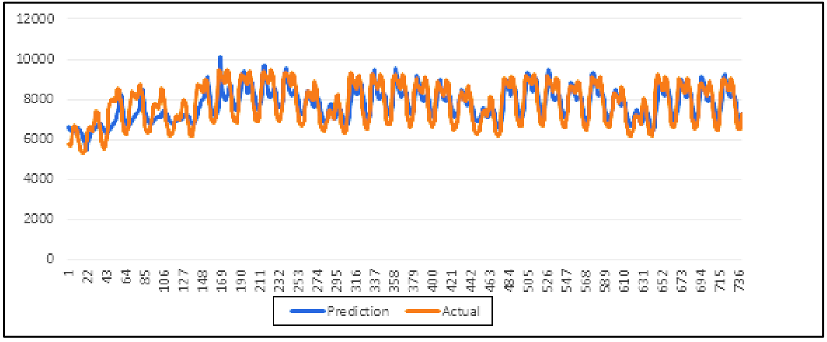


Figure 3: The training result of the model for 7k rows

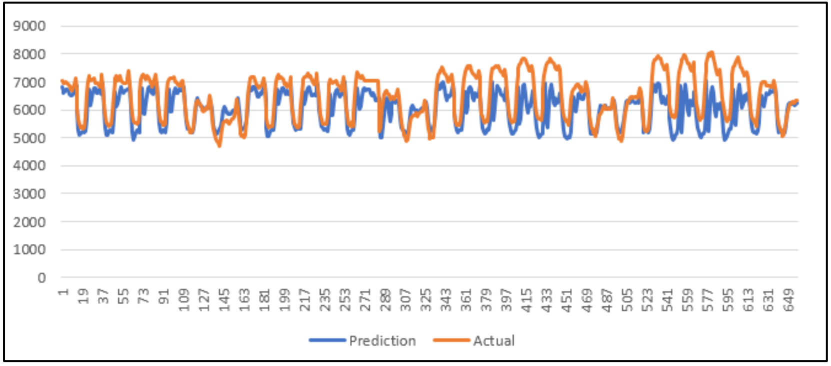


Figure 4: The testing result of the model for 7k rows

The following Figures 5 and 6 show the model performance when trained on 4k data samples.

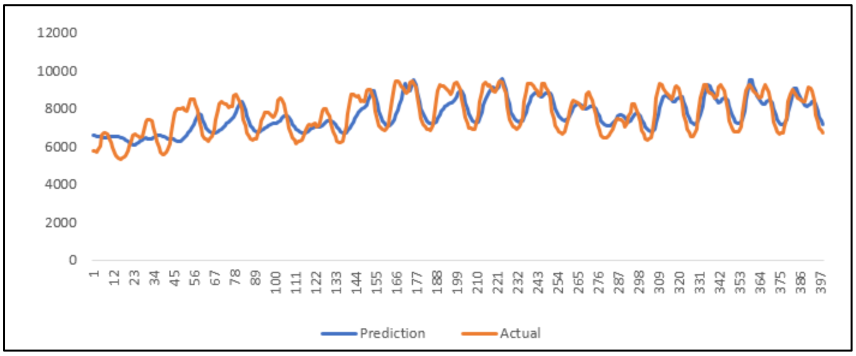


Figure 5: The training curves for 4000 rows

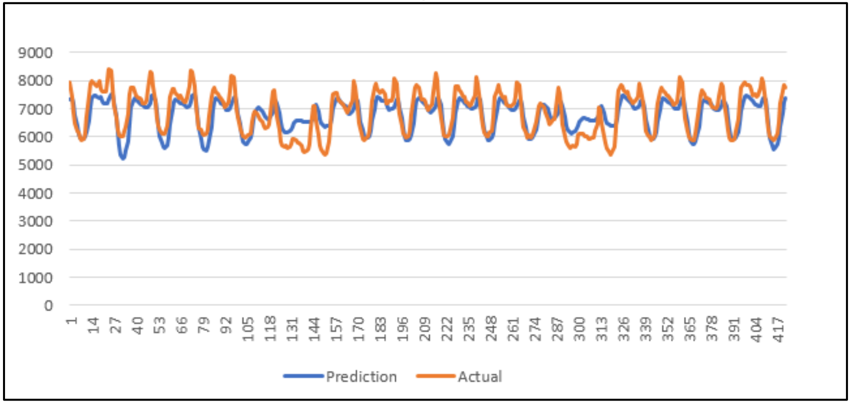


Figure 6: Testing curves for 4k rows

Table 1: RNN-LSTM model results

Model	Training/Testing	Input/output	MAE	MAPE
4k rows	Train C	12 inputs 1 output	363.47	4.86
	Test C		324.41	5.04
	Test P		340.91	5.21
	Train C	12 inputs 3 outputs	489.87	**6.53**
	Test C		450.11	**7.46**
	Test P		445.65	6.75
7k rows	Train C	12 inputs 1 output	279.813	3.92
	Test C		465.18	7.01
	Test P		392.77	6.69
	Train C	12 inputs 3 outputs	382.16	5.36
	Test C		678,81	10.2
	Test P		586.93	9.95

Table 1 presents a performance comparison of both of the experiments. The models are trained and tested on 4k and 7k data samples, providing one- and three-hour predictions into the future. The errors are presented in Megawatt hours (MWh). The model is trained and tested on both “Consumption” and “Production,” as you can see in the above table 1. The model is run on 4k and 7k rows from the dataset. “Train C” represents that the model is trained on “Consumption” historical data, “Test C” indicates that the model is tested on “Consumption” data, and “Test P” represents that the model is tested on “Production” data. In the experiment with 4k data points, we obtained an accuracy of **93.47 %** for training and **92.54%** for testing to predict the next three hours. The best results in the table are shown in bold text.

CGP-LSTM:

The CGP-LSTM training was carried out on the production data, and validation was performed on both the training and testing data. Figure 7 shows the actual and predicted values; the predicted values curve follows the actual values curve very closely.

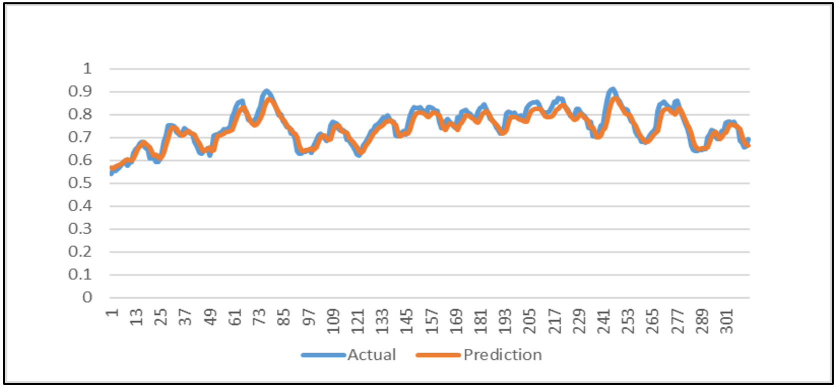


Figure 7: Testing result curve with CGP-LSTM algorithm

Table 2: Results of the CGP-LSTM model

Model	Training/ Testing	Input /Output	MAE	MAPE
50 Nodes Full datasets	Train- P	6 input 6 Output	0.064384	9.407604
	Test- P		0.062959	9.409408
	Test- C		0.048834	7.790334
	Train- P	6 input 3 output	0.045892	6.679367
	Test- P		0.044945	6.684573
	Test- C		0.035013	5.558505
	Train- P	6 input 1 one output	0.031806	4.589071
	Test- P		0.031083	4.583583
	Test- C		0.024525	3.869808

CGPLSTM used 50 nodes, although the actual nodes used in the model are fewer. The model is trained on the “Production” column from the dataset then it is tested on both “Production” and “Consumption”, Train- P (Model trained on “Production”), Test-P (Model test on “Production), and Test –C (Model test on “Consumption”). As you see in Table 2, we have six inputs and 6 outputs, which means that based on the previous six hours, we are predicting the next 6 hours. Then we have three inputs and three outputs, which means that the model predicts the next three hours based on the previous six hours which have both observed and predicted values as explained in the custom sliding window approach. Finally, we have six inputs and one output, which means that we are predicting the next hour based on the previous six hours. When the model was first trained and tested on the entire dataset, the outcomes are displayed in Table 2.

Conclusion:

In conclusion, the various approaches and algorithms employed in this study present unique advantages and drawbacks, contributing to a detailed understanding of their applicability. Convolutional Neural Networks and Recurrent Neural Networks in the prediction process offer a robust foundation for capturing spatial and temporal dependencies in the data. CNNs stand out in extracting spatial features, whereas RNNs are adept at modeling temporal patterns. However, the dependency on vast training data and the possibility of overfitting are noteworthy drawbacks. Moreover, the autoregressive nature of the models presents challenges in precisely predicting distant subsequent values. Finding a balance between model complexity and forecasting accuracy is a key concern across all approaches. These insights help the continued discussion about the optimization of predictive modeling for electricity consumption and production prediction, leading the way for future improvements and refinements in the field. On the other hand, with its iterative prediction methodology using RNN-LSTM and CGP-LSTM, the Custom Sliding Window approach excels in its adaptability in yielding an output of varying lengths. This adaptability verifies advantageous in scenarios demanding several output predictions. However, the recurrent nature of the approach may present higher computational complexity. Comparing the results of these two established models, CGP-LSTM gave good results compared to RNN-LSTM. However, it must be mentioned that both models can be improved by experimenting with different combinations of hyperparameters. CGP-LSTM gave a training MAPE of **6.67** and a testing MAPE of **6.68** for the next three hours; on the other hand, RNN-LSTM gave a training MAPE of **6.53** and a testing MAPE of **7.46** for the next three hours. We have validated the RNN-LSTM model on the Jungle-Inn dataset, which contains hourly data spanning 65 days. The results are promising, and in the future, when more data is available, the same method can be extended.

References:

[1] A. K. Singh, Ibraheem, S. Khatoon, M. Muazzam, and D. K. Chaturvedi, "Load forecasting techniques and methodologies: A review," ICPCES 2012 - 2012 2nd Int. Conf. Power, Control Embed. Syst., 2012, doi: 10.1109/ICPCES.2012.6508132.

[2] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," Int. J. Forecast., vol. 30, no. 4, pp. 1030–1081, Oct. 2014, doi: 10.1016/J.IJFORECAST.2014.08.008.

[3] M. Rafik, A. Fentis, T. Khalili, M. Youssfi, and O. Bouattane, "Learning and Predictive Energy Consumption Model based on LSTM recursive neural networks," 4th Int. Conf. Intell. Comput. Data Sci. ICDS 2020, Oct. 2020, doi: 10.1109/ICDS50568.2020.9268733.

[4] J. F. Chen, W. M. Wang, and C. M. Huang, "Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting," Electr. Power Syst. Res., vol. 34, no. 3, pp. 187–196, Sep. 1995, doi: 10.1016/0378-7796(95)00977-1.

[5] G. Juberias, R. Yunta, J. Garcia Moreno, and C. Mendivil, "New arima model for hourly load forecasting," Proc. IEEE Power Eng. Soc. Transm. Distrib. Conf., vol. 1, pp. 314–319, 1999, doi: 10.1109/TDC.1999.755371.

[6] "Forecasting of a hydropower plant energy production with Fuzzy logic Case for Albania", [Online]. Available: https://www.jmest.org/wp-content/uploads/JMESTN42352182.pdf

[7] K. Kandananond, "Forecasting Electricity Demand in Thailand with an Artificial Neural Network Approach," Energies 2011, Vol. 4, Pages 1246-1257, vol. 4, no. 8, pp. 1246–1257, Aug. 2011, doi: 10.3390/EN4081246.

[8] N. Amjady and F. Keynia, "A New Neural Network Approach to Short Term Load Forecasting of Electrical Power Systems," Energies 2011, Vol. 4, Pages 488-503, vol. 4, no. 3, pp. 488–503, Mar. 2011, doi: 10.3390/EN4030488.

[9] P. P. et Al., "Systematic analysis and review of stock market prediction techniques," Comput. Sci. Rev., vol. 34, p. 100190, Nov. 2019, doi: 10.1016/J.COSREV.2019.08.001.

[10] E. Yuniarti, Nurmaini, B. Y. Suprapto, and M. Naufal Rachmatullah, "Short Term Electrical Energy Consumption Forecasting using RNN-LSTM," ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding, pp. 287–292, Oct. 2019, doi: 10.1109/ICECOS47637.2019.8984496.

[11] F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, "Short-Term Prediction of Residential Power Energy Consumption via CNN and Multi-Layer Bi-Directional LSTM Networks," IEEE Access, vol. 8, pp. 123369–123380, 2020, doi: 10.1109/ACCESS.2019.2963045.

[12] E. Izgi, A. Öztopal, B. Yerli, M. K. Kaymak, and A. D. Şahin, "Determination of the Representative Time Horizons for Short-term Wind Power Prediction by Using Artificial Neural Networks," Energy Sources, Part A Recover. Util. Environ. Eff., vol. 36, no. 16, pp. 1800–1809, Aug. 2014, doi: 10.1080/15567036.2011.561274.

[13] F. A. Olivencia Polo, J. Ferrero Bermejo, J. F. Gómez Fernández, and A. Crespo Márquez, "Failure mode prediction and energy forecasting of PV plants to assist dynamic maintenance tasks by ANN based models," Renew. Energy, vol. 81, pp. 227–238, Sep. 2015, doi: 10.1016/J.RENENE.2015.03.023.

Combatting Illegal Logging with AI-powered IoT Devices for Forest Monitoring

Abdullah Khan, Hamza Ali, Maham Jadoon, Zain Ul Abideen, Nasru Minuallah

Computer Systems Engineering University of Engineering and Technology, Peshawar, Pakistan

***Correspondence**: 20pwcse1916@uetpeshawar.edu.pk, hamzaali.dcse@gmail.com, 20pwcse1875@uetpeshawar.edu.pk, zainikhan3434@gmail.com, n.minallah@uetpeshawar.edu.pk.

Citation | Khan. A, Ali. H, Jadoon. M, Abideen. Z, Minuallah. N, "Combatting Illegal Logging with AI Powered IoT Devices for Forest Monitoring", IJIST, Special Issue pp 134-142, May 2024

Received| May 05, 2024, **Revised|** May 12, 2024, **Accepted|** May 18, 2024, **Published|** May 22, 2024.

This research presents a comprehensive strategy for tackling illegal logging by leveraging Artificial Intelligence (AI) and Internet of Things (IoT) technologies. In high-risk forestry areas, sensors-equipped Internet of Things devices are used to continuously monitor and detect the sound of the surroundings. The AI component uses machine learning methods to identify potential unlawful logging activities by accurately detecting and distinguishing sound patterns associated with chainsaw and logging operations such as tree cutting and also detecting natural disasters like wildfires. When such activities are detected by these smart AI-powered IoT devices installed in the forest, real-time notifications are generated after such activity which allows surrounding enforcement agencies, such as the forest department, to intervene promptly. By providing a targeted and prompt solution to the issue of illicit logging, this strategy supports biodiversity preservation and sustainable forest management.

Keywords: Forest Monitoring; AI Against Illegal Logging; Real-time Alerts; Environmental Conservation; IoT for Anti-Logging.

Introduction:

These Forests are woven into our history. It is the source of fresh air, strong materials for houses, fresh water, and fertile soil for growing crops. But forests are being challenged. Illegal logging driven by uncontrolled greed for money challenges them. It throws nature's balance off give and take. The results are damaging to our ecosystems causing disasters like habitat losses as well as social entity indifferences. Deforestation initiates climate imbalance messing up the weather regime as well as disturbing the process of equilibrium in the ecosystem. All this careless act performed risks the same resources for its gain. Big-time cutting of the trees combined with illegal chopping of the wood provisions to the problems in saving worldwide vine variance, keeping equal ecology, and protecting the homes of countless wild beings. This destroys the system of world waters, reduces nature's distinction by the destruction of homes, and creates a lot of conflicts and other effects on social matters. Alarmingly, every year an expanse of country-sized land is afflicted by deforestation either from illegal logging or fires, especially in growing countries like Brazil and Indonesia. The records of vast forests have always been a challenge in the history books. Walking through dense woods is risky, requires a number of people, and usually leaves out far places. Regular surveillance like cameras can help, but they don't capture everything and can miss things as an action is taking place. The quiet noises in the forest, often covered by leaves and distance, can slip on by. This allows lasting damage to happen before we can step in. What if tech could boost these soft sounds, turning them into warnings of unfairness and signals of hope for our vanishing woods? Imagine hearing - really hearing – the hidden messages in the moving leaves. This is the big dream driving our project. We try to appeal to the cutting edge of the Internet of Things (IoT) and voice classification power [1]. We want to stand guard for peaceful green defenders. Now let's imagine intelligent sensors scattered in a forest, as shown in Figure 1. They're run by the chip ESP32 [2], a powerful, widely-used engine. Their job is to beckon and learn, with advanced tech that recognizes voices. They're always on the lookout for injurious things like noisy chainsaws, falling logs, or people talking up to no good. Before, these damaging noises would go unnoticed. Now they are translated into useful information. Straightaway, this information fires through the unseen paths of the IoT to a main cloud base. The soft sounds would rise into a loud noise in the woods, slowly but surely. Alerts go off immediately, leading forest rangers to exactly where the action was improper. Now, lawbreakers who prowl and come unseen are seen; their quiet deeds demand responsibility loudly and effect action, reverberating in the digital world.

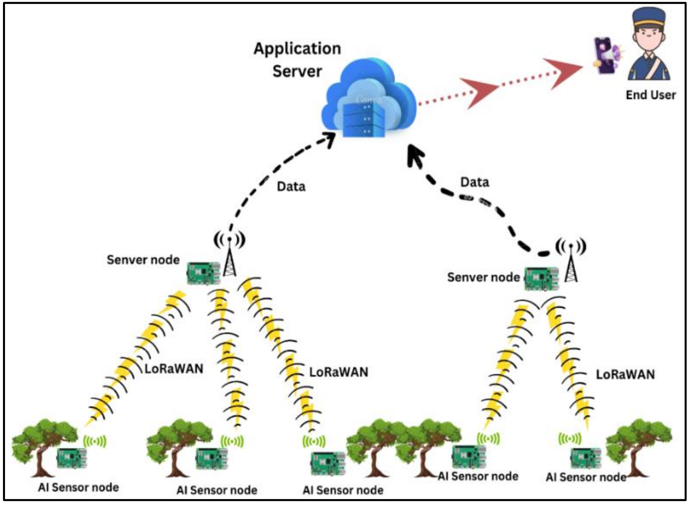


Figure 1: Use Case Diagram

Illegal logging is a pressing threat to global forests, imperiling their sustainability. A solution leveraging artificial intelligence (AI) and Internet of Things (IoT) devices [3] is proposed to combat this issue. The project aims to reduce unauthorized logging through real-time monitoring, and deploying IoT devices strategically. AI algorithms [4] will analyze data, identifying patterns and potential logging incidents. Preserving biodiversity is a key focus, preventing deforestation from illegal logging. The project enables data-driven decision-making for forest management, emphasizing collaboration among stakeholders. Capacity-building initiatives prioritize enhancing local communities' capabilities in monitoring and combating illegal logging. Awareness initiatives highlight the impacts of illegal logging, emphasizing forest conservation. Specific objectives include deploying IoT devices [3], implementing AI-powered analysis, and fostering collaboration. Advocacy for supportive policies is integral to combating illegal logging. Community engagement involves training local communities in monitoring and protecting forests. Ongoing evaluation ensures the effectiveness of AI-powered IoT devices in combating illegal logging.

Literature Review:

In the world of audio classification, which has made great strides since the last decades of the 20th century, applications have been found for distance learning and digitized libraries [5]. The new generation methodologies that have been pioneered include the Hidden Markov Model by Lu Jian with its intricate sorting of audio files and also label-free approach by Zhao Xueyan. Li, S.Z provides a refined sound identification system using Support Vector Machines (SVMs) and Mel Frequency Cepstral Coefficients (MFCC) in the domain of sound understanding [6]. Features from the time and frequency domain form the focus area of the study for sorting the sounds. Using Short-Time Average ZCR and Short-Time Energy in the time domain, the important features for sound or silence determination are extracted on the paper. In the frequency domain, there exist repeated features like the Centroid of the Audio Frequency Spectrum and Sub-Band Energy Ratio that give useful pointers. The idea of the suggested method of sound sorting is grounded in SVM, and thus it requires the careful preprocessing of audio samples and many features to be extracted, including MFCC. The system proved effectual as it was tested for a mixed set of 2500 sound pieces and obtained an average commendable accuracy of 92.14%. In conclusion, this article reveals some insight into the historical context of audio sorting, a deep sound sorting system mixing time and frequency traits, augmented with SVM processing. Tiny Machine Learning (Tiny ML) challenges the problem of pitching machine learning into microcontroller integration in domains such as sound recognition [5], image classification, and motion classification/anomaly detection [1]. In sound recognition, the most powerful keyword spotting (KWS) takes center stage whereby individuals gain hands-free access control for better user accessibility [6]. In projects that depict exemplary Tiny ML applications, such as the project with XIAO nRF52840 Sense, implementation of Tiny ML applications as feasible is demonstrated even in cases of limited resources. Regarding other projects, Tiny ML has an application in image classification where XIAO ESP32S3 Sense additionally supports camera access and therefore can be used in projects related to object recognition too. Tiny ML enhanced by an Ensenso Pico for motion classification/anomaly detection enabled the real-time analysis of sensor data and has been exemplified in an innovative cargo damage prevention tool utilizing XIAO nRF52840 Sense. These, as well as those purpose-developed in this area, illustrate the transforming potential of Tiny ML hence empowering low-power embedded systems across various sectors.

We provide audio classification in this literature review [1], from understanding basic processes to the application of advanced machine learning models. It focuses on voice conversion into quality digital format, stress on analysis through spectrogram and other tool selections like Mel spectrogram and Short-Time Fourier Transform (STFT). Within the study, the loudness differences are normalized using techniques, and ideas of Data Augmentation and also the use of mechanisms such as Depth Separable Convolutional Neural Networks (DS-CNN) and Fast GRNN are presented to improve the overall capabilities of listening to audio tasks.

This investigation into Enhanced Sound Recognition (ESR) with the application of Deep Learning (DL) techniques underscores the emergent interest in leveraging DL in enhancing Machine Learning (ML) models [5]. This study will review the different approaches toward sound recognition from spectrograms/MFCCs as inputs to ANN/CNN classifiers and raw waveform directly. In one of the most illustrative experiments, sound classifying in a Wireless Sensor Network (WSN) with Raspberry Pi (RPi) nodes is performed using a Convolutional Neural Network (CNN). The research has proven that even feature extraction and sound classification can be executed on embedded high-level devices, so it highly accentuates searching computational capabilities [6]. Addressing the dearth in discourse of the benefits and the feasibility of the introduction to machine learning techniques of sound classification [1]. Other proposed solutions for the Internet of Things (IoT) are specially designed models like CNN [3], with a separate focus on model inference time and resource efficiency on low-power microcontrollers. Hardware accelerators such as Tensor Processing Units (TPUs) and Field Programmable Gate Arrays (FPGAs) [7] are exploited to enhance the performance of audio applications of Deep Learning. Lastly, the study concludes with a better approach for looking at damage identification and location in multiple classes by considering the various levels of severity of damages as well as scenarios. These techniques along with the windowing data augmentation and a unique majority voting technique accompanied by the global CNN-1D model prove to be adequate for dealing with the problems of limited data.

Methodology:

The methodology consists of a series of related steps from the data collection to the real-world deployment designed to develop an effective system to fight against the issue of illegal logging with the help of AI-enabled IoT devices [8]. In our proposed methodology cutting-edge technology is used by utilizing the low-power microcontroller and making it a powerful tool with the use of Machine learning. Like in the first step through the data collection and preprocessing before training the supposed model.

Data Collection and Preprocessing:

First, a rich dataset comprising 75 samples of each of the following audio classes - Silence, Axe, Chainsaw, and Fire has been collected from different websites. Each audio recorded voice has 5 seconds and the total data collected is shown in Table [1]. The process involved in data collection uses Mel-filter bank energy features that are very important in audio classification. These features provide vital information about environmental sounds and extract speech signals for use in recognition tasks like those associated with illegal logging.

Table 1: Data Collection

Data Collected	Train / Test Split
25m 0s	79% / 21%

Training Deep Learning Model:

In this step, the convolutional neural network model (CNN) is trained and selected filter bank energy features (MEF) as a feature to detect the surrounding voices and make meaningful insight from recorded voices. In Step 1, obtain a balanced dataset where each class would contain at least 75 samples: Silence, Axe, Chainsaw. The audio samples consisted of files with durations of exactly 5 seconds each. This dataset is important in training a strong classifier for the accurate classification of environmental sounds associated with the activities of illegal logging.

Mel-filter bank Energy Features (MEF):

The Mel-filter bank energy features (MEF) have been used as the choice feature for this audio classification. The method of extraction of the features used in this work is to capture the frequency content of the audio signals using the Edge Impulse Platform [9]. This process can be achieved by using a filter bank to divide this audio signal into several narrow frequency bands. The resulting energy values across these bands serve as essential input features for the machine learning model. Parameters for MEF are listed below

- **Frame Length:** 0.02 seconds
- **Frame Stride:** 0.01 seconds
- **Filter Number:** 40
- **FFT Length:** 256
- **Low Frequency:** 0

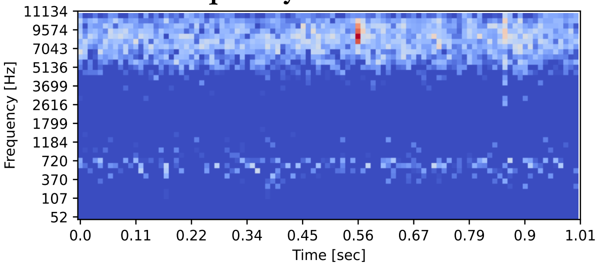


Figure 2: MEL Energies (DSP Output)

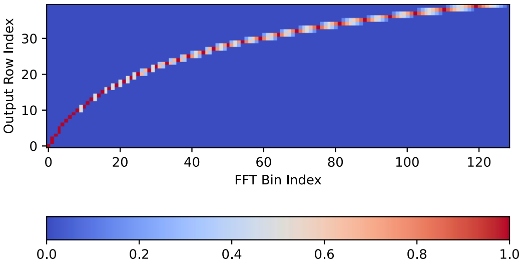


Figure 3: FFT Bin Weightings

Figures 2 and 3 illustrate the Mel Eng Mel Energies and FFT Bin Weighting of some of the voices from a data set. These parameters should be chosen very carefully in such a way as to optimize the extraction of Mel filter bank [10] energy features from the audio data, such that ultimately a set of very representative and informative features is obtained for the purpose of training the deep learning models.

Neural Network Architecture:

The neural network architecture was designed to effectively process the Mel- filter bank energy features classifying the audio into their corresponding classes. The architecture includes:

- **Input Layer (3,960 features):** The model reshapes the features so that it can be processed by the model.
- **Reshape Layer (40 columns):** 40 filters that are part of the MEF.
- **1D Conv/Pool Layer (8 neurons, 3 kernel size, 1 layer):** First incidence of convoluting.
- **Dropout (Rate 0.25):** Introduces regularization to prevent overfitting.
- **Conv/Pool Layer (16 neurons, 3 kernel size, 1 layer):** Extra convolutions for better feature extraction.
- **Dropout (Rate 0.25):** Further regularization.
- **Flatten Layer:** It converts the output into a one-dimensional array/tensor for the next series of layers.
- **Additional Layer**: Introduces complexity and abstraction for improved model performance.
- **Output Layer (4 classes):** Represents the four classes (Silence, Axe, Chainsaw, Fire).

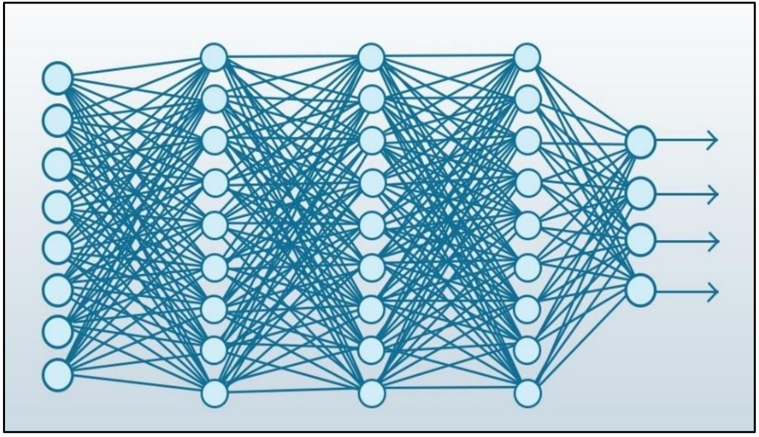


Figure 4: Model with Hidden Layer

Table 2: Evaluation Metrics

Accuracy	Loss
96.2%	0.10

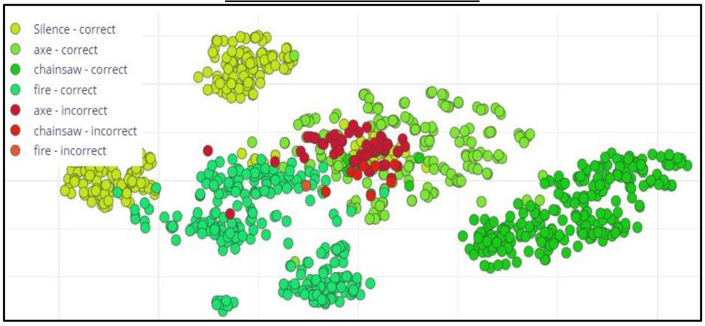


Figure 5: Data Explorer

Training Settings:

- **Number of Training Cycles:** 100
- **Learning Rate:** 0.005

In processing that, it is possible to realize the consensus and accuracy during the training process to see that the network learns effectively the Mel-filterbank energy features. Figure 5 below shows the model training accuracy which also shows the incorrect classification with red points.

Integration Model with ESP32 Microcontroller as a Sensor Node:

Using the trained model that we have developed in the previous section, the model is exported as an Arduino and deployed to the ESP32 [2] (ESP32 is a series of low-cost, low-power systems on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth) microcontroller functioning as a sensor node. For the recording of audio data of surroundings, an enabled INMP441 microphone has been used The INMP441 is a high-performance, low-power, digital-output, omnidirectional MEMS microphone. The recorded audio data, therefore undergoes processing by use of a device machine learning model. This integration ensures a classification of sounds that falls into categorization related to the activities of illegal logging by ESP32 autonomously.

Table 3: On Device Performance

Interfacing Time	Peak RAM Usage	Flash Usage
21ms	10.6Kb	32.6Kb

In Table 3, on-device performance is illustrated. On ESP32 [2] SoC. The latter uses LoRa communication for a secure transmission of the voice identified, and node location (longitude, latitude). In this scenario, the ESP32 being a node sensor, sends its data, using LoRa technology to the central server node which is another ESP32, and works as a server as it is handling the database of that project. This way of communication is needed to establish networking abilities over wide forested areas so that monitoring can occur in real-time.

Server Node (ESP32) and Database Integration

In this step, ESP32 works as the server node and is to receive signals from the ESP32 sensor nodes. The server node connects Firebase which is a cloud platform by Google. Firebase is used for real-time data storage, updating node locations, and other relevant information that exists in the field of operations. This integration ensures a centralized and accessible repository of data. The mobile application is built to interface with the Firebase database and give real-time notifications on detected activities and node locations. It is the user interface for the app, where different users like members of the local community, law enforcement agencies, as well as environmental groups interact. Figure 6 illustrates the project top overview and clear flow of our project. This is the real-world-based scenario that is used to test how good the model is and has trained on several localization and classification tasks in a real-world case. The testing phase also tests whether LoRa communication is robust as well as ensuring data moves from LoRa to Firebase successfully.

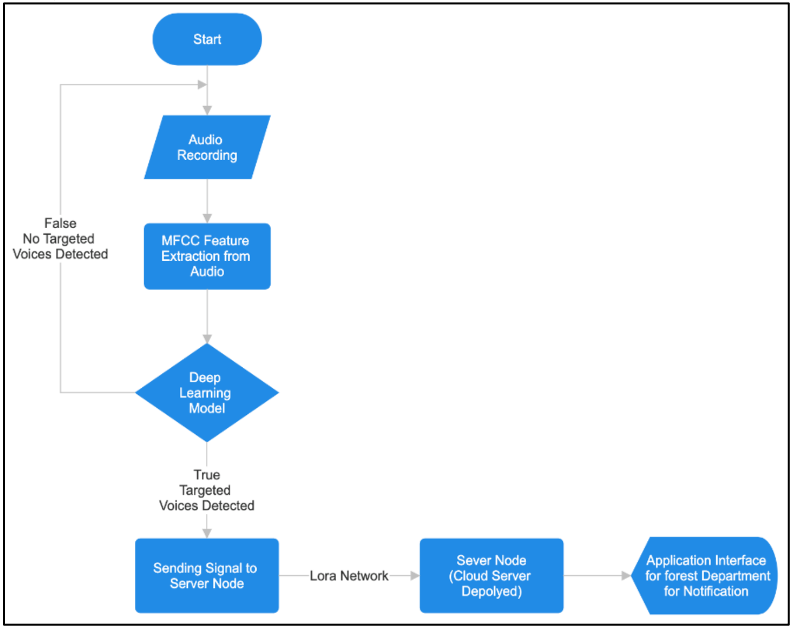

Start
Audio Recording
MFCC Feature Extraction from Audio
Deep Learning Model
False
No Targeted Voices Detected
True
Targeted Voices Detected
Sending Signal to Server Node
Lora Network
Sever Node (Cloud Server Depolyed)
Application Interface for forest Department for Notification

Figure 6: Workflow diagram

Results:

The model performed exceptionally well in various aspects. It achieved an accuracy of 96.2% on the validation set along with low loss, that is, 0.10. The confusion matrix reflected the accurate different class abilities of the model, with high percentages for SILENCE, AXE, CHAINSAW, and FIRE. The corresponding efficiency in terms of F1 scores was further validated through the table below. During the on-device performance, the model showed efficient inferencing that took very short of 21 MS. Minimal resource usage was identified by peak RAM at 10.6K and flash usage at 32.6K as already shown in Table 3. This makes for deployment in real-time constrained devices.

Table 4: 82% Accuracy

	Silence	**Axe**	**Chainsaw**	**Fire**	**Uncertainty**
Silence	73.5%	9.8%	0%	0%	16.7%
Axe	1.8%	70.2%	0%	4.0%	24.1%
Chainsaw	0%	2.9%	95.2%	0%	1.9%
Fire	0%	1.2%	0%	98.4%	0.4%
F1 score	0.84	0.77	0.98	0.96	-

Table 5: Model Efficiency in Utilization

	MFE	**Classifier**	**Total**
Latency	450ms	21ms	471ms
Ram	20.4K	10.6K	20.4K
Flash	-	32.6K	-
Accuracy	-	-	-

In each of the testing scenarios, the model maintained a high level of accuracy at 82.57%. The performance behavior of the classification algorithm in classifying into groups the audio samples under the different classes including instances of uncertainty was depicted by the confusion matrix. The F1 scores for each class also confirmed the reliability of the model under different setup conditions during testing. The latencies distribution during the inference process of the model displays as shown in Table 5 a well-distributed figure, with an overall of 471 MS made up of 450 MS that are contributed by the Mel-filter bank energy features (MFE), another 21 MS for the classifier. Resource usage in terms of RAM and FLASH stood at 20.4K and 32.6K respectively unveiling further efficacy of the model in memory allocation as well as storage. Overall, the results underscore the strong generalization ability of the model showing it as being promising in a real-world setting combating illegal logging with its effective audio classification on IoT devices.

Conclusion:

The application of the Internet of Things (IoT) and Artificial Intelligence (AI) technology [4] offers an achievable approach to combating illegal logging, which is a major danger to world forests and biodiversity. We have demonstrated an innovative strategy for real-time monitoring and detecting actions related to illegal logging, such as chainsaw running and tree cutting down, through the deployment of smart sensors installed with AI algorithms [4]. By harnessing advanced machine learning techniques which include convolutional neural networks (CNNs) trained on Mel-filter bank energy features (MEF), the developed model attained a spectacular accuracy of 96.2% on the validation set. The model efficient inferencing process with minimal resource usage on low-powered microcontrollers like the ESP32 which spectacle its suitability for real-world deployment in forest environments. Integration with IoT devices like ESP32 [2] microcontroller serving as a sensor node enables autonomous classification of environmental noises related to illegal logging activities. Employing LoRa communication technology, data transmission to a centralized server node deployed with the Firebase database ensures real-time data storage and accessibility. The model's overall accuracy of 82.57% as well as successful memory allocation and memory resource usage have been confirmed by the testing scenarios' results, which also highlight the model's robustness and generalization capacity. The model is a useful tool for monitoring and preventing illicit logging in real-world forest situations because of its low-latency inference process and ability to reliably classify audio samples into distinct classifications. In conclusion, our research demonstrates the effectiveness of AI-powered IoT devices in preserving biodiversity promoting sustainable forest management, and also safeguarding against the dangerous effects of illegal logging. By providing timely detection and intervention capabilities this strategy contributes to the conservation of forests, ecosystems, and a lot of benefits they provide to humanity and the planet as a whole.

References:

[1] C. Anjanappa, S. Parameshwara, M. K. Vishwanath, M. Shrimali, and C. Ashwini, “AI and IoT based Garbage classification for the smart city using ESP32 cam,” Int. J. Health Sci. (Qassim)., vol. 6, no. S3, pp. 4575–4585, May 2022, doi: 10.53730/IJHS.V6NS3.6905.

[2] “ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems.” Accessed: May 06, 2024. [Online]. Available: https://www.espressif.com/en/products/socs/esp32

[3] S. B. Calo, M. Touna, D. C. Verma, and A. Cullen, “Edge computing architecture for applying AI to IoT,” Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017, vol. 2018-January, pp. 3012–3016, Jul. 2017, doi: 10.1109/BIGDATA.2017.8258272.

[4] M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim, and Y. Kim, Eds., “Current Approaches in Applied Artificial Intelligence,” vol. 9101, 2015, doi: 10.1007/978-3-319-19066-2.

[5] M. Merenda, C. Porcaro, and D. Iero, “Edge Machine Learning for AI-Enabled IoT Devices: A Review,” Sensors 2020, Vol. 20, Page 2533, vol. 20, no. 9, p. 2533, Apr. 2020, doi: 10.3390/S20092533.

[6] X. Yang, H. Xing, and X. Su, “AI-based sound source localization system with higher accuracy,” Futur. Gener. Comput. Syst., vol. 141, pp. 1–15, Apr. 2023, doi: 10.1016/J.FUTURE.2022.10.023.

[7] D. Roggen, R. Cobden, A. Pouryazdan, and M. Zeeshan, “Wearable FPGA Platform for Accelerated DSP and AI Applications,” 2022 IEEE Int. Conf. Pervasive Comput.

Commun. Work. other Affil. Events, PerCom Work. 2022, pp. 66–69, 2022, doi: 10.1109/PERCOMWORKSHOPS53856.2022.9767398.

[8] I. Ahmad, S. E. Shariffudin, A. F. Ramli, S. M. M. Maharum, Z. Mansor, and K. A. Kadir, “Intelligent Plant Monitoring System Via IoT and Fuzzy System,” 2021 IEEE 7th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2021, pp. 123–127, Aug. 2021, doi: 10.1109/ICSIMA50015.2021.9526312.

[9] “Edge Impulse - The Leading edge AI platform.” Accessed: May 06, 2024. [Online]. Available: https://edgeimpulse.com/

[10] S. R. Madikeri and H. A. Murthy, “Mel filter bank energy-based slope feature and its application to speaker recognition,” 2011 Natl. Conf. Commun. NCC 2011, 2011, doi: 10.1109/NCC.2011.5734713.

RESEARCH & INNOVATION ESSA DIVISION
IJIST

NEUROSCAN: Revolutionizing Brain Tumor Detection Using Vision-Transformer

Kamran Khan, Najam Aziz, Afaq Ahmad, Munib-ur-Rehman, Yasir Saleem Afridi
Department of Computer Systems Engineering, University of Engineering & Technology, Peshawar, Pakistan

*** Correspondence:** Kamran Khan (20pwcse1895@uetpeshawar.edu.pk)

Citation | Khan. K, Aziz. N, Ahmad. A, Rehman. M, Afridi. Y. S, “NEUROSCAN: Revolutionizing Brain Tumor Detection Using Vision-Transformer”, IJIST, Special Issue pp 143-151, May 2024

Received | May 14, 2024, **Revised |** May 18, 2024, **Accepted |** May 20, 2024, **Published |** May 23, 2024.

