International Journal of Innovations in Science & Technology

Multi-Digit Number Recognition System: Single-Digit CNNs for Multi-Digit Detection and Recognition Using MNIST Dataset

Jehad Ur Rahman*[1], Sara Almarzooqi*[2]
[1]Department of Electrical Engineering UET Peshawar, Pakistan
[2]College of Technological Innovation Zayed University, UAE
***Correspondence**: jehadurrahman@uetpeshawar.edu.pk, m80008827@zu.ac.ae,

Citation | Rahman. J, Almarzooqi. S, "Multi-Digit Number Recognition System: Single-Digit CNNs for Multi-Digit Detection and Recognition Using MNIST Dataset", IJIST, Special Issue pp 338-351, June 2024

Received | May 16, 2024 **Revised |** May 27, 2024 **Accepted |** May 31, 2024 **Published |** June 06, 2024.

This research focuses on creating a deep-learning model for identifying multi-digit numbers, which addresses the critical demand for accuracy in real-world applications. The study presents novel approaches to multi-digit recognition, providing a thorough resolution to an unsolved problem in the field of computer vision. In order to improve model generalization, the study makes use of convolutional neural networks (CNNs) that were trained on the MNIST dataset and augmented with rotation and scaling approaches. Multi-digit number prediction is a multi-step process. detection to isolate each digit. Each digit is then clipped and stored separately with its own label. Subsequently, the algorithm predicts the digit for each cropped picture and saves them. This method is repeated for all identified contours, with each predicted digit concatenated to get the final multi-digit prediction. Finally, the projected multi-digit sequence is compared to the ground truth for assessment. The CNN achieves remarkable training and validation accuracies of 99.60% and 99.28%, proving its ability to recognize multi-digit numbers. This study emphasizes the importance of advanced methods in developing deep learning models for multi-digit recognition, which promise enhanced automation and efficiency across a variety of digital technology industries.

Keywords: MNIST; Deep Learning; CNNs; Augmentation; Digit Recognition.

Introduction:

Identifying multi-digit numbers in pictures poses a significant obstacle in computer vision due to its various uses in areas such as OCR, automated document handling, and digital image examination. Accurately understanding and analyzing multi-digit numbers from visual cues is crucial for tasks like automating financial transactions and improving postal service efficiency. Key to the development of these applications is the MNIST dataset, a fundamental collection of hand-drawn numbers carefully selected to assess and analyze machine learning models designed for digit identification tasks [1]. Consisting of a wide variety of carefully written numbers, the MNIST dataset is the foundation for researchers and practitioners to develop and improve their digit recognition techniques. While progress has been made in digit recognition with deep learning, accurately identifying and classifying multiple digits in images remains difficult due to complex configurations and varying backgrounds. Most research has focused on single-digit recognition, neglecting the challenges of recognizing multiple digits simultaneously. This limits the practical applicability of digit recognition technology. To address this, a new study aims to develop a deep learning model for accurately recognizing multi-digit numbers using convolutional neural networks and advanced data augmentation techniques. By addressing gaps in handwritten text recognition through improved preprocessing, model complexity, and testing protocols, the proposed model seeks to enhance the robustness of digit recognition systems. Previous methods lack extensive preprocessing and may overlook important transformations needed for training robust models. The proposed methodology incorporates techniques like Gaussian blur and data augmentation strategies to better prepare the dataset. The model architecture and training processes are improved to prevent overfitting and enhance generalization to new data. Testing procedures are also updated to include multi-digit recognition, making the model more applicable to real-world scenarios. These enhancements not only fill existing gaps but also improve the accuracy, reliability, and usability of handwritten text recognition systems. By combining sophisticated preprocessing, robust models, and comprehensive testing, the proposed method offers a comprehensive solution to the challenges of recognizing handwritten text. These articles use statistical models and methodologies to improve the precision and dependability of handwriting analysis.