Brain tumor detection is a pivotal component of neuroimaging, with significant implications for clinical diagnosis and patient care. In this study, we introduce an innovative deep-learning approach that leverages the cutting-edge Vision Transformer model, renowned for its ability to capture complex patterns and dependencies in images. Our dataset, consisting of 3000 images evenly split between tumor and non-tumor classes, serves as the foundation for our methodology. Employing Vision Transformer architecture, we processed high-resolution brain scans through patching and self-attention mechanisms. The model is trained through supervised learning to perform binary classification tasks. Our employed model achieved a high of 98.37% in tumor detection. While interpretability analysis was not explicitly performed, the inherent use of attention mechanisms in the Vision Transformer model suggests a focus on important brain regions and enhances its potential for prioritizing crucial information in brain tumor detection.

Keywords: Brain Tumor Detection, Medical Imaging, Classification, Vision Transformers, ViT, Machine Learning, Deep Learning.

Introduction:

Brain tumors are a major global health concern, and by early identification, patient healthcare can be greatly enhanced. Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are commonly used medical imaging techniques for tumor diagnosis in the human brain. However, the diagnosis still relies on the doctor or radiologist's decision, which is an error-prone and time-consuming process. In order to support medical personnel in making fast and precise diagnoses, the increasing amount of brain imaging data needs sophisticated computational algorithms. This research presents the incorporation of a type of artificial neural network called vision transformer (ViT) for brain tumor detection. ViT is a viable option for medical image analysis since they have demonstrated exceptional performance in a variety of computer vision tasks. The purpose of the research is to assess the accuracy of vision transformers while detecting brain tumors.

Vision transformers are a useful tool for detecting brain tumors because of their capacity to extract contextual information and spatial correlations from medical images. In contrast to conventional convolutional neural networks (CNNs), which analyze images using a fixed grid-like structure, vision transformers use a self-attention mechanism that enables them to take interactions and long-range relationships between various image regions into account. Additionally, vision transformers have proven to perform remarkably well in a variety of computer vision applications, such as segmentation, object identification, and image classification. They are a desirable alternative for medical image analysis, where reliable and accurate abnormality detection is crucial, due to their capacity to learn from large-scale datasets and generalize well to unseen data.

This research aims to assess the effectiveness of ViT for identifying brain tumors through extensive experimentation and achieve an acceptable accuracy by ViT for the said task. Our research will demonstrate the potential of ViT as a valuable tool for diagnosing brain tumors in the clinical setting. The research paper has been divided into five sections. Section 1 elaborates on the Introduction, Section 2 covers the Literature Review, Section 3 covers the methodology, Section 4 discusses results and Section 5 concludes the paper.

Literature Review:

During the previous decade, deep learning algorithms made great hype for their high performance in various domains ranging from textual analysis to image analysis. These algorithms are also extensively used for healthcare data and help in automating various tasks related to healthcare while achieving comparable accuracy to that of humans. The incorporation of deep learning techniques has resulted in a remarkable revolution in the field of brain tumor identification. Conventional approaches in this field mostly depend on manually designed features and rule-based algorithms, which frequently prove inadequate for managing the complex and diverse nature of medical images. In a research study, Kumar et al. [1], have made significant progress in proving that Convolutional Neural Networks (CNNs) are effective at automating the process of identifying brain tumors. Conventional methods have difficulties when it comes to capturing spatial hierarchies and patterns inside medical images. CNNs have proven essential in overcoming these challenges.

Even with CNN's significant success, it is still necessary to investigate new approaches. Hossain et al. [2] developed the concept of vision transformers and provided a new dimension to improve the detection of brain tumors. Vision transformers provide a comprehensive viewpoint by removing global dependencies from images, which may enhance the model's capacity to identify complex and subtle patterns suggestive of malignancies. By customizing vision transformers for the classification of medical images, Manzari et al. [3] have made a significant contribution to this investigation. Their research highlights how adaptable these transformer models are at handling the difficulties presented by intricate medical imagery.

The quick advancement of medical imaging technology emphasizes how crucial it is to identify brain tumors with accuracy. Puttagunta et al. [4] offer an extensive overview of deep learning methods for medical image interpretation. Their work describes the changing environment in which precise identification becomes more and more important, in addition to highlighting the revolutionary potential of these tools. Furthermore, the work of Sadad et al. [5] offers insightful information about brain tumor detection through a wider variety of machine learning algorithms, offering a comprehensive overview of the difficulties faced and prospects in the field.

Ahn et al. [6] in their study, provide the theoretical foundation for comprehending attention mechanisms, which are a crucial aspect of vision transformers. The model can detect global dependencies in images, thanks to this attention mechanism, which is in line with the complex needs of brain tumor diagnosis. The field of brain tumor diagnosis has experienced a dynamic shift due to the incorporation of state-of-the-art deep learning techniques. This development was prompted by the limits of traditional approaches that relied on manually created features and rule-based algorithms. Recent research has demonstrated the revolutionary potential of convolutional neural networks (CNNs) in automating the complex process of recognizing brain tumors, as demonstrated by the work of Kumar et al. [1]. With their capacity to extract intricate patterns and spatial hierarchies from medical images, CNNs are a key component in the development of diagnostic techniques.

The study of Brownlee et al [7], focuses on mastering machine learning techniques by providing useful information that can support the continued advancement and deployment of deep learning techniques in the particular field of brain tumor identification. The notion of vision transformers was created by Akinyelu et al. [8] added a new layer to enhance ViT for brain tumor identification. By eliminating global dependencies from images, vision transformers offer a holistic perspective that may improve the model's ability to recognize intricate and nuanced patterns suggestive of tumors. A noteworthy contribution to this study has been made by Wanget al. [9], who customized vision transformers for the classification of medical images. Their study demonstrates how versatile these transformer models are in managing the challenges posed by complex medical images.

Jiang, et al. [10], in their work, provided an in-depth review of deep learning techniques for the interpretation of medical imaging data. Their work highlights the revolutionary potential of these techniques and outlines the evolving environment in which exact identification becomes increasingly vital. Moreover, Tummala et al. [11] provide informative data regarding the identification of brain tumors using a greater range of machine learning algorithms, providing a thorough summary of the challenges encountered and future prospects in the field. Asiriet al. [12] utilized an attention mechanism to enable the model to identify global dependencies in images, which is consistent with the requirements of the complicated task of brain tumor diagnosis. The work of [13][14][15][16] provides a variety of perspectives and approaches that greatly enhance our knowledge and progress in the field of brain tumor detection using deep learning techniques.

Methodology:

This section explains the overall methodology of this research and is presented in Figure 1 by the methodology flow diagram.

Data Description:

The dataset that has been utilized for our research study is the Br35H Brain Tumor Detection 2020 Dataset which is available on Kaggle. The dataset contains 3000 T1 weighted images of tumorous (yes) class and healthy (no) brain scans. Each class has 1500 images and an equal distribution of the Dataset. Sample MRI brain scans of tumors and without tumors are shown in Figures 2(a) and (b), respectively.

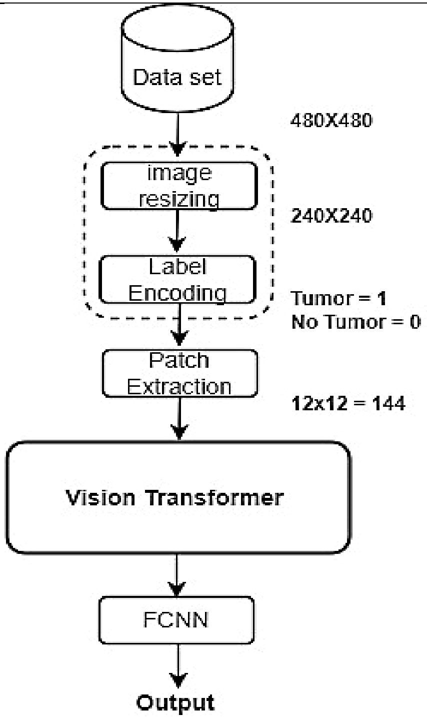
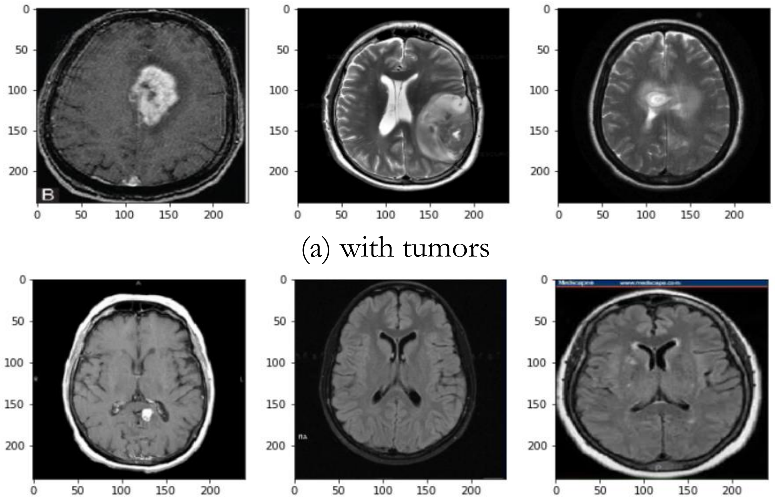


Figure 1: Methodology flow diagram



(a) with tumors

(b) without tumors

Figure 2: Samples from the dataset

Preprocessing:

Data preprocessing is a vital step in preparing data that is suitable for model training. This enhances data quality, and compatibility and ensures optimal performance by the model being trained. Various preprocessing steps that are applied to the utilized dataset are explained as follows.

Resizing:

The dataset contains images that are of varying sizes. Hence, resizing is done to bring every image to 240 × 240 pixels, which was an important first step. This consistency allowed our model to have consistent input dimensions, which made processing easier.

Normalization:

To scale data to a common range, normalization is done. The process ensures that all the input features have a similar influence on the model, preventing certain features from dominating the learning process due to differences in their original scales. This aids in effective learning and performance improvement of the model. To achieve normalization, we used the following mathematical transformation.

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

Where the resized image is represented by X and the normalized image is represented by X′.

Label Encoding:

By label encoding, we converted the categorical labels yes and no for brain tumor presence and absence, into numerical values to make sure they are compatible with our model.

Vision Transformer (ViT):

The Vision Transformer, or ViT, is a model for image classification that employs a Transformer-like architecture over patches of the image. An image is split into fixed-size patches, each of them is then linearly embedded, position embeddings are added, and the resulting sequence of vectors is fed to a standard Transformer encoder. In order to perform classification, the standard approach of adding an extra learnable "classification token" to the sequence is used [17]. The intricate details of medical images are catered to in the vision transformer architecture. In order to process image patches and enable the network to capture both local and global information, the model is composed of a number of transformers blocks as shown in Figure 3. The model can assess the relative importance of various regions in the input images using the self-attention process.

Model Architecture:

The model architecture we have implemented for this study is depicted in Figure 3. and is explained as follows.

- **Image Patching**: The input to the model is split up into smaller patches as part of the patching process. This separation makes processing more efficient and makes it possible for the model to successfully collect local spatial information.
- **Image Flattening:** After patching, the two-dimensional spatial information is converted into a linear representation by flattening the patches. The data is now ready for additional processing inside the model.
- **Patch Encoder:** The Patch Encoder module processes the flattened patches. This module's dense projection layer and positional encoding encode the patches' spatial information and give the locations of the patches within the image context.
- **Positional Encoding:** An essential part of embedding positional information into the patch representations is positional encoding. As a result, the model is able to comprehend the spatial relationships between patches and gather crucial categorization context.
- **Transformer Encoder Block:** The Transformer Encoder Blocks are the central component of the model. These blocks are made up of feedforward neural network layers and multi-head self-attention layers. They facilitate the model's ability to extract high-level characteristics from the encoded patches and to capture global dependencies.
- **Aggregate Representation:** Upon completion of several Transformer Encoder Blocks, the representations obtained from every patch are combined. The contextual data that is acquired from every patch is combined throughout this aggregation phase to provide a complete representation of the full image.
- **Classification:** In the end, a classification head made up of dense layers receives the combined representation. To ascertain whether a brain tumor is present in the input image, these layers carry out categorization.

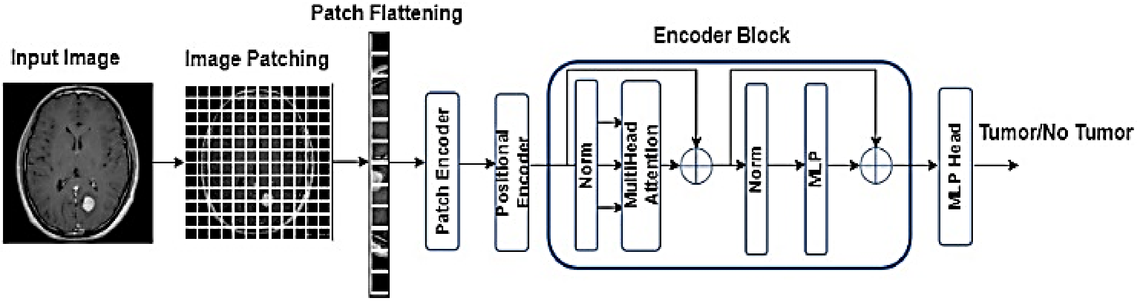


Figure 3: Model Architecture

Training Strategy:

Here, we describe the training approach that we used to train our brain tumor identification model using the Br35H: Brain Tumor Detection 2020 Kaggle dataset in an efficient manner.

Data Splitting:

We divided the dataset into separate training and testing sections in order to create a strong training regimen. To ensure there was enough data for the model to learn, we set aside 80% of the dataset for training. The remaining 20% was set aside for testing, allowing for an objective assessment of the performance of the trained model.

Hyperparameters Settings:

- **Learning Rate:** We regulate the rate at which the model modifies its parameters during optimization by setting the learning rate to 0.001. The trade-off between parameter stability and convergence speed is balanced by this decision.
- **Weight Decay:** To prevent overfitting inclinations, a weight decay of 0.0001 was used as a regularization strategy. Model generalization is promoted by weight decay, which penalizes large parameter values.
- **Batch Size:** The number of samples processed in each training iteration was determined by our training procedure, which included a batch size of 32. This batch size balances gradient stability with computational efficiency.
- **Number of Epochs:** The training process took place over fifty epochs, allowing for iterative dataset learning. The model is exposed to the data throughout several epochs, which promotes model convergence and parameter refining.
- **Image Size and Patch Size**: After resizing the input images to 240 by 240 pixels, patches with a size of 20 by 20 pixels were extracted. These dimensions played a crucial role in specifying the input organization and spatial resolution of the feature extraction process of the model.
- **Transformer Architecture:** Eight transformer layers, each incorporating four attention heads, made up our model architecture. Dense layers with units [2048, 1024] were featured in the final classifier, while the transformer units had size [128, 64]. These architectural decisions were carefully considered in order to maximize the expressiveness of the model and its ability to identify complex patterns in the data.

Result and Discussion:

The implemented ViT model shows impressive performance while utilizing the Br35H dataset. The model performance plots are shown in Figure 4. and the classification report is shown in Table 1.

Training and Validation Loss Graph:

The Graph shows how training and validation loss changed over the course of 50 epochs while our brain tumor detection model was being trained. At first, there is a declining trend in both the training and validation losses, which suggests that the model is successfully learning from the data. Both trajectories, however, converge at the 50th epoch, indicating that the model has reached a stage at which more training no longer considerably enhances its performance. This convergence shows that the model is performing at its best and has successfully captured the underlying patterns in the data. The losses then level out or maybe even begin to rise, indicating that the model has taken in all the information it can from the training set. Overall, the convergence after 50 epochs shows that training was successful and efficient by the model.

Training and Validation Accuracy Graph:

The accuracy of our brain tumor detection model during training and validation is shown in the graph. Both accuracy levels rise gradually, peaking close to the 50th epoch. At this stage, the model obtains an impressive testing accuracy of 98.37%, demonstrating its efficacy in correctly categorizing photos of brain tumors. This great accuracy shows how reliable and appropriate the model is for practical use in clinical settings.

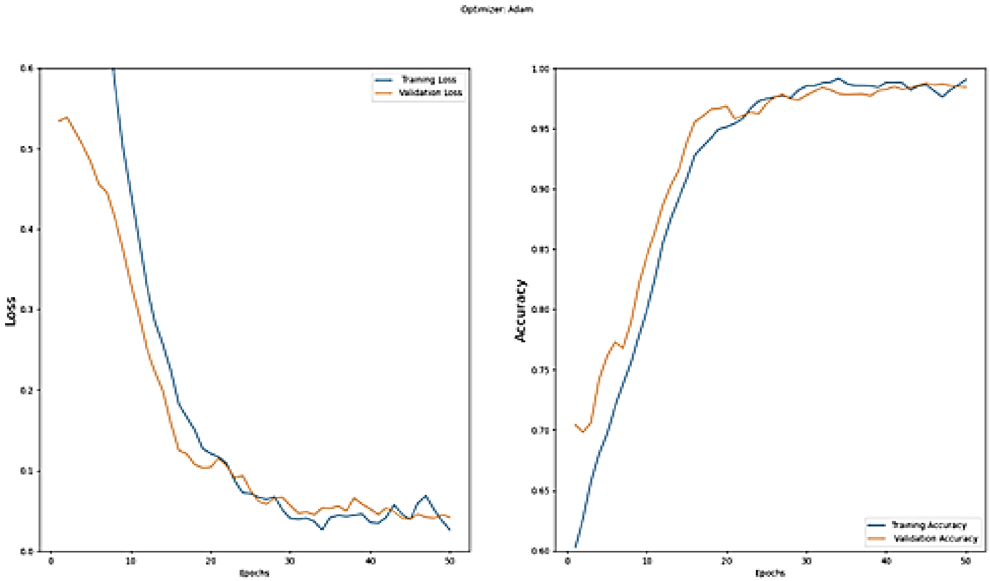


Figure 4: ViT Training and validation (a) loss (b) accuracy

Table 1: Classification Report

	Precision	Recall	F1-Score	Support
Yes	1.00	0.99	0.99	300
No	0.97	0.99	0.98	300
Accuracy			0.98	600

Explanation of Classification Report:

- **Precision:** Precision is defined as the percentage of actual positive forecasts among all positive predictions. The precision for the "Yes" class (brain tumor presence) in this instance is 1.00, meaning that all of the cases that were predicted to have brain tumors were accurate. Similarly, the precision for the "No" class (absence of brain tumor) is 0.97, meaning that 97% of cases that were predicted to be brain tumor-free were accurate.
- **Recall:** The percentage of true positive predictions among all actual positive cases is measured by recall, which is sometimes referred to as sensitivity. With a recall of 0.99 for the "Yes" class, the model was able to accurately identify 99% of real brain tumor cases. The recall for the "No" class is likewise 0.99, meaning that 99% of real cases.
- **Support:** In the dataset, Support is the total number of instances of each class. Our test data consists of 300 MRI scans from each class, totaling 600.
- **Accuracy:** Measured as the percentage of correctly classified occurrences over the total number of examples, accuracy assesses the overall correctness of the model's

predictions. With an accuracy of 0.98, 98% of the cases in the dataset were properly classified by the model.

Conclusion:

The field of medical image processing is undergoing a fundamental paradigm change with the integration of vision transformers for brain tumor diagnosis. Vision transformers are extremely useful tools for healthcare practitioners because of their special qualities, which include their interpretability and ability to understand complex relationships among images. One important function of the attention mechanism, which is a distinguishing characteristic of vision transformers, is to shed light on the characteristics that influence the model's judgment. In the medical industry, confidence and comprehension of the diagnostic procedure are greatly enhanced by transparency. The vision transformer model is being improved and optimized on a continuous basis. To evaluate the model's generalizability and robustness, extensive research is being conducted across a wide range of datasets and clinical settings.

Finally, this study emphasizes how important it is to incorporate vision transformers into the field of brain tumor identification. The model's proven performance on a variety of complex medical images represents a significant breakthrough in diagnostic skills. In addition to their exceptional performance, vision transformers can be interpreted, giving medical professionals useful information on the decision-making process. In addition to having the capacity to interpret intricate patterns, these models have the potential to improve accuracy, which makes them revolutionary instruments for medical practitioners. This study represents not only a significant advancement but also a possible paradigm changes in the field of brain tumor detection.

Acknowledgement:

Our project's success is mainly due to the outstanding advice and assistance of several people. We sincerely thank our Supervisor, Dr Yasir Saleem Afridi, and our Advisor, Engineer Najam Aziz, for their invaluable support. We also acknowledge the University of Engineering and Technology Peshawar, our parents, and our instructors for their constant support and for creating a supportive environment in which we may work.

Author's Contribution:

All the Authors have contributed equally to this work.

Conflict of Interest: The authors hold no conflict of interest in publishing this manuscript in IJIST.

References:

[1] S. Kumar, R. Dhir, and N. Chaurasia, "Brain Tumor Detection Analysis Using CNN: A Review," Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021, pp. 1061–1067, Mar. 2021, doi: 10.1109/ICAIS50930.2021.9395920.

[2] S. Hossain, A. Chakrabarty, T. R. Gadekallu, M. Alazab, and M. J. Piran, "Vision Transformers, Ensemble Model, and Transfer Learning Leveraging Explainable AI for Brain Tumor Detection and Classification," IEEE J. Biomed. Heal. Informatics, vol. 28, no. 3, pp. 1261–1272, Mar. 2024, doi: 10.1109/JBHI.2023.3266614.

[3] O. N. Manzari, H. Ahmadabadi, H. Kashiani, S. B. Shokouhi, and A. Ayatollahi, "MedViT: A robust vision transformer for generalized medical image classification," Comput. Biol. Med., vol. 157, p. 106791, May 2023, doi: 10.1016/J.COMPBIOMED.2023.106791.

[4] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," Multimed. Tools Appl. 2021 8016, vol. 80, no. 16, pp. 24365–24398, Apr. 2021, doi: 10.1007/S11042-021-10707-4.

[5] T. Sadad et al., "Brain tumor detection and multi-classification using advanced deep learning techniques," Microsc. Res. Tech., vol. 84, no. 6, pp. 1296–1308, Jun. 2021, doi: 10.1002/JEMT.23688.

[6] K. Ahn, X. Cheng, M. Song, C. Yun, A. Jadbabaie, and S. Sra, "Linear attention is (maybe) all you need (to understand transformer optimization)," Oct. 2023, Accessed: May 16, 2024. [Online]. Available: https://arxiv.org/abs/2310.01082v2

[7] "Data-Scientist-Books/Data Preparation for Machine Learning Data Cleaning, Feature Selection, and Data Transforms in Python by Jason Brownlee (z-lib.org).pdf at main · aaaastark/Data-Scientist-Books · GitHub." Accessed: May 16, 2024. [Online]. Available: https://github.com/aaaastark/Data-Scientist-Books/blob/main/Data Preparation for Machine Learning Data Cleaning%2C Feature Selection%2C and Data Transforms in Python by Jason Brownlee (z-lib.org).pdf

[8] A. A. Akinyelu, F. Zaccagna, J. T. Grist, M. Castelli, and L. Rundo, "Brain Tumor Diagnosis Using Machine Learning, Convolutional Neural Networks, Capsule Neural Networks and Vision Transformers, Applied to MRI: A Survey," J. Imaging 2022, Vol. 8, Page 205, vol. 8, no. 8, p. 205, Jul. 2022, doi: 10.3390/JIMAGING8080205.

[9] W. Wang, C. Chen, M. Ding, H. Yu, S. Zha, and J. Li, "TransBTS: Multimodal Brain Tumor Segmentation Using Transformer," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12901 LNCS, pp. 109–119, 2021, doi: 10.1007/978-3-030-87193-2_11/COVER.

[10] Y. Jiang, Y. Zhang, X. Lin, J. Dong, T. Cheng, and J. Liang, "SwinBTS: A Method for 3D Multimodal Brain Tumor Segmentation Using Swin Transformer," Brain Sci. 2022, Vol. 12, Page 797, vol. 12, no. 6, p. 797, Jun. 2022, doi: 10.3390/BRAINSCI12060797.

[11] S. Tummala, S. Kadry, S. A. C. Bukhari, and H. T. Rauf, "Classification of Brain Tumor from Magnetic Resonance Imaging Using Vision Transformers Ensembling," Curr. Oncol. 2022, Vol. 29, Pages 7498-7511, vol. 29, no. 10, pp. 7498–7511, Oct. 2022, doi: 10.3390/CURRONCOL29100590.

[12] A. A. Asiri et al., "Exploring the Power of Deep Learning: Fine-Tuned Vision Transformer for Accurate and Efficient Brain Tumor Detection in MRI Scans," Diagnostics 2023, Vol. 13, Page 2094, vol. 13, no. 12, p. 2094, Jun. 2023, doi: 10.3390/DIAGNOSTICS13122094.

[13] E. A. Albadawy, A. Saha, and M. A. Mazurowski, "Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing," Med. Phys., vol. 45, no. 3, pp. 1150–1158, Mar. 2018, doi: 10.1002/MP.12752.

[14] R. A. Zeineldin, M. E. Karar, J. Coburger, C. R. Wirtz, and O. Burgert, "DeepSeg: deep neural network framework for automatic brain tumor segmentation using magnetic resonance FLAIR images," Int. J. Comput. Assist. Radiol. Surg., vol. 15, no. 6, pp. 909–920, Jun. 2020, doi: 10.1007/S11548-020-02186-Z/TABLES/4.

[15] M. I. Sharif, J. P. Li, J. Amin, and A. Sharif, "An improved framework for brain tumor analysis using MRI based on YOLOv2 and convolutional neural network," Complex Intell. Syst., vol. 7, no. 4, pp. 2023–2036, Aug. 2021, doi: 10.1007/S40747-021-00310-3/TABLES/13.

[16] A. A. Asiri et al., "Block-Wise Neural Network for Brain Tumor Identification in Magnetic Resonance Images," Comput. Mater. Contin., vol. 73, no. 3, pp. 5735–5753, Jul. 2022, doi: 10.32604/CMC.2022.031747.

[17] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," ICLR 2021 - 9th Int. Conf. Learn. Represent., Oct. 2020, Accessed: Feb. 21, 2024. [Online]. Available: https://arxiv.org/abs/2010.11929v2

Home Automation Using Internet of Things and Machine Learning

Laila Amin, Nasru Minallah

Department of Computer Systems Engineering University of Engineering and Technology Peshawar, Pakistan

*__Correspondence__: lailaamin.cse@uetpeshawar.edu.pk

Citation | Amin. L, Minallah. N, "Home Automation Using Internet of Things and Machine Learning", IJIST, Special Issue pp 152-157, May 2024

Received| May 06, 2024, **Revised|** May 12, 2024, **Accepted|** May 20, 2024, **Published|** May 23, 2024.

This paper proposes an energy-efficient home automation system leveraging the Internet of Things (IoT) and machine learning. The system, implemented in Python on a Raspberry Pi, enables remote control of appliances (lights, televisions, air conditioners) via a web interface accessible from any local network device. Machine learning is introduced in the second phase, utilizing linear regression to automate appliance management based on historical data stored in a database. This work demonstrates the feasibility of IoT and machine learning for cost-effective and efficient home automation, laying the groundwork for the future development of database-driven smart homes with advanced machine learning algorithms.

Keywords: Home Automation, Internet of Things, Machine Learning.

Introduction:

Home automation systems have gained significant traction in recent years, offering various functionalities and specifications [2][3][4][5][6][7][8][9]. Existing literature explores a range of approaches, including ZigBee-Arduino control for small appliances [2], low-cost Arduino UNO-based systems with web server functionality and environmental sensors [3], X10 controller and IEEE 1394 AV framework integration forremote access [4], Java-based structures for PC server control [5], PIC microcontroller-driven systems with web interfaces for slave control points [6], Bluetooth-over-internet appliance control [7], and RF module-Arduino UNO-WiFi-cloud- mobile application systems with environmental sensing for fan and light control [8]. Additionally, research has been conducted on minimizing energy usage through multiband antennas, energy-efficient sensors, and thermal management [9]. However, limitations exist in many proposed systems. Some prioritize simplicity, focusing on basic controls for fans and lights, while others target complex devices like intelligent doors. Issues arise with user-friendliness, basic functionality, cost, and energy consumption, leading to a gap between user requirements and system capabilities.

This paper addresses these limitations by proposing a cost-effective, user-friendly Internet of Things (IoT) home automation system. The system offers internet, PC, and mobile application control for home appliances, promoting energy efficiency. We leverage a Raspberry Pi as a bridge device, translating user interaction through a web page into signals for end-device control. The system utilizes Wi-Fi technology for local network communication and, with an internet-connected server, allows remote access via a web browser. This paper details the software and hardware implementation, focusing on the initial control of three key appliances: fan, air conditioner, and light. The proposed system promises to be cost-effective, reliable, and easy to implement, aiming to bridge the gap between user needs and existing home automation solutions.

Figure 1: Possibilities with Smart Security [1]

Electronics and their Application in Automation:

Raspberry Pi:

This work utilizes a Raspberry Pi 3B+ as the central hub for communication and processing. The Raspberry Pi offers connectivity through one RJ45 connector, four USB ports, and an integrated Wi-Fi adapter. This allows for interaction with various communicating terminals within the smart home environment. The Raspberry Pi serves two primary functions, it facilitates communication between all connected appliances and sensors. It stores historical data in its memory for machine-learning purposes. A Flask server running on the Raspberry Pi manages the database and facilitates model training.

Passive Infrared Sensor:

A Passive Infrared (PIR) sensor is employed to detect infrared radiation, enabling the system to turn appliances on or off based on user presence. Ideally, upon entering a room, lights should illuminate basedon a pre-set schedule retrieved from the database. Conversely, if a user leaves the room unexpectedly, the PIR sensor detects the absence of human movement and triggers the Raspberry Pi to turn off appliances, promoting energy efficiency. The sensor transmits a binary value ("1" for presence, "0" for absence) to a designated pin on the Raspberry Pi, which is also displayed on the web interface foruser convenience. This eliminates the need for intrusive camera installations for presence detection.

Relays:

Relays are crucial for switching connected devices on and off. The control signal received by the relay determines whether a specific pin on the IoT device is ICTIS 2024 activated or deactivated. This enables remote control of appliances within the smart home system. This project utilizes 5-volt, 10-ampere relay modules for the aforementioned functionality.

Flask Server:

Flask, a web framework, provides the necessary libraries for building web applications. In thisproject, a Flask server is implemented to host all connected devices. It facilitates sending "ON" and "OFF" signals to these devices. The server is hosted on the Raspberry Pi with a local IP address, restricting access to devices within the same local subnet for security purposes. This necessitates the smartphone controlling the system to be connected to the same subnet. Alternatively, a VPN connection could be established for remote control from outside the local network

Implementation of Home Automation System:

Prototype Development:

The prototype for the machine learning-based home automation system using a Raspberry Pi is connected to three electrical devices (fan, television, and light source) the connections to the Raspberry Pi are through a breadboard andjumper cables. A Flask server running on the Pi provides a web interface for remote control (on/off) of these devices. Inparallel, a Python script maintains a database logging deviceactivation times and associated GPIO pins. This data serves as the foundation for a routine learning algorithm, enabling the system to predict and automate device behavior after a seven-day training period. The web interface remains accessible from laptops or smartphones within the same localnetwork for manual control or system monitoring. The Implementation is completed in two phases

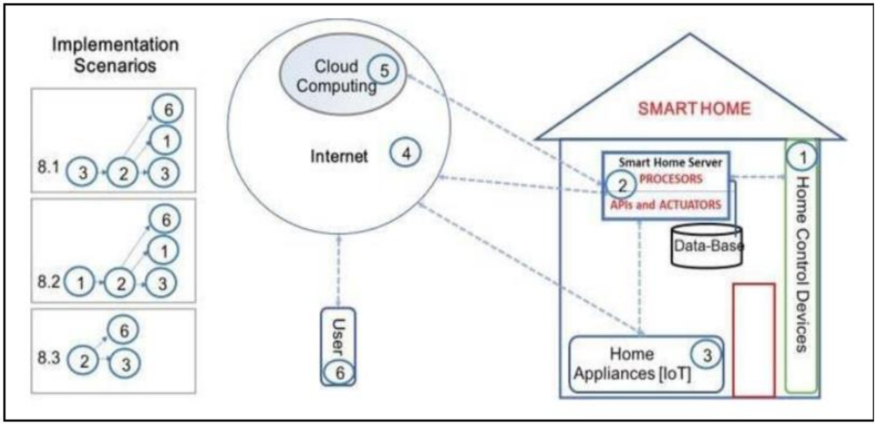


Figure 2: Smart home implementation chart

Phase 1:

Controlling all the appliances in a smart homeenvironment through any internet-enabled device like a Smartphone or Laptop etc. The connectivity is done by having a Raspberry Pi as a bridge device to translate signals generated by human interaction through web pages to the end terminal devices.

Phase 2:

Phase 1 is enhanced in phase 2 by automatinghuman interaction in the introduction of databases that aremaintained through a machine learning algorithm

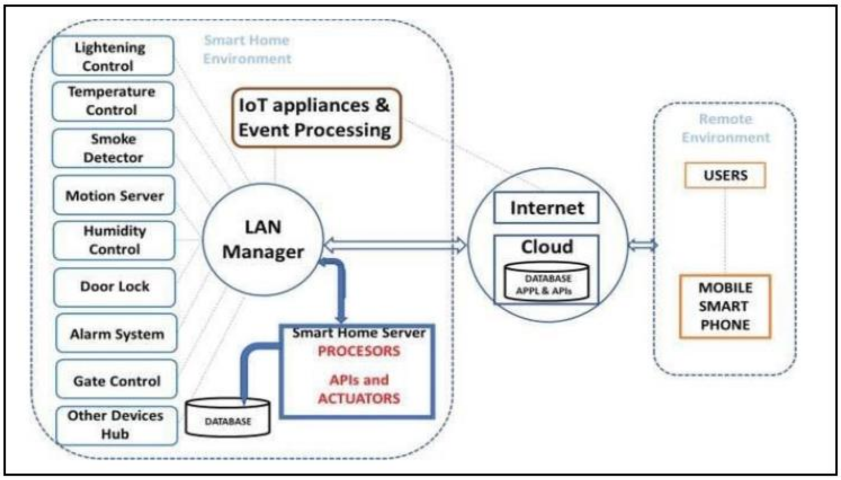


Figure 3: Detailed implementation with the introduction ofmachine learning

Setup for Smart Home Project Implementation:

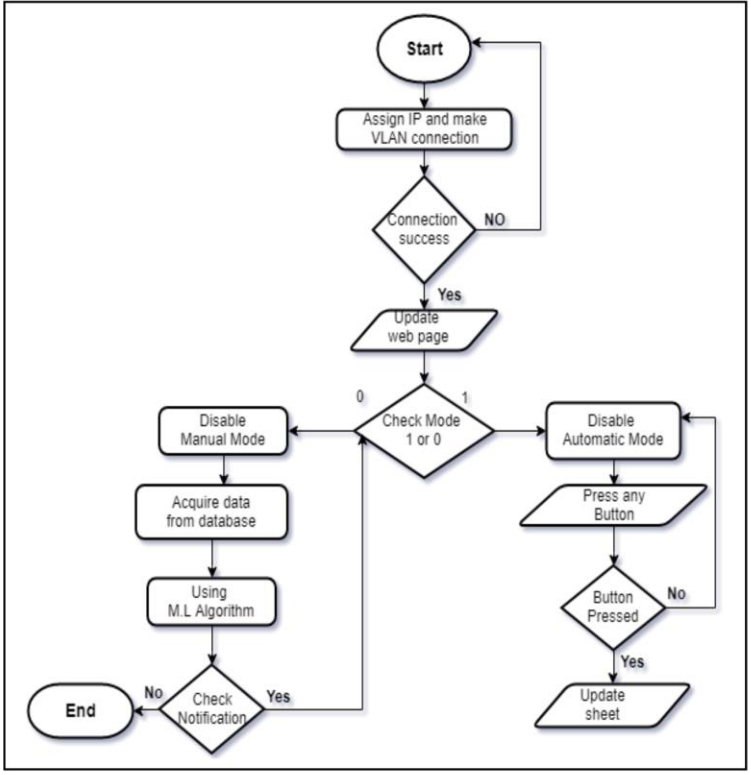


Figure 4: Flow Chart for Implementation of the Smart Home

When the device (Raspberry PI) is switched on. The most important element is its connection to the Local Area Network. In this way, it can communicate with the existing communicating devices. After the device is allocated an IP either through DHCP or assigning a static IP. The web GUI of the project can be accessed. This GUI enables the user to control the connected appliances. In the web GUI, the user canselect the mode of operation to be Manual or Automatic. The first mode enables the project to use data from the databases for the corresponding behavior of the project while the second mode needs the interaction from the user to turn on /off the connected devices. Once the interaction is complete the existing database is updated and the flow ends.

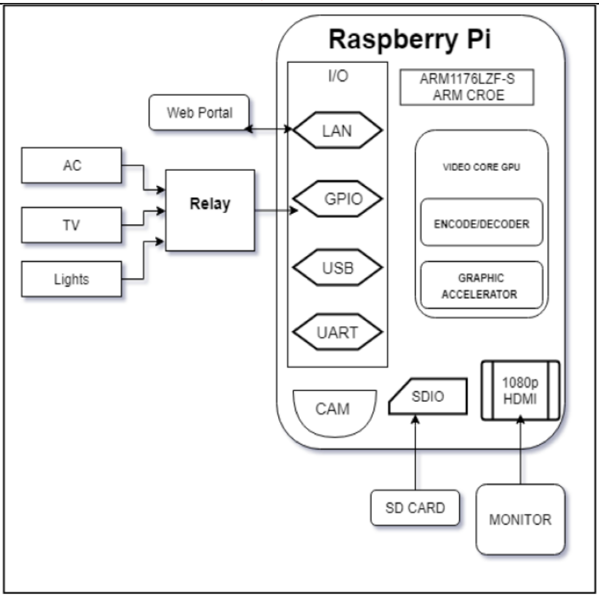


Figure 5: Procedure of Setup

The development of a machine learning-based home automation system using a Raspberry Pi 3B+ leverages Ubuntu 16.04 for visualization and Python 3.5 for core functionality. Code development is facilitated by Spyder 3.3.0. A user-friendly web interface is implemented using HTML to enable remote control and facilitate future application integration. Python scripts interact with the Raspberry Pi's GPIO pins to control connected appliances (fan, television, light source) via IoT protocols. A basic data logging system is established, utilizing a Python script to record device activation times and associated timestamps into a text file (.txt). This data serves as the foundation for a futuremachine learning algorithm, enabling the system to learn userbehavior and automate device operation.

Results:

The data of the sensor are sent to the web browser for monitoring of the system after the successful connection to the server. Thewebpage will appear when we enter the IP address in the web browser. The web server provides important information aboutthe status of the home appliances that are connected to the internet remotely. Home automation makes it possible to automate tasks relatedto security, well-being, and comfort through a smart system installed in a home or building. In other words, it integrates technology into the design of a space. One of the main advantages of home automation systems is energy efficiency.

Conclusion:

This paper contributes to the growing field of smart homeautomation by leveraging the Internet of Things (IoT) andmachine learning. We present a Raspberry Pi-based system that interfaces with various electrical appliances, enablingtheir remote control and automated operation. In the initial phase, the system provides a user-friendly interface for manual control. Subsequently, a machine learning component is introduced to analyze historical usage patterns and automate appliance switching based on learned schedules. This approach not only promotes energy efficiency but also enhances user convenience by reducing manual intervention and offering increased flexibility in managing time and resources.

Future Work:

This work's inherent scalability allows for the integration of "smarter" appliances, controllable either through smartphones or autonomously based on historical data. This opens doors to numerous domestic applications, including smart cameras, self-activating cleaning robots, and intelligentwashing machines. By leveraging machine learning algorithms, household chores can be automated through theseappliances, enhancing user convenience and promoting significant cost savings via efficient resource management.

Acknowledgments:

Special thanks to Dr. Nasru Minuallah, who acted as my project supervisor and examiner respectively, for his ideas, patience, support, and guidelines to come up with this project.I would also like to thank my parents for their support.

References:

[1] A. C. Muller and S. Guido, "Introduction to machine learning with Python : a guide for data scientists," p. 376, Accessed: May 11, 2024. [Online]. Available: https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/

[2] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, "Participatory Sensing," From Mach. to Internet Things, pp. 295–305, Jan. 2014, doi: 10.1016/B978-0-12-407684-6.00015-2.

[3] H. Kopetz, "Internet of Things," pp. 307–323, 2011, doi: 10.1007/978-1-4419-8237-7_13.

[4] "smart security iot - Google Search." Accessed: May 11, 2024. [Online]. Available: https://www.google.com/search?q=smart+security+iot&client=firefox-b-d&source=lnms&sa=X&biw=1920&bih=944&udm=2

[5] "Upton, E. and Halfacree, G., 2014. Raspberry Pi user guide. John Wiley & Sons", [Online]. Available: https://dn.odroid.com/IoT/other_doc.pdf

[6] S. Jain, A. Vaibhav, and L. Goyal, "Raspberry Pi based interactive home automation system through E-mail," ICROIT 2014 - Proc. 2014 Int. Conf. Reliab. Optim. Inf. Technol., pp. 277–280, 2014, doi: 10.1109/ICROIT.2014.6798330.

[7] "Building the Web of Things." Accessed: May 11, 2024. [Online]. Available: https://www.manning.com/books/building-the-web-of-things

[8] "Raspberry Pi as Internet of Things hardware: Performances and Constraints." Accessed: May 11, 2024. [Online]. Available: https://www.researchgate.net/publication/272175660_Raspberry_Pi_as_Internet_of_Things_hardware_Performances_and_Constraints

[9] G. . Montgomery, D.C., Peck, E.A, Vining, "Introduction to linear regression analysis," John Wiley Sons.No match, vol. 821, 2012.

Relevance Classification of Flood-Related Tweets Using XLNET Deep Learning Model

Aneela Habib, Yasir Saleem Afridi, Madiha Sher, Tiham Khan

Department of Computer Systems Engineering University of Engineering and Technology Peshawar

***Correspondence**: aneelahabib2@gmail.com, yasirsaleem@uetpeshawar.edu.pk, madiha@uetpeshawar.edu.pk, tihamkhan3b@gmail.com.

Citation | Habib. A, Afridi. Y. S, Sher. M, Khan. T, "Relevance Classification of Flood-Related Tweets Using XLNET Deep Learning Model", IJIST, Special Issue pp 158-164, May 2024

Received | May 10, 2024, **Revised |** May 15, 2024, **Accepted |** May 19, 2024, **Published |** May 23, 2024.

Floods, being among nature's most significant and recurring phenomena, profoundly impact the lives and properties of tens of millions of people worldwide. As a result of such events, social media structures like Twitter often emerge as the most essential channels for real-time information sharing. However, the total volume of tweets makes it hard to manually distinguish between those relating to floods and those that are not. This poses a large obstacle for responsible government officials who need to make timely and well-knowledgeable decisions. This study attempts to overcome this challenge by utilizing advanced techniques in natural language processing to effectively sort through the extensive volume of tweets. The outcome we obtained from this process is promising, as the XLNET model achieved an extraordinary F1 rating of 0.96. This high degree of overall performance illustrates the model's usefulness in classifying flood-related tweets. By leveraging the abilities of the XLNET model, we aim to provide a valuable guide for responsible governance, aiding in making timely and well-informed choices during flood situations. This, in turn, will assist reduce the impact of floods on the lives and property-affected communities around the world.

Keywords: Text classification, LSTM, Multi-head Attention, Flood, Tweets

Introduction:

Natural disasters, like floods, can cause excessive destruction to communities and residences. In the contemporary generation, social media platforms have emerged as treasured resources of statistics during and after such disasters. Twitter, in particular, plays an important position in disseminating real-time updates. However, the full extent of tweets generated through these events can overwhelm responsible governance, making it hard to identify pertinent information [1]

To deal with this problem, we advocate a text class framework making use of a machine learning model known as XLNET. XLNET is one of the latest deep learning models well-known for its splendid performance across various natural language processing (NLP) responsibilities. By leveraging XLNET's competencies, our goal is to construct an effective solution for classifying flood-related tweets. This framework could be an effective tool that can be utilized by government agencies, helping them quickly identify the relevant information amidst the vast amount of information on Twitter, thereby helping to enable appropriate and timely decision-making.

Literature Review:

Text classification is a key task in the broader framework of NLP, which aims at grouping text into predefined classes or categories. This task constitutes various jobs, such as spam detection, sentiment analysis, and theme categorization, among others. It is noteworthy, that there have been significant studies carried out on detecting natural disasters and the usage of social media and satellite imagery [2]. In recent years, social media structures, specifically Twitter, have emerged as precious assets of facts, in particular, especially during times of crisis [3]. Several studies have been carried out on taking benefit from social media for disaster reaction and management. For instance, Palen et al. (2010) [4] analyzed Twitter usage during the 2009 Red River Valley flood, identifying various sorts of tweets, together with situational recognition updates, legitimate alerts, requests for help, and emotional support. Similarly, Imran et al. (2015) [5] tested Twitter's function during the 2013 Colorado floods and highlighted its significance in supplying precious facts for disaster reaction and control.

The utility of NLP techniques in analyzing social media statistics for the duration of disasters has been widespread. Caragea et al. (2011) [6] applied machine learning methods to categorize catastrophe-related tweets, attaining an F1 score of 0.79 on a dataset comprising 9000 tweets from three exclusive disasters. Furthermore, the use of gadget-mastering models for flood detection, as demonstrated in Flood Detection in Urban Areas Using Satellite Imagery and Machine Mastering [7], signifies the ability of such procedures to improve situational attention in the course of emergencies. Extracting geographically rich know-how from micro texts like tweets becomes important for location-based structures in emergency services to correctly respond to diverse natural and man-made disasters, including earthquakes, floods, pandemics, vehicle accidents, terrorist attacks, and shooting incidents [4].

Methodology:

Figure 1 represents the block diagram of the overall methodology used in this study to illustrate the sequential steps involved in achieving our objectives. These steps are explained in the following subsections.

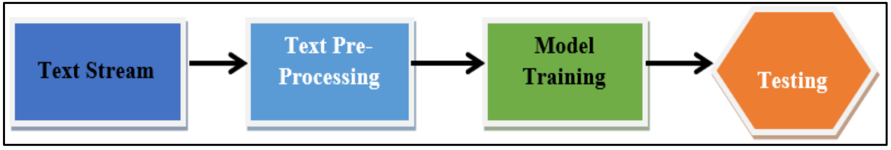


Figure 1: Block diagram of our steps to be followed

Text Stream:

In our investigation, we aim to discover the capacity of using tweets as a precious source of facts and communique during and after disasters. To accomplish this, we rent Twitter APIs to extract relevant tweets associated with particular catastrophe events. For the purpose of model training, we've amassed and processed a substantial dataset of 5313 tweets. The preprocessing step entails cleansing the data, removing noise, and transforming the text into a suitable format that can be used for training the model efficiently.

Text Pre-Processing:

In the sphere of NLP, the pre-processing stage holds massive significance as it includes important tasks consisting of data cleansing and transformation. The objective here is to convert the raw text statistics into a format suitable for training the model [8].

Text Normalization:

Normalization of the text (the next stage in natural language analysis) normalizes the text information which later on is used to run the analysis. It is surrounded by a chain of operations that involves punctuation removal, lower-casing of all text, and special characters removal. By using these normalization techniques, we attain a greater uniform representation of the textual content, facilitating powerful evaluation and enabling NLP models, including advanced ones like XLNET, to recognize the underlying semantic content. This system lays the foundation for robust model training, complementing the overall accuracy and comprehension of the language used within the data.

Tokenization:

Tokenization can be seen as the foundational technique in NLP. It contributes a lot by working with texts such as tweets. It involves segmenting text into tokens, which are the basic units upon which most NLP processors operate, facilitating various tasks including morphological analysis. Therefore, tokenizing our tweet dataset provides the necessary structure and training for the required analysis and processing of the text. Hence, we proceed to the next step which is very essential for an actualization of the various roles within the NLP tasks and the extraction of meaningful information from the Tweets posted.

Model Training:

After preprocessing, the next step taken was the training of processed data. The choice of a 2-way type model was motivated by [8].

XLNET Model:

XLNET is a complex transformer-based language model that employs a permutation-based training approach to efficiently capture dependencies amongst all tokens within an input series. Interestingly, the architecture of XLNET is comparable with many transformer-based models like BERT. However, these models have an additional feature that sets them apart from their counterparts. Besides the above-mentioned attributes performance, this language processing model can be identified as unique and indelible in the field of NLP [9].

XLNET distinguishes itself from BERT in phrases of its training goal. While BERT relies upon left-to-right or masked left-to-proper training to predict masked tokens in a set order, XLNET employs a sequence-based training method [10]. In this method during learning, XLNET considers all viable changes of the input tokens in preference to being constrained to a specific order. As a result, XLNET captures two-way context and dependencies among all tokens in the enter dataset, main to a greater complete and nuanced learning of the language.

The architecture of XLNET is a hit fusion of the strengths discovered in automated fashions like Transformer-XL and masked language fashions like BERT. This composition boosts XLNET as the top neural language representation model, thus achieving a range of activities from NLP tasks at large [11]. This approach that shares both the input and the output contexts reinforces its ability to handle sophisticated language modalities, hence it is a proper tool in the realm of multi-lingual natural processing.

This self-attenuation structure is designed as an improvement over the conventional transformer version to overcome its limitations. The first component is the content movement representation, which is similar to the same old self-attention in Transformer. It considers both the content x_{z_t} and the position information z_t of the target token within the input sequence as depicted in Figure 2. By doing this, the model can capture even more relevant interactions between objects and topics within the context. Another approach, termed illustration, is for this model to function as a substitute for BERT's [MASK] mechanism. It employs query circulation attention to predict the target token x_{z_t} based solely on its positional information, excluding the actual content. The model has entry to the placement information of the goal token and the context statistics before that token for making predictions.

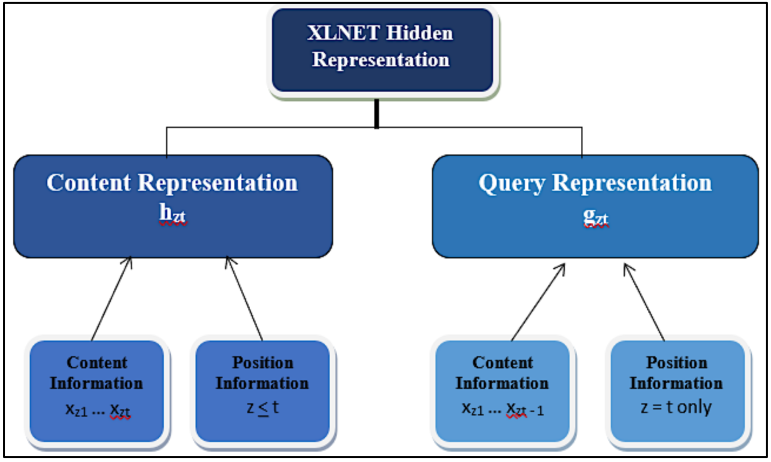

XLNET Hidden Representation
Content Representation h_{zt}
Query Representation g_{zt}
Content Information $x_{z1} \dots x_{zt}$
Position Information $z \leq t$
Content Information $x_{z1} \dots x_{zt-1}$
Position Information $z = t$ only

Figure 1: XLNET with 2 sets of hidden representations

By incorporating those two types of self-attention mechanisms, the Two-Stream Self-Attention architecture pursuits to better seize both local and global dependencies in the input sequence, thereby enhancing the accuracy of predictions for the target tokens. The Two-Stream attention mechanism in XLNET leads to a target-conscious prediction distribution. The key difference between XLNET and BERT lies in their pretraining strategies. Unlike BERT, XLNET does not rely on data corruption and masking during pre-training. By avoiding BERT's masking limitations, as referred to earlier in the autoencoder model, XLNET achieves its target focus and offers improved overall performance.

XLNET improves its ability to capture lengthy-variety dependencies in comparison to RNNs and general Transformers by incorporating Transformer-XL's relative encoding scheme and section repetition mechanism. The relative positional encoding is implemented primarily based on the original sequence, while the segment-level repetition mechanism prevents context fragmentation and enables the reuse of past sentence segments with new ones, thereby maintaining long-term period context. By including phase-stage repetition in hidden states, XLNET stands apart from Transformer, resulting in enhanced overall performance and better handling of remote dependencies.

XLNET integrates Transformer-XL into its improved training framework, permitting the incorporation of the repetition mechanism. This mechanism is used in the proposed permutation setting of XLNET to reuse hidden states from previous segments. However, the factorization order within the permutation from preceding segments is not cached and reused in future computations. Only the content representation of the section is retained in the hidden states, allowing for efficient handling of long-range dependencies without the need to store and reuse the whole factorization order.

LSM and Multi-Head Attention Layer: This layer is comprised of two sub-layers working together.

LSM (Likely Long Short-Term Memory): This is a type of recurrent neural network (RNN) adept at handling sequential data like text. LSTMs can capture long-term dependencies within sentences, crucial for tasks like sentiment analysis or machine translation.

Multi-head Attention:

This is a mechanism that allows the model to focus on specific parts of the input text that are most relevant to the current processing step. It essentially helps the model pay attention to different aspects of the input simultaneously.

Output Layer:

This layer takes the processed data from the previous layer and generates the final output, which could be a variety of things depending on the specific NLP task. Examples include classification which is classifying the sentiment of a text review (positive, negative, neutral), machine translation which is converting text from one language to another and text summarization which is creating a concise summary of a lengthy piece of text [12].

Overall, Figure 3 represents a simplified illustration of how a deep learning model can be structured to process and analyze textual data. By leveraging LSTMs for capturing long-term dependencies and multi-head attention to focus on relevant parts of the input, the model can learn complex patterns within the language and perform various NLP tasks.

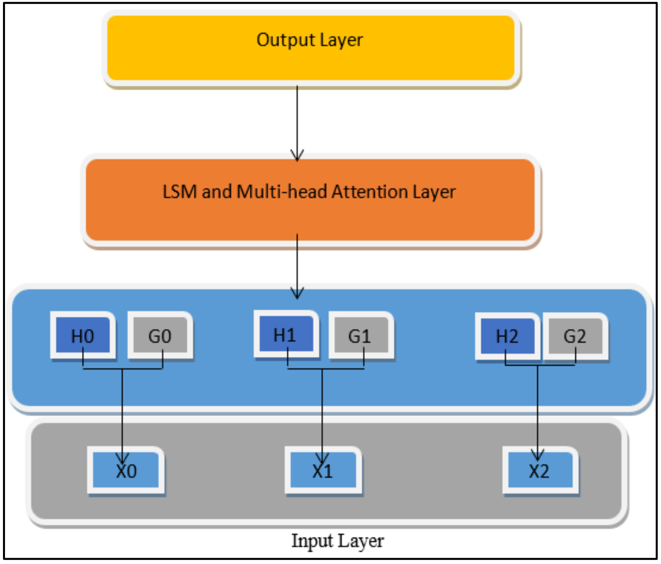


Figure 3: Layers of XLNET

Model Structure:

The XLNET model architecture can be divided into four major parts. The initial stage involves processing the entered text, where it is tokenized, meaning it is divided into individual units or words for further analysis. This process essentially operates with word vectors, resulting in low-dimensional representations of the textual input being used. Subsequently, the XLNET layer dissects the text representation and further extracts features. Such operations capture crucial patterns from the input text and then the result is transferred into Long Short-Term Memory (LSTM) and Multi-head Attention layers.

The LSTM is a type of recurrent neural network designed to capture context information from input sequences. It achieves this by using input gates, forget gates, and output gates to control the flow of information. The LSTM selects relevant memory feature vectors and integrates them to generate meaningful output. Finally, the Multi-head Attention Layer calculates the probability of attention from multiple perspectives. This lets the model assign distinctive weights to the extracted feature vectors, essentially giving greater importance to certain parts of the input text. By doing so, the model aims to acquire effective text classification.

Data Splitting:

We split our dataset into two components, training, and validation sets, as shown in Figure 4.

Training Dataset: This is a subset of the data used to train the model. The model learns patterns and relationships from this data.

Dataset
Training Dataset
Validation Dataset

Figure 4: Splitting of the datasets

Validation Dataset:

This is another subset of the data used to assess the model's performance on unseen data. It's essential to prevent overfitting, which occurs when a model performs well on the training data but poorly on new data. The training dataset constituted 80% of the complete dataset, at the same time as the validation dataset consisted of 20% of the dataset. These separate datasets were used to train the model, investigate its performance at some point of validation, and eventually, compare its generalization on the test set.

Testing:

To evaluate the effectiveness of our model in classifying tweets as they should be, we tested it on a set of tweets. The model achieved an outstanding F1 score of 0.96 on the test dataset. This high F1 score shows that the model demonstrates sturdy overall performance and reliability in classifying tweets with an excessive stage of accuracy.

Results and Discussion:

From the results presented in Table 1, it is evident that our model achieved an accuracy of 94.16% on the test set. This accuracy rate reflects the model's ability to correctly classify tweets with a high level of precision. Looking ahead, we intend to further improve the model's performance by fine-tuning various parameters. Despite the impressive accuracy achieved, we also evaluated the model's performance using the F1 score metric, which demonstrated an outstanding score of 96%. This high F1 score indicates a robust performance in both precision and recall, underscoring the effectiveness of our model in classifying tweets accurately.

Table 1: Results of XLNET Model on test set

Method	F1 Score	Accuracy
XLNET	0.960	0.9416

Conclusion:

In our paper, we tackled the significant challenge of identifying relevant tweets after a flood occurrence, recognizing the important role of timely and accurate information for informed decision-making by governmental authorities. Leveraging advanced techniques in NLP, particularly XLNET, enabled us to efficiently sift through the immense volume of tweets generated during such events. Our findings underscore the potential of deep learning models, such as XLNET, in addressing complex challenges at the intersection of natural disasters and social media. By effectively harnessing these technologies, we have demonstrated the capability to extract valuable insights from large-scale social media data, thereby facilitating more effective disaster response strategies.

Furthermore, our research highlights the importance of ongoing innovation in the field of NLP, particularly in the context of disaster management and response. As the volume and complexity of social media data continue to grow, leveraging cutting-edge techniques like XLNET will be essential for improving the efficiency and accuracy of information extraction and decision-making processes in disaster scenarios.

References:

[1] N. Pourebrahim, S. Sultana, J. Edwards, A. Gochanour, and S. Mohanty, "Understanding communication dynamics on Twitter during natural disasters: A case

study of Hurricane Sandy," Int. J. Disaster Risk Reduct., vol. 37, p. 101176, Jul. 2019, doi: 10.1016/J.IJDRR.2019.101176.

[2] N. Said et al., "Natural disasters detection in social media and satellite imagery: a survey," Multimed. Tools Appl., vol. 78, no. 22, pp. 31267–31302, Nov. 2019, doi: 10.1007/S11042-019-07942-1/METRICS.

[3] "Using Twitter as a data source." Accessed: May 11, 2024. [Online]. Available: https://eprints.whiterose.ac.uk/126729/8/Normal_-_Ethics_Book_Chapter_WA_PB_GD_Peer_Review_comments_implemented__1_.pdf

[4] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: What twitter may contribute to situational awareness," Conf. Hum. Factors Comput. Syst. - Proc., vol. 2, pp. 1079–1088, 2010, doi: 10.1145/1753326.1753486.

[5] M. Imran and C. Castillo, "Towards a data-driven approach to identify crisis-related topics in social media streams," WWW 2015 Companion - Proc. 24th Int. Conf. World Wide Web, pp. 1205–1210, May 2015, doi: 10.1145/2740908.2741729.

[6] S. Zhang, D. Caragea, and X. Ou, "An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6860 LNCS, no. PART 1, pp. 217–231, 2011, doi: 10.1007/978-3-642-23088-2_15.

[7] A. H. Tanim, C. B. McRae, H. Tavakol-davani, and E. Goharian, "Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning," Water 2022, Vol. 14, Page 1140, vol. 14, no. 7, p. 1140, Apr. 2022, doi: 10.3390/W14071140.

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," Adv. Neural Inf. Process. Syst., vol. 32, 2019, Accessed: May 11, 2024. [Online]. Available: https://github.com/zihangdai/xlnet

[9] W. J. Luo Pengcheng, Wang Yibo, "Research on automatic classification of literature subjects based on deep pre-trained language model [J]," J. Inf. Sci., vol. 10, 2010.

[10] Z. Chong, "Research on Text Classification Technology Based on Attention-Based LSTM Model [D]," Nanjing Nanjing Univ., 2016.

[11] et al. Liang Xiaobo, Ren Feiliang, Liu Yongkang, "Machine reading comprehension model based on double- layer Self-attention [J]," Chinese J. Inf., vol. 32, no. 10, pp. 130–137, 2018.

[12] S. Yu, D. Liu, W. Zhu, Y. Zhang, and S. Zhao, "Attention-based LSTM, GRU and CNN for short text classification," J. Intell. Fuzzy Syst., vol. 39, no. 1, pp. 333–340, Jan. 2020, doi: 10.3233/JIFS-191171.

Beyond CNNs: Encoded Context for Image Inpainting with LSTMs and Pixel CNNs

Taneem Ullah Jan, Ayesha Noor
Computer Science & IT University of Engineering and Technology Peshawar, Pakistan
*Correspondence: taneemishere@gmail.com, ayesha.noor2324@gmail.com.

Citation | Jan. T. U, Noor. A, "Beyond CNNs: Encoded Context for Image Inpainting with LSTMs and Pixel CNNs", IJIST, Special Issue pp 165-179, May 2024
Received | May 04, 2024 **Revised |** May 15, 2024 **Accepted |** May 20, 2024 **Published |** May 26, 2024.

Our paper presents some creative advancements in the image in-painting techniques for small, simple images for example from the CIFAR10 dataset. This study primarily targeted on improving the performance of the context encoders through the utilization of several major training methods on Generative Adversarial Networks (GANs). To achieve this, we upscaled the network Wasserstein GAN (WGAN) and compared the discriminators and encoders with the current state-of-the-art models, alongside standard Convolutional Neural Network (CNN) architectures. Side by side to this, we also explored methods of Latent Variable Models and developed several different models, namely Pixel CNN, Row Long Short Term Memory (LSTM), and Diagonal Bidirectional Long Short-Term Memory (BiLSTM). Moreover, we proposed a model based on the Pixel CNN architectures and developed a faster yet easy approach called Row-wise Flat Pixel LSTM. Our experiments demonstrate that the proposed models generate high-quality images on CIFAR10 while conforming the L2 loss and visual quality measurement.

Keywords: Index Terms—Image Inpainting; GAN; Pixel CNN; LSTM

Introduction:

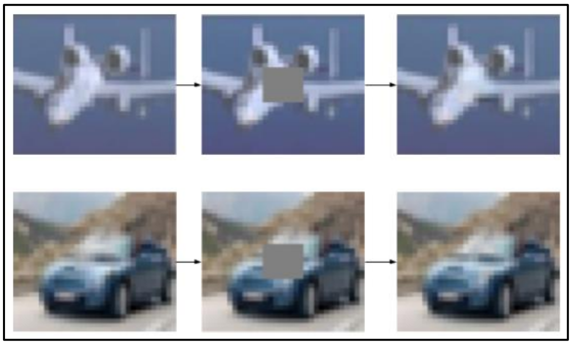
Image in-painting involves reconstructing damaged or missing parts within an image, commonly applied in the restoration of old photos or paintings and image editing tasks. Notably, tools like Photoshop feature a robust completion tool that doubles as a removal tool. Despite Convolutional Neural Networks (CNNs) surpassing human classification accuracy on ImageNet [1], in-painting outcomes still fall short of human predictions. The challenge lies in the vast number of possible ways to fill an $8 \times 8 \times 3$ section, around $50{,}000$ possible ways, while ImageNet has only $32{,}000$ classes. Interestingly, humans effortlessly mentally reconstruct missing image sections by comparing context with their knowledge of the world, enabling scene and object recognition, as well as extrapolating missing elements from memory. Artificial and computer-based methods leverage similar principles in their approach.

There exist two categories of approaches: local methods [2][3] solely rely on contextual information, such as color or texture, and aim to extend and blend these details seamlessly. These techniques demand minimal previous perception and training. For instance, if there are no eyes on the head part, a local method might replace it with a patch of skin-textured pixels. However, these methods face limitations in scenarios where larger patches are absent, excelling primarily in tasks like watermark removal. On the other hand, more advanced methods adopt a global, context-based, and semantic approach [4][5][6]. These methods identify patterns within images, like a door or a cabin, which leverage this understanding to infuse the empty spots. Unlike local techniques, they hold the importance of specific elements, such as the necessity of a nose in a particular facial position, constructing a fitting replacement based on their broader knowledge of the context.

An attractive aspect of such problems lies in the effortless generation of extensive datasets for training. Some datasets of images like ImageNet and CIFAR10 [7], which we used for our convenience, can be readily pre-processed by introducing alterations to the images. This approach allows the generation of hundreds of millions of training examples, enabling the training of larger deep networks.

The Problem:

Every image is divided into two segments: the portion that is absent and under reconstruction, and the contextual part. To enhance simplicity, we suppose that the section which is missing, is a square of dimension $n \times n$. However, it is worth mentioning that the functionality of the network remains consistent even when dealing with arbitrary removals. See figure 1.



Original Image Input Images Output Images

Figure 1: Image In-Painting

The concept of image in-painting is commonly presented as a constrained image generation challenge. The network is tasked with receiving a contextual input and generating an image with identical dimensions as the absent patch. The ultimate assessment hinges on the average element-wise L2 [8], the distance between the original missing section $Y \in R^{n \times n \times 3}$ and the predicted counterpart $\widehat{Y} \in R^{n\times n\times 3}$. In our illustrative instance using CIFAR10, $n = 8$. For a given sample i, the loss is calculated as follows:

$$L = \frac{1}{n^2}\sum_{p,q,r} \left(Y^{(i)}{}_{p,q,r} - \widehat{Y}^{(i)}{}_{p,q,r}\right)^2$$

We introduce an evaluation metric termed approximate exact-match (AEM), which is solely used for assessment purposes. We observe that a slight move of one or two elements in the value of the pixel channel has minimal visible impression, the effect can be seen in figure 2. Therefore, if a calculated pixel value falls from the accurate image value in the range of ± 5 at each channel, this qualifies as the same and or equal. We present the mean-AEM, denoted as MAEM, where a value of 100% implies that the visual impression of the image is almost identical to the original image, a simpler version of Generated Image Quality Assessment (GIQA) [9].

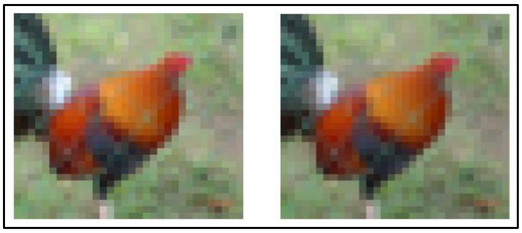


Figure 2: On the left is ground truth example and on the right is a ±5 randomly added to each pixel channel

In-painting comes in two forms: blind [10][11], where the network lacks information about the position and shape of the missing area. On the other hand, in the non-blind [12], such details are provided within the inputs. Extensive research indicates that blind in-painting poses a hard challenge. While non-blind in-painting is more extensively documented, there remains considerable scope for enhancement. Consequently, our emphasis is placed on the latter, reflecting a deliberate choice to concentrate efforts on non-blind in-painting, recognizing its potential for further advancements.

Our primary goal is to make use of the latest computer vision methodologies to develop a resilient and well run in-painter. Here we aimed to attain satisfactory outputs in the form of mean square error or $L2$ loss, benchmarked against current models. Initial results and the existing methods indicate minimal empirical distinctions between utilizing a square-shaped mask and employing randomly selected rectangular shape masks in the middle of given images. Therefore, for implementation simplicity, our focus primarily revolves around centered square-shaped cover ups of a consistent size. Specifically, on dataset like CIFAR10, this involves the removal of a patch from every image at the center of size 8×8.

Related Work:

Various researchers have investigated a diverse range of methods to tackle such challenges. A notable work by Pathak et al. [13], from where our initial inspiration took root. They adapted the conventional Generative Adversarial Network (GAN) [14] model by incorporating contextual information of image, rather than using stochastic noise, for predicting the incomplete section. Highlighting the importance of Leaky ReLU as detailed in [15] within the discriminator, and exclusion of pooling layers, they implemented compression and decompression operations with strides differing from 1. Their model training involved a combination of $L2$ loss and adversarial loss, measuring the success of the generator in deceiving the discriminator. However, our study revealed that this increased the risk of overfitting by utilizing fully connected layers in several instances. On the other hand, they have used relatively simple CNN model architecture as for encoders and decoders. In our work, our aim was to not only explore this but also investigate modern methodologies renowned for state-of-the-art results in domains like VGG [16], Inception [17], and others similar.

In a publication [18], their proposed approach revolves around modeling images as the conditional probability product distribution. Here the objective was to calculate the image pixels in sequential order, such as to the bottom-right point from the top-left. These functions which calculate conditional probability are shaped to either capture the contextual information of pixels within the upper rows with the help of recurrent networks or by employing CNNs to operate on local pixels (Pixel CNN [18]). Although originally designed for image generation, this method proves adaptable for our objective of optimizing the probability of the reconstructed image by providing pixel data.

Yang et al. [19] present a technique aimed at addressing the in-painting of large sections within extensive images. Traditional models encounter challenges in producing sharp results for such tasks, often resulting in blurry outputs with noticeable edges between the contextual information and the reconstructed region. The authors reduce this issue by incorporating hierarchical approaches to introduce fine-grained features above the regenerated spots, enhancing the resolution of in-paintings. Their approach involves training two distinct networks; first is a feature extractor that is assessed using a content-based loss same as in [13], second is texture-based network which employs minimizing the texture-based local loss. The incorporation of perceptual equivalence ensures that the patterns within the generated spots closely align with the context of local texture. This strategy significantly improves visual sharpness, it may be considered unnecessary in our case, given the context of working with smaller images.

The structure and process of our work represent a substantial adaptation of the CIFAR10 classification pipeline, originally derived from resources. The only retained elements are the queuing system and monitoring session. It is noteworthy that the autoencoders, GANs, and Wasserstein GANs (WGAN) [20] utilized in our study are also publicly available through standard sources. Moreover, there are adaptations of [13] in existence, their inefficiency and non-functionality led us to avoid their use. In the exploration of advanced models such as VGG, Inception, and ResNet [21], we adapted them from model repositories, and made adjustments to tailor them to CIFAR10 specifications.

Methods:

Dataset:

Our primary dataset is CIFAR10, comprising images of dimensions $32 \times 32 \ \times \ 32$. It consists of around 50 thousand training and about 10 thousand test samples. Notably, the CIFAR10 has been taken from the Tiny Images dataset [22]. When our model achieves stability, we extend our training efforts to leverage this larger dataset. The inclusion of a more extensive variety of images from the larger dataset proves helpful, particularly given that CIFAR10 is limited to only 10 classes. The substantial number of samples in the larger dataset serves as an efficient countermeasure against overfitting. Our model facilitates seamless scaling for training on this expanded dataset, resulting in high training performance.

Additionally, in the middle we also incorporate data augmentation steps on our CIFAR10 that helps in amplifying the dataset further enhancing the adaptability of this model for minor variations effectively. This involves introducing small, random adjustments to hue, saturation, contrast, and applying random Gaussian blur to the images. Also we put zeros in place of the middle $8 \times 8 \times 3$ crop.

Autoencoders:

In-painting constitutes a subset of a broader category of image generation problems involving the creation or modification of pixels. Tasks like deblurring, denoising, and small-scale blind in-painting, such as text removal, are commonly addressed using autoencoders. Autoencoders typically consist of two main components, an encoder and a decoder. Initially, an image is encoded in a latent feature space, or embedding. On the other hand, the decoder reconstructs the original image based on this embedding. The training process involves jointly optimizing both networks to decrease the input-output disparity. This architectural configuration requires the encoder to maximize the information encoded into the embedding. The encoder must learn the abstract intelligent features to compress the information in least possible loss, that too within the limited size of this latent space. In CIFAR10 experiments, where input images consist of vectors with a length of 3072, it has been found that employing a bottleneck size of 512 and or 1024 produces satisfactory results. Larger bottlenecks do not provide any attraction to autoencoders to discover accurate representations but keeping raw sections of pixels proves to be sufficient. On the other hand, using smaller bottlenecks lack sufficient capacity to encode the input information. This can be seen in figure 3.

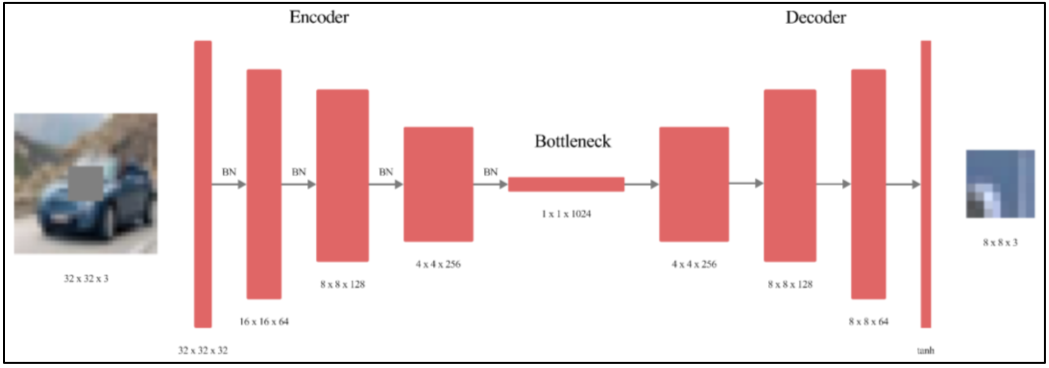


Figure 3: Autoencoder architecture of our model

The typical ratio between the image and latent space dimension in the existing literature often exceeds than what we employ, ranging from 3 to 6. Empirically, our choice stems from working with small images, introducing a challenge in identifying unified entities and components in the target image. Achieving improved encoding ratios, as seen in ImageNet; 12 or more, for instance, is more feasible because it is comparatively easier to isolate high-level features in larger images like those in ImageNet.

To establish a baseline, we initially constructed a straightforward CNN architecture. The input dimension is $32 \times 32 \times 3$, and the output dimension is $8 \times 8 \times 3$. Guidance from the literature advocates for the use of several small filters rather than larger ones. The rationale behind this approach lies in the ability to achieve an equivalent receptive field with deeper networks. Throughout, we utilized filters of size 3 exclusively. While filters of size 5 and 7 were tested on multiple occasions, their performance consistently lagged behind that of size 3 filters.

Our experimentation involved a $[\mathrm{Conv} - \mathrm{Conv} - \mathrm{Pool2}] \times 3$ architecture, followed by either 2 fully connected layers or 2 convolutional layers. This architecture closely resembles a well-performing design on CIFAR10, featuring smooth dimension reduction coupled with an increment in the number of filters, initiating at 32 or 64 and doubling at each pooling step. Subsequently, we opted to eliminate max pooling by entirely substituting it with convolutional layers of stride 2, maintaining the same filter progression. Transitioning from the $4 \times 4 \times 256$ convolutions to the bottleneck can be conceptualized as a stride 4 operation with 1024 filters. Across all scenarios, ReLU activation consistently outperformed alternatives. The incorporation of Batch Normalization on each layer given substantial performance improvements, approximately around $\pm 15\%$, with comparable execution times.

In terms of qualitative assessment, as previously mentioned, a prominent issue is blurriness: while colors are generally accurate, details and textures tend to be lost, resulting in predicted sections that often resemble indistinct dots, failing to seamlessly blend into the image. To address this challenge, our objective is to diminish visible continuity errors between the predicted section and the context. We aim to achieve this by predicting a patch that slightly overlaps with the context and imposing a robust penalty on the loss specifically within this overlap region.

Generative Adversarial Networks (GANs):

The first figured-out method demonstrated notable L2 loss. Nevertheless, the generated output images consistently exhibited blurriness and a lack of intricate details, as illustrated in figure 4. The inherent nature of L2 loss encourages the network to adopt a risk-averse approach, generating safe predictions characterized by a lack of sharp shapes and substantial changes across the patch. The inclination to generate blurred, average-color images derived from context minimizes the occurrence of substantial errors. Consequently, while the model excels in terms of the L2 norm, the generated outputs fall short of realism when evaluated by a human eye.

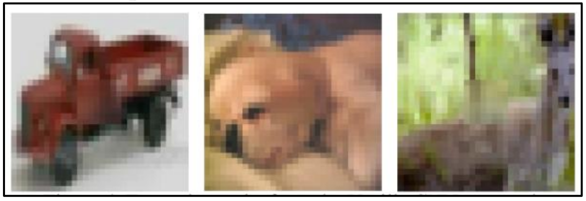


Figure 4: Results from normal CNN architecture

Deep Convolutional GANs (DCGANs):

To encourage our network to take more risks and generate realistic outputs, we opted to explore GANs. Here the underlying objective is the emulation of visual and perceptual evaluation and assessment. For instance, if a model predicts a human head without eyes or mouth, this is considered superior to generating an image with a black spot at the center. The rationale is that an averagely blurry image will never appear realistic to a human observer.

In the realm of GANs, the key concept involves training a discriminator network (D) concurrently with a generator network (G). The discriminator learns to assess the authenticity of an image, distinguishing between real and or generated spots and dots. Normally they are on both real and generated examples with distinct labels for each. The generator faces a penalty with an increased loss if the output image is viewed as generated by the discriminator. To outsmart the discriminator, our generator aims to create natural looking and visually clear images. As the discriminator improves, both networks benefit from the feedback loop, driving mutual enhancement.

The introduction of adversarial networks may not necessarily lead to an improvement in $L2$ loss, as the generator could throw significant errors. Although, the primary objective is to enhance the realism of the generated images. In the end, the crucial aspect is whether the predicted human eye, for example, appears authentic within the context of the image. Even if the $L2$ loss indicates a substantial difference from the ground truth; it constitutes an acceptable result, if the predicted sections are admissible in the context of the human head. Here the focus shifts from minimizing pixel-wise differences to creating outputs that are visually convincing and contextually appropriate.

Now, loss $L = \alpha L_{rec} + (1-\alpha) L_{dics}$, here L_{rec} is the same mean square error, while L_{disc} known as the probabilistic output from the discriminator for the generated images. This also can be called sigmoid cross entropy. Given that α changes the scale of the loss, using the loss itself for the optimization of this crucial hyperparameter is impractical. This challenge is to intricately linked to the observation that a lower $L2$ loss should not automatically correlate with script and visually better output. The choice of α is aimed at optimizing the results of our output samples and ensuring. Overall, this can be calculated as:

$$L_{disc} = -\sum_{i} log(p_i) = -\sum_{i} log\big(D\,(\hat{Y}^{\,(i)})\big)$$

The discriminator, in our setup, is a network consisting of six convolutional layers, having a stride value of 2. After this, there is a final convolutional layer that reduces the depth value to 1, resulting in a normal value output. Conceptually, a typical classification task is addressed by our discriminator. Therefore, various recent methods can allow them to function as a typical generator network. We leveraged pre-trained models, including VGG, Inception, and ResNet, to enhance our discriminator. Since these models are designed for larger inputs like ImageNet, we attempted to pad our images, but the results were unsatisfactory. As a solution, we adjusted these techniques to operate on our smaller inputs. This involved removing the first layers with dimensions higher than 32×32 and injecting our input into the initial smaller layers. To prevent overfitting and improve execution time, we also reduced the depth and the number of filters in consideration of the fact that our input contains less information. The same adaptation strategy was applied to the encoder, where we experimented with a variety of high-performing models.

Learning Tricks:

As usual we encountered challenges in effectively training a GAN. Visual inspection of the predicted images revealed the presence of colored artifacts, as illustrated in figure 5. Upon closer inspection, it became apparent that shapes and color gradients aligned with the context, but the colors were distorted, particularly noticeable in the case of the plane. Due to difficulties in achieving satisfactory convergence with GANs, the $L2$ loss was higher compared to vanilla autoencoders. The current architecture exhibits a sensitivity to randomness, where the exact same method may result in convergence or divergence under different circumstances.

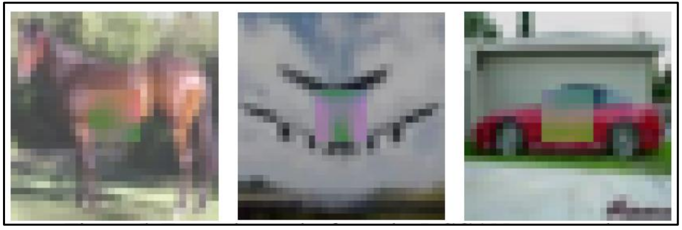


Figure 5: Results from DCGAN architecture

Training GANs poses inherent challenges as they are often unstable and highly sensitive to network architectures and parameters setting. Finding the right hyperparameters and architecture details can be a tedious task. Another common issue encountered is the imbalance between the discriminator and the generator, where one may exceed and overwhelm the other. As an example, if out network's discriminator becomes excessively robust, then our generator will struggle to deceive it, leading to a scenario where the adversarial loss sharply rises at high values, increasing the $L2$ loss. Conversely, if the generator becomes too dominant, it can successfully outsmart the discriminator, causing it to halt learning and assign identical probabilistic values to both fake and real samples. Then, it is symmetrical to have no adversarial loss. To address these challenges, we experimented with a few learning tricks aimed at mitigating these issues.

Separate Batches:

We provided two distinct batches: one containing solely real samples and the other composed entirely of fake samples. This approach facilitates clearer and more direct updates for each batch, allowing the network to focus distinctly on improving its understanding of real examples and enhancing the detection of fake ones.