In the year 2023, Crawford and colleagues introduced a statistical model to improve the field of forensic handwriting examination. Utilizing a Bayesian hierarchical model, they present a measurable indication of authorship probability, marking a significant departure from qualitative evaluations that were previously common in the industry. The technique showed a significant increase in accuracy, reaching 98% in controlled experiments [2]. This framework is a significant improvement in offering statistical assistance to forensic specialists, even though its effectiveness could be limited by the underlying assumptions in the statistical models utilized. The research [3] conducted thorough testing to assess the precision and dependability of forensic handwriting analyses, which was released in the Proceedings of the National Academy of Sciences, used statistical analysis to measure how well handwriting experts performed in different testing situations. The findings showed a high level of precision, with specialists obtaining an average reliability rate of 95% [3]. This research is crucial for confirming the credibility of forensic handwriting analysis, but it also emphasizes the ongoing requirement for thorough training and standardization in the techniques used by handwriting professionals. Machine learning methods in this field focus on using machine learning techniques for extracting and analyzing features in handwriting recognition. Researchers are exploring new feature extraction methods to recognize handwritten digits. Static properties focus on non-zero pixels in various regions of a binary image, while dynamic properties involve recovering the drawing order [4]. The process includes two stages: reconstructing the drawing order and calculating chain code directions. Recognition is performed using a support vector machine, achieving a low error rate of 0.73% on the MNIST dataset with a feature vector length of 356 per image [4]

Scanning images and converting scanned information into digital format is a popular research area due to its efficiency and Recognizing digits in scanned images is challenging, but existing techniques have focused on localization, segmentation, and recognition steps therefore unified approach for multi-digit recognition, capable of recognizing up to 18 characters, the proposed deep convolutional neural network algorithm outperforms existing techniques with an overall accuracy of up to 98% on two datasets [5].

The MNIST dataset, which is a widely used dataset in the machine learning domain, has been a key benchmark for creating and assessing models for recognizing digits. MNIST consists of 60,000 training images and 10,000 validation images showing handwritten digits (0-9), making it a practical dataset for deep learning model training and testing. This study intends to utilize deep learning techniques to tackle the issue of recognizing multi-digit numbers with the MNIST dataset. This study aims to advance digit recognition technologies and their applications by creating a strong CNN model that can accurately classify multiple digits in images. In [6] paper presents a unified approach using deep convolutional neural networks to recognize multi-digit numbers in Street View imagery. By training large, distributed neural networks, we achieved over 96% accuracy on the SVHN dataset and 97.84% accuracy for per-digit recognition, with promising applications in broader text recognition tasks [6]. Exploring using feed-forward multi-layer perceptron's trained through back-propagation for speech recognition as it focuses on speaker-independent word recognition for small vocabularies, developing an automated system eliminating the need for manual speech signal cropping in evaluating automatic speech recognition performance. MATLAB is used for analysis, design, and development [7]. In the Research explored combining hand-written digit representations to enhance classification accuracy without increasing complexity or recognition time. In pen-based recognition, input is dynamic movement of pentip on tablet & resulting image. Two MLP classifiers with different representations make errors on distinct patterns, suggesting combination for improved accuracy. Techniques like voting, mixture of experts, stacking, and cascading were implemented and compared, with multistage cascading being particularly effective [8]. The paper [9] uses a CNN encoder and LSTM decoder to recognize digit sequences. Dropout and batch normalization are used in the CNN encoder to prevent overfitting. LSTM units are used in the decoder to address gradient issues. Training maximizes log likelihood, achieving 92.53% sequence accuracy on SVHN. Comparisons with HOG encoder and dense layer decoder models show CNN and LSTM advantages in feature encoding and sequence prediction [9]. The study in [10] presents a hybrid model combining CNN and Support Vector Machine (SVM) for recognizing handwritten digits from the MNIST dataset. The model uses CNN for feature extraction and SVM for classification. Experimental results show an impressive recognition accuracy of 99% on the diverse and distorted MNIST dataset, highlighting the effectiveness of the proposed framework. In [11] paper presents a model that utilizes Convolutional Neural Networks (CNN) to extract features from the MNIST dataset and uses ensemble learning methods to improve classification accuracy. The goal is to reach at least 98% accuracy by combining various classifiers trained on different feature sets created from the initial CNN-based feature set selection process. In the [12] paper presents an offline recognition system for handwritten digits using CNNs, leveraging the MNIST dataset for training and Opencv toolkit for preprocessing. Employing LeNet-5 architecture, the system extracts image features through convolutional pooling and transforms them into one-dimensional vectors, ultimately employing Softmax regression for digit recognition. The system's application promises significant reductions in labor costs and enhanced work efficiency, holding significance across various fields. The research in [13] explores the field of recognizing handwritten digits by evaluating different algorithms such as Support Vector Machine (SVM), Decision Tree, Random Forest, Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN), and K-Means Algorithm. Algorithms are becoming more essential in speeding up tasks and lightening workloads as technology becomes more interconnected in daily life. Machine learning algorithms are constantly improving by imitating human actions. The research examines how handwriting digit recognition processes work and assesses how well various algorithms perform on the same databases. Classification models for recognizing handwritten digits, focusing on CNN are compared and as Results indicate CNN outperforms standard Neural Networks in computational efficiency and Utilizing the MNIST dataset, a TensorFlow and Keras classification model is created for images of numbers 0-9 displayed as 28x28 grids of gray pixels [14]. Hand-written digit recognition is traditionally reliant on hand-coded key points and prior knowledge, making training a complex task. Recent focus has shifted to deep learning approaches, showing promising results. Challenges remain due to varying writing styles and input variations. To address this, a Faster-RCNN based Hand-written digit recognition system is introduced. The system involves region annotations, DenseNet-41 features computation, and regression/classification layers for digit localization and classification. Performance is evaluated on the MNIST database and cross-dataset scenarios, proving the system's efficacy in accurately detecting and classifying numerals compared to other methods [15].