Soft and Noisy Labels:

Soft and noisy labels were employed following the approach outlined in [23]. True images were assigned random labels between 0.9 and 1.1, while fake images received labels ranging from 0 to 0.2. Additionally, labels were occasionally flipped randomly between classes.

These strategies introduce noise and enhance the robustness of the discriminator, contributing to improved training stability.

Maximizing log D Instead of Minimizing log (1 − D):

Rather than minimizing $log(1 - D)$, we chose to maximize $logD$. Here D represents the discriminator's output. While these formulations are equivalent in this context, the latter avoids the issue of vanishing gradients early in the training process, enhancing the stability and effectiveness of the learning procedure.

WGANs:

To address the challenges in training GANs, as discussed above, WGANs [20] have also been explored as an alternative to traditional GANs, relying on the Earth-Mover distance [24] for learning distributions. WGANs minimize the Earth-Mover distance, allowing for the estimation of this metric during training, which correlates well with visual quality. WGANs also eliminate the need for a realness probability from the discriminator, using an unbounded score, and enforce Lipschitz continuity through gradient clipping. Importantly, WGANs do not require a delicate balance between the generator (G) and discriminator (D), allowing for training the discriminator to convergence at each step. In practice, we train the discriminator 10 times at each iteration for stability and convergence.

Reducing Continuity Errors:

The adoption of WGANs marked a significant improvement in training a robust adversarial network. However, a noticeable issue persisted; the distinct visibility of the border between the context and the reconstructed region. Despite the challenge in visually explaining this clear border appearance, a detailed analysis revealed the absence of an organized color correction with pixels was highly noticeable, but a random error tended to blend more seamlessly, refer to figure 2. This border problem remained even in instances with otherwise accurate predictions.

To address this issue, we implemented a clever trick: predicting a 16×16 center square that encompasses the original 8×8 target along with a an each side overlap of 4-pixels. While we penalized our reconstruction loss more than 20 times, it is relatively simple for the model to accurately calculate and predict it. Here this strategic approach significantly enhances the quality of border merging. By ensuring that there are no major discontinuities within the output, this effectively reduced gaps between the input and prediction. It is important to note that, for visualization purposes, we retain and display only the 8×8 target in the final results.

Prior to using this strategic trick, feeding the entire image to the discriminator posed challenges, as the obvious border served as a key indicator for determining the authenticity of the image. However, with the introduction of the overlap and the subsequent reduction in the visibility of borders, we successfully transitioned to providing the complete image to the discriminator. In this improved setting, the discriminator could assess the entire image as a cohesive unit. Notably, an 8×8 square, either from a real or fake image, typically lacks sufficient meaningful information on its own, making it challenging for the discriminator to make a decisive meaning. In contrast, evaluating the entire image facilitates a clear and more straightforward return for the discriminator. Overall the results can be seen in figure 6.

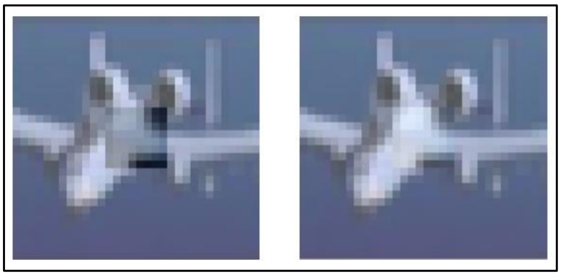


Figure 6: On the left is the result without overlap trick and on the right is with overlap

Density Based Models:

The Idea:

We opted to implement unsupervised models alongside our supervised methods for filling the contextual portion in the center of an image. In this approach, our objective is to approximate the function of Probability Mass (PMF), p of all images by multiplying conditional probabilities. Therefore, for a given image x of dimension $n \times n$, where the pixels are denoted as $\{x1, x2, \dots, xn^2\}$, the PMF is expressed as:

$$p(x) = \prod_{i=1}^{n^2} p(x \mid x_i, \dots, x_{i-1})$$

Where $p(x \mid x_i, \dots, x_{i-1})$ denotes the probability of the i^{th} pixel or x_i assuming the values $x1, \dots, x_{i-1}$ have been observed in the previous pixels.

Each pixel is represented by a triplet of integers as: $(R, G, B) \in \{0, 256\}^3$. Now using this notation $x_i = (x_{i,R}, x_{i,G}, x_{i,B})$ and $x < i = (x1, \dots, x_{i-1})$, the conditional probabilities can be written as:

$$p(x_i \mid x < i) = p(x_{i,R} \mid x < i) \times p(x_{i,G} \mid x < i, x_{i,R}) \times p(x_{i,B} \mid x < i, x_{i,R}, x_{i,G})$$

Therefore, the final probability $p(x)$ is obtained by multiplying $3n^2$ terms. Consequently, while in-painting images, our aim is to populate the space with the pixels by maximizing this probability.

The approach we take is greedy, which means that assuming we have learned the characteristics of the function p and are given an image with m missing pixels at positions $i_1 < i_2 < \dots < i_m$, we will first set the value $x_{i1,R}$ given $x < i_1$, then the value of $x_{i1,G}$ given $(x < i_1, x_{i1,R})$, and eventually we set $x_{im,B}$ given $(x < i_m, x_{im,R}, x_{im,G})$. Hence, as this approach is easy to implement and is computationally efficient, there is no assurance that our reconstructed image maximizes the likelihood $p(x)$.

An intuitive approach is to establish an order on the pixels, progressing from the top-left corner to the bottom-right corner, scanning rows consecutively. Once the order is defined, the next step involves selecting the type of conditional probability to be learned. We will introduce two distinct methods, drawing inspiration from [18]. The first method involves a Pixel CNN, where the conditional probability of a pixel is determined through a convolutional neural network operating on pixels in the surrounding neighborhood. The second method is a flattened row Long Short-Term Memory (LSTM), changing significantly from the one proposed in [18]. In this alternative approach, the probability is computed based on pixels from preceding rows, which are then fed into an LSTM network.

Pixel-CNN:

In order to calculate the conditional probability $p(x_i \mid x < i)$, a convolutional neural network is implemented. Various architectures were explored, incorporating combinations of convolutional and leaky ReLU layers. The model concludes with three fully connected layers, each followed by a softmax layer corresponding to the three-color channels (R, G, B). Our model applies softmax loss function, computing the estimated probabilities for the true pixel values $\hat{p}_{i,true}$ in an image with n^2, are computed as:

$$L_{softmax} = -\sum_{i=1}^{n^2} log(\hat{p}_{i,true})$$

An alternative perspective argues that the SoftMax loss may not be directly applicable, considering that the task at hand is not a classification one. In this context, calculating the value of 117 rather than 118 could not be as crucial as predicting 117 instead of 245. While it is obvious that a low probability $\hat{p}(x_{i,R} = 120)$ may not pose a significant issue. If the emphasis on values within the range (112, 124) is sufficiently enough, the softmax loss function has shown interesting results, even given the extensive training examples compared to the 256 possible values for each channel. The model we came up with finally incorporates a 5x5 section within the image for the initial convolutional layer, positioned at the top-left corner of the pixel grid. See figure 7.

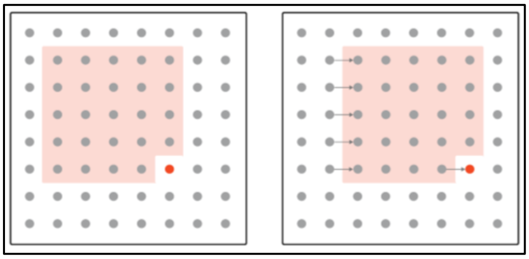


Figure 7: The red pixels are surrounded by gray dots; these are used for predicting the conditional distribution which form the neighborhood. Subsequently, to estimate the distribution of the next pixel we the same network, shifted one step to the right

Our method offers the advantage of utilizing a similar model, not only this but provides a similar set of learning parameters to compute the final conditional probabilities for each. The complexity remains persistent, irrespective of the image size, as we consistently use the same number of pixels; 24 to calculate the distribution of the next pixel. However, in the case of resolutions, like 128×128 images, this may be necessary to employ a greater relative section for more accurate predictions.

The outcomes obtained with our Pixel CNN model exhibit a slightly lower performance compared to GANs when evaluating the $L2$ loss. Two factors contribute to this disparity:

- Our Pixel CNN was trained using a SoftMax loss, which explains the higher $L2$ loss value of 6.98, as opposed to the results that came from our GANs.
- Iterating through the image from left to right, the Pixel CNN determines the value of each pixel based on its 24 neighbors located at the top-left corner. Consequently, during the gap-filling process the pixels at the bottom of the image remain unused. See figure 8.

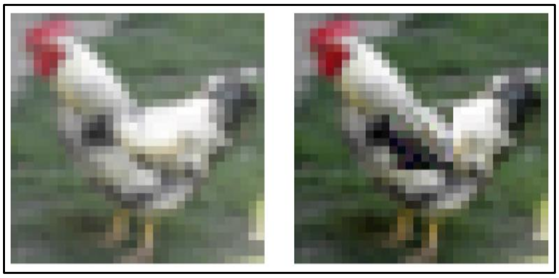


Figure 8: On the left is ground truth example and on the right is a reconstructed image from the Pixel CNN model

At the bottom and right portions of the image, clearly, the reconstruction of the bird reveals a neglect that is predominantly characterized with colors like light gray and white. This omission is particularly noticeable in the reconstructed square, where the discontinuity is more apparent at the bottom and right sides. Additionally, the limited number of dark feathers initially situated at the top-left corner of the absent section have now expanded to encompass a significant portion of the reconstructed square.

We also recognize the presence of the Row LSTM and Diagonal BiLSTM [25] from the same article. We successfully implemented and executed these methods, which yielded satisfactory results. While we refrain from providing an exhaustive report on these techniques due to their complexity in terms of implementation and performance, we attempted to propose a significantly simpler architecture. Despite its simplicity, this alternative architecture demonstrates the capability to deliver relatively good performance on CIFAR10, enabling insightful analyses.

Flattened Row LSTM:

Finally, we present a model that incorporates data from all preceding rows of the image to compute the probability distribution of each pixel. Termed as the Row Flattened LSTM, the architecture of this model can be seen in figure 9.

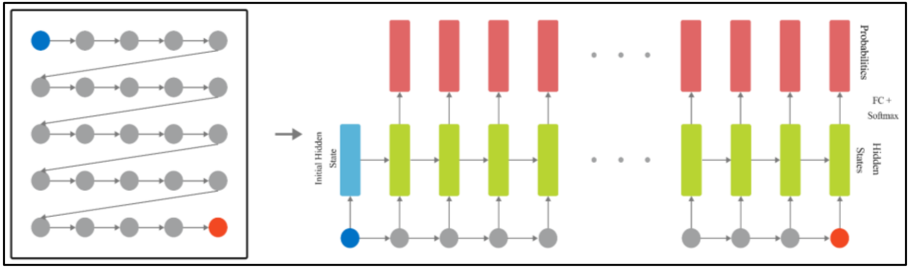


Figure 9: Row Flattened LSTM Architecture

The entire image is flattened, executing to the bottom-right from the top-left corner. Supplied to the LSTM are pixel channels, represented as 256-dimensional one-hot vectors, in the prescribed RGB sequence. With a set of 64 hidden dimensions, we proceed to convert these hidden vectors into output vectors from a fully connected softmax layer, each comprising 256 dimensions. Our Flattened Row LSTM reconstructed image example can be seen in figure 10.

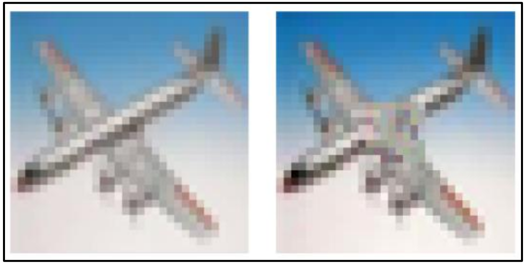


Figure 10: On the left is ground truth example and on the right is a reconstructed image from our Row-wise Flat Pixel LSTM

Our obtained result exhibits certain limitations compared to the Pixel CNN model. Several factors may contribute to these observations:

- The model assigns more influence to the pixels immediately on the left of the reconstructed pixel the way the LSTM scans. This differs from the Pixel CNN approach, where we employed a concentrated region of 24 neighboring pixels located at the top-left corner of our target pixel.
- As our current LSTM model may be too simplistic to effectively interpret this information, its strength lies in capturing all information from previous rows before making predictions. Enhancements, such as incorporating multiple LSTM networks or increasing the hidden dimensions, could potentially lead to more accurate predictions.

Results:

Autoencoders:

We used the Adam optimizer for training and conducted cross-validation for the learning rate. Learning rate decay proved beneficial across all our tests, striking a balance between swift initial learning and a gradual reduction in loss during later stages. However, we encountered challenges in cross-validating the decay rate and decay factor due to the intricacies involved in this process. Furthermore, we implemented dropout at a rate of 0.5 for every activation, and our experiments revealed no substantial changes in accuracy within the reasonable range of values for the probability drop from 40% to 80%. Also, we investigated the impact of the bottleneck size on our fundamental architecture, recognizing it as one of the most critical parameters. But it is worth noting that the best results, shown in table 1; did not emerge from the basic architecture.

Table 1: Train Vs. Val L2 Loss

Size of a Bottleneck	L2 Loss (Training)	L2 Loss (Validation)
256	7.02	7.96
512	6.36	7.41
1024	4.89	7.48
2048	4.23	8.14

It is apparent that overfitting poses a challenge. Significantly decreasing the size of bottleneck, and consequently it reduces the size of fully connected layer. This leads to a substantial decrease in the number of parameters, this is because this layer constitutes the majority of training parameters of the model. Overfitting diminishes notably from 2048 to 512, with 512 appearing to be the optimal choice, while 256 proves to be too small and potentially insufficiently expressive. Furthermore, to combat over-fitting more effectively, we plan to investigate strategies such as expanding the dataset and employing the data augmentation, integrating explicit L2 regularization, and experimenting with techniques like drop-connect, which involves randomly dropping connections in CNNs during training.

WGANs:

As previously mentioned, unlike our vanilla autoencoder, GANs do not exclusively optimize for $L2$ loss. Theoretically, $L2$ loss results should not be superior for GANs. However, we dedicated more time to refining GANs because our best $L2$ only architecture gave visually poor results; the optimized loss function and visual quality did not align. Consequently, our GANs exhibit improved results as a consequence of this experimental bias.

Due to challenges in training a Deep Convolutional GAN (DCGAN [26]), we exclusively present results for our implementation of WGAN, which has proven effective. The reported score in the paper was slightly inconsistent, representing a coefficient of 0 for the adversarial loss, making it a normal CNN. The most favorable outcomes were achieved with a VGG-like architecture. It is important to note that we do not argue that this is the optimal architecture, as our exploration of alternative options has been limited.

Density Methods:

Our conditional probability function in the Pixel CNN uses several convolutional layers with dropout. We opted not to include any pooling layers, as their addition did not appear to enhance the quality of our reconstructed images. Given that the parameters of the CNN are shared across all pixels, training the Pixel CNN essentially involves a classification task using a 24-pixel section; $5 \times 5 \times -1$, as explained earlier, as input and an integer within the range $[0, 256]$ as the target. For optimization, we utilized the Adam Optimizer with cross-entropy loss, and the learning rate was cross-validated. The dropout rate was set at 50%.

Our Flattened Row LSTM model achieved a quadratic loss of 7.63 on the test set, employing the softmax cross-entropy loss. The hidden dimension 64 goes with cross-validation, although due to computational constraints, we could not test as many values as desired. With the incorporation of several LSTM networks, it is likely that we could have achieved a lower $L2$ loss.

Comparison:

The outcomes of the optimal run for each model type on the test set are depicted in table 2 below. Additionally, several outcomes from our top-performing run are illustrated in figure 11.

Table 2: Different models' L2 Loss values

Model	L2 Loss
Row Flattened LSTM CNN	7.63

Pixel CNN	6.98
Wasserstein GAN	4.26
CNN	7.49

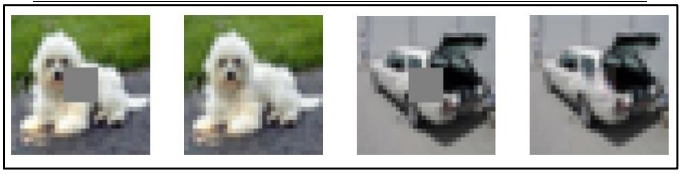


Figure 11: Results from Out-of-sample example

Conclusion:

Our unique CNN-based image in-painter with notable efficiency recognizes the significance of adversarial loss, as we incorporated a GAN into our model. Employing various tricks, we extended the approach from [13] to incorporate WGANs and utilized well-established architectures like VGG, Inception, and ResNet. Additionally, our overlap trick enhances border smoothing and also aids the training of discriminators. Our exploration not only delved into density-based methods, implementing Pixel CNNs based on [18], and introducing our model, Flattened Row LSTM. Through qualitative and quantitative comparisons, we looked to comprehend the limitations of these models. Overall, we are highly content with the outcomes achieved by our proposed architectures.

Future improvements involve adapting our models for larger images, offering exciting possibilities for handling more complex scenes with larger objects. Despite potential performance challenges, scaling up would open directions for addressing intricate scenarios. Additionally, refining our Flattened Row LSTM model to enhance symmetry and reinforce its generative capabilities stands as another goal for future improvement.

References:

[1] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/S11263-015-0816-Y/FIGURES/16.

[2] T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using Markov random field modeling," IEEE Trans. Image Process., vol. 24, no. 1, pp. 444–456, Jan. 2015, doi: 10.1109/TIP.2014.2372479.

[3] K. H. Jin and J. C. Ye, "Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting," IEEE Trans. Image Process., vol. 24, no. 11, pp. 3498–3511, Nov. 2015, doi: 10.1109/TIP.2015.2446943.

[4] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-Net: Image Inpainting via Deep Feature Rearrangement," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11218 LNCS, pp. 3–19, Jan. 2018, doi: 10.1007/978-3-030-01264-9_1.

[5] C. S. Weerasekera, T. Dharmasiri, R. Garg, T. Drummond, and I. Reid, "Just-in-Time Reconstruction: Inpainting Sparse Maps using Single View Depth Predictors as Priors," Proc. - IEEE Int. Conf. Robot. Autom., pp. 4977–4984, May 2018, doi: 10.1109/ICRA.2018.8460549.

[6] J. Zhao, Z. Chen, L. Zhang, and X. Jin, "Unsupervised Learnable Sinogram Inpainting Network (SIN) for Limited Angle CT reconstruction," Nov. 2018, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1811.03911v1

[7] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009, [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[8] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 Regularization for Learning Kernels," Proc. 25th Conf. Uncertain. Artif. Intell. UAI 2009, pp. 109–116, May 2012, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1205.2653v1

[9] S. Gu, J. Bao, D. Chen, and F. Wen, "GIQA: Generated Image Quality Assessment," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12356 LNCS, pp. 369–385, 2020, doi: 10.1007/978-3-030-58621-8_22.

[10] Y. Wang, Y. C. Chen, X. Tao, and J. Jia, "VCNet: A Robust Approach to Blind Image Inpainting," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12370 LNCS, pp. 752–768, 2020, doi: 10.1007/978-3-030-58595-2_45.

[11] Y. Liu, J. Pan, and Z. Su, "Deep Blind Image Inpainting," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11935 LNCS, pp. 128–141, 2019, doi: 10.1007/978-3-030-36189-1_11.

[12] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 1067–1074, 2013, doi: 10.1109/CVPR.2013.142.

[13] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 2536–2544, Dec. 2016, doi: 10.1109/CVPR.2016.278.

[14] I. J. Goodfellow et al., "Generative Adversarial Nets," Adv. Neural Inf. Process. Syst., vol. 27, 2014, Accessed: Oct. 02, 2023. [Online]. Available: http://www.github.com/goodfeli/adversarial

[15] B. Xu, N. Wang, H. Kong, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolution Network", Accessed: May 06, 2024. [Online]. Available: https://github.com/

[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., Sep. 2014, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1409.1556v6

[17] C. Szegedy et al., "Going deeper with convolutions," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June-2015, pp. 1–9, Oct. 2015, doi: 10.1109/CVPR.2015.7298594.

[18] A. Van Den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional Image Generation with PixelCNN Decoders," Adv. Neural Inf. Process. Syst., pp. 4797–4805, Jun. 2016, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1606.05328v2

[19] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 4076–4084, Nov. 2017, doi: 10.1109/CVPR.2017.434.

[20] "Wasserstein generative adversarial networks | Proceedings of the 34th International Conference on Machine Learning - Volume 70." Accessed: May 06, 2024. [Online]. Available: https://dl.acm.org/doi/10.5555/3305381.3305404

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.

[22] A. Birhane and V. U. Prabhu, "Large image datasets: A pyrrhic win for computer vision?," Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021, pp. 1536–1546, Jun. 2020, doi: 10.1109/WACV48630.2021.00158.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," Adv. Neural Inf. Process. Syst., pp. 2234–

2242, Jun. 2016, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/1606.03498v1

[24] "(PDF) The Earth Mover's Distance as a Metric for Image Retrieval." Accessed: May 06, 2024. [Online]. Available: https://www.researchgate.net/publication/220659330_The_Earth_Mover's_Distance_as_a_Metric_for_Image_Retrieval

[25] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks." PMLR, pp. 1747–1756, Jun. 11, 2016. Accessed: May 06, 2024. [Online]. Available: https://proceedings.mlr.press/v48/oord16.html

[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," Int. Conf. Learn. Represent., 2015.

OPEN ACCESS

International Journal of Innovations in Science & Technology

Deep Learning-based Skin Lesion Segmentation and Classification

Ayesha Rani[1], Haseeb Ullah[1], Hafiza Atika Shahab[1], Amaad Khalil[1], M. Abeer Irfan[1], Yaser Ali Shah[2],

[1]Computer System Engineering (University of Engineering and Technology Peshawar)

[2]Department of Compute Science (COMSATS University Islamabad, Attock Campus)

*__Correspondence__: ayesharani6788@gmail.com, haseebullahbj@gmail.com, atikashahab4455@gmail.com, amaadkhalil@uetpeshawar.edu.pk, abeer.irfan@uetpeshawar.edu.pk, Yaser@cuiatk.edu.pk

Citation | Rani. A, Ullah. H, Shahab. H. A, Khalil. A, Irfan. M. A, Shah. Y. A, "Deep Learning-based Skin Lesion Segmentation and Classification", IJIST, Special Issue pp 180-188, May 2024

Received | May 04, 2024, **Revised |** May 16, 2024, **Accepted |** May 20, 2024, **Published |** May 26, 2024.

By using deep learning to automate skin lesion segmentation, this work aims to improve the classification of melanoma. By properly segmenting lesions and utilizing the U-Net algorithm's preprocessing capabilities, our research aims to improve the accuracy of skin cancer diagnosis. During preprocessing, raw dermoscopic pictures from the HAM10000 dataset are enhanced and normalized early. Next, the U-Net model is used to accurately segment lesions. Advanced deep learning approaches are applied after segmentation segmented images are subjected to classification, such as Convolutional Neural Networks (CNN) and Vision Transformer (VIT) models. The VIT model demonstrated a high training accuracy of 0.94, indicating its effectiveness in learning from the training data. However, its validation and testing accuracies were at 0.73. The CNN model showed a training accuracy of 0.95, implying its ability to learn the training data effectively. However, its validation and testing accuracies were at 0.73. This all-encompassing method not only improves dermatological image analysis's dependability and effectiveness, but it also shows promise for enhancing clinical outcomes in the diagnosis and management of different forms of skin cancer. Our work is a significant step toward the creation of more reliable techniques in this important area, opening the door for improvements in patient care and healthcare diagnostics.

Keywords: Melanoma, Skin cancer, lesion segmentation, deep learning, lesion classification, image processing, vision transformer.

Introduction:

Skin cancer originates in skin cells due to the uncontrollable growth of skin cells. Skin cancer is primarily caused by UV radiation damage to the cell's DNA, which is passed on by exposure to sunlight or artificial UV radiation sources. Melanoma, squamous cell carcinoma (SCC), and basal cell carcinoma (BCC) are the most common forms of skin cancer. According to [1], each year the number of cases of melanoma identified, increased by 53% between 2008 and 2018. If the skin cancer is detected later, the survival rate is less than 14%. Over the next decade, there will likely be an increase in the death rate. On the other hand, if skin cancer is diagnosed in the early stages, the survival rate will increase to 97%.

The increasing global prevalence of skin cancer highlights the urgent need for efficient techniques for both detection and categorization. With the advancement of technology, computer-aided diagnostic systems started to be used in the identification of skin cancer. Inspired by the critical need for detecting skin cancer in an early stage, scientists began to employ machine learning methods in unconventional ways [2]. Deep learning, a subset of AI (artificial intelligence), has been integrated into skin cancer detection in recent years, raising the bar for accuracy and efficiency in skin cancer diagnosis through automated, data-driven insights [3].

For the diagnosis of skin cancer segmentation is the most critical step for extracting features and isolating the skin lesion. The conventional method for classifying skin lesion images involves segmenting and pre-processing the image. After extracting characteristics from the region of interest, the lesion is classified using different classifiers [4]. The segmentation approach reduces the image processing steps. Seeja et al. [5] suggest that deep learning can also improve diagnostic efficiency by simplifying the process of interpreting images by using a model to extract representative features from lesions by combining the input image with a segmentation mask.

In this paper, we present a novel approach for the segmentation of skin cancer utilizing the deep convolutional neural network based on the U-Net algorithm in preprocessing. This work focuses on the segmentation and classification of different seven classes of skin cancer by employing the HAM10000 Dataset for training and evaluation. The experimental results showcase the effectiveness of our proposed method achieving the notable segmentation accuracy of seven classes of skin cancer. In the pre-processing step, we utilized the U-Net architecture for segmentation. For classification, we employed VIT and CNN architectures.

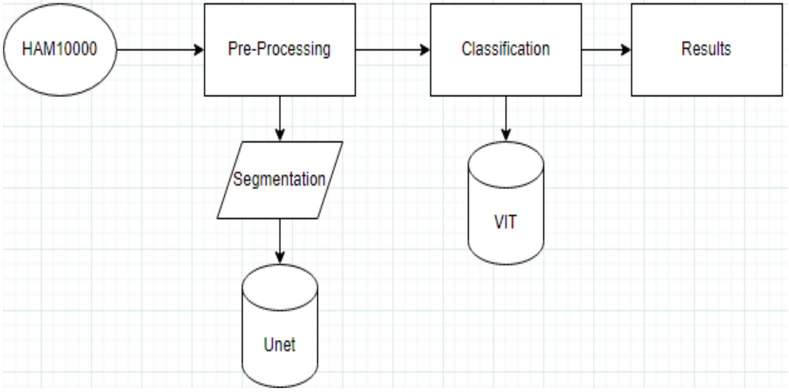


Figure 1: Flow chart of proposed methodology

Related Work:

Rafael Luz Ara´ujo et.al [6] proposed a segmentation method for melanoma skin lesions using modified U-net along with post-processing techniques. The research was conducted on two datasets, PH2 and DermIS involving acquisition and segmentation with a highly effective U-net network, and for further improvement, they performed post-processing techniques that disconnected extra regions, filled holes, and removed loose regions. In the PH2 dataset, they were able to acquire a dice coefficient of 0.933, and in the DermIS dataset, 0.872.

For the fast and accurate detection and segmentation of melanoma skin lesions, M. Taghizadeh et al. [7] proposed a method using SegNet and Yolov3 Based on Deep Transfer Learning. He suggests a two-phase procedure for melanoma detection. F-YoloV3 is applied for melanoma localization and F-SegNet for segmentation. Bisla et al. [8] suggested a deep learning model for both classification and segmentation for the diagnosis of skin lesions. The segmentation technique was to mask away portions of the image that weren't needed. He utilized the U-net architecture for segmentation. To improve the accuracy of the automated diagnosis of melanoma through deep learning image segmentation technique, Aleksandra Dzieniszeewska et al [4] combined the segmentation mask and skin lesion images using Gaussian blur. They employed the deeplabV3 and U-net segmentation networks for the segmentation process. On the combined ISC dataset, they obtained an accuracy of 84.85%. The classification performance of melanoma was enhanced by Seeja R D, and Suresh A [5] using deep learning-based automatic skin lesion segmentation. The process unfolds in three stages: Segmentation using U-Net, Feature Extraction (color, texture, shape) using HOG, LBP, Edge Histogram, and Gabor methods. , and Classification using SVM, Random Forest, K-NN and naïve Bayes. The SVM classifier yields the best result on the ISBI 2016 dataset based on F1-score and accuracy. For image segmentation, they achieved a Dice co-efficiency value of 77.59% and the SVM classifier produced 85.19 % accuracy. It is observed that classification with segmentation achieves much better accuracy, sensitivity, and specificity of the model compared to classification unsegmented images.

Mehwish Zafar et al. [9] proposed a segmentation model in which features are extracted through a pre-trained MobilebetV2 model. This model acts as a base of Deeplabv3+ for boundary extraction. Using the ISIC 2016, 2017, 2018, and PH2 datasets, the suggested segmentation strategy is assessed based on Mean Accuracy, Global Accuracy, BF Score, Weighted IoU, and Mean IoU. These metrics yield global accuracy values of 0.97481, 0.97297, 0.98642, and 0.95914, respectively. Nojus Dimša et al [10] suggested the automatic segmentation of skin lesions using deep learning, which explores the crucial field of melanoma diagnosis and highlights the important significance of early detection. MultiResUNet outperforms U-Net++ by a tiny amount (0.86%), all things considered, these U-Net variations show promise for improving the traditional U-Net model in skin lesion segmentation. Still, multi-class segmentation in skin lesions is a difficult field of study due to its complexity.

Researchers proposed a deep learning-based approach, comprising fuzzy k-means clustering (FKM) and region-based convolutional neural network (RCNN), to classify skin melanoma at an early phase [11]. The PH2, ISBI-2016, and ISIC-2017 datasets were utilized to evaluate the efficacy of the methodology that was offered. It beat existing state-of-the-art approaches, according to the data, with an average accuracy of 95.40%, 93.1%, and 95.6%. Mohammad Ali Kadampur et al [1] proposed a skin cancer detection method using Deep Learning Studio (DLS). Their approach involved data preparation, model construction, tuning, and deployment as a REST API. The DLS model achieved an exceptional AUC of 99.7% in skin cancer detection, demonstrating its effectiveness and ease of use. Ameri A [12] presented a robust deep-convolutional neural network utilizing AlexNet to classify skin lesions as benign or malignant. This model shows significant potential for assisting dermatologists in skin cancer detection with a classification accuracy of 84%, sensitivity of 81%, and specificity of 88%.

Using the Vision Transformer architecture, [13] research presents a unique method for classifying skin cancer that achieves an impressive 96.15% accuracy on the HAM10000 dataset. The work outperforms conventional deep learning techniques by utilizing pre-trained models like ViT patch-32, which has promising potential for dermatological diagnostics. The model's performance is further improved by using the Segment Anything Model for lesion segmentation, proving its usefulness in computer-aided skin cancer diagnosis. This study demonstrates the noteworthy advancements in deep learning applications for the interpretation of medical images, especially in the field of dermatology, which enhances patient outcomes by enabling prompt and precise diagnosis.

Material and Methods:

In our approach, we utilize deep learning techniques for automatic skin lesion segmentation using the U-Net algorithm as a preprocessing step to improve the classification of melanoma. Convolutional neural network architecture U-Net performs exceptionally well in biomedical image segmentation applications [4][5][14]. We employ the HAM10000 Skin Cancer dataset, which consists of various classifications of skin lesions, including melanocytic nevi, melanoma, basal cell carcinoma, actinic keratoses, vascular lesions, and dermatofibroma. The methodology involves the following steps:

Dataset:

We use the HAM10000 dataset, which consists of 10,015 dermoscopic pictures with extensive clinical metadata, in our study. This collection includes a variety of skin lesion representations, including benign nevi, malignant melanoma, and other dermatological diseases [14]. The HAM10000 dataset makes it easier to train and assess machine learning models for automated skin lesion recognition and segmentation because of its diverse population sources and clinical contexts.

Process:

Some separate subtasks that are suited for various input skin image types comprise the classification process:

- **Unaltered Lesion Classification:** In this subtask, skin lesions are categorized without any segmentation or preprocessing. It acts as a reference point for comparing segmented lesion classifications.
- **U-Net Segmented Lesion Classification:** Lesions are automatically separated from input images by utilizing the U-Net segmentation model. The U-Net architecture is trained to precisely identify skin lesions and is well-known for its efficacy in biomedical image segmentation.

Pre-Processing:

Our preprocessing stage is used for segmentation tasks by preparing the raw dermoscopic pictures. By means of the given service, we analyze the data to remove biases, the adjustment of the intensity levels, and also improve the image quality. The delineation procedure is performed in the next step to draw a border around a skin lesion. We will before segregation and classification do certain preprocessing tasks so as to make an accurate and consistent process.

Data Cleaning

To ensure data integrity and dependability for our research, we used data cleaning techniques during the preprocessing step to correct mistakes, inconsistencies, and missing values in the dataset.

Distribution of 7 Different Classes:

Subsequently, we Analyze the dataset's distribution of the seven distinct groups of skin lesions to identify any class imbalances.

Addressing Class Imbalance:

To balance the distribution, we decide to employ both up-sampling and down-sampling methods. A more equal distribution of the classes resulted in down-sampling, in which randomly selected samples of samples from the majority class were used to trim the size of the minority class. Moreover, we ensured that each class of the minority samples was replicated using such methods as up-sampling techniques to the level of attaining the balance of the dataset and a proper representation of each class.

Analysis of Spatial and Demographic Factors:

We performed an analysis to understand the distribution of skin lesions in several localized fields (e.g., arms, legs, torso) as well as the demographic features (gender, age) of skin lesioned individuals. The goal of the comprehensive study was to find geographic patterns or variances in the distribution of lesions and demographic trends or correlations, offering significant data for the classification of skin lesions.

Segmentation:

A crucial preprocessing step in our work is segmentation, which is used to precisely identify skin lesions from dermoscopic pictures. Each of the numerous subtasks that make up the segmentation approach contributes to the process's improvement and optimization.

Model Definition and Training:

Our approach relies heavily on the U-Net segmentation model to precisely identify skin lesions from dermoscopic images. The U-Net architecture, which is well-known for its effectiveness in biomedical image segmentation, is described inside a function that enables the customization of parameters such as the number of epochs. By iteratively modifying its parameters throughout the training phase and utilizing the rich contextual data that its architecture captures, the U-Net model gains the ability to accurately recognize skin lesions.

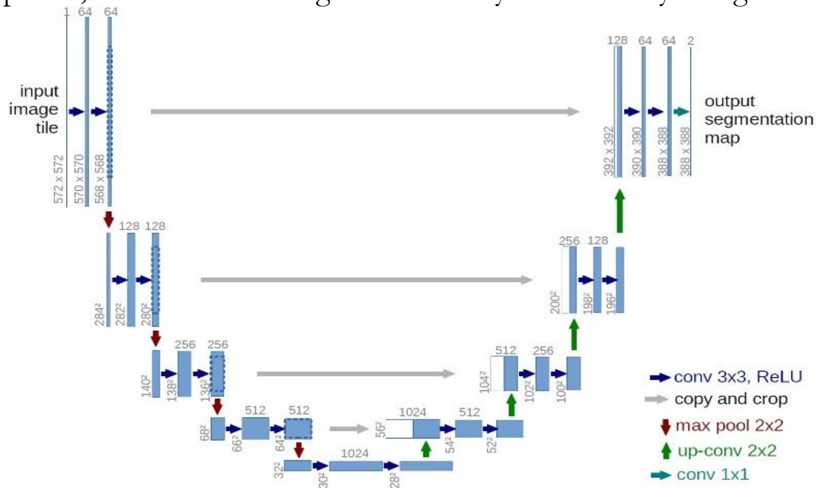


Figure 2: Illustration of U-Net Architecture [4]

Data Loading and Preparation

To provide consistency and structure in data handling, a function is defined to load the dataset in a sorted order. The dataset is divided into training and test sets to evaluate model performance impartially and prevent biases.

Evaluation Metrics:

As evaluation metrics, intersection over union (IoU), Jaccard Index, and dice coefficient are used to measure how accurate and similar segmented lesions are to ground truth annotations. The Jaccard Index, also known as the Jaccard similarity coefficient, is calculated as the ratio of the size of the intersection between the two sets to the size of their union. It is essentially the same as IoU but sometimes computed slightly differently. A statistic used to assess the similarity and diversity of sample sets is the Jaccard index, sometimes referred to as the Jaccard similarity coefficient and Intersection over Union. Like accuracy, the Dice score penalizes for false positives that the algorithm detects in addition to counting how many positives you find.

$$\mathrm{Dice} = 2 * \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} + (\mathrm{tp} + \mathrm{fn})$$

Visualization and Post-Processing:

To evaluate the implemented model for skin lesion segmentation, predictions are made on unknown data that have not been trained upon. We apply extra postprocessing approaches, for example, to improve lesion boundary feature visibility and then to adjust segmentation mask projections afterward the segmentation process. The proposed method is also suitable for smoothing segmentation masks and increasing visual interpretation accuracy with imprecise lesion boundaries. Thus, these after-processing procedures make the skin cancer detection method based upon the accurate segmentation results both reliable and aesthetically appealing.

Application of Masks:

To precisely define skin lesions for subsequent classification tasks, we employed the segmentation masks produced by the U-Net model in the last segmentation stage. Precise lesion borders were provided by the segmentation masks, which made feature extraction and classification easier. the application of segmentation masks generated by the U-Net model for accurate delineation of skin lesions is illustrated in Figure 3.

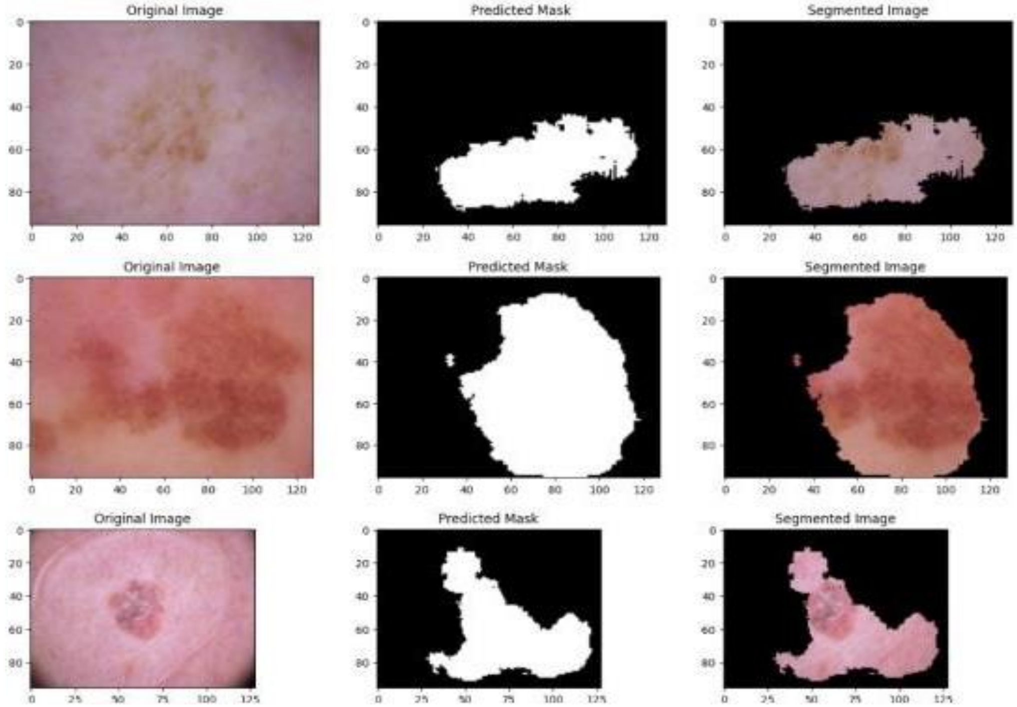


Figure 3: Predicted Mask and Segmented Images

Classification Using VIT:

Several separate segments were produced when we trained a model to segment VIT images. The VIT model was then trained to distinguish between melanoma and melanocytic nevi using these segments. Accuracy measurements displayed in tables and figures helped to clarify the findings. The segmentation-based classification strategy is shown to be reliable and successful by this thorough analysis. Significant progress toward improving our understanding and identification of skin lesions has been accomplished through this research, likely leading to improvements in the results of dermatological healthcare.

To train our classification model, segmented images were sent to a Vision Transformer (VIT) model. This method took advantage of VIT models' ability to efficiently handle image segmentation tasks. Our model was trained with the segmented representations to discriminate between two classes: melanoma and melanocytic nevi. This approach demonstrates a possible path toward increasing binary classification accuracy in medical image analysis by leveraging the advantages of both transformer-based models and segmentation. From this model, the training accuracy is 0.73, and the validation accuracy is 0.73.

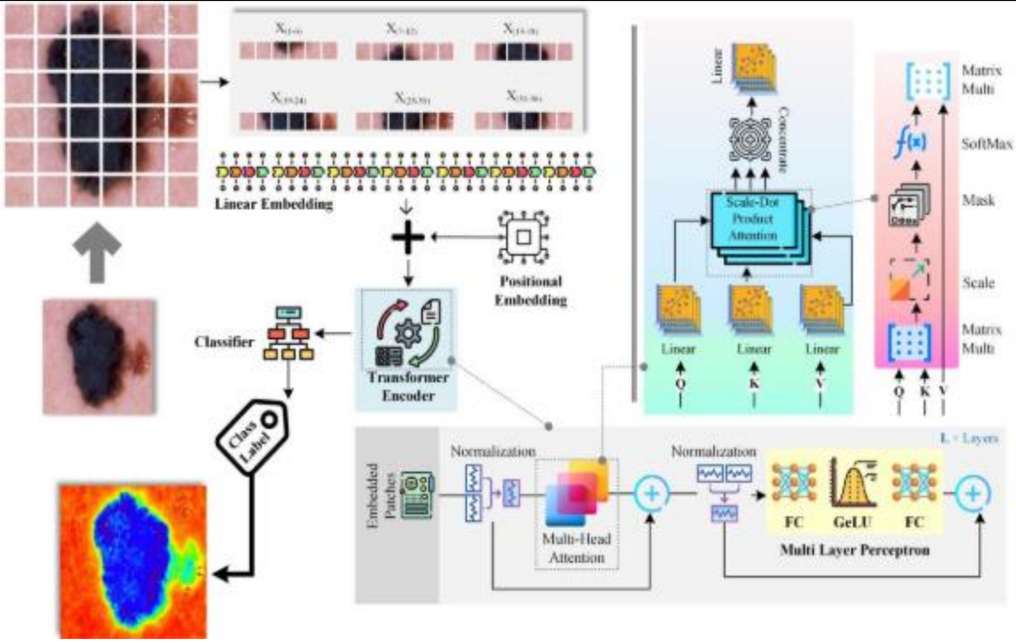


Figure 4: (Vision Transformer-based Skin Cancer Classification Model) [13]

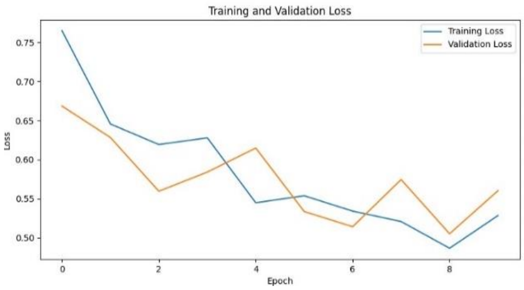


Figure 5: (Training & validation loss)

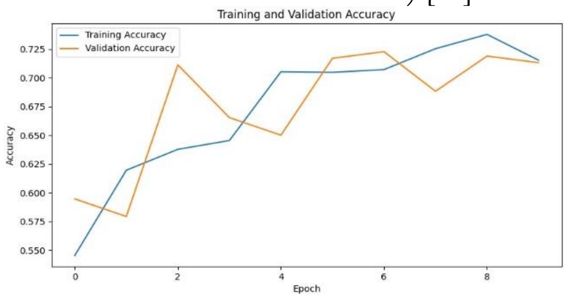


Figure 6: (Training & validation Accuracy)

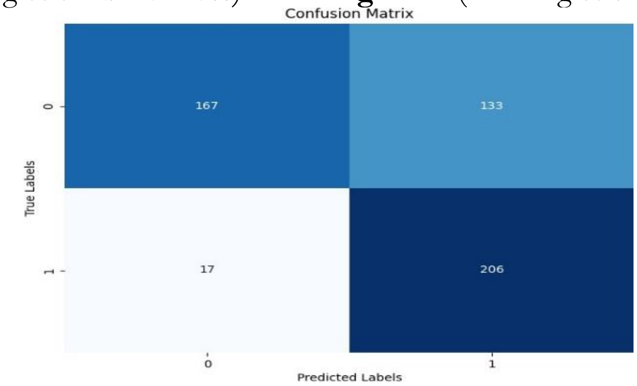


Figure 7: Confusion Matrix

The confusion matrix above Thirteen of the actual class one 167 images were wrongly classified as class 0 133 were accurately predicted as class 1 and 206 images that were truly class 0 were accurately classified, whereas 17 images were mistakenly classed as class 1.

Classification Using CNN:

Because of dataset imbalances, we trained a CNN model for a 7-class classification challenge using the segmented images. This all-inclusive strategy sought to address the data's unpredictability more successfully. A comprehensive table and figure below give the results of a thorough analysis of the resulting categories, together with pertinent metrics. By providing insights into how CNN models can handle unbalanced datasets and improve classification accuracy across several classes, this approach makes a substantial contribution to the area of medical image analysis.

It uses pre-trained weights from ImageNet to initialize Mobile Net, leaving out the fully connected layers. It reduces geographic dimensions by adding a layer of global average pooling. We add a completely connected layer with ReLU activation and 256 units at the end. It includes SoftMax activation and an output layer with 7 units (for a classification problem with 7 classes). We build a new model with the custom classification layers and the input from the Mobile Net. Essentially, it adds unique layers for classification to modify the robust Mobile Net architecture for a particular picture classification application.

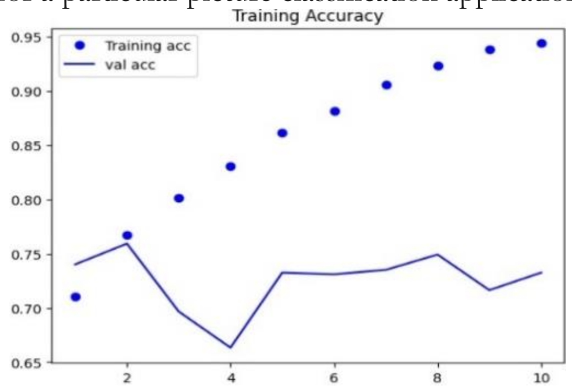


Figure 8: Training and Validation Accuracy

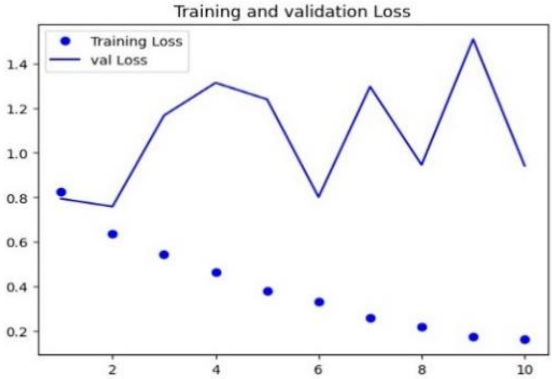


Figure 9: Training and Validation Loss

Result and Discussion:

The VIT model demonstrated a high training accuracy of 0.94, indicating its effectiveness in learning from the training data. However, its validation and testing accuracies were at 0.73. The VIT model still achieved a notable testing accuracy, indicating its capability to generalize well to unseen data.

Table 1: Classification Using VIT Model

Training Accuracy	Validation Accuracy	Testing Accuracy
0.73	0.71	0.71

On the other hand, the CNN model showed a comparable training accuracy of 0.95 to the VIT model, implying its ability to learn the training data effectively. However, its validation and testing accuracies were lower at 0.73 respectively.

Table 2: Classification Using CNN Model

Training Accuracy	Validation Accuracy	Testing Accuracy
0.94	0.73	0.73

Conclusion:

In conclusion, this work presents a method for skin lesion segmentation and classification using state-of-the-art deep learning techniques. Using Convolutional Neural Networks (CNN) and Vision Transformer (VIT) models for classification and the U-Net technique for segmentation, we have shown encouraging results in correctly classifying different kinds of skin lesions. The metrics acquired for segmentation accuracy verify the efficacy of our method in medical picture analysis. Our approach has a lot of potential to boost clinical outcomes by increasing the precision and dependability of skin cancer diagnosis. better work will concentrate on enlarging the classification task to support a wider variety of lesion classes, and continuous efforts will be made to better hone and optimize our methodology. This work represents a major advancement in the use of deep learning in dermatological analysis and emphasizes the need for ongoing research and development in this area.

Acknowledgement: Acknowledgments are considered necessary.

Author's Contribution: The corresponding author should explain the contribution of each co-author completely.

Conflict of Interest: Authors are advised to explain that there exists no conflict of interest for publishing this manuscript in IJIST.

Project Details: If this research was conducted as a result of a project, please give details like project number, project cost completion date, etc.

References:

[1] M. A. Kadampur and S. Al Riyaee, "Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images," Informatics Med. Unlocked, vol. 18, p. 100282, Jan. 2020, doi: 10.1016/J.IMU.2019.100282.

[2] M. A. Kassem, K. M. Hosny, R. Damaševičius, and M. M. Eltoukhy, "Machine Learning and Deep Learning Methods for Skin Lesion Classification and Diagnosis: A Systematic Review," Diagnostics 2021, Vol. 11, Page 1390, vol. 11, no. 8, p. 1390, Jul. 2021, doi: 10.3390/DIAGNOSTICS11081390.

[3] R. A. Mehr and A. Ameri, "Skin Cancer Detection Based on Deep Learning," J. Biomed. Phys. Eng., vol. 12, no. 6, p. 559, Dec. 2022, doi: 10.31661/JBPE.V0I0.2207-1517.

[4] A. Dzieniszewska, P. Garbat, and R. Piramidowicz, "Skin Lesion Classification Based on Segmented Image," 2023 12th Int. Conf. Image Process. Theory, Tools Appl. IPTA 2023, 2023, doi: 10.1109/IPTA59101.2023.10320004.

[5] R. D. Seeja and A. Suresh, "Deep Learning Based Skin Lesion Segmentation and Classification of Melanoma Using Support Vector Machine (SVM)," Asian Pac. J. Cancer Prev., vol. 20, no. 5, p. 1555, 2019, doi: 10.31557/APJCP.2019.20.5.1555.

[6] R. L. Araujo, R. D. A. L. Rabelo, J. J. P. C. Rodrigues, and R. R. V. E. Silva, "Automatic segmentation of melanoma skin cancer using deep learning," 2020 IEEE Int. Conf. E-Health Networking, Appl. Serv. Heal. 2020, Mar. 2021, doi: 10.1109/HEALTHCOM49281.2021.9398926.

[7] M. Taghizadeh and K. Mohammadi, "The Fast and Accurate Approach to Detection and Segmentation of Melanoma Skin Cancer using Fine-tuned Yolov3 and SegNet Based on Deep Transfer Learning," Oct. 2022, Accessed: May 06, 2024. [Online]. Available: https://arxiv.org/abs/2210.05167v2

[8] L. Bi, J. Kim, E. Ahn, A. Kumar, M. Fulham, and D. Feng, "Dermoscopic Image Segmentation via Multistage Fully Convolutional Networks," IEEE Trans. Biomed. Eng., vol. 64, no. 9, pp. 2065–2074, Sep. 2017, doi: 10.1109/TBME.2017.2712771.

[9] M. Zafar, J. Amin, M. Sharif, M. A. Anjum, G. A. Mallah, and S. Kadry, "DeepLabv3+-Based Segmentation and Best Features Selection Using Slime Mould Algorithm for Multi-Class Skin Lesion Classification," Math. 2023, Vol. 11, Page 364, vol. 11, no. 2, p. 364, Jan. 2023, doi: 10.3390/MATH11020364.

[10] N. Dimša and A. Paulauskaitė-Tarasevičienė, "Melanoma multi class segmentation using different U-Net type architectures," 2021.

[11] M. Nawaz et al., "Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering," Microsc. Res. Tech., vol. 85, no. 1, pp. 339–351, Jan. 2022, doi: 10.1002/JEMT.23908.

[12] A. Ameri, "A Deep Learning Approach to Skin Cancer Detection in Dermoscopy Images," J. Biomed. Phys. Eng., vol. 10, no. 6, p. 801, 2020, doi: 10.31661/JBPE.V0I0.2004-1107.

[13] G. M. S. Himel, M. M. Islam, K. A. Al-Aff, S. I. Karim, and M. K. U. Sikder, "Skin Cancer Segmentation and Classification Using Vision Transformer for Automatic Analysis in Dermatoscopy-Based Noninvasive Digital System," Int. J. Biomed. Imaging, vol. 2024, 2024, doi: 10.1155/2024/3022192.

[14] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," Sci. Data 2018 51, vol. 5, no. 1, pp. 1–9, Aug. 2018, doi: 10.1038/sdata.2018.161.

Investigating Threats to ICS and SCADA Systems Via Honeypot Data Analysis and SIEM

Tameem ud Din[1], Usman Zia[1], Mahnoor[1], Laiq Hasan[1], Syed M. Ali Uddin Hafee[2],
[1]University of Engineering and Technology Peshawar Pakistan,
[2]NED University of Engineering and Technology Karachi, Pakistan,
***Correspondence**: Tameem Ud Din - 20PWCSE1866@uetpeshawar.edu.pk

Citation | Din. T, Zia. U, Mahnoor, Hasan. L, Hafee. S. M. A. U, Investigating Threats to ICS and SCADA Systems Via Honeypot Data Analysis and SIEM, IJIST, Special Issue. pp 189-197, April 2024

Received | May 04, 2024, **Revised |** May 12, 2024, **Accepted |** May 18, 2024, **Published |** May 27, 2024.

Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems (ICS) are crucial for managing essential infrastructure, but their exposure to the internet has made them vulnerable to cyber threats, which can lead to significant consequences. This study presents an innovative approach to investigating cyber threats to SCADA and ICS systems by combining open-source honeypot deployment, log analysis, and integration with open-source SIEM solutions to enhance threat detection capabilities and incident response. A Conpot honeypot was deployed in a containerized environment on a cloud platform and exposed to the internet to collect real-world threat data, which was then analyzed by the Wazuh SIEM solution and integrated with TheHive for security orchestration and automated response. The analysis of the honeypot logs and SIEM alerts revealed various types of attacks, including brute force login attempts, reconnaissance and vulnerability scanning, and unauthorized access attempts, originating from multiple countries and targeting different industrial protocols. The integration with TheHive enabled the creation of playbooks for automating response actions, such as blocking malicious IP addresses or isolating infected systems. The study demonstrates the effectiveness of this combined approach using open-source tools in protecting critical infrastructure and enhancing cybersecurity posture for SCADA and ICS systems.

Keywords: ICS; SCADA; OT; Honeypot; Critical Infrastructure Protection.

Introduction:

Supervisory control and data acquisition (SCADA) systems are considered a type of industrial control system that allows users to monitor, acquire data from, and control industrial processes locally or remotely through sensors and actuators [1]. Power plants and water treatment facilities are examples of traditional industrial systems that were designed to operate in highly controlled and separated settings. However, the recent exposure of Industrial Control Systems (ICS) to the Internet has made access and technological adaptation easier, which has led to the exploitation of security holes by attackers to launch attacks against ICSs [2]. These attacks can significantly impact the economics and national security of countries. To identify possible threats and comprehend the terrain of these assaults, ICS honeypots are deployed [3]. Honeypots are an interesting security concept; instead of keeping attackers out, you want to invite them in [4]. They are typically divided into three categories: low-, medium-, and high-interaction. The key distinctions between these types are their capacities for data collection, maintenance, and deployment [5]. The higher the level of interaction, the more data the honeypot can capture. High interaction ones are most similar to real systems and can collect the most data. In industrial environments where attacks occur, such as ICS/SCADA, honeypots need to be very similar to real systems to reduce detection risk [6]. This paper investigates the threats to SCADA and ICS systems using open-source honeypot deployment, log analysis, and integration with open source SIEM solutions to enhance threat detection capabilities and incident response. Furthermore, it studies the viability of putting open-source honeypots in the cloud to detect attacks on these systems. In addition, the study investigates how to integrate these honeypots with an open-source SIEM solution. This combination strategy tries to improve threat detection by connecting honeypot data with other security logs. Finally, the research extends beyond simply recognizing attackers. It digs into a deep examination of the collected traffic to determine attack kinds, sources, and other useful information for creating defensive tactics. The core contributions of this paper are: 1) Evaluation of Open-Source Honeypot Deployment on Cloud 2) Integration of Honeypot with Open-Source SIEM for Threat Intelligence 3) In-Depth Analysis of Attack Patterns.

Background And Related Work:

This section gives an overview of honeypots, and SIEM Solutions, summarizes relevant efforts, and honeypot deployments.

Honeypots:

Honeypots are software applications designed to mimic real systems, deceiving attackers while serving as a decoy. Honeypots are classified into two types: production honeypots, which are installed within a network to deceive insiders, and research honeypots, which are deployed on the internet to attract attackers from the outside. Furthermore, honeypots can be classed according to how closely they engage with attackers. Low-interaction honeypots simulate basic services but do not completely function, whereas high-interaction honeypots provide a full operating system with real services [7]. To set up a honeypot, a specific honeypot image is deployed on the machine. Once configured, a port is assigned to the honeypot. Any attempt to access this port redirects the attacker to the honeypot, which appears to be a genuine system. Through this redirection, the attacker’s machine ID, IP address, and other critical parameters are logged in the honeypot’s log file, using these logs we can move to further analysis. We used conpost, an open-source, general-purpose ICS honeypot.

Conpot:

The ICS honeypot we have used is called Conpot. Conpot is a SCADA honeypot that serves as a valuable tool for emulating SCADA systems and detecting potential threats within SCADA device networks. Developed by The Honeynet Project, Conpot is designed to be easy to implement and provides simulation capabilities for protocols like HTTP, Modbus, and SNMP, as well as integration with programmable logic controllers (PLC). It features a logging system that monitors and records any unauthorized changes made by intruders, offering detailed event logs with millisecond accuracy. By mimicking the behavior of real SCADA systems, Conpot utilizes a logging system to monitor any changes that are made by intruders. The honeypot logs events of HTTP, SNMP, and Modbus services with millisecond accuracy and offers basic tracking information such as source address, request type, and resource requested in the case of HTTP [8].

Security Information and Event Management:

Security Information and Event Management (SIEM) is a vital security software/platform that analyzes security events. It works by examining log files sent to it and following the paths we set. SIEM analyzes log files using established rules and generates alerts depending on the results. Custom rules can be written for specific types of logs to generate alerts when certain circumstances are satisfied. To successfully analyze logs, we must create a decoder for any log types that do not follow the standard format. SIEM does, however, include preconfigured decoders for JSON structured logs.

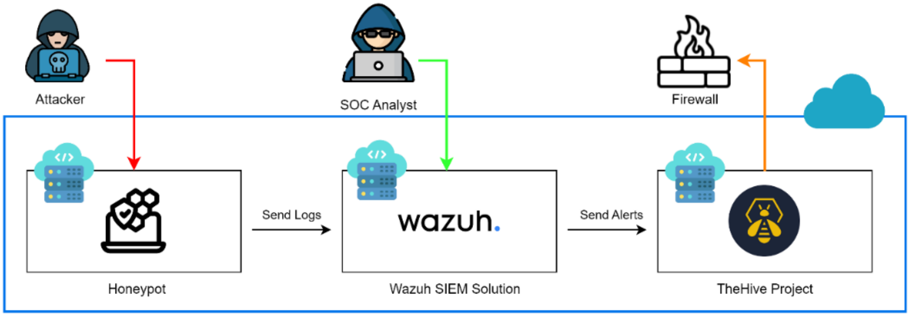


Figure 1: Holistic Approach to SCADA/ICS Threat Investigation with Honeypot, Wazuh, and the Hive.

Related Work:

The use of honeypots as a security precaution and research tool has been extensively studied, including industrial control systems (ICS) and supervisory control and data acquisition (SCADA). for protecting Operational Technology (OT) environments. The following literature review focuses on significant studies and research efforts on the subject of honeypots and their use to secure critical infrastructure. Nasution et al. concentrate on the low-interaction honeypot HONEYD's potential for enhancing security systems in their paper [9]. They go through how HONEYD is configured and deployed, as well as how it can identify and record assaults, giving security analysts important information. A thorough analysis of the application of honeypots and honeynets in Cyber-Physical Systems (CPS), Industrial IoT (IIoT), and the Internet of Things (IoT) is provided by Franco et al. [10]. Their research emphasizes the benefits and drawbacks of setting up honeypots in these settings and stresses the significance of acquiring information on new dangers. Jicha et al. [8] investigate the characteristics of SCADA honeypots, with a special emphasis on the Conpot variation. Their research provides a thorough examination of Conpot's capabilities, including its ability to mimic numerous industrial protocols and collect important information on prospective assaults on SCADA systems. Sharma and Kaul [11] conduct a detailed investigation on intrusion detection systems and honeypot-based proactive security measures in Vehicular Ad hoc Networks (VANETs) and VANET clouds. Their findings emphasize the necessity of honeypots for recognizing and mitigating risks in these dynamic and diverse contexts. Uitto et al. [12] describe the anti-honeypot strategies and procedures used by attackers to discover and avoid honeypots. Their research underlines the significance of establishing strong and resilient honeypot systems to fight such evasion strategies while maintaining effective threat monitoring and analysis. Cao et al. [13] describe DiPot, a distributed industrial honeypot system that monitors internet scanning and attack behaviors on industrial control systems. DiPot's strengths are its high-level modeling, broad data analysis capabilities, and accessible graphical frontend, which allow users to obtain insights into the present condition of ICS security. Lopez-Morales et al. [14] introduce HoneyPLC, a highly interactive, adaptable, and malware-collecting honeypot that supports a wide range of PLC models and manufacturers. Their trials show HoneyPLC's capacity to successfully disguise itself as a real device, enticing and misleading attackers while collecting vital data samples for further study. Radoglou-Grammatikis et al. [7] provide TRUSTY, a strategic custom honeypot deployment and analysis framework. It uses an adversarial game model between the attacker and the defense to optimize the number of honeypots deployed based on the attacker's behavior and the available computing resources.

Research Gap:

These studies show the growing importance of honeypots in critical infrastructure security, notably for SCADA and ICS systems. They provide useful insights on honeypot deployment, configuration, and analysis, as well as the problems and tactics involved in their effective use in detecting and mitigating cyber threats. While earlier research has examined the impact of honeypot deployment, there is still a gap in building a real-time threat analysis system. This system would use open-source and free tools to continually monitor honeypot data and extract threat intelligence to proactively mitigate threats. This is how the remainder of the paper is organized. Section 2 explains the approach and the experimental setting. The findings are examined and described in Section 3. Section 5 concludes the paper.

Material and Methods:

This study offers a comprehensive approach to investigating cyber hazards to SCADA and ICS systems. As shown in Fig. 1, it uses honeypot technology in conjunction with Wazuh, an open-source SIEM system, to analyze threats and attacks in real-time. For one week, the honeypot was deployed in a containerized environment on a cloud platform and exposed to the internet in order to collect logs. A persistent storage solution within the container ensured log retention for Wazuh's real-time analysis. Wazuh alerts were also linked to the Hive project, allowing for the building of playbooks. These playbooks managed the blocking of new malicious IP addresses detected by the honeypot using TheHive's SOAR platform, automating the response to possible threats.

The experiment was set up by installing and configuring the Conpot honeypot, emulating a SCADA environment with various industrial protocols such as Modbus, SNMP, and HTTP. The honeypot was then exposed to the internet to attract potential attackers and gather real-world threat data. In parallel, the Wazuh SIEM agent was installed on the same system hosting the Conpot honeypot. The Wazuh agent was configured to monitor and process the log files generated by the honeypot, enabling the analysis of security events and potential threats in the dashboard of Wazuh. Further, the Wazuh was integrated with TheHive for security orchestration and automation purposes. The details of each stage are discussed in the subsections below.

Conpot Deployment:

Selection of Honeypot: The Conpot honeypot was used for this study because it is open source and can efficiently simulate SCADA systems. Conpot supports a variety of industrial protocols, including Modbus, SNMP, and HTTP, making it ideal for replicating a realistic SCADA system. Installation and Configuration: The Conpot honeypot was packed into a Docker container and placed on a dedicated system, with all necessary dependencies such as Python, Twisted, Scrapy, and PyYAML. The honeypot was then set up to imitate the needed SCADA system characteristics, such as supported protocols and services. The IEC 60870 5-104 protocol was disabled, and the default template for the configuration was utilized. Exposure to the Internet: To attract real-world cyber-attacks and collect vital threat intelligence, a honeypot with a small footprint (Conpot) was deliberately exposed to the Internet. This included opening critical ports and making the honeypot accessible via external networks. We mapped the typical ports used by common SCADA protocols to the Docker container running Conpot, leaving its default settings for a realistic attack surface.

SIEM Solution Installation and Configuration:

SIEM Solution Selection: The Wazuh SIEM Solution was chosen for this study because it is open-source, scalable, and compatible with a variety of log formats, including JSON, the major format utilized by the Conpot honeypot. Installation and Configuration: The Wazuh agent was installed on the server that ran the Conpot honeypot, and the Wazuh SIEM was configured to monitor and process not only the Conpot log files but also monitor the systems logs of the server running the Conpot. Furthermore, the Wazuh indexer server and dashboard were installed on a separate server. Rule Definition: Custom rules were built within the Wazuh SIEM to help with the examination of the Conpot log files. These rules were designed to detect specific patterns, abnormalities, or occurrences of interest associated with SCADA system attacks and vulnerabilities.

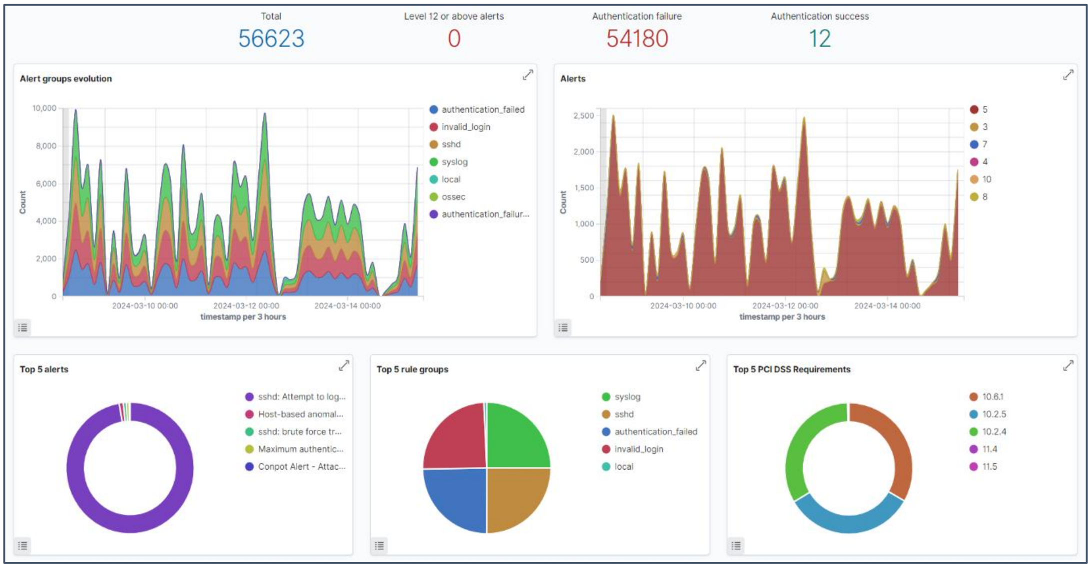


Figure 2: Wazuh SIEM Dashboard: Real-time Monitoring and Analysis of Security Alerts and Events.

Data Collection and Analysis:

Logs Monitoring: The Conpot honeypot logs were constantly monitored, and any interactions or attacks were recorded with millisecond delay, including source IP addresses, requested resources, and any attempt to make unauthorized changes by possible attackers. Logs Analysis through Wazuh Dashboard: The Wazuh SIEM dashboard was used to examine the log data collected from the Conpot honeypot, making use of specified rules and decoders. Figure 2 illustrates the Wazuh SIEM dashboard, which enables real-time monitoring and analysis of security alerts and events. This investigation sought to uncover potential threats, attack patterns, and vulnerabilities associated with SCADA systems.

Threat Intelligence Gathering: The analysis of the Conpot honeypot logs and SIEM alerts provided valuable threat intelligence, including the tactics, techniques, and procedures (TTPs) employed by malicious actors targeting SCADA systems. Additionally, the public IP addresses of potential threat actors were collected for further investigation or potential blocking.

Integration with the Hive:

Installation and Configuration:

The Hive integration involved setting up a dedicated server for its deployment. Within The Hive, a new organization was created specifically for managing security incidents. A dedicated user account was also created and assigned to this organization. To establish communication between the Hive and Wazuh, an API-based connection was configured. This two-way communication channel allows Wazuh to automatically send security alerts to the Hive for further analysis and response.

Writing Playbooks:

The Hive's playbook functionality was leveraged to automate response actions based on incoming alerts. Playbooks were written to orchestrate specific actions, such as blocking malicious IP addresses identified by the honeypot or SIEM system directly on the firewall. This automated response streamlines the security workflow, allowing for faster threat mitigation and minimizing potential damage to SCADA/ICS systems. When the honeypot or SIEM detects a potential threat, such as a hostile IP address attempting to exploit a vulnerability, the Hive playbooks can be automatically activated. These playbooks would then execute a predetermined set of steps, such as blocking the malicious IP address at the network layer or isolating infected systems. This quicker response helped to reduce the impact of cyberattacks and prevent potential harm to SCADA/ICS systems.

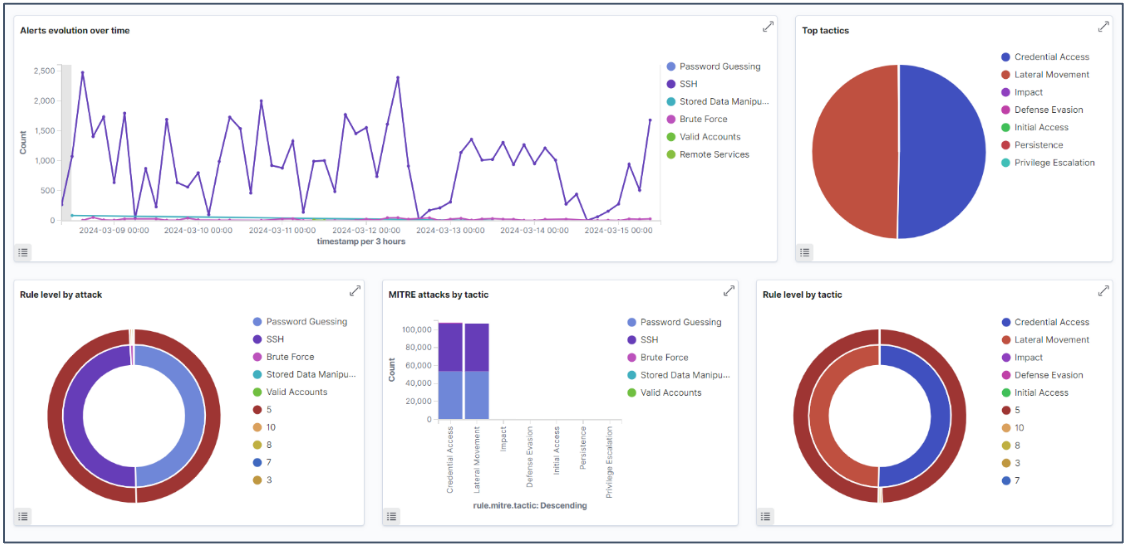


Figure 3: Wazuh SIEM Dashboard: Top Tactics and Techniques Observed in Attacks Based on the MITRE ATT&CK Framework

Result and Discussion:

Observed Attacks and Threats:

The analysis of the Conpot honeypot logs and Wazuh SIEM alerts provided valuable insights into the tactics, techniques, and procedures (TTPs) employed by malicious actors targeting SCADA systems. Figure 3 showcases the top tactics and techniques observed in attacks, categorized according to the MITRE ATT&CK framework. The following subsections provide a detailed overview of the findings.

Types of Attacks:

The study identified several types of attacks attempted against the simulated SCADA environment, including Brute Force Login Attempts: A significant number of brute force attacks (7,548) were observed, where attackers attempted to gain unauthorized access to the system by trying multiple combinations of non-existent usernames and passwords. Reconnaissance and Vulnerability Scanning: Numerous attempts were made to probe the honeypot for open ports, services, and potential vulnerabilities. These reconnaissance efforts are often precursors to more advanced attacks. Exceeding Authentication Limits: In some instances, attackers exceeded the maximum allowed authentication attempts, indicating their persistence in trying to gain unauthorized access.

Attacker's Region:

The analysis of the source IP addresses revealed that the attacks originated from various regions across the globe. Table 1 presents the distribution of attack sources by country or region.

As shown in Table 1,

Table 1: Distribution of Attack Sources by Country/Region.

Country/Region	Number of Unique IP Addresses
United States (US)	201
China (CN)	12
United Kingdom (GB)	10
Germany (DE)	10
Pakistan (PK)	8
Netherlands (NL)	8
Russia (RU)	8
Others	52

The majority of attacks originated from the United States, with 201 unique IP addresses involved. Other notable sources include China, the United Kingdom, Germany, Pakistan, the Netherlands, and Russia.

Protocols:

The honeypot was configured to emulate various industrial protocols commonly used in SCADA systems. Table 2 presents the distribution of attacks based on the targeted protocols and the corresponding port numbers. As shown in Table 2, the majority of attacks (1,168) targeted the HTTP protocol on port 8800, while 150 attacks aimed at the SNMP protocol on port 16100. It is important to note that the honeypot was intentionally exposed to the internet to attract potential attackers and collect real-world threat data. The observed attacks and their distribution provide valuable insights into the tactics, techniques, and procedures (TTPs) employed by malicious actors targeting SCADA and ICS systems.

Table 2. Distribution of Attack Sources by Protocol and Port Number.

Protocol	Port Number	Number of Attacks
HTTP	80	1168
SNMP	161	150

Discussion:

The study revealed a significantly higher number of attacks targeting the HTTP protocol on port 8800 compared to other protocols like SNMP. This observation can be attributed to several factors. Firstly, HTTP is a widely adopted protocol, and many attackers attempt to exploit web-based vulnerabilities or misconfigurations, making it an attractive target for reconnaissance and potential exploitation.

Another potential reason for the high HTTP traffic could be the detectability of the default Conpot web interface. While Conpot aims to emulate a realistic SCADA environment, the default configuration and web interface may have been recognized by attackers as a honeypot, prompting them to focus their efforts on the HTTP service. To enhance the authenticity of the simulated environment and attract a more diverse range of attack patterns, customizing the Conpot configuration to better mimic specific SCADA systems or industrial environments could be beneficial.

Furthermore, the study did not explicitly mention attacks targeting the SSH protocol, which is often used for remote administration. Leaving the SSH port open, although not directly related to SCADA protocols, could potentially attract unwanted brute-force attacks or unauthorized access attempts, which may not be relevant to the study's scope. It is recommended to avoid exposing unnecessary ports to the internet when deploying honeypots or simulating SCADA environments to maintain a focused and relevant attack surface.

Lastly, the relatively lower traffic observed for protocols like SNMP compared to HTTP could be due to the difficulty in accurately emulating the behavior and nuances of industrial protocols within a low-interaction honeypot environment with a default configuration template.

To address this challenge, future studies could explore incorporating more realistic industrial protocol implementations, potentially through the use of virtualized or emulated hardware components and integrating additional data sources or logs from real SCADA systems (with appropriate anonymization and security measures) to enrich the honeypot's behavior and log patterns.

Conclusion:

This study successfully investigated threats to SCADA/ICS systems by combining honeypots, SIEM, and security orchestration. A honeypot like Conpot acted as a decoy, collecting real-world attack data when exposed on a cloud platform. This data was then analyzed by Wazuh, a SIEM solution, to identify attack patterns and sources. Finally, Hive, a security orchestration platform, automated responses based on predefined protocols, such as blocking malicious IPs and isolating infected systems. This study demonstrates the effectiveness of this combined approach, and importantly, the value of open-source tools (Conpot, Wazuh, the Hive) in protecting critical infrastructure.

Future work could focus on enhancing the realism of the honeypot deployment, exploring the detection and analysis of more advanced attack techniques, investigating the scalability and performance of the proposed solution, integrating with external threat intelligence platforms, and fostering collaboration and information sharing among organizations and security communities.

Acknowledgement:

We would like to express our sincere gratitude to the University of Engineering and Technology, Peshawar, and NED University of Engineering and Technology, Karachi, for providing the necessary resources and support for this research project. We are thankful to our supervisors, Dr. Laiq Hasan and Syed M. Ali Uddin Hafee, for their invaluable guidance and mentorship throughout the project. Additionally, we acknowledge the open-source communities behind the tools and technologies used in this study, particularly Conpot, Wazuh, and TheHive, for their dedication to advancing cybersecurity research and development.

Author's Contribution:

Tameem Ud Din, Usman Zia, and Mahnoor were responsible for the conceptualization, implementation, and execution of the project. They conducted extensive research, deployed the honeypot and SIEM solution, performed data analysis, and integrated the automated response capabilities. Dr. Laiq Hasan provided overall supervision, guidance, and valuable insights throughout the project, ensuring its successful completion. Syed M. Ali Uddin Hafee contributed his expertise in cybersecurity and industrial control systems, providing critical feedback and recommendations to enhance the project's effectiveness.

Conflict of interest:

The authors declare that there is no conflict of interest in publishing this manuscript in the International Journal of Information Security and Threat Modeling (IJIST). The research was conducted solely for academic and scientific purposes, and the authors have no financial or other interests that could influence the objectivity or integrity of the work presented.

References:

[1] M. Mesbah, M. S. Elsayed, A. D. Jurcut, and M. Azer, "Analysis of ICS and SCADA Systems Attacks Using Honeypots," Futur. Internet 2023, Vol. 15, Page 241, vol. 15, no. 7, p. 241, Jul. 2023, doi: 10.3390/FI15070241.

[2] A. Nechibvute and H. D. Mafukidze, "Integration of SCADA and Industrial IoT: Opportunities and Challenges," IETE Tech. Rev., May 2024, doi: 10.1080/02564602.2023.2246426.

[3] A. Ara, "Security in Supervisory Control and Data Acquisition (SCADA) based Industrial Control Systems: Challenges and Solutions," IOP Conf. Ser. Earth Environ. Sci., vol. 1026, no. 1, p. 012030, May 2022, doi: 10.1088/1755-1315/1026/1/012030.

[4] "Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community: Honeynet Project: 9780201746136: Amazon.com: Books." Accessed: May 06, 2024. [Online]. Available: https://www.amazon.com/Know-Your-Enemy-Revealing-Community/dp/0201746131

[5] "World Wide ICS Honeypots: A Study into the Deployment of Conpot Honeypots." Accessed: May 06, 2024. [Online]. Available: https://www.researchgate.net/publication/358166067_World_Wide_ICS_Honeypots_A_Study_into_the_Deployment_of_Conpot_Honeypots

[6] S. Chamotra, J. S. Bhatia, R. Kamal, and A. K. Ramani, "Deployment of a low interaction honeypot in an organizational private network," Proc. 2011 Int. Conf. Emerg. Trends Networks Comput. Commun. ETNCC2011, pp. 130–135, 2011, doi: 10.1109/ETNCC.2011.5958501.

[7] P. Radoglou-Grammatikis et al., "TRUSTY: A solution for threat hunting using data analysis in critical infrastructures," Proc. 2021 IEEE Int. Conf. Cyber Secur. Resilience, CSR 2021, pp. 485–490, Jul. 2021, doi: 10.1109/CSR51186.2021.9527936.

[8] A. Jicha, M. Patton, and H. Chen, "SCADA honeypots: An in-depth analysis of Conpot," IEEE Int. Conf. Intell. Secur. Informatics Cybersecurity Big Data, ISI 2016, pp. 196–198, Nov. 2016, doi: 10.1109/ISI.2016.7745468.

[9] A. M. Nasution, M. Zarlis, and S. Suherman, "Analysis and Implementation of Honeyd as a Low-Interaction Honeypot in Enhancing Security Systems," Randwick Int. Soc. Sci. J., vol. 2, no. 1, pp. 124–135, Jan. 2021, doi: 10.47175/RISSJ.V2I1.209.

[10] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems," IEEE Commun. Surv. Tutorials, vol. 23, no. 4, pp. 2351–2383, 2021, doi: 10.1109/COMST.2021.3106669.

[11] S. Sharma and A. Kaul, "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud," Veh. Commun., vol. 12, pp. 138–164, Apr. 2018, doi: 10.1016/J.VEHCOM.2018.04.005.

[12] J. Uitto, S. Rauti, S. Laurén, and V. Leppänen, "A Survey on Anti-honeypot and Anti-introspection Methods," Adv. Intell. Syst. Comput., vol. 570, pp. 125–134, 2017, doi: 10.1007/978-3-319-56538-5_13.

[13] J. Cao, W. Li, J. Li, and B. Li, "DiPot: A Distributed Industrial Honeypot System," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10699 LNCS, pp. 300–309, 2018, doi: 10.1007/978-3-319-73830-7_30.