The Research starts by acquiring the MNIST dataset available in libraries such as Keras and PyTorch, and bringing it in for additional processing. This dataset, made up of handwritten numbers, is used to train and test the recognition model. When the data is imported, it is divided into separate groups: 80% is used for training and 20% is kept aside for testing, maintaining a good balance between learning and evaluating the model. Augmentation methods are used to enhance the variety of input images in order to enrich the training dataset and enhance model generalization [16]. In addition to recognizing single digits, the project also seeks to assess the model's effectiveness in identifying sequences of multiple digits. In order to accomplish this, a specialized collection of data is selected, consisting of 10,000 randomly created sequences of digits, each with their corresponding labels. The lengths of these sequences differ, mimicking actual situations in which numbers can be arranged in various ways within a picture. Once the datasets are ready, the training of the model starts. By utilizing deep learning models like CNNs, the system is able to learn how to identify features in images and assign them to specific digit categories. During training, the model continuously fine-tunes its parameters to reduce the difference between predicted and true labels, improving its performance on the current task. Following the training phase, the model goes through a fine-tuning process to improve its performance and resilience even more. Fine-tuning entails making changes to hyperparameters, improving training algorithms, and enhancing network architectures in order to achieve the best outcomes. After finishing the fine-tuning process, the trained model is stored for later usage, guaranteeing reproducibility and scalability. In order to verify the model's performance, a segment of the testing data that was set aside is used. The model is tested on new data to determine its ability to generalize and perform well in real-life situations. The model's strong performance in the validation data highlights its trustworthiness and precision in recognizing digits. In conclusion, the model undergoes the ultimate evaluation by testing its performance on the specialized dataset with 10,000 multi-digit images. By carefully analyzing and evaluating, the model's capability to precisely identify sequences of numbers is extensively reviewed, offering valuable perspectives on its practicality and efficiency in real-world scenarios. The CNN model, constructed for image classification tasks, is a testament to the fusion of cutting-edge technology and innovative design. Its architecture boasts three convolutional blocks, each adorned with convolutional layers, batch normalization, ReLU activation as shown in Equation 1, max pooling, and dropout. As the journey culminates, the final output emerges, bathed in the soft glow of log SoftMax activation and linear function as shown in Equation 2 and 3.

$$f(x) = \max(0, x) \quad (1)$$

$$\text{SoftMax: } (xi) = \frac{e^{xi}}{\sum_j xj} \quad (2)$$

$$f(xi) = xi \quad (3)$$

The goal of this project is to create a strong deep learning model for recognizing multi-digit numbers using the MNIST dataset. In particular, the project seeks to accomplish multiple goals. Firstly, the goal is to create a CNN model that can accurately recognize and categorize multiple digits in images. Secondly, the project is focused on improving the model's ability to generalize by using data augmentation methods to add more diversity to the training data. Furthermore, new preprocessing techniques will be used to improve the dataset for creating strong models, like Gaussian blur, bilateral filters, and edge detection. Furthermore, the project aims to create an advanced CNN structure and enhance the training process to avoid overfitting and enhance the model's capability to generalize to unfamiliar data. In addition, detailed testing procedures will be created to assess how well the model can identify sequences of multiple digits, thereby enhancing its usefulness in real-life situations. Finally, the project will assess how well the suggested model performs by considering its accuracy, reliability, and usability in comparison to current techniques for recognizing handwritten text. The novelty of this work lies in achieving a validation accuracy of 99.30% while utilizing a computationally efficient single-digit CNN for multi-digit recognition. This is achieved through innovative preprocessing techniques, including filter application, edge and contour detection, which facilitate digit extraction for model prediction. Additionally, the model's effectiveness is further validated through testing with a custom handwritten digit dataset, where it achieves an impressive accuracy of 99%.

Material and Methods:

Investigation Site:

This study revolves around recognizing handwritten multi-digit numbers