[14] E. López-Morales et al., "HoneyPLC: A Next-Generation Honeypot for Industrial Control Systems," Proc. ACM Conf. Comput. Commun. Secur., pp. 279–291, Oct. 2020, doi: 10.1145/3372297.3423356.

Breast Masses Detection Using YOLOv8

Abdul Moiz, Hikmat Ullah, Hannia Naseem, Umar Sadique, Muhammad Abeer Irfan, Amaad Khalil

Dept. of CSE, UET Peshawar, Pakistan

***Correspondence**: moizmansoor68@gmail.com, hikmatcse1919@gmail.com, hanninaseem836@gmail.com, umar.sadique@uetpeshawar.edu.pk, abeer.irfan@uetpeshawar.edu.pk, amaadkhalil@uetpeshawar.edu.pk.

Citation | Moiz. A, Ullah. H, Naseem. H, Sadique. U, Irfan. M. A, Khalil. A, "Breast Masses Detection Using YOLOv8", IJIST, Special Issue. pp 198-206, May 2024

Received | May 05, 2024 **Revised |** May 13, 2024 **Accepted |** May 19, 2024 **Published |** May 27, 2024.

Breast cancer stands as a formidable global health challenge, necessitating swift and precise diagnostic measures to combat its devastating impact. In this study, we delve into the efficacy of YOLOv8, a cutting-edge artificial intelligence model, for the precise detection and localizing of breast masses in digital mammography images. YOLOv8's inherent capability to simultaneously detect and localize masses showcases accurate pinpointing of the exact locations of abnormalities within mammographic scans. Our comprehensive evaluation reveals compelling performance metrics, including an F1 score of 0.91 and a mean Average Precision (mAP) of 0.942. These results depict the robustness of the YOLOv8 in mass detection but also show better results than the conventional clinical methods, offering higher accuracy and efficiency in the diagnostic process. This study explains the transformative potential of YOLOv8 in revolutionizing breast cancer detection paradigms, presenting a promising pathway toward enhancing early detection rates and ultimately improving patient outcomes.

Keywords: YOLOv8, Mass Detection and Localization, Digital Mammography

Introduction:

Breast cancer poses a significant global health challenge, underscoring the urgent need for advancements in diagnostic techniques to ensure early identification and improved patient outcomes [1]. While traditional methods like mammography have been pivotal in breast cancer screening, their limitations in sensitivity and specificity have prompted increased interest in leveraging technological breakthroughs, particularly the application of YOLO (You Only Look Once) and its latest iteration, YOLO v8, in medical imaging for enhanced breast cancer detection.

The intricate development of malignant breast masses stems from aberrant cell division within human tissues, leading to the emergence of benign and malignant masses. Benign masses, non-cancerous in nature, exhibit localized growth without aggressive tendencies. Conversely, malignant breast masses, driven by cancerous cells, possess an uncontrolled propensity to multiply and potentially spread to different body parts and adjacent tissues [2]. YOLO v8, as a cutting-edge object detection system, has emerged as a transformative force in medical imaging, heralding improved capabilities for disease detection [3].

In the realm of deep learning for breast cancer detection, the primary focus is on utilizing a diverse dataset to train YOLO v8 with representations from various mammography views. This strategic approach aims to augment the model's practi- cal performance by fostering a comprehensive understanding of breast cancer lesions. Overcoming challenges associated with traditional diagnostic techniques is crucial for achieving greater accuracy and efficacy [4]. The deliberate inclusion of mediolateral oblique and craniocaudal views is deemed essential, enhancing the model's ability to detect subtle pat- terns indicative of malignant growth. This comprehensive strategy elevates sensitivity and equips YOLO v8 to navigate complexities in identifying and classifying breast cancer [2].

This study is different from conventional approaches by taking leverage of a carefully curated breast mass dataset obtained from Roboflow to accurately annotate them which leads to achieving better results; further, we also conducted a validation process to ensure the dataset's quality and accuracy. Furthermore, collaborating with a radiologist for the result validation strengthens the clinical relevance of our findings. By employing YOLOv8 on this dataset and incorporating expert validation, our research offers valuable insights into the efficacy of YOLOv8 for breast mass detection. This investigation paves the way for further exploration of deep learning in breast cancer screening, potentially leading to more accurate diagnoses and improved patient care.

Literature Review:

Each year, the American Cancer Society estimates the numbers of new cancer cases and deaths in the United States and compiles the most recent data on population-based cancer occurrence and outcomes using incidence data collected by central cancer registries (through 2020) and mortality data collected by the National Center for Health Statistics (through 2021). In 2024, 2,001,140 new cancer cases and 611,720 cancer deaths are projected to occur in the United States [5]. The im- portance of mammography images in the diagnosis of breast cancer has led to a thorough investigation of developments in detection and classification.

Breast cancer impacts more than one in ten women globally, but it is particularly prominent—across all racial and ethnic groups—in the United States. The need for focused diagnostic efforts is highlighted by differences in incidence rates amongst ethnic groups [2][6]. Breast lesions are complex, three-dimensional anomalies that reflect a variety of radio- logically defined illnesses. The distinction between benign and malignant lesions must be made early to improve the prognosis of patients with this cancer, which is the most common in women and the second largest cause of cancer-related fatalities [7][8]. The use of deep learning techniques in computer vision, segmentation, detection, and image identification has increased dramatically in recent years, overcoming the drawbacks of conventional computer-aided diagnosis (CAD) methods [9][10][11]. Despite these developments, problems in manually identifying lesions and controlling memory complexity during training still exist.

Surprisingly, when compared to shallower models, deep learning models like Alex Net, Res Net, VGG16, Inception, Google Net, and Dense Net have shown improved classification performance. The classification accuracies attained by VGG16, ResNet50, and Inception-V3 were 95%, 92.5%, and 95.5%, respectively. Although these deep learning techniques perform better than shallow models, problems with memory complexity during training and manual detection still exist. Building upon the landscape of breast cancer detection, recent advancements by Mahoro and Akhloufi [12] showcase the potential of YOLOv7 and YOLOv8 in breast mass detection. The utilization of the Vin Dr-Mammo dataset, coupled with innovative image enhancement techniques, positions YOLOv8 as a superior model, outperforming its predecessor YOLOv7 and contributing to enhanced breast cancer diagnostics.

Al-antari et al.'s [13] important study involved estimating a Full Resolution Convolutional Network (FrCN) using a Computer-Aided Diagnosis (CAD) model. Their method, which used X-ray mammography and a four-fold cross- validation, showed a high accuracy of 95.96%. One approach to identifying breast cancer is called Diverse Features-Based Detection (DFeBCD), proposed by Chouhan and colleagues [14]. They assessed their method on the IRMA mammography dataset, and it attained an accuracy rate of 80.30% by combining an emotion learning-inspired integrated classifier (ELiEC) with the Support Vector Machine (SVM). Through the application of the Lifting Wavelet Transform (LWT) for feature extraction from breast images, Muduli et al. [15] made a substantial contribution to the field. With the use of the Extreme Learning Machine (ELM) and moth flame optimization methodology, their method, which combined Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), produced remarkable accuracy rates of 98.76% and 95.80% on the DDSM and MIAS databases, respectively.

A technique for detecting breast cancer based on diversity analysis, geostatistics, and alpha form was presented by Junior et al. [16]. They achieved a 96.30% detection accuracy using the Support Vector Machine (SVM) classifier on the DDSM and MIAS datasets. An approach for segmentation using a Mult granulation rough set and intuitionistic fuzzy soft set was presented by Ghosh et al. [17]. Their method distinguished between malignant and unaffected tissue in mammograms, hence addressing ambiguity in pixels. An effective Adaboost deep-learning technique for identifying breast cancer was created by Zheng et al. [18]. Their strategy, which combined multiple deep learning techniques with feature extraction and selection, produced a noteworthy 97.2% accuracy. Mahoro and Akhloufi investigated sophisticated deep-learning methods for breast mass identification, particularly the YOLO (You Only Look Once) framework. The promise of YOLO- based techniques in improving breast cancer diagnostics was demonstrated by the researchers by utilizing the Vin Dr- Mammo dataset and incorporating the YOLOv7 and YOLOv8 architectures.

Methodology:

In this study, a dataset containing breast mass mammograms was obtained from Rob flow publically available [19]. The dataset was carefully observed, ensuring that each image was annotated with bounding boxes around the identified masses. Block diagram of overall system is shown in figure 1.

Preprocessing:

Mammograms were preprocessed using functionality such as Auto-Orient and resizing the dataset. An auto-orient operation ensures that images are oriented correctly, and resizing is applied to stretch images to a constant size of 640x640 pixels. CLAHE (Contrast Limited Adaptive Histogram Equalization) was applied to enhance mammogram quality and visual- ize masses. Preprocessed mammograms can be seen in figure 2.

Data Splitting:

The previously developed dataset consists of a total of 1025 mammographic images. This dataset is divided into three groups: training, validation, and testing, to ensure robust model training, performance validation, and unbiased evaluation. The dataset is split as follows:

Train Set: 80% of the dataset, totaling 823 images, is allocated for training the model.

Valid Set: 13% of the dataset, totaling 135 images, is reserved for validating the model's performance during training.

Test Set: 7% of the dataset, totaling 67 images, is kept separate for final evaluation on unseen data.

Furthermore, each training example underwent data augmentation to enhance the model's robustness. The augmentation process included horizontal flipping, resulting in three augmented outputs per training example. Regarding the categories of the dataset, it comprises two classes:" mass" and" null." The" mass" class represents cases where a mass is present in the mammogram, indicating a potential abnormality, while the" null" class denotes mammograms without any detectable masses, indicating normal cases.

Model Selection and Training:

The YOLOv8 model was selected for its superior ability in object detection tasks. The YOLOv8 architecture allows for the simultaneous detection of several objects in one image, which is convenient for efficient breast mass detection, so the pre- processed data set for model training is transferred to Google Colab and the YOLOv8 model is trained on the training set.

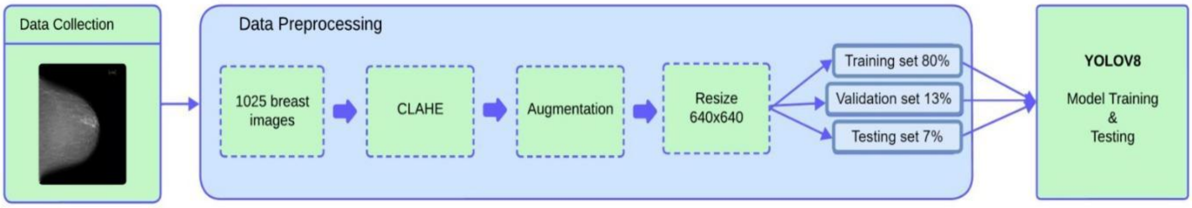


Figure 1: Block Diagram of YOLOv8-based Breast Mass Detection System

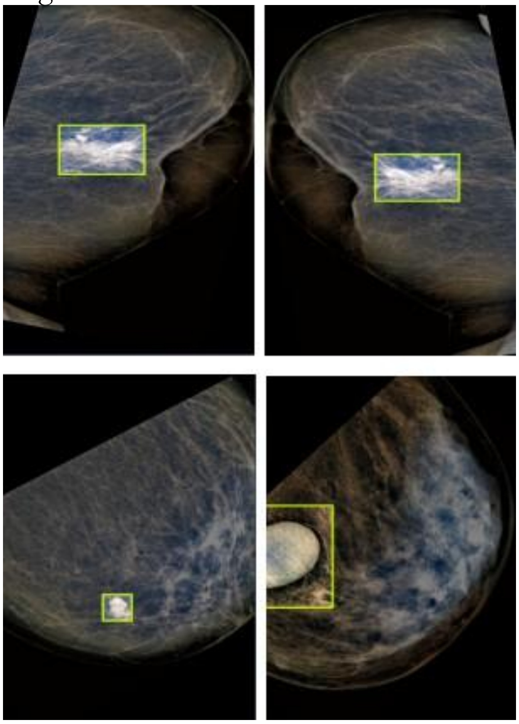


Figure 2: Preprocessed Mammograms

Testing Unseen Mammograms:

To test the generalization ability of the model, it was tested on images that were not used during training or validation. This step was taken to measure and visualize the ability of the model to detect breast mass accurately in real-world and unseen scenarios.

Result and Discussion:

The breast mass detection system using YOLOv8 was evaluated accurately and comprehensively, and the results showed its effectiveness in detecting and localizing masses in mammographic images.

Performance Metrics:

To test the robustness and generalization of the system, the model is evaluated using matrices such as F1-scores, maps (average accuracy) and PR curves.

F1-Score: F1-score is measured as the harmonic mean of precision and recall and is a valuable metric for assessing the balance between false positives and false negatives. It is calculated by the formula as shown in equation 1.

$$F1 = \frac{2 \times P \times R}{P + R} \qquad (1)$$

The F1 score for the breast mass detection system is 0.91. This score is an important indicator of the model's ability to achieve high accuracy and recall in breast mass detection, as shown in Figure 3.

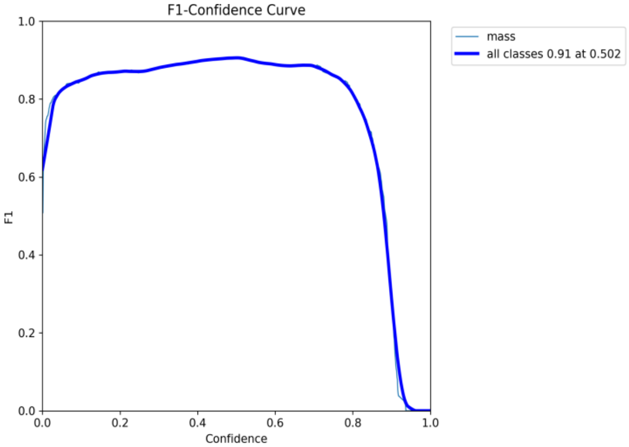


Figure 3: F1-Score Variation with Confidence Threshold

Figure 3 also shows the variation of F1-score with different confidence limits. It provides insight into model performance at different confidence points, showing robustness in balancing accuracy and recall at different operating points.

MAP (Mean Average Precision):

The map is calculated because it is a comprehensive measure that takes into account accuracy at different levels of confidence. The system achieved a commendable map of 0.942 as shown in Figure 4, confirming its high retention ability.

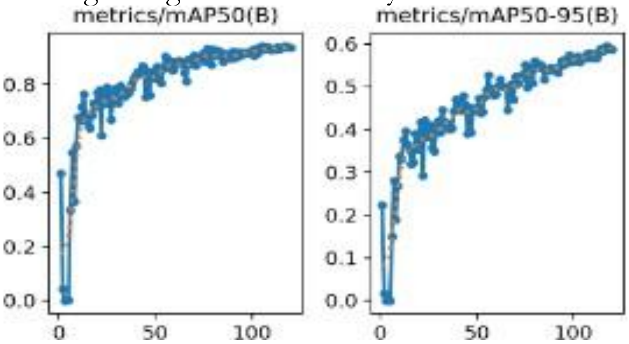


Figure 4: Mean Average Precision with IoU Threshold on the x-axis and Mean Average Precision on the y-axis

Recall and Precision Confidence Curves:

Recall and precision confidence curves provide a detailed analysis of the model's behavior at different confidence limits, showing how precision and recall vary. This curve is necessary to under- stand the trade-off between precision and recall at different confidence levels. Figure 5 shows the confidence Recall curve for the breast mass detection system. This curve represents the relationship between the return (R) and the confidence limit, helping to determine the optimal return point for the model without compromising accuracy.

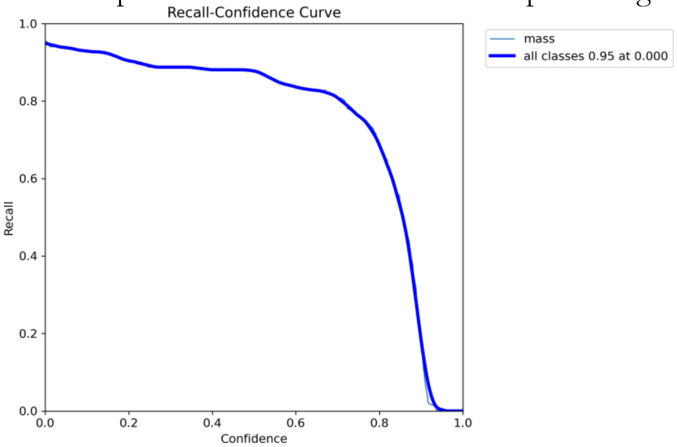


Figure 5: Recall Confidence Curve for Breast Masses Detection From equation 2 Recall (R) is defined as:

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (2)$$

Similarly, Figure 6 shows the precision confidence curve showing how precision (P) varies with different confidence limits. This curve helps to choose an appropriate operating point based on the desired balance between accuracy and recall. The formula for Precision (P) is given in equation 3 as:

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (3)$$

PR Curve:

Likewise, the Precision-Recall (PR) curve is an important visual tool that shows the model's performance across various precision-recall tradeoffs. Precision is produced by plotting against recall at different confidence limits. The PR curve of this system is shown in Figure 7, which is equal to 1.00, indicating its ability to provide high accuracy even at high recall levels.

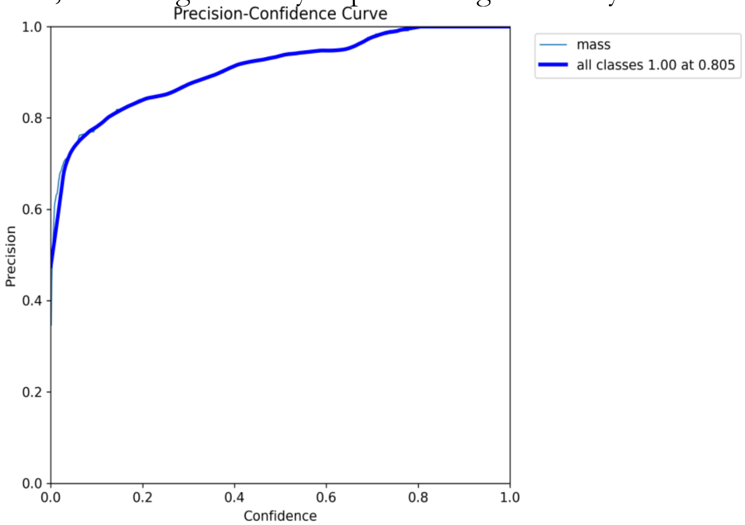


Figure 6: Precision Confidence Curve for Breast Masses Detection

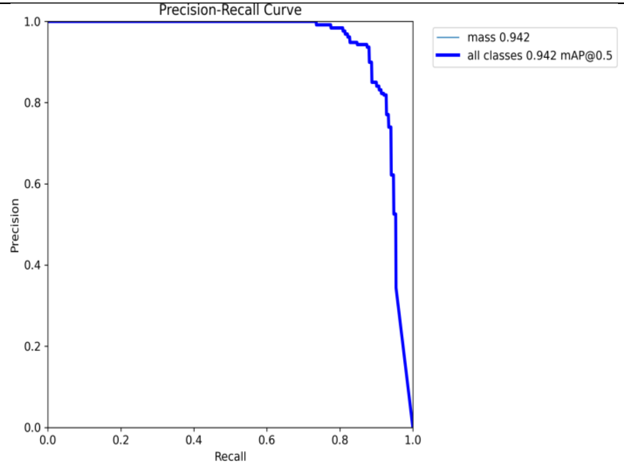


Figure 7: Precision-Recall Curve for Breast Masses Detection

Overall System Performance:

The overall performance of the system is carefully evaluated through a combination of quantitative measures, and various learning and evaluation metrics are considered to gain a deeper understanding of system performance. Figure 8 shows the plot of training and validation loss (Lbox, Lcls, Ldfl), recall and map (average accuracy) score during training.

Box-Loss (Bounding Box Loss):

Measures the localization accuracy of predicted bounding boxes and it is defined by the formula mentioned in equation 4.

$$L_{\text{box}} = \frac{1}{N \cdot B} \sum_{i=1}^{N} \sum_{j=1}^{B} \Big[(x_{i,j} - x^*_{i,j})^2 + (y_{i,j} - y^*_{i,j})^2 + (w_{i,j} - w^*_{i,j})^2 + (h_{i,j} - h^*_{i,j})^2 \Big] \quad (4)$$

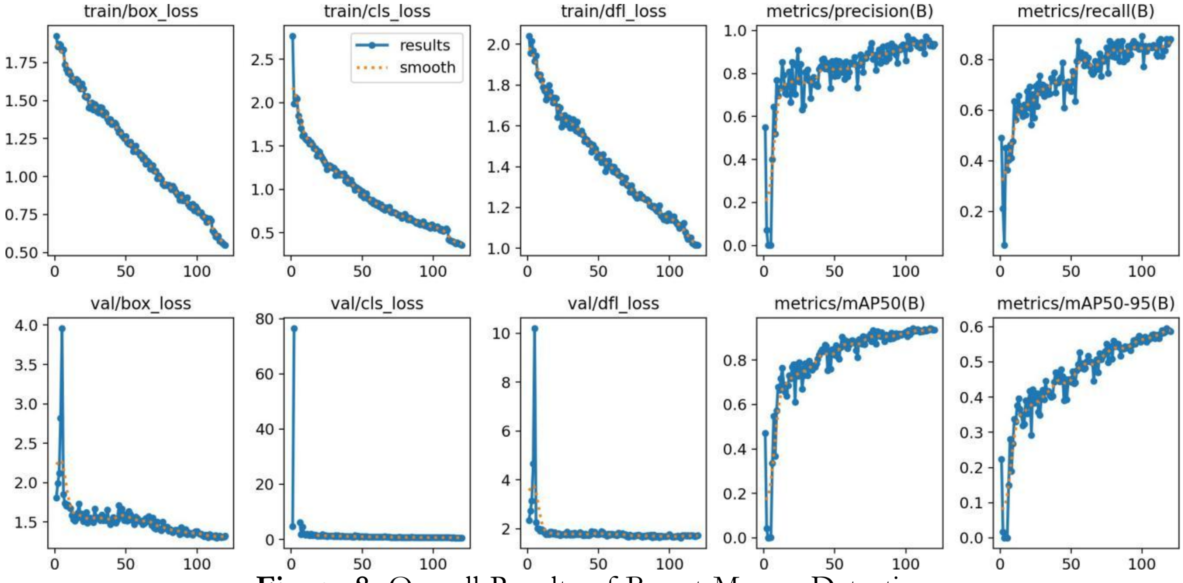


Figure 8: Overall Results of Breast Masses Detection

N: Number of samples.

B: Number of bounding boxes.

$x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j}$: Predicted coordinates.

$x^*_{i,j}, y^*_{i,j}, w^*_{i,j}, h^*_{i,j}$: Ground truth coordinates.

Cls-Loss (Classification Loss):

Evaluates the accuracy of object class predictions and it is calculated as in equation 5.

$$L_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big[y_{i,c}\cdot\log(p_{i,c}) + (1-y_{i,c})\cdot \log(1-p_{i,c})\Big] \quad (5)$$

N: Number of samples.
C: Number of classes.
Y$_{i,c}$: Ground truth class.
P$_{i,c}$: Predicted probability.

Dfl-Loss (Detection Focal Loss):

Represents the overall detection performance, combining localization and classification and its formula can be seen in equation 6.

$$L_{\text{dfl}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big[y_{i,c}\cdot(1-p_{i,c})^{\gamma}\cdot\log(p_{i,c}) + (1-y_{i,c})\cdot p_{i,c}^{\gamma}\cdot\log(1-p_{i,c})\Big] \quad (6)$$

N: Number of samples.
C: Number of classes.
Y$_{i,c}$: Ground truth class.
P$_{i,c}$: Predicted probability.
P: Tunable parameter.

MAP (Mean Average Precision):

Quantifies the precision-recall trade-off across different confidence thresholds. (Computed numerically using algorithms like the trapezoidal rule)

Conclusion:

Our results demonstrate the effectiveness of the advanced YOLOv8-based breast mass detection system. This model shows promising performance in mass localization in breast mammograms showing the potential to help in the diagnosis of breast cancer. Validation of system results with radiologists strengthens its clinical utility and reliability. However, it is important to acknowledge that the quality and quantity of descriptive images play an important role in model performance. In Rob flow, a more comprehensive and annotated dataset can improve the accuracy and reliability of the model. Our collab- oration with radiologists in the annotation process ensures that the database faithfully reflects real-world scenarios.

Future Directions:

For the future direction, it is recommended to expand interpretation efforts to include supplementary mammographic views like axillary tail, and tangential views. Adding these additional views to existing mammograms can significantly improve the model's ability to detect breast masses in a wider range of scenarios. In addition, the study of advanced imaging modalities such as tomosynthesis and ultrasound may contribute to a more comprehensive and multimodal breast cancer detection system.

References:

[1] G. Hamed, M. A. E. R. Marey, S. E. S. Amin, and M. F. Tolba, "Deep Learning in Breast Cancer Detection and Classification," Adv. Intell. Syst. Comput., vol. 1153 AISC, pp. 322–333, 2020, doi: 10.1007/978-3-030-44289-7_30.

[2] "World Health Organization," 2020, [Online]. Available: https://www.who.int/news-room/fact- sheets/detail/cancer

[3] J. Li, Q., Smith, "Deep Learning in Medical Imaging: A Comprehensive Review," J. Comput. Assist. Tomogr., vol. 42, no. 2, pp. 223–233, 2018.

[4] et al Johnson, M., "Dataset Diversity in Deep Learning for Medical Image Analysis," Int. J.

Comput. Vis., vol. 128, no. 3, pp. 777–795, 2020.

[5] "Rebecca L. Siegel, Surveillance Research, American Cancer Society, 270 Peachtree Street, Atlanta, GA 30303, USA".

[6] L. A. Torre, F. Bray, R. L. Siegel, J. Ferlay, J. Lortet- Tieulent, and A. Jemal, "Global cancer statistics, 2012," CA. Cancer J. Clin., vol. 65, no. 2, pp. 87–108, Mar. 2015, doi: 10.3322/CAAC.21262.

[7] E. Ward et al., "Cancer Disparities by Race/Ethnicity and Socioeconomic Status," CA. Cancer J. Clin., vol. 54, no. 2, pp. 78–93, Mar. 2004, doi: 10.3322/CANJCLIN.54.2.78.

[8] et al Dalaker, "Bureau of the Census, current population report, series P60–210. Poverty in the United States; 1997. U.S," Gov. Print. Off. Wash- ington, DC, 1999.

[9] "Cancer in North America: 2008–2012. Volume one: combined Cancer incidence for the United States, Canada and North America." Accessed: May 08, 2024. [Online]. Available: https://www.naaccr.org/wp-content/uploads/2016/11/Cina2015.v1.combined-incidence.pdf

[10] H. D. Cheng, J. Shan, W. Ju, Y. Guo, and L. Zhang, "Automated breast cancer detection and classification using ultrasound images: A survey," Pattern Recognit., vol. 43, no. 1, pp. 299–317, Jan. 2010, doi: 10.1016/J.PATCOG.2009.05.012.

[11] M. A. Al-antari, M. A. Al-masni, M. T. Choi, S. M. Han, and T. S. Kim, "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification," Int. J. Med. Inform., vol. 117, pp. 44–54, Sep. 2018, doi: 10.1016/J.IJMEDINF.2018.06.003.

[12] E. Mahoro and M. A. Akhloufi, "Breast masses detection on mammograms using recent one-shot deep object detectors," BioSMART 2023 - Proc. 5th Int. Conf. Bio-Engineering Smart Technol., 2023, doi: 10.1109/BIOSMART58455.2023.10162036.

[13] M. A. Al-antari, M. A. Al-masni, and T. S. Kim, "Deep Learning Computer-Aided Diagnosis for Breast Lesion in Digital Mammogram," Adv. Exp. Med. Biol., vol. 1213, pp. 59–72, 2020, doi: 10.1007/978-3-030-33128-3_4.

[14] N. Chouhan, A. Khan, J. Z. Shah, M. Hussnain, and M. W. Khan, "Deep convolutional neural network and emotional learning based breast cancer detection using digital mammography," Comput. Biol. Med., vol. 132, p. 104318, May 2021, doi: 10.1016/J.COMPBIOMED.2021.104318.

[15] D. Muduli, R. Dash, and B. Majhi, "Automated diagnosis of breast cancer using multi-modal datasets: A deep convolution neural network based approach," Biomed. Signal Process. Control, vol. 71, Jan. 2022, doi: 10.1016/J.BSPC.2021.102825.

[16] G. Braz Junior, S. V. Da Rocha, M. Gattass, A. C. Silva, and A. C. De Paiva, "A mass classification using spatial diversity approaches in mammography images for false positive reduction," Expert Syst. Appl., vol. 40, no. 18, pp. 7534–7543, Dec. 2013, doi: 10.1016/J.ESWA.2013.07.034.

[17] P. Ghosh, M. Mitchell, J. A. Tanyi, and A. Y. Hung, "Incorporating priors for medical image segmentation using a genetic algorithm," Neurocomputing, vol. 195, pp. 181–194, Jun. 2016, doi: 10.1016/J.NEUCOM.2015.09.123.

[18] "Correction to 'Deep Learning Assisted Efficient AdaBoost Algorithm for Breast Cancer Detection and Early Diagnosis' | IEEE Journals & Magazine | IEEE Xplore." Accessed: May 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9285266

[19] "Breast Cancer. "Breast Cancer Dataset." Open Source Dataset. Roboflow Universe, Roboflow", [Online]. Available: https://universe.roboflow.com/breast- cancer-ce1zx/breast-cancer-jtuaz

OPEN ACCESS

International Journal of Innovations in Science & Technology

RESEARCH & INNOVATION DIVISION

IJIST

Stress Detection and Prediction Using CNNs from Electrocardiogram Signals

Jehad Ur Rahman[1] Samad Riaz[1], Salma[2],
[1]Department of Electrical Engineering UET Peshawar, Pakistan,
[2]Institute of Nursing Science KMU Peshawar, Pakistan,
*Correspondence: jehadurrahman@uetpeshawar.edu.pk

Citation | Rahman. J. U, Riaz. S, Salma, "Stress Detection and Prediction Using CNNs from Electrocardiogram Signals", IJIST, Special Issue. pp 207-215, May 2024

Received | May 05, 2024 **Revised |** May 14, 2024 **Accepted |** May 20, 2024 **Published |** May 27, 2024.

Stress prediction is a crucial aspect of mental health monitoring, with consequences for both psychological well- being and productivity. This work presents a unique way for stress prediction that uses binary and multiclass classification models. Through extensive experimentations with different durations and frequencies of Electrocardiogram Signal (ECG) signals, we identified a 5-second dataset sampled at 200Hz as the optimal configuration for our model. Moreover, we introduced an innovative feature i.e., the prediction of stress scores ranging from 0 to 100, providing nuanced insights into stress levels, where 0 represents no stress and 100 indicates high stress levels. The model obtains 95.04% accuracy, 95.27% precision, 94.95% F1 score, 86.69% sensitivity, and 99.44% specificity for the binary classification. With "Fun" added to the list of stress categories in addition to "Base" and "TSST," the model continues to perform well in the multiclass classification scenario, with accuracy of 88.10%, precision of 87.60%, F1 score of 87.35%, sensitivity of 95.97%, and specificity of 79.23%. These findings highlight how well this applied strategy predicts stress levels, providing important information for mental health and stress management strategies.

Keywords: Stress Detection, Stress Score, ECG Signals, Stress Levels, CNNs.

Introduction:

People become increasingly stressed as societies expand because of the increased competition. This stress can have negative consequences for work, relationships, and safety. Our rapid and demanding society has made mental stress a widespread problem that has an impact on people's productivity and general well-being. Uncontrolled chronic stress can cause a variety of health issues, such as cardiovascular disease, depression, and anxiety [1]. Long- term stress can lead to depression, addiction, and heart and brain disorders [2]. Emotional stress is currently a major issue for both physical and mental health. Thus, creating efficient techniques for stress evaluation and management is essential to preserving public health. Stress is a physiological reaction to challenging events. It is distinguished by a sequence of physical and emotional changes, such as elevated heart rate, muscle tension, and anxiety. While acute stress can help mobilize resources to deal with urgent dangers, chronic stress, when extended, can be harmful to both physical and mental health. The neurological system in our bodies responds differently when we are under stress. Stress stimulates the sympathetic nervous system (SNS), which regulates heart rate and breathing. After stress, the parasympathetic nervous system (PNS) takes over to calm things down. We can detect stress by observing changes in parameters such as heart rate. Researchers have been looking into how electrocardiogram (ECG) signals, which assess heart activity, can help detect stress. Traditionally, they used five minutes of ECG data, which is too long for real-time monitoring [3]. Some studies have successfully detected stress using only one minute of ECG data, but this is still not ideal because it requires wearing uncomfortable equipment and is too slow for real-time monitoring.

Healthcare is one of the many industries that artificial intelligence (AI) has changed. AI has shown great promise in the field of stress assessment as a means of detecting stress patterns and forecasting stress levels. Complex patterns can be extracted and analyzed from a variety of data sources, such as physiological signals, behavioral data, and self-reported assessments, by AI models, especially deep learning algorithms. Because ECG signals are constant, freely accessible, and non-invasive, they have become more important in AI-based stress evaluation. ECG signals are the electrical activity of the heart [4]. Rich information on physiological changes linked to stress can be found in ECG signals, including heart rate variability (HRV), heart rate (HR), and signal complexity. These minute variations in ECG signals can be examined by AI models to precisely identify and categorize stress levels. Traditional approaches for assessing stress from ECG signals typically rely on hand- crafted time or frequency domain features [5]. These approaches, however, may be limited in their ability to capture complex patterns and correlations within ECG signals

We've developed a new method for detecting mental stress and predicting stress scores based on a Convolutional Neural Networks (CNN) architecture. Our method entails obtaining ECG signals, cleaning them using a bandpass filter to reduce noise, and altering the frequency from 700Hz to 100Hz. These preprocessed signals are then sent into a CNN, which extracts unique stress patterns from the temporal data of ECG signals. We trained two models: one for binary classification, which distinguishes between stressed and non-stressed individuals, and another for multiclass stress prediction. The binary classification technique has produced ground- breaking results in predicting stress levels. We use the obtained information to reliably diagnose stress levels and predict stress scores.

Literature Review:

In this article [6], the author investigated the analysis of ECG Raw Signal and Spectrogram pictures, using a dual method combining Raw ECG with 1D CNN and Spectrograms with ResNet-18 architecture. Their analysis produced complex results, with an accuracy of 66.6%, precision of 67.6%, and recall of 66.6% across three unique categories: neutral, tension, and amusement. This extensive study combined Leave-One-Subject-Out (LOSO) methods with chest-worn ECG data. Furthermore, the study expanded its investigation to the RML dataset, where deep learning models showed notable performance measures, including an accuracy of 72.7%, precision of 76.6%, and recall of 72.7%. Notably, this study used datasets from LESO, RML, and WESAD, allowing for both binary and three-class classification. In [5] emphasis is on the use of raw ECG signal data, which was analyzed using CNN and Bidirectional Long Short-Term Memory (BiLSTM) architecture. The results of this investigation were positive, with an overall accuracy of 86.5% and a specificity of 92.8%. Furthermore, the study carefully classified stress levels into three categories: low (91.3%), moderate (89.4%), and high (79.8%). These conclusions are based on locally acquired data, demonstrating the study's relevance and applicability. Article [7] analyzed ECG and HRV data using CNN for categorization purposes. Their investigation produced remarkable performance measures, including 97% accuracy, precision, recall, and F1-Score. Notably, the study got its data locally, which ensured the dataset's validity and dependability. Furthermore, the categorization assignment had three unique classes, which provided insights into subtle changes in the dataset. In the [8] raw ECG data is used, which was classified using CNN and VGG-inspired architectures. The study produced strong results, with claimed accuracies of 83.55% for three classes and 93.77% for two classes. Notable is the use of the Drive DB and Arachnophobia datasets, using a VGG-inspired architecture for binary classification and a 1D CNN for categorization into three classes. This strategic approach demonstrated the flexibility and versatility of the approaches used across a variety of datasets. The [9] performed a detailed investigation of ECG and HRV features using K-Nearest Neighbors (KNN) and Probabilistic Neural Network (PNN) classifiers. The study found impressive accuracies of 91.66% (ECG) and 94.66% (HRV), along with thorough specificity and sensitivity data for both modalities. The use of locally obtained data is significant since it increases the study's relevance and application to real-world circumstances. Furthermore, the study's emphasis on binary categorization highlighted its practical applications in the healthcare domain. The study described in [10] included the integration of ECG and EEG data using a Radial Basis Function Support Vector Machine (RBF-SVM) and KNN classifiers. The results showed significant accuracies ranging from 86.13% to 87.75% across various stress characteristics, as defined in the Kaggle dataset. This extensive research enabled binary categorization scenarios, revealing light on different stress levels and their physiological manifestations. The study's thorough approach to feature integration and categorization has shown its importance in the field of stress detection and management.

In [11], the authors conducted a thorough study of ECG plot pictures, investigating both time and frequency domains using CNN and Long Short-Term Memory (LSTM) architectures. The study revealed appealing performance data, including accuracies of 94.8% in the time domain and 98.3% in the frequency domain. Notable is the precise characterization of accuracy, sensitivity, and specificity measurements for each domain, which provides insight into the efficacy of the approaches used. The study's focus on binary classification tasks, which used the ST Change and WESAD datasets [12], emphasized its practical applications in healthcare and diagnostic contexts.

Within the scope of [13], the study focused on raw ECG data and used the CNN architecture for classification purposes. The research produced respectable findings, with a stated accuracy of 88.4% and an F1-score of 0.90. Notably, the study used data from the PhysioNet and SWELL databases, which allowed for categorization into three unique groups. This thorough technique demonstrated the resilience and usefulness of the used methodology in detecting small alterations within the dataset.

The study [14] investigated HRV features using Artificial Neural Network (ANN) and Naive Bayes (NB) classifiers. The study revealed impressive performance metrics, including an accuracy of 95.75% on the WESAD and SWELL-KW datasets for binary classification tasks. The full study of HRV characteristics is noteworthy, as it takes advantage of a synergistic method that combines the strengths of both ANN and NB classifiers. This intentional combination highlighted the study's effectiveness in detecting subtle patterns within the dataset, increasing its usefulness in therapeutic and diagnostic contexts. The research conducted [15] focused on the analysis of HRV features using a Support Vector Machine (SVM) classifier. The study provided insights into the dataset, with a reported accuracy of 72.82% on the SWELL-KW dataset [16] for binary classification tasks. Notably, the study's emphasis on HRV characteristics highlighted their importance in detecting minor alterations within the dataset, hence increasing its usefulness in therapeutic and diagnostic situations. Using CNN architecture, a thorough study of HRV Features was initiated in the [17]. On the Spider Fear dataset, the study produced impressive performance metrics: 83.29% accuracy, 85% precision, and 82% recall for classifying the data into three different categories. Of particular note is the careful characterization of the accuracy, recall, and sensitivity measures, which sheds light on how well the used algorithm distinguishes minute differences in the dataset. This thorough analysis highlighted how important the study was in clarifying subtle patterns in the dataset, which increased its use in diagnostic and clinical contexts.

Methodology:

The methodology includes custom data collection and preprocessing, selection of model architecture, and concluded results as shown in Figure 1.

Data Collection:

The dataset employed in this research comprises raw sensor data recorded using a chest-worn device (RespiBAN) and a wrist-worn device (Empatica E4). Synchronization of these devices was achieved by having subjects perform a double tapping gesture on their chest, creating a characteristic pattern in the acceleration signal. The synchronized raw sensor data and labels were stored in files labeled SX.pkl. The dataset includes various physiological modalities such as ACC (acceleration), ECG, EDA (electrodermal activity), EMG (electromyography), RESP (respiration), TEMP (temperature), and BVP (blood volume pulse). Labels were assigned to different study protocol conditions, with 0 = not defined / transient, 1 = baseline, 2 = stress, 3 = amusement, 4 = meditation, and 5/6/7 = disregarded conditions. Ground truth information was available in SX_quest.csv.

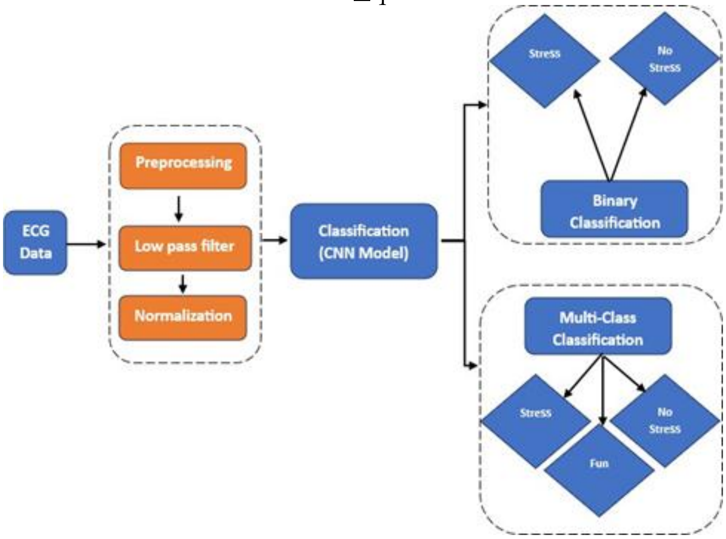


Figure 1: Block Diagram of the used methodology

Data Preprocessing:

Data Extraction:

From ECG recordings the data for binary classification focusing on stress and baseline conditions, and for multiclass classification stress, baseline, and amusement conditions were extracted from the dataset. There are different duration signals for each class then the signal was chunked to specific time durations of 30 seconds, 20 seconds, 15 seconds, 10 seconds, 5 seconds, and 3 seconds, these all are used for creating different datasets of different duration and finding the best time for stress prediction. The WESAD dataset ECG signal frequency is 700Hz this is used as one dataset and then different sampling frequencies 350Hz, 250Hz, 200Hz, and 100Hz were experimented with to find the optimal configuration, and best model using these all datasets.

Removing Noise from Signals:

We implemented a bandpass filter to increase the quality of the ECG data. As shown in Figure 2, this filter was designed to allow frequencies ranging from 0.5Hz to 50Hz while rejecting others. We effectively reduced high-frequency noise from the data, retaining only the desired frequency range for further analysis.

Normalizing Data:

The clean and preprocessed data are then normalized using the following mathematical formula (1) bring the data in the range of 0 to 1 here.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \qquad (1)$$

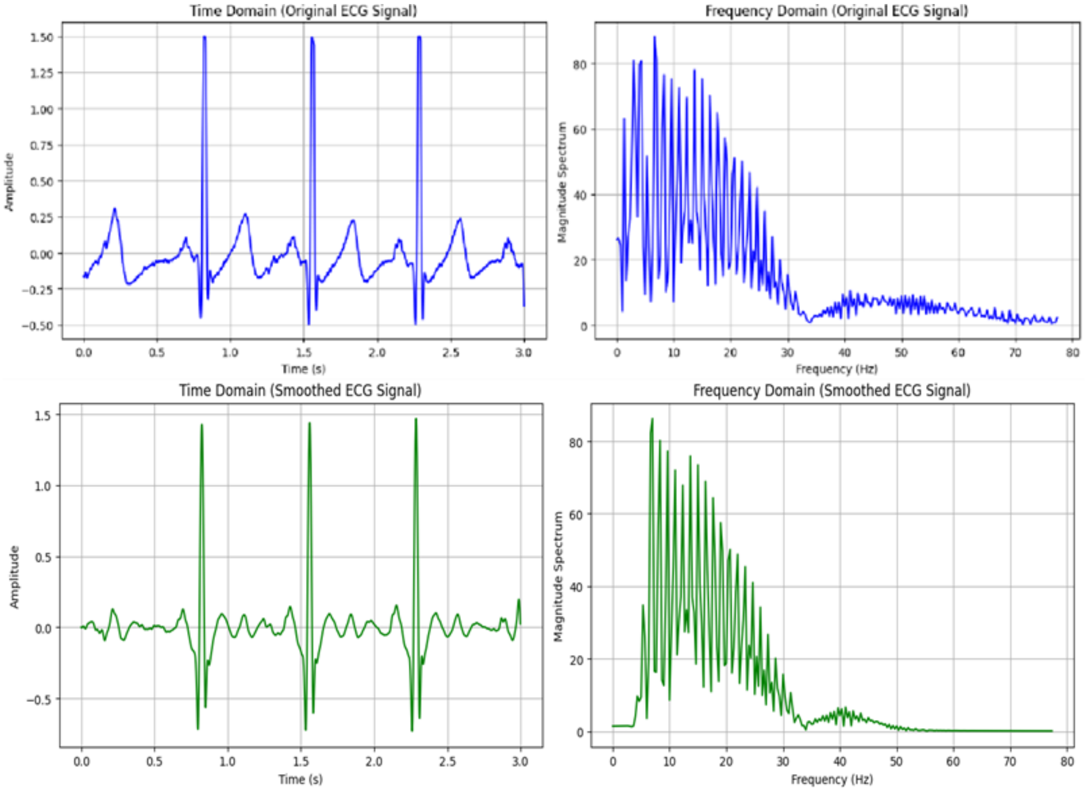


Figure 2: Raw and Filtered ECG signals

Where min and max are the minimum and maximum values in the dataset. This normalized dataset is used as input to the model.

Model Architectures and Selection:

Several neural network architectures were explored, including CNN, Long Short-Term Memory (LSTM) [18], and combinations like ANN with LSTM [19] and CNN with LSTM, Resnet34, and ResNet50 [20]. Each of these models was trained on all datasets, as after preprocessing we get datasets 30 seconds dataset with 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz and the same for 15 seconds, 10 seconds 5 seconds, and 3-second datasets. We have a total of 25 datasets and we applied each model on each dataset to get the best dataset duration and frequency and the best model that is less computational and accurate. And then we have the same datasets for multiclass classification.

For binary classification the selected CNN architecture shown in Figure 3 exhibited superior performance, achieving a training accuracy of 96.07%, a validation accuracy of 95.04%, and a test accuracy of 94.59%. while for multiclass classification the same CNN architecture shown in Figure 4 exhibits the best result 93.38% on training data, 88.67% on validation data, and 87.60% on test data. The complexity of the model depends on the size of the fed input sample size, this 5-second size and 200 Hz frequency selection methodology is 7 times more computationally efficient than the existing methodologies. For stress score prediction the binary model was used as the sigmoid activation function in the output layer was employed to predict stress scores in the range of 0 to 1, then it's multiplied by 100 to ensure a range from 0 to 100. A stress score of 100 means high stress and 0 means no stress.

Result and Discussion:

To detect stress levels in real-time, we created a deep neural network and compared its performance to more traditional methods that rely on manually built features. We proposed a 1D-CNN base model that takes the Raw ECG data of 5 seconds and a frequency of 200Hz. We implement different models for getting the optimal model for the data, the dataset of 30 seconds is used and the ANN model is trained the result of the ANN model for the 30 seconds datasets 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz has accuracy 67.43%, 68.43%, 66.56%, 71.21% and 69.21% respectively. For multiclass classification the accuracy for this dataset of 30 seconds with frequencies of 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz having an accuracy of 57.13%, 59.93%, 61.56%, 62.91%, and 59.01% respectively. All the models were trained in the same way CNN model with 9 layers and with batch normalization of each layer used for all the data the result is for 30 seconds it had the highest accuracy for 250Hz and the accuracy for this was 83.34% for binary and 78.67% for multiclass classification. For 15 seconds dataset, the CNN has the highest accuracy for 200Hz, 82.23% for binary, and 72.39% for multiclass classification, for the dataset of 10 seconds the 200HZ has good results, 90.32% for binary and 82.32% for multiclass classification, for the dataset of 5 second the model has the accuracy of 95.04 for binary and 88.67% for multiclass and the frequency for this result was 200Hz and this is our selected data and model. We also tried 3 second dataset with all frequencies but its result was not good as 5 seconds. The dataset for the 5 seconds contains 6595 examples for binary and 7987 examples are used for multiclass classification. As we have an additional feature in our method which is stress score prediction for that purpose, we used a binary model and the output layer used the sigmoid activation function (2).

$$S(z) = \frac{1}{1 + e^{-z}} \qquad (2)$$

This equation gives the value from 0 to 1 range to classify it as no stress or stress the threshold is 0.5 and then predicts the stress score from 0 to 100 using that model output multiplied by 100. The final training and validation accuracy and loss plots are in Figure 5 for binary classification.

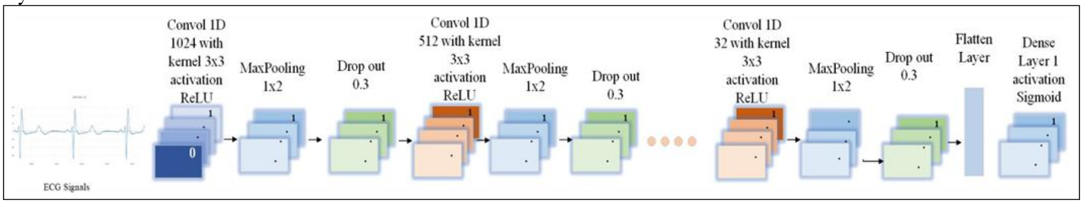

Convol 1D 1024 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Convol 1D 512 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Convol 1D 32 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Flatten Layer
Dense Layer 1 activation Sigmoid
0
1
ECG Signals

Figure 3: 1-D CNN architecture for binary class classification and prediction.

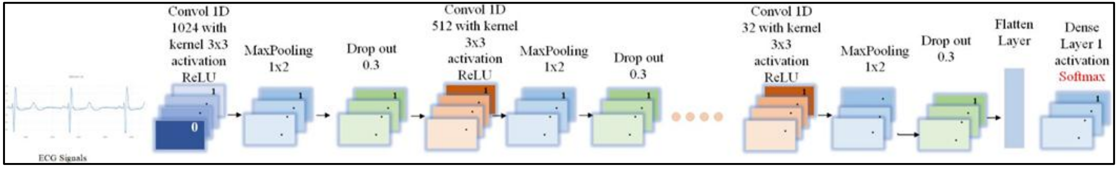

Convol 1D 1024 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Convol 1D 512 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Convol 1D 32 with kernel 3x3 activation ReLU
MaxPooling 1x2
Drop out 0.3
Flatten Layer
Dense Layer 1 activation Softmax
0
1
ECG Signals

Figure 4: 1-D CNN architecture for multiclass classification and prediction.

The performance of the model is determined by the accuracy (3), precision (4), F1 score (5), sensitivity (6) and specificity (7).

$$Accuray = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3)$$

$$Precision = \frac{TP}{TP + FP} \qquad (4)$$

$$F1\,Score = 2\,x\,\frac{Recall\ x\ Precision}{Recall + Precision} \qquad (5)$$

$$Sensitivity = \frac{TP}{TP + FN} \qquad (6)$$

$$Specificity = \frac{TN}{TN + FP} \qquad (7)$$

- True positive (TP) = the number of cases accurately identified as stress.
- False positive (FP) = the number of cases wrongly diagnosed as stress.
- True Negative (TN) = the number of instances correctly diagnosed as having no stress.
- False negative (FN) = the number of cases mistakenly categorized as "no stress."
- The performance of the binary and multiclass model in terms of accuracy, precision, F1 score, sensitivity and specificity are mentioned in table 1.
- The confusion matrix of the binary model is on validation test data is shown in the Figure 5.
- Figure 6 shows the multiclass classification model's confusion matrix based on validation and test data.

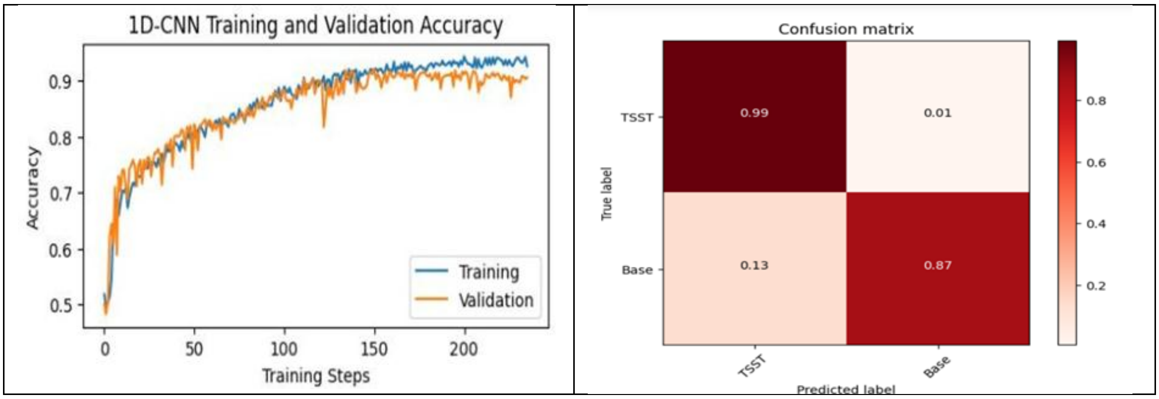


Figure 5: Performance metrics of the binary classification model on 5 seconds 200Hz dataset

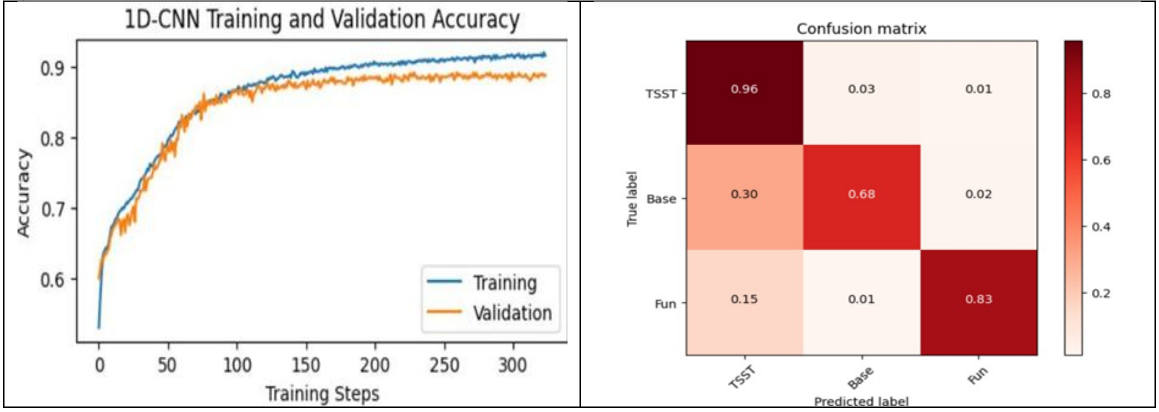


Figure 6: Performance metrics of multiclass classification model on 5 seconds 200Hz dataset.

Table 1: Different evaluation metrics of binary and multiclass model.

Performance Matrices	Binary Model	Multiclass Model
Accuracy	95.04%	88.10%
Precision	95.27%	87.60%
F1 score	94/95%	87.35%
Sensitivity	86.69%	95.97%
Specificity	99.44%	79.23%
PPV	98.96%	85.55%
NPV	93.64%	78.97%

Conclusion:

In this study, we have developed a deep neural network-based approach for real-time stress detection utilizing electrocardiogram (ECG) data. By comparing our proposed 1D-CNN model against traditional methods relying on manually engineered features, we demonstrated superior performance in stress prediction. Our methodology involved rigorous experimentation with different dataset durations and sampling frequencies, aiming to optimize model accuracy and computational efficiency. Through extensive model selection and evaluation, we found that a Simple 1D Convolutional Neural Network architecture yielded the best results for both binary and multiclass stress classification tasks. Specifically, our model achieved notable accuracies across various dataset configurations, with the highest accuracy obtained for a 5-second dataset sampled at 200Hz, demonstrating the effectiveness of our approach in capturing temporal dynamics of stress patterns.

Additionally, we introduced an innovative aspect to our methodology by incorporating stress score prediction, enabling a finer-grained understanding of stress levels ranging from 0 to 100. Leveraging the sigmoid activation function in the output layer of our binary model, we accurately predicted stress scores, further enhancing the utility of our approach for comprehensive stress assessment. Our study underscores the importance of leveraging deep learning techniques for stress detection, offering valuable insights into individuals' well-being and mental health. The ability to predict stress levels in real-time has significant implications for personalized stress management interventions and improving overall quality of life. Moving forward, further research may explore additional physiological modalities and sensor data fusion techniques to enhance the robustness and generalizability of stress detection models in diverse real-world settings.

References:

[1] A. Mariotti, “The effects of chronic stress on health: New insights into the molecular mechanisms of brain-body communication,” Futur. Sci. OA, vol. 1, no. 3, Nov. 2015, doi: 10.4155/FSO.15.21/ASSET/IMAGES/LARGE/FIGURE2.JPEG.

[2] S. L. Sauter, L. R. Murphy, and J. J. Hurrell, “Prevention of work-related psychological disorders: A national strategy proposed by the National Institute for Occupational Safety and Health (NIOSH).,” Work well-being An agenda 1990s., pp. 17–40, Oct. 2004, doi: 10.1037/10108-002.

[3] A. H. Khandoker, H. F. Jelinek, and M. Palaniswami, “Heart rate variability and complexity in people with diabetes associated cardiac autonomic neuropathy,” Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS’08 - "Personalized Healthc. through Technol., pp. 4696–4699, 2008, doi: 10.1109/iembs.2008.4650261.

[4] M. A. Serhani, H. T. El Kassabi, H. Ismail, and A. N. Navaz, “ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges,” Sensors 2020, Vol. 20, Page 1796, vol. 20, no. 6, p. 1796, Mar. 2020, doi: 10.3390/S20061796.

[5] P. Zhang et al., “Real-Time Psychological Stress Detection According to ECG Using Deep Learning,” Appl. Sci. 2021, Vol. 11, Page 3838, vol. 11, no. 9, p. 3838, Apr. 2021, doi: 10.3390/APP11093838.

[6] Z. Ahmad, S. Rabbani, M. R. Zafar, S. Ishaque, S. Krishnan, and N. Khan, “Multilevel Stress Assessment from ECG in a Virtual Reality Environment Using Multimodal Fusion,” IEEE Sens.

J., vol. 23, no. 23, pp. 29559–29570, Dec. 2023, doi: 10.1109/JSEN.2023.3323290.

[7] R. Zhou et al., "ECG-based biometric under different psychological stress states," Comput. Methods Programs Biomed., vol. 202, p. 106005, Apr. 2021, doi: 10.1016/J.CMPB.2021.106005.

[8] K. Tzevelekakis, Z. Stefanidi, and G. Margetis, "Real-Time Stress Level Feedback from Raw Ecg Signals for Personalised, Context-Aware Applications Using Lightweight Convolutional Neural Network Architectures," Sensors 2021, Vol. 21, Page 7802, vol. 21, no. 23, p. 7802, Nov. 2021, doi: 10.3390/S21237802.

[9] P. Karthikeyan, M. Murugappan, and S. Yaacob, "DETECTION OF HUMAN STRESS USING SHORT-TERM ECG AND HRV SIGNALS," https://doi.org/10.1142/S0219519413500383, vol. 13, no. 2, Apr. 2013, doi: 10.1142/S0219519413500383.

[10] A. Hemakom, D. Atiwiwat, and P. Israsena, "ECG and EEG based detection and multilevel classification of stress using machine learning for specified genders: A preliminary study," PLoS One, vol. 18, no. 9, p. e0291070, Sep. 2023, doi: 10.1371/JOURNAL.PONE.0291070.

[11] M. Kang, S. Shin, J. Jung, and Y. T. Kim, "Classification of Mental Stress Using CNN-LSTM Algorithms with Electrocardiogram Signals," J. Healthc. Eng., vol. 2021, 2021, doi: 10.1155/2021/9951905.

[12] P. Schmidt, A. Reiss, R. Duerichen, and K. Van Laerhoven, "Introducing WeSAD, a multimodal dataset for wearable stress and affect detection," ICMI 2018 - Proc. 2018 Int. Conf. Multimodal Interact., pp. 400–408, Oct. 2018, doi: 10.1145/3242969.3242985.

[13] M. Donati, M. Olivelli, R. Giovannini, and L. Fanucci, "ECG-Based Stress Detection and Productivity Factors Monitoring: The Real-Time Production Factory System," Sensors 2023, Vol. 23, Page 5502, vol. 23, no. 12, p. 5502, Jun. 2023, doi: 10.3390/S23125502.

[14] M. R. S. Zawad, C. S. A. Rony, M. Y. Haque, M. H. Al Banna, M. Mahmud, and M. S. Kaiser, "A Hybrid Approach for Stress Prediction from Heart Rate Variability," Lect. Notes Networks Syst., vol. 519 LNNS, pp. 111–121, 2023, doi: 10.1007/978-981-19-5191-6_10.

[15] S. Sriramprakash, V. D. Prasanna, and O. V. R. Murthy, "Stress Detection in Working People," Procedia Comput. Sci., vol. 115, pp. 359–366, Jan. 2017, doi: 10.1016/J.PROCS.2017.09.090.

[16] S. Koldijk, M. Sappelli, S. Verberne, M. A. Neerincx, and W. Kraaij, "The Swell knowledge work dataset for stress and user modeling research," ICMI 2014 - Proc. 2014 Int. Conf. Multimodal Interact., pp. 291–298, Nov. 2014, doi: 10.1145/2663204.2663257.

[17] A. Vulpe-Grigorasi and O. Grigore, "A Neural Network Approach for Anxiety Detection Based on ECG," 2021 9th E-Health Bioeng. Conf. EHB 2021, 2021, doi: 10.1109/EHB52898.2021.9657544.

[18] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," Neural Comput., vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/NECO_A_01199.

[19] N. Faris Ali and M. Atef, "An efficient hybrid LSTM-ANN joint classification-regression model for PPG based blood pressure monitoring," Biomed. Signal Process. Control, vol. 84, p. 104782, Jul. 2023, doi: 10.1016/J.BSPC.2023.104782.

[20] B. Koonce, "ResNet 50," Convolutional Neural Networks with Swift Tensorflow, pp. 63–72, 2021, doi: 10.1007/978-1-4842-6168-2_6.

OPEN ACCESS
International Journal of Innovations in Science & Technology

Classification of Medical Images Through Convolutional Neural Network Modification Method

Syed Kashif Badshah[1] Noor Badshah[1]
[1]Department of Basic Sciences and Islamiat, University of Engineering and Technology, Peshawar, Pakistan
*Correspondence: skashif.bsi@uetpeshawar.edu.pk, noorbadshah@uetpeshawar.edu.pk

Citation | Badshah. S. K, Badshah. N, "Classification of Medical Images Through Convolutional Neural Network Modification Method", IJIST, Special Issue pp 216-226, May 2024 May

Received| May 04, 2024, **Revised|** May 12, 2024, **Accepted|** May 21, 2024, **Published|** May 27, 2024.

The COVID-19 positive, tuberculosis and pneumonia, share the trait of being able to be identified using radiological investigations, such as Chest X-ray (CXR) images. This paper aims to distinguish between four classes, including tuberculosis (TB), COVID-19 positive, healthy, and pneumonia using CXR images. Many deep-learning models such as a Convolutional Neural Network (CNN) have been developed for the Classification of CXR images. Deep learning-based models such as CNN offer significant advantages over traditional methods in the classification of diseases like TB, COVID-19, pneumonia, and healthy states. They provide higher accuracy, automation, early detection, reduced subjectivity, and resource efficiency, ultimately leading to improved patient care and outcomes. However, well-liked CNNs are massive models that require a lot of data to achieve optimal accuracy. In this paper, we propose a new CNN model that can be used to distinguish between different classes of CXR images. This model proves to be effective in classifying different diseases such as pneumonia, COVID-19, and tuberculosis. This study has used 6326 CXR images dataset containing COVID-19 positive, tuberculosis, and pneumonia and has normal images. In this dataset, 80% of the CXR images are taken for the training purpose and 20% are taken for the validation purpose, of the proposed CNN model. The proposed CNN modified model with parameter adjustment as well as using categorical cross-entropy as a loss function obtains the highest classification accuracy of 98.51% with a precision, recall, and F1 score of 0.98, 0.985, and 0.98 respectively.

Keywords: Image Classification; Fuzzy Membership; VGG-19 Modified Model

Introduction:

COVID-19 has set historical records on a global scale. More than 116 million confirmed cases and more than 2.5 million fatalities had been recorded by the World Health Organization (WHO) as of 8 March 2021. The SARS-CoV2 virus, which causes the infectious disease COVID-19, is easily spread by contact and the air and has a serious impact on the lungs of those who contract it. The COVID-19 virus can develop consequences, including pneumonia, as well as other symptoms that may be mistaken for those of other infections [1].

In contrast, the infectious disease tuberculosis, which is caused by Mycobacterium tuberculosis, also results in antibiotic resistance and the death of tissue in various sections of the body, primarily the lungs. The WHO estimates that tuberculosis kills over 1.5 million people worldwide each year, making it the most lethal infectious disease. An estimated 10 million people contracted TB in 2019 alone [1].

As a result, COVID-19, pneumonia, and tuberculosis all have the ability to be identified by radiological procedures like CXR images. Prior to the development of deep learning (DL) frameworks, feature extraction and classification methods were used to classify medical images. Medical image classification through Convolutional Neural Networks (CNNs) is an application of deep learning that has revolutionized the field of medical imaging. With the increase in availability and quality of medical imaging data, there has been a growing demand for automated and accurate image classification tools that can assist clinicians in their diagnoses and treatment decisions. CNN is a neural network that is designed specifically for image analysis tasks. They are made up of several layers of interlinked nodes able to recognize and extract features from pictures. In medical image classification, CNN is typically trained on large datasets of labeled images, where the labels represent the different diagnostic categories or pathologies of interest. The process of medical image classification through CNN involves several steps. First, the input medical images are preprocessed to ensure that they are in a suitable format for analysis. Next, the CNN is trained on a dataset of labeled medical images, using a process called backpropagation to adjust the weights of the network to optimize its performance on the training data. Once the CNN has been trained, it can be used to classify new medical images by passing them through the network and observing the output of the final layer, which represents the predicted class label.

CNN-based models have been successfully applied and used to create dependable, quick, and accurate detection methods against COVID-19 and other respiratory diseases, demonstrating their potential for transforming medical diagnostic procedures. This is due to the models' deep learning capabilities and intricate architectures [2][3][4][5][6][7]. This is why the current study's objective is to determine whether it is possible to distinguish between healthy patients and those with COVID-19 positive, pneumonia, and TB using early automated classification of CXR images. To determine if a patient is normal or has a lung illness, we have built a deep transfer learning pipeline called the VGG-19 modified model. The VGG-l9 is ideally modified in the suggested network. Transfer learning techniques are implemented using pre-trained networks on the VGG-19 model. We put our suggested network to the test for four ICTIS 2024 class classification problems (TB, pneumonia, healthy, and COVID-19 positive).

To the best of our current understanding, this work is an important attempt to investigate the viability and effectiveness of using early automatic identification and differentiation methods with a particular focus on differentiating between people who are COVID-19 positive, suffering from pneumonia or tuberculosis, and those who are considered healthy using only CXR images as the primary diagnostic modality. The suggested model has demonstrated robust and enhanced performance over the state-of-the-art methods for the classification of 1ung disorders in all of our datasets and has been able to perform optimally in a variety of multi-class tasks. Moreover, it has demonstrated its versatility and efficacy in intricate diagnostic circumstances by continuously outperforming a wide range of multi-class tasks. This work represents a major step forward in using computational approaches to support early and precise identification of respiratory diseases, especially when it comes to using CXR imaging to differentiate COVID-19 from other related disorders. The following are this paper's major contributions:

- The building of a new convolutional neural network, named VGG-19 modified model, for robustness and more precise classification.
- Evaluating the VGG-19 modified model's precision and robustness on CXR image datasets with many classes of labels (TB, pneumonia, COVID-19 positive, and healthy).
- The comparison of the VGG-19 modified model with the other state-of-the-art architectures such as VGG-16, DenseNet-121, and ResNet-50.

The pattern of the paper is organized as follows:

- Section I introduces the specified title.
- The associated works are briefly summarized in Section II.
- The preprocessing approach and the suggested methodology are presented in Sections III and IV respectively.
- The numerical outcomes of our methodology on the CXR dataset are shown in Section V.
- Section VI presents the final conclusion.

Literature Review:

The classification of CXR images has been the subject of extensive research in recent years. A brief summary of these research initiatives is provided below: Jaiswal et al. [8] introduced a Mask-Region-based CNN model, aimed at automating the classification process of pneumonia cases utilizing CXR images. Bharati et al. [9] concentrated on making use of a hybrid deep learning system that blends several modern techniques, such as data augmentation approaches, spatial transformer networks (STNs), and Convolutional Neural Networks (CNNs) with VGG architecture. Bharati et al. sought to create a complete framework that may improve the resilience and accuracy of classification tasks by combining these disparate components, especially when it came to the diagnosis of medical disorders from CXR pictures.

CNNs are well known for their ability to extract features and recognize patterns; this helped to provide the groundwork for the proposed system's learning capabilities. Additionally, the utilization of VGG architecture, known for its depth and effectiveness in image classification tasks, further bolstered the model's performance. To address potential spatial variations and distortions within the CXR images, Bharati et al. integrated a spatial transformer network (STN), enabling the network to dynamically adapt and rectify spatial transformations to enhance its adaptability and resilience to image variations. Furthermore, to augment the training dataset and mitigate overfitting, data augmentation techniques were employed, facilitating the generation of diverse training samples by applying transformations such as rotations, translations, and scaling. Through extensive experimentation and training on the National Institutes of Health (NIH) CXR dataset, Bharati et al. reported notable results, achieving an accuracy rate of 73%. They trained their network in the NIH CXR dataset and achieved 73% accuracy.

Pereira et al. [10] introduced the RYDLS-20 network model, representing a significant advancement in the field of medical image analysis. The primary focus of their study was on the diagnosis of COVID-19 utilizing deep learning techniques. The RYDLS-20 model, meticulously designed and optimized by Pereira et al., achieved an impressive F1 value of 89%, underscoring its efficacy and reliability in accurately identifying COVID-19 cases. Notably, the dataset utilized in their study exhibited a considerable imbalance, consisting of 2000 healthy cases compared with only 180 patients afflicted by COVID-19. Without using any cross-validation steps, the RYDLS-20 model's classification performance was shown, demonstrating the model's capacity to attain high accuracy even in the absence of such validation processes. This omission may cause questions about how well the model generalizes to new data, but the stated F1 value indicates that there is a good chance that it will be useful in everyday life.

Song et al. [2] built a COVID-19 patient identification system for computed tomography (CT) scans called deep pneumonia that is based on deep learning. After manually segmenting the lung area using a DL network, they classified healthy or COVID-19 patients. They combined ResNet50 with a feature pyramid network (FPN) and an attention model to create their own network, which they called DRE-Net. The investigation's primary strengths were highlighted by the study, which used mu1ti-vendor datasets from three different hospitals and showed impressive sensitivity (95%) and specificity (96%), as well as a quick diagnostic time of only 30 seconds per patient. Several drawbacks can be identified in this study. Firstly, the reliance on semi-automatic lung segmentation raises concerns about the consistency and accuracy of the segmentation process, potentially introducing variability in the analysis. Secondly, the c1assification of datasets solely based on CT images without stratification according to factors such as advanced age, underlying diseases, or the presence of pleural effusions could lead to biased results.

Chen et al. [7] applied 46, 095 anonymized images of 106 hospitalized patients at Renmin Hospital of the University of Wuhan to train their deep network. Of them, 51 patients had COVID-19 pneumonia confirmed by a laboratory, and the remaining 55 patients had various illnesses. The lungs were divided into sections, and any scar tissue was located using a U-net++ network. The time difference between the radiologist and the model was compared using a two-tailed paired Student's t-test with a significance level of 0.05. The main strengths of this work are the huge and equally distributed training dataset, the good classification accuracy (above 95%), and the use of three experienced radio1ogists who considered inter-observer variability to obtain the ground truth.

Ozturk et al. [11] leveraged the Dark Covid Net model as a tool to assist radiologists and medical professionals in diagnosing COVID-19. This model demonstrated a remarkable accuracy in binary classification, achieving an impressive 98.08% accuracy rate in distinguishing between COVID-19 cases and healthy individuals. Additionally, in more complex multi-class classification scenarios where the model had to differentiate between pneumonia, COVID-19, and healthy cases, it maintained a respectable accuracy of 87.02%.

A brand-new transfer learning-based mode1 for the categorization and detection of pneumonia (both viral and bacterial) was put out by Rahman et al. [12]. To determine which pre-designed CNN architecture had the greatest performance, they suggested comparing the various ones. The results demonstrated that, of all the utilized architectures, DenseNet201 demonstrated remarkable accuracy rates, with a noteworthy 98% accuracy in differentiating between chest X-rays that showed no symptoms and those that showed indicators of pneumonia (bacterial or viral). In addition, when the model was required to distinguish between cases of bacterial, viral, and normal pneumonia, it performed admirably, scoring 93.3%. In this particular subset, the model had a robust accuracy rate of 95% when it came to differentiating between pneumonia caused by bacteria and viruses.

Michail et al. [13] presented the DenResCov-19 deep transfer learning model to identify patients with Pneumonia, TB, COVID-19, or health based on the CXR images dataset. Their method combines the previous ResNet-50 and DenseNet-121 architectures with an extra layer of convolutional neural network (CNN) building blocks to improve the model's functionality. They were able to leverage the advantages of both networks by combining these topologies, using ResNet's residual connections to mitigate vanishing gradient problems and DenseNet's dense connections for feature reuse. Their network was tested on the CXR image dataset, which included several classifications such as COVID-19, pneumonia, TB, and normal cases. With careful tweaking and training, the network demonstrated an amazing accuracy rate of 96.40%. This significant accuracy highlights the potential of deep transfer learning approaches in the field of medica1 image analysis, providing promising paths for the early and accurate diagnosis of a variety of respiratory conditions, including COVID-19, and enabling prompt intervention and treatment plans.

Dataset Description:

We have collected a large set of CXR images and applied data augmentation techniques to increase data diversity in an attempt to improve classification accuracy. The study employed a dataset consisting of 6326 CXR images that were obtained from the free software platform Kaggle. This dataset contains pictures of a variety of diseases, such as pneumonia, tuberculosis, and COVID-19-positive patients, in addition to typical cases.

Preprocessing:

Fuzzy set theory and Gaussian kernel-based enhancement are used to improve the performance of CNN-based image classification. Fuzzy set theory is a mathematical framework that deals with uncertainty and imprecision. It allows for the representation of a concept with degrees of membership instead of a binary yes/no value. In the context of image classification, this means that instead of assigning a single label to an image, we can assign multiple labels with different degrees of certainty. This approach can be particularly useful when dealing with images that contain ambiguous or overlapping features.

Fuzzy set theory involves the use of membership functions to assign degrees of membership to elements of a set. Let X be the set of possible image features, and let A be a fuzzy subset of X. The membership function for A is denoted by μA(x), where x is an element of X, and μA(x) is a value between 0 and 1 that represents the degree to which x belongs to A. For each image feature x, we compute the membership function values $\mu A_1(x)$, $\mu A_2(x)$, ..., $\mu A_n(x)$ for each of the fuzzy subsets A_1, A_2, ..., A_n. Then we apply the Gaussian kernel function to the image to smooth it and reduce noise.

To improve the quality of the images, we experimented with a range of image enhancement methods, such as Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), and Fuzzy Contrast Enhance (FCE). Each of these methods was applied to a single dataset image. Figure 1 displays the outcomes of various methods. HE and CLAHE have demonstrated impressive results in denoising and improving the images, as illustrated in Figure 1. But it may also be possible to see that using CLAHE intensifies the color of the bones, which could have an impact on how well the classification model performs because the rib and sternum bones may be identified by the neural network as the main X-ray detection features [14].

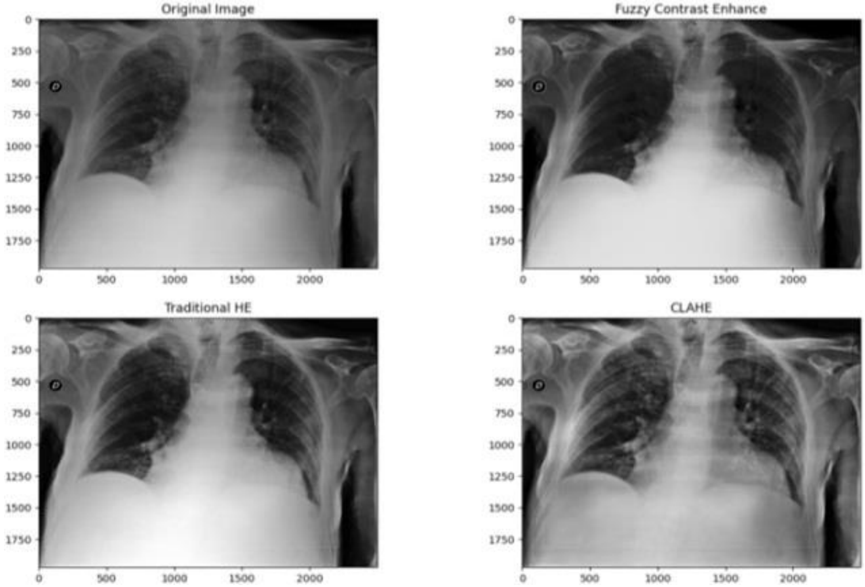


Figure 1: Comparison of different image enhancement techniques

Thus in our suggested framework, histogram equalization was chosen as the image-enhancing method. Our second step is to split the dataset. The dataset of CXR images is divided into two groups. 80% of the CXR images (5061 images) are used for training the proposed model, and 20% (1265 images) are reserved for validating the model. The images in the dataset originally had different sizes (512 × 512) pixels. To make them uniform and suitable for deep learning, they are resized to a smaller size of 112 × 112 pixels. This resizing step ensures that all images are in the same format and size, making it easier for the model to analyze them. Resizing techniques' underlying theory is covered in [15].

Methodology:

In this section, we will discuss the methodology of the proposed network. We started with the VGG-19 model, which is CNN architecture, designed for image classification. VGG-19 is a well-known CNN architecture with 19 layers, consisting of 16 convo1utional layers followed by 3 fully connected layers. The model VGG-19 is known for its effective feature extraction. It uses 16 convolutional layers organized into 5 groups. After each group of convolutional layers, there is a max-pooling layer. These convolutional layers are designed to capture the important features from the input images. They use (3, 3) filters with Rectified Linear Units (ReLUs) as activation function. Max-pooling is employed with a (2, 2) kernel and a stride of 2 pixels for downscaling.

We have changed some layers in VGG-19 architecture to adapt it to our X-ray image classification task. One of the convolutional layers in the VGG-19 was replaced with a dropout layer and also skipped one or more than one convolutional layer in each group. Dropout is a regularization technique that helps to prevent overfitting by randomly deactivating a fraction of neurons during training. We added one more group for feature extraction, making a total of six groups, each one followed by a max-pooling layer. These layers are responsible for extracting important features from the input images. In the first layer of our modified model, we used a 2D convolution operation with a (5, 5) kernel size, applying the same padding, followed by max-pooling with a (2, 2) kernel for reducing the spatial dimensions. ReLU activation function and dropout are applied in this layer. We continued to build the model with similar convolutional layers for feature extraction. The final layer in the features extraction process uses a 2D convolution with a (3, 3) kernel size, (2, 2) stride, and additional (2, 2) max-pooling which is shown in Figure 2.

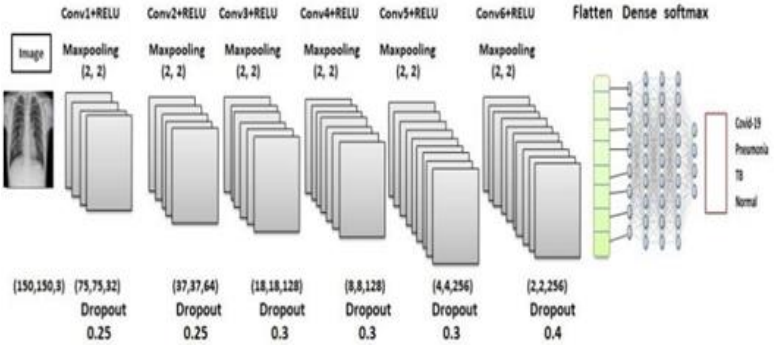
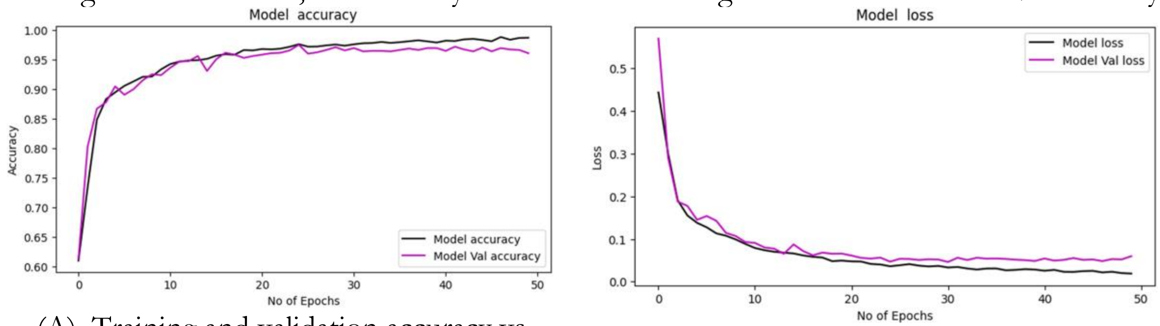


Figure 2: Architecture of the proposed model

After these convolutional and max-pooling layers, we used a flattened layer to convert the extracted features into a one-dimensional vector. Following the flattened layer, weadded dense (fully connected) layers for the classification task. The first dense layer has 1024 features and the final dense layer has only 4 features, indicating the number of classes in our X-ray image classification. Following the last dense layer, there is a softmax layer with the same number of outputs as the classes. The softmax layer computes class probabilities. In this work, categorica1 cross-entropy is used as the loss function, which is common for multi-class classification problems. We specified a learning rate of 0.00150 for the optimizer. The learning rate controls how quickly the model adjusts its weights during training. Hyperparameter tuning was performed to optimizeparameters like step size, kernel size, number of dropouts, and number of channels. We obtained results as output inthe output layer and achieved optimal accuracy in CXRimage classification.

Experimental Results:

In this study, we classified CXR images using a CNN-modified model. Adamax optimizer is used to compile the CNN model after it has been constructed. The experimentalresults of the proposed model are compared with the existing mode1 [13] and also with other state-of-the-art architectures. It is a very potent optimization method for deep learning networks. The suggested model, which utilized the Adamax optimizer with a learning rate of 0.00150, successfully classified CXR images and obtained 98.51% accuracy.



(A). Training and validation accuracy vs epochs

(B). Training and validation loss vs epochs

Figure 3: Training and validation loss and accuracy

Our study's main training goal is to reduce the averageprobability error between each pixel's anticipated and the actual values for CXR images. This was accomplished by using categorical cross-entropy as the loss function [16]. This method works well for multi-class classification problems such as our CXR image classification, as it penalizes discrepancies between predicted and true classlabels, making it easier to optimize the CNN model's parameters. With the help of this loss function, which efficiently measures the difference between the ground truthlabels and the projected probability distributions, the network can iteratively modify its weights and biases in order to reduce this difference over the course of subsequent training epochs. This progression across the 50 epochs is graphically depicted in Figure 3, which provides insights into the convergence behavior and generalizability of the model tonew data. We can evaluate the learning dynamics of the proposed CNN model, to spot any overfitting or underfitting scenarios and adjust hyperparameters to maximize performance by examining the trends shown in the graphical output of the model.

Confusion Matrix:

A confusion matrix is used as a fundamental tool in classification algorithms. In essence, it's a square matrix that can handle several classes; normally, it's 2 × 2 in size for binary classification issues. The results of aclassification task are concisely summarized in this matrix, where the rows represent the actua1 class 1abels and the columns represent the anticipated class 1abes. Figure 4 shows the Confusion Matrix of the proposed network. The four fundamental values within the confusion matrix are as follows:

True Positive (TP): This describes situations in which both the expected class and the actual class are positive. For example, in the medical domain, this might mean accurately diagnosing patients with a certain illness.

True Negative (TN): In these cases, the predicted and actual classes are both negative. In the context of medical diagnostics, this could mean accurately identifying people who do not have a specific illness.

False Positive (FP): This is also referred to as a Type I errorand happens when the actual class is negative but the projected class is positive.

False Negative (FN): When the expected class is negative but the actua1 class is positive, this results in a false negative (Type II mistake). This could indicate, in the context of healthcare applications, that a patient was not correctly diagnosed with a condition.

Researchers can assess the accuracy, precision, recall, and other performance parameters of the classification algorithm by examining these four variables, which offer crucial insights into the algorithm's operation. The distribution of examples inside the confusion matrix can be analyzed to spot misclassification trends and gauge how well the algorithm performs generally in differentiating between classes. As a result, the confusion matrix is an essential analytical tool for evaluating the advantages and disadvantages of classification algorithms, which helps to improve and optimize them for a range of practical uses.

Precision:

In the context of classification, precision is a metric that calculates the proportion of all cases that were correctly anticipated to be positive, that a classifier predicts as positive. It is calculated as using Eq. (1)

$$\text{Precision} = \frac{TP}{TP+} \qquad (1)$$

Recal1 (Sensitivity or true positive rate): In the context of classification, Recall is a performance metric that measures how well the classifier identified and retrieved all instances in the dataset that belong to a particular class. It is calculatedas using Eq. (2):

$$\text{Recall or true positive rate} = \frac{TP}{TP+FN} \qquad (2)$$

F1-Score: The Fl-score is the harmonic mean of precision and recal1. This provides a fair evaluation of a classifier's performance by taking into account both precision and recall. It is calculated as using Eq. (3):

$$\text{F1-score} = \frac{2\,(Precision\ x\ Recall)}{Precision + Recall} = \frac{2TP}{2TP + Fp + FN} \qquad (3)$$

The overall training performance of the proposed model derived from the confusion matrix is presented in Table. 1 while the validation performance is shown in Table. 2. Table. 3 shows the comparative Analysis of the proposed network with the other deep learning networks performing classification of TB, pneumonia, COVID-19, and healthy.

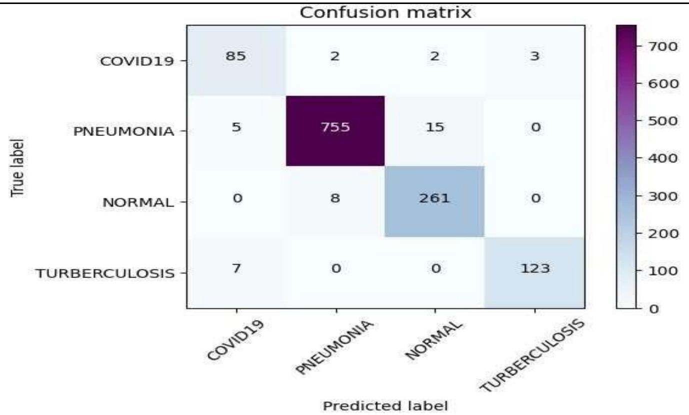


Figure 4: Confusion matrix

Table 1. Training results of the proposed CNN modification method

Class Name	Precision	Recall	F1-Score	Support
Covid-19	0.98	0.98	0.98	368
Pneumonia	1.00	0.98	0.99	3100
Normal	0.94	1.00	0.97	1072
Tuberculosis	1.00	0.98	0.99	520

Table 2. Validation results of proposed CNN modification method.

Class Name	Precision	Recall	F1-Score	Support
Covid-19	0.88	0.92	0.90	92
Pneumonia	0.99	0.95	0.97	775
Normal	0.88	1.00	0.94	269
Tuberculosis	0.98	0.92	0.95	130

Table 3. Comparative analysis of the proposed network with other deep learning networks performing classification of pneumonia, TB, COVID-19, and health.

Model	Precision (%)	Recall (%)	AUC-ROC(%)	F1(%)
Proposed Model	98	98.5	98.51	98.25
DenResCov-19	82.90	69.7	95.00	75.75
Dense Net-121	79.35	62.70	91.00	70.07
Res Net-50	78.60	62.00	93.21	69.51

Conclusion:

In the field of medical science, classifying CXR images is crucial. This research project aims to create a new type of computer algorithm called a Convolutional Neural Network (CNN) that can distinguish between four different categories of CXR images: Pneumonia, Covid-19, TB, and Normal. The process starts with preparing the images by making them a consistent size and reducing noise. Then, the dataset is divided into two parts: one part for training the algorithm and another part for validating its performance. After that, the CNN is used to automatically categorize the CXR images. Impressively, the model achieved a high accuracy rate of 98.51% in experimental tests using the CXR image dataset. Looking ahead, future research should aim to find new and innovative ways to classify CXR images in the medical field. This work will contribute to efforts to investigate the potential of artificial intelligence (AI) in the future. Additionally, it contains a number of research avenues that could be pursued in the future that are pertinent to the study of medical sciences.

References:

[1] J. E. Luján-García, Y. Villuendas-Rey, I. López-Yáñez, O. Camacho-Nieto, and C. Yáñez-Márquez, "NanoChest-Net: A Simple Convolutional Network for Radiological Studies Classification,"

Diagnostics 2021, Vol. 11, Page 775, vol. 11, no. 5, p. 775, Apr. 2021, doi: 10.3390/DIAGNOSTICS11050775.

[2] J. W. Song et al., "Omics-Driven Systems Interrogation of Metabolic Dysregulation in COVID-19 Pathogenesis," Cell Metab., vol. 32, no. 2, pp. 188-202.e5, Aug. 2020, doi: 10.1016/J.CMET.2020.06.016.

[3] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review," Chaos, Solitons & Fractals, vol. 139, p. 110059, Oct. 2020, doi: 10.1016/J.CHAOS.2020.110059.

[4] D. Das, K. C. Santosh, and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest X-rays," Phys. Eng. Sci. Med., vol. 43, no. 3, pp. 915–925, Sep. 2020, doi: 10.1007/S13246-020-00888-X/TABLES/6.

[5] L. Sarker, M. M. Islam, T. Hannan, and Z. Ahmed, "COVID-DenseNet: A Deep Learning Architecture to Detect COVID-19 from Chest Radiology Images," May 2020, doi: 10.20944/PREPRINTS202005.0151.V1.

[6] K. Li et al., "CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19)," Eur. Radiol., vol. 30, no. 8, pp. 4407–4416, Aug. 2020, doi: 10.1007/S00330-020-06817-6/FIGURES/5.

[7] J. Chen et al., "Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: a prospective study," medRxiv, p. 2020.02.25.20021568, Mar. 2020, doi: 10.1101/2020.02.25.20021568.

[8] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach," Measurement, vol. 145, pp. 511–518, Oct. 2019, doi: 10.1016/J.MEASUREMENT.2019.05.076.

[9] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images," Informatics Med. Unlocked, vol. 20, p. 100391, Jan. 2020, doi: 10.1016/J.IMU.2020.100391.

[10] R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. Silla, and Y. M. G. Costa, "COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios," Comput. Methods Programs Biomed., vol. 194, p. 105532, Oct. 2020, doi: 10.1016/J.CMPB.2020.105532.

[11] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," Comput. Biol. Med., vol. 121, p. 103792, Jun. 2020, doi: 10.1016/J.COMPBIOMED.2020.103792.

[12] T. Rahman et al., "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray," Appl. Sci. 2020, Vol. 10, Page 3233, vol. 10, no. 9, p. 3233, May 2020, doi: 10.3390/APP10093233.

[13] M. Mamalakis et al., "DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays," Comput. Med. Imaging Graph., vol. 94, p. 102008, Dec. 2021, doi: 10.1016/J.COMPMEDIMAG.2021.102008.

[14] K. Verma, G. Sikka, A. Swaraj, S. Kumar, and A. Kumar, "Classification of COVID-19 on Chest X-Ray Images Using Deep Learning Model with Histogram Equalization and Lung Segmentation," SN Comput. Sci., vol. 5, no. 4, pp. 1–15, Apr. 2024, doi: 10.1007/S42979-024-02695-7/METRICS.

[15] "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation | IEEE Conference Publication | IEEE Xplore." Accessed: May 05, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/6909475

[16] "Binary cross-entropy with deep learning technique for image classification." Accessed: May 05, 2024. [Online]. Available: https://www.warse.org/IJATCSE/static/pdf/file/ijatcse175942020.pdf

Skin Scan: Cutting-edge AI-Powered Skin Cancer Classification App for Early Diagnosis and Prevention

Maria Sial[1], Salman Shakeel[1], Muhammad Asim[1], Amaad Khalil[1] Muhammad Abeer Irfan[1], Atif Jan[2]

[1]Departement of Computer Systems Engineering University of Engineering and Technology, Peshawar

[2]Department of Electrical Engineering University of Engineering and Technology, Peshawar

***Correspondence**:20pwcse1910@uetpeshawar.edu.pk, 20pwcse1925@uetpeshawar.edu.pk, 20pwcse1944@uetpeshawar.edu.pk, amaadkhalil@uetpeshawar.edu.com, abeer.irfan@uetpeshawar.edu.pk, atifjan@uetpeshawar.edu.pk

Citation | Sial. M, Shakeel. S, Asim. M, Khalil. A, Irfan. M. A, Jan. A, "Skin Scan: Cutting-edge AI-Powered Skin Cancer Classification App for Early Diagnosis and Prevention", IJIST, Special Issue pp 227-235, May 2024

Received | May 03, 2024, **Revised |** May 13, 2024, **Accepted |** May 22, 2024, **Published |** May 27, 2024.

Mobile health applications (mHealth) use machine learning (AI)-based algorithms to classify skin lesions; nevertheless, the influence on healthcare systems is unknown. In 2019, a large Dutch health insurance provider provided 2.2 million people with free mHealth software for skin cancer screening. To evaluate the effects on dermatological care consumption, the research conducted a practical transitional and population-based study. To evaluate dermatological needs between the two groups throughout the first year of free access, the research compared 18,960 mHealth users who completed at least one successful evaluation with the app to 56,880 controls who did not use the app. The odds ratios (OR) were then computed. A cost-effectiveness analysis was conducted in the near term to find out the expense for each extra-diagnosed premalignancy. Here, results indicate that mHealth users had a three-fold greater incidence of requests for benign tumors on the skin and the nevi (5.9% vs 1.7%, OR 3.7 (95% CI 3.4–4.1)), and they had greater numbers of claims for (pre)malignant skin cancers as groups (6.0% vs 4.6%, OR 1.3 (95% CI 1.2– 1.4)). Compared to the existing standard of care, the expenses associated with using the app to detect one additional (pre) malignant skin lesion were €2567. These results suggest that AI in m Health may help identify more dermatological (pre)malignancies, but this could be weighed against the current greater rise in the need for care for benign tumors of the skin and nevi.

Keywords: Skin Cancer; AI-Powered; Skin Lesions; Skin Cancer Types (Basal Cell Carcinoma, Squamous Cell Carcinoma, Melanoma); Image Classification.

Introduction:

One of the most popular forms of cancer, skin cancer is becoming more common and more widespread, which puts considerable pressure on healthcare systems [1][2][3][4] Medical technology advancements like teledermatology and artificial intelligence (AI)-powered mobile health apps are being incorporated into medical treatments as a potential way to reduce this load. AI application could relieve stress on physicians and further minimize associated healthcare expenses by reducing the number of appointments for benign skin cancers and boosting the possibility of early identification of skin cancer [5].

The idea of using AI-based algorithms to identify skin cancer has a lot of assurance, but it has only been studied in controlled laboratory environments. Furthermore, even though AI is comparable to Dermatologists can identify cancers of the skin on dermoscopy-based images [6][7][8], but it's not yet clear how or for whom to use this technique in clinical care. The public can now use this method because AI-based algorithms for identifying skin cancer have been used in several mHealth apps in recent years [9]. The Netherlands is in a unique situation because of the quick advancement of AI-based mobile health apps in population-based settings. A mobile app for skin care is being paid for by multiple major health insurance carriers. Cancer detection for their consumers [10], allows consumers to assess whether to see a GP (general practitioner) for a potentially malignant skin lesion using an AI-based mHealth app. By evaluating how these real-world data affect doctors and their patients when they utilize such an app, we can gain a better understanding of the potential impact on healthcare consumption [11]. Thus, the purpose of this study is to assess how a mHealth app for questionable skin lesions affects the use of dermatological healthcare in a population-based scenario.

Related Work:

Skin cancer is a global issue affecting one in five people, with millions of new cases reported annually. It occurs when normal cells grow uncontrolled and disorderly, leading to contact inhibition of proliferation. There are two main categories of skin cancer: non-melanoma and melanoma. Melanoma skin cancer is further divided into two subtypes: squamous cell carcinoma (SCC) and basal cell carcinoma (BCC). Deep learning algorithms have become popular for early skin cancer detection. Tahir et al. introduce the DSCC_Net, a deep learning-based skin cancer classification network, aiming to identify different types of skin cancer. The model is tested on three standard datasets and provides a robust evaluation framework for six deep networks. DSCC_Net outperforms other models in skin cancer diagnosis, achieving a 99.43% AUC and outperforming baseline models in accuracy, recall, precision, and F1-score, making it a valuable tool for dermatologists and healthcare professionals. [12] Maad M. Mijwil's paper "Skin cancer disease images classification using deep learning solutions" focuses on image classification using deep learning models, which is a fundamental pillar of medical progress. The study uses a convolutional neural network (ConvNet) to detect skin cancer pictures, covering over 24,000 instances. Three architectures—InceptionV3, ResNet, and VGG19— are carefully used to identify the most effective architecture for the classification of benign and malignant cancer types with high accuracy. The research methodology focuses on artificial intelligence, specifically machine, and deep learning, with self-learning algorithms forming the foundation of artificial neural networks. TensorFlow, a flexible open-source library, is used for machine and deep learning tasks. The study uses a large dataset of high-pixel images from the ISIC collection from 2019-2020. The authors propose an enhanced framework for early skin cancer detection using the well-known CNN-based architecture, VGG-16. The model operates on the foundation of the VGG-16 network but introduces enhancements to improve accuracy. The International Skin Image Collaboration dataset is used for assessment. [13].

The paper "Detection and optimization of skin cancer using deep learning" by Balambigai, Elavarasi, Abarna, Abinaya, and Arun Vignesh focuses on improving Convolutional Neural Network (CNN) models for skin cancer classification. The authors used a dataset from

Kaggle and applied random search optimization for hyper-parameter selection, resulting in an improved accuracy of 77.17%. In the article "Classification of Skin Cancer Lesions Using Explainable Deep Learning," Muhammad Zia Ur Rehman introduced a unique technique by incorporating extra convolutional layers into pre-trained models. The modified DenseNet201 model showed a remarkable accuracy of 95.50%, demonstrating state-of-the-art performance compared to other approaches. The study highlights the importance of computer-aided diagnostic solutions in enhancing the detection process and supporting dermatologists in making informed decisions for early intervention and better patient outcomes.[14]

Types of Skin Cancer:

Skin cancer can be of three main forms. Melanoma, squamous cell carcinoma, and basal cell carcinoma are Thames of these varieties.

Basal Cell Carcinoma:

This is the type of skin cancer that occurs in the Basel cells. Basal cells are present in the lower part of the **outer layer** called the epidermis. On the skin, basal cell carcinoma appears as a tiny, usually shiny bump or scaly, flat patch that slowly expands which we can see in Figure 1.

Squamous Cell Carcinoma:

Squamous cell carcinoma (SCC), also known as cutaneous carcinoma of squamous cells (CSCC), is the second most common skin cancer which we can see in Figure 2, primarily affecting the **outermost layer** of the skin, often found in areas exposed to the sun.

Melanoma:

Melanoma, also known as "Black Tumor," is a dangerous skin cancer caused by melanocytes, which produce **melanin**, a black substance that gives skin shade. It spreads easily and expands quickly in Figure 1.

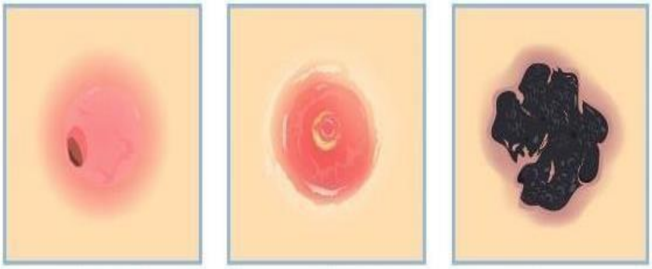


Figure 1: Main Types of Skin Cancer (Basel Cell Carcinoma, Squamous Cell Carcinoma, Melanoma)

AI algorithms are being developed to improve early skin cancer diagnosis and user experience. The **Skin Scan app** will be tested for accessibility, engagement, and impact on preventative healthcare measures. Ethical considerations like user consent and data privacy will be considered when releasing AI-powered health apps. The "Skin Scan" project combines advanced technology and user-centered design to improve skin cancer prevention and detection. It uses artificial intelligence algorithms, machine learning models, and ethical guidelines to ensure early diagnosis and tracking of health outcomes.

Methodology:

The proposed skin cancer detection system consists of the following features.

User Login/Register:

Doctors and patients can start by creating an account and entering basic information like their username and password. Users returning can safely log in. The system confirms user identity with a strong authentication procedure, ensuring allowed access to Skin Scan features.

Skin Scan Interface:

Skin Scan interface offers skin cancer screening services, including scanning body areas, viewing results, learning about forms, taking pictures, uploading, and providing details on different types of skin cancer.

Selecting Body Part:

Users select a body area for skin cancer analysis using their device's camera. They can capture or upload images, which are preprocessed for analysis. These images are then imported into the Skin Scan program for further examination.

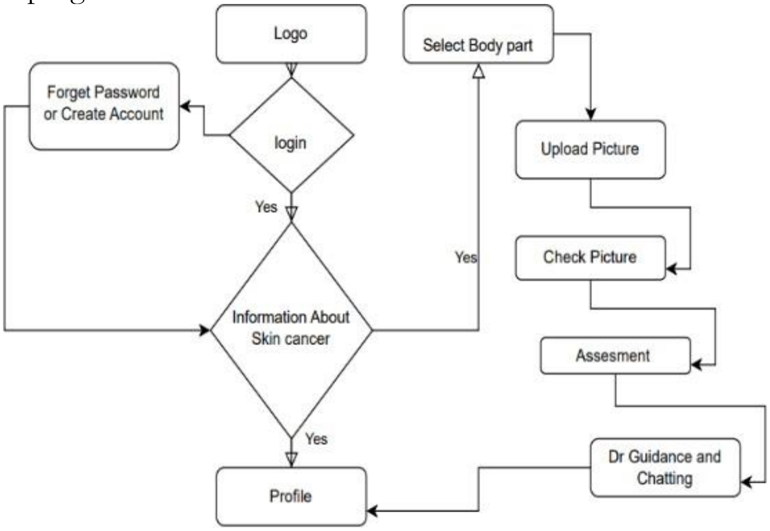


Figure 2: Skin Scan Flow Diagram

View Result:

Users obtain the results of a skin scan, which may include details regarding anomalies found, potential forms of cancer of the skin, and suggested courses of action. To help users comprehend scan results, the system may indicate areas of concern on uploaded photos using remarks or visualizations and the flow diagram can be seen in Figure 2. This figure explains the functionality of the overall proposed system for skin cancer.

Connecting Patients and Doctors:

The application's chat feature facilitates easy communication between doctors and patients, promoting a patient-centered approach to healthcare delivery, enhancing access, and empowering patients. Patients can easily communicate with the doctor for further guidance.

Model Architecture:

In our investigation, we utilized an enhanced version of the Swin Transformer architecture. The process of fine-tuning involves retraining a pre-existing model on a specific dataset or task, thereby enhancing its effectiveness for that particular objective. The model classifies various dermatological conditions, including actinic keratoses, basal cell carcinoma, benign keratosis-like lesions, dermatofibroma, melanocytic nevi, melanoma, and vascular lesions which we can also see in Figure 4, aiding early diagnosis and treatment planning.

Architecture of Swin Transformers:

With a linear computing complexity resulting from self-attention processing within each local window, the Swin Transformer is a Vision Transformer that may be used for image identification and classification tasks. It builds hierarchical feature maps by combining picture areas in layers whose classification can be seen in Figure 3.

Data Set:

The HAM10000 ("Human Against Machine with 10000 training images") dataset, consisting of 10,015 images, aims to address the scarcity of dermatoscopic images for the automated diagnosis of pigmented skin lesions. It covers diagnostic categories like Actinic keratoses, Basal cell carcinoma, benign keratosis-like lesions, dermatofibroma, melanoma, and vascular lesions which can be seen in figure 4 with all types of skin cancer.

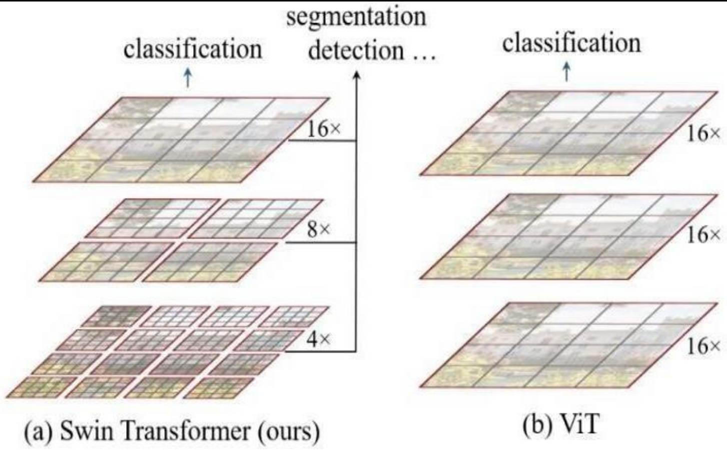


Figure3: Swin Transformer

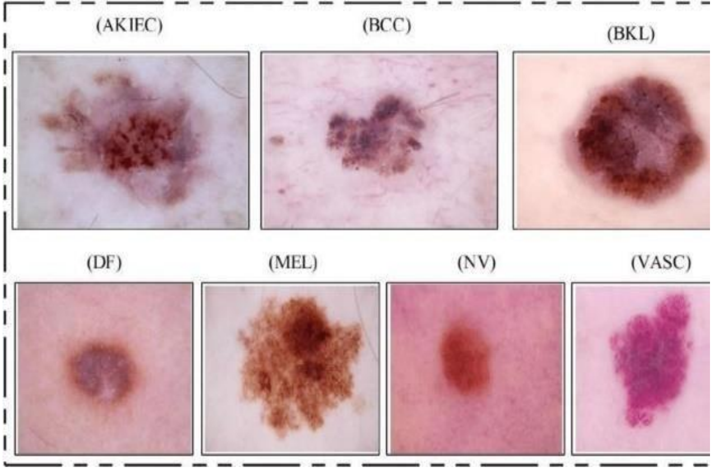


Figure 4: Types of Skin Cancer

Training Procedure:

We carefully chose and adjusted several hyperparameters during the training process to maximize the effectiveness of our skin cancer classification model

Training Hyperparameters:

Learning Rate:

The number of steps taken during optimization to update the model's parameters is determined by the learning rate which is 5e-05 in this case.

$$\text{New Weight}=\text{Old Weight}-\text{Learning Rate} \times \text{Gradient} \quad (1)$$

Train Batch Size:

The number of training examples used in each training cycle is determined by the batch size. The train batch size of this model is **32**.

Eval Batch Size:

It determines the number of examples utilized for evaluation in each iteration, much as the train batch size. The Eval batch size of this model is 32.

Seed:

For reproducibility, the random number generator is initialized with a seed in Figure 4, which is a random number. The seed in this model is **42**.

Gradient Accumulation Steps:

In advance of changing the model's parameters, gradient accumulation permits gradients to build across several batches. The value is 4.

Total Train Batch Size:

The product of train batch size and gradient accumulation steps yields the effective batch size that is used for parameter updates. The total train batch size is 128.

Train Batch Size × Gradient Accumulation Steps = Total Train Batch Size (2)

Table 1: Training Results & Accuracy of Swin Transformer model

Training Result & Accuracy of Model	
Training Loss	0.3261
Epoch	1.0
Step	3.30
Validation Loss	0.2744
Accuracy	0.8984
Precision	0.9030
F1	0.8985
AUC-Roc Curve	0.9935

Optimizer:

The Adam optimizer, with betas = (0.9, 0.999) and epsilon=1e-08, uses adaptive methods to adjust learning rates and momentum, ensuring stability during parameter updates.

Learning Rate Scheduler:

Type: A linear learning rate scheduler adjusts the learning rate linearly during training epochs.

Learning Rate Scheduler Warmup Ratio:

The value of 0.1 indicates that **10%** of the epochs are allocated for the learning rate warmup.

Number of Epochs:

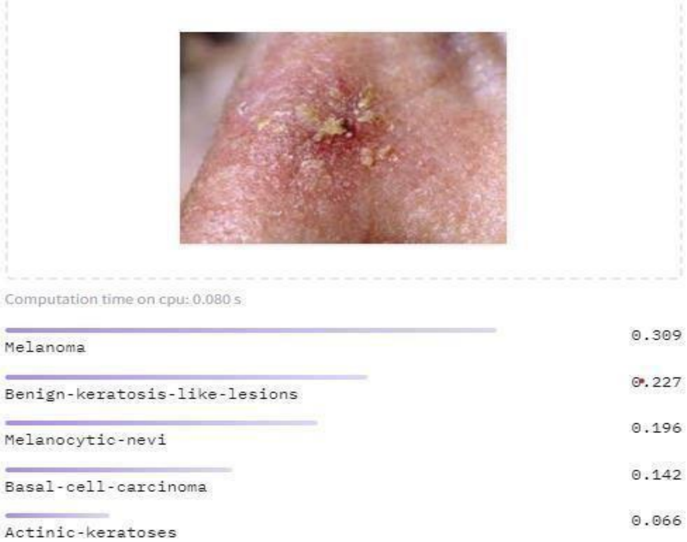
The model is trained for one epoch, completing one pass through the entire training dataset. In our proposed system the training loss after one epoch is 0.3261, while the validation loss is 0.2744 as shown in table 1. The training method achieved an accuracy of 89.84%, precision of 90.30%, F1 score of 89.85%, and Area Under the Curve (AUC ROC) of 99.35%. Additionally, it implies that the value at the given stage is 3.30 as given in Table 1. These indicators collectively reflect the model's performance and efficacy after the stipulated training period, providing information about its capacity to make accurate predictions and manage the given data.

Working:

The application features a logo screen, a login screen, and password reset options. It raises consumer awareness about skin cancer and provides access to important information. Users can create accounts, customize profiles, and upload pictures for skin assessment. They can preview submitted photos before starting the assessment via the Check Picture screen in Figure 2. The Skin Scan program offers a user-friendly interface with distinct functions, allowing users to create accounts, access instructional materials, upload photos, and receive detailed skin scan findings after processing supplied photographs.

Results:

Presenting Skin Scan, a cutting-edge AI-powered software for prevention and early detection of skin cancer. We have reached significant goals through thorough research and testing, proving the app's performance in precisely detecting possible skin problems. The artificial intelligence model utilized in Skin Scan has exhibited remarkable efficiency, attaining elevated levels of precision in identifying skin cancer in various datasets. The measures for accuracy, recall, and F1-score consistently show how reliable our model is in differentiating between various kinds of skin lesions. Early feedback from users and engagement data demonstrate how well-liked Skin Scan's functionality and user experience is. Users value the app's ability to raise awareness and its simplicity of use, as well as its clear and instructional content regarding skin cancer. One of the main concerns with Skin Scan has been its picture processing pipeline efficiency which can be seen in Figures 5 and 6, we have provided it with images, and it has shown us the results. Our findings demonstrate that the program offers quick analysis, guaranteeing a smooth user experience. The app's usefulness for daily use is enhanced by its real-time processing capabilities. The image was classified using a model, predicting it to be a Melanoma or Melanocytic-nevi, with lower confidence values for other classifications, aiding in skin lesion diagnosis in Figures 5 and 6, based on the precision and accuracy results we conclude that this is the desired type of cancer.



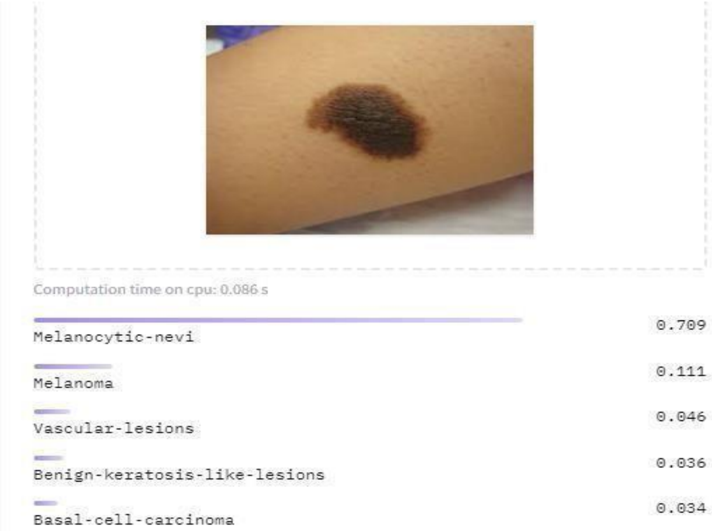
```
Computation time on cpu: 0.080 s

Melanoma                                   0.309
Benign-keratosis-like-lesions              0.227
Melanocytic-nevi                           0.196
Basal-cell-carcinoma                       0.142
Actinic-keratoses                          0.066
```

Figure 5: Skin Cancer Classification Output

Discussion:

There was a 32% rise in reports for (pre)malignant lesions of the skin among users of the app in comparison to a comparable number of those who avoided using the mHealth app. However, a three- to four-fold increased likelihood of claims for benign skin cancers and nevi among mHealth users also offset this effect. Based on the previously published diagnostic precision of the analyzed app [15][16], these results were anticipated. Additionally, they align with other highly recognized population-based cancer screening programs that strike a balance between accurately diagnosing malignancies and producing false positive results [17], as well as contemporary clinical dermatological practice, which excises about 8 nevi for every melanoma [18].



```
Computation time on cpu: 0.086 s

Melanocytic-nevi                           0.709
Melanoma                                   0.111
Vascular-lesions                           0.046
Benign-keratosis-like-lesions              0.036
Basal-cell-carcinoma                       0.034
```

Figure 6: Classification Results for Skin Cancer

Using a mHealth app could be an option even though traditional skin cancer screening according to a full body check by a qualified healthcare provider is not advised [19]. a stage in between to think about focused screening for high-risk lesions. According to this study, using the app was associated with a rise in (pre)malignancy claims; as a result, it may be a useful first step in enhancing skin cancer detection. But as it stands, the app can now detect any cutaneous (pre)malignancies, including actinic keratosis, keratinocyte carcinomas, and melanomas. The morbidities and fatalities of these (pre)malignancies vary greatly [20][21][22][23], and early diagnosis of cutaneous pre-malignancies such as actinic keratosis is medically less significant. The incorrect diagnosis and hence inefficient use of limited healthcare resources could be a major drawback of implementing these kinds of apps across the population [24].

Conclusion:

The SkinScan app, using the Swin Transformer model, is a groundbreaking solution for early skin cancer detection and prevention. With an accuracy rate of 0.89, it facilitates user-friendly skin examinations, enabling timely medical interventions and potentially life-saving treatments. SkinScan also raises awareness and promotes proactive health behaviors, potentially improving patient outcomes and reshaping skin cancer management.

References:

[1] M. huidkankerrapport I. Schreuder, K., de Groot, J., Hollestein, L. M., Louwman, “No Title”, [Online]. Available: https://iknl.nl/nieuws/2019/steedsvaker-huidkanker,- nationaal-plan-nodig (2019)

[2] S. Tokez, L. Hollestein, M. Louwman, T. Nijsten, and M. Wakkee, “Incidence of Multiple vs First Cutaneous Squamous Cell Carcinoma on a Nationwide Scale and Estimation of Future Incidences of Cutaneous Squamous Cell Carcinoma,” JAMA Dermatology, vol. 156, no. 12, pp. 1300–1306, Dec. 2020, doi: 10.1001/JAMADERMATOL.2020.3677.

[3] A. Lomas, J. Leonardi-Bee, and F. Bath-Hextall, “A systematic review of worldwide incidence of nonmelanoma skin cancer,” Br. J. Dermatol., vol. 166, no. 5, pp. 1069–1080, May 2012, doi: 10.1111/J.1365-2133.2012.10830.X.

[4] S. T. Chen, A. C. Geller, and H. Tsao, “Update on the Epidemiology of Melanoma,” Curr. Dermatol. Rep., vol. 2, no. 1, pp. 24–34, Mar. 2013, doi: 10.1007/S13671-012-0035-5.

[5] M. Janda and H. P. Soyer, “Can clinical decision making be enhanced by artificial intelligence?,” Br. J. Dermatol., vol. 180, no. 2, pp. 247–248, Feb. 2019, doi: 10.1111/BJD.17110.

[6] P. Tschandl et al., “Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study,” Lancet Oncol., vol. 20, no. 7, pp. 938–947, Jul. 2019, doi: 10.1016/S1470-2045(19)30333-X.

[7] A. Esteva et al., “Dermatologist-level classification of skin cancer with deep neural networks,” Nat. 2017 5427639, vol. 542, no. 7639, pp. 115–118, Jan. 2017, doi: 10.1038/nature21056.

[8] L. Oakden-Rayner, “Reply to ‘Man against machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists’ by Haenssle et al.,” Ann. Oncol., vol. 30, no. 5, p. 854, May 2019, doi: 10.1093/annonc/mdy519.

[9] “Algorithm based smartphone apps to assess risk of skin cancer in adults: systematic review of diagnostic accuracy studies,” BMJ, vol. 368, Feb. 2020, doi: 10.1136/BMJ.M645.

[10] “Sorry. De pagina die u bezoekt bestaat niet (meer) - CZ.” Accessed: May 04, 2024. [Online]. Available: https://www.cz.nl/404

[11] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol, “AI in health and medicine,” Nat. Med., vol. 28, no. 1, pp. 31–38, Jan. 2022, doi: 10.1038/S41591-021-01614-0.

[12] M. Tahir, A. Naeem, H. Malik, J. Tanveer, R. A. Naqvi, and S. W. Lee, “DSCC_Net: Multi-Classification Deep Learning Models for Diagnosing of Skin Cancer Using Dermoscopic Images,” Cancers 2023, Vol. 15, Page 2179, vol. 15, no. 7, p. 2179, Apr. 2023, doi: 10.3390/CANCERS15072179.

[13] M. M. Mijwil, "Skin cancer disease images classification using deep learning solutions," Multimed. Tools Appl., vol. 80, no. 17, pp. 26255–26271, Jul. 2021, doi: 10.1007/S11042-021-10952-7/METRICS.

[14] M. Zia Ur Rehman, F. Ahmed, S. A. Alsuhibany, S. S. Jamal, M. Zulfiqar Ali, and J. Ahmad, "Classification of Skin Cancer Lesions Using Explainable Deep Learning," Sensors 2022, Vol. 22, Page 6915, vol. 22, no. 18, p. 6915, Sep. 2022, doi: 10.3390/S22186915.

[15] A. Udrea et al., "Accuracy of a smartphone application for triage of skin lesions based on machine learning algorithms," J. Eur. Acad. Dermatology Venereol., vol. 34, no. 3, pp. 648–655, Mar. 2020, doi: 10.1111/JDV.15935.

[16] T. Sangers et al., "Validation of a Market-Approved Artificial Intelligence Mobile Health App for Skin Cancer Screening: A Prospective Multicenter Diagnostic Accuracy Study," Dermatology, vol. 238, no. 4, pp. 649–656, Jul. 2022, doi: 10.1159/000520474.

[17] G. B. Taksler, N. L. Keating, and M. B. Rothberg, "Implications of false-positive results for future cancer screenings," Cancer, vol. 124, no. 11, pp. 2390–2398, Jun. 2018, doi: 10.1002/CNCR.31271.

[18] K. C. Nelson, S. M. Swetter, K. Saboda, S. C. Chen, and C. Curiel-Lewandrowski, "Evaluation of the Number-Needed-to-Biopsy Metric for the Diagnosis of Cutaneous Melanoma: A Systematic Review and Meta-analysis," JAMA Dermatology, vol. 155, no. 10, pp. 1167–1174, Oct. 2019, doi: 10.1001/JAMADERMATOL.2019.1514.

[19] M. Johansson, J. Brodersen, P. C. Gøtzsche, and K. J. Jørgensen, "Screening for reducing morbidity and mortality in malignant melanoma," Cochrane Database Syst. Rev., vol. 2019, no. 6, Jun. 2019, doi: 10.1002/14651858.CD012352.PUB2/MEDIA/CDSR/CD012352/IMAGE_T/TCD012352-AFIG-FIG03.PNG.

[20] A. S. Adamson, E. A. Suarez, and H. G. Welch, "Estimating Overdiagnosis of Melanoma Using Trends Among Black and White Patients in the US," JAMA Dermatology, vol. 158, no. 4, pp. 426–431, Apr. 2022, doi: 10.1001/JAMADERMATOL.2022.0139.

[21] M. Boniol, P. Autier, and S. Gandini, "Melanoma mortality following skin cancer screening in Germany," BMJ Open, vol. 5, no. 9, p. e008158, Sep. 2015, doi: 10.1136/BMJOPEN-2015-008158.

[22] A. Stang and K. H. Jöckel, "Does skin cancer screening save lives? A detailed analysis of mortality time trends in Schleswig-Holstein and Germany," Cancer, vol. 122, no. 3, pp. 432–437, Feb. 2016, doi: 10.1002/CNCR.29755.

[23] H. G. Welch, B. L. Mazer, and A. S. Adamson, "The Rapid Rise in Cutaneous Melanoma Diagnoses," N. Engl. J. Med., vol. 384, no. 1, pp. 72–79, Jan. 2021, doi: 10.1056/NEJMSB2019760/SUPPL_FILE/NEJMSB2019760_DISCLOSURES.PDF.

[24] A. S. Adamson and H. G. Welch, "Machine Learning and the Cancer-Diagnosis Problem — No Gold Standard," N. Engl. J. Med., vol. 381, no. 24, pp. 2285–2287, Dec. 2019, doi: 10.1056/NEJMP1907407/SUPPL_FILE/NEJMP1907407_DISCLOSURES.PDF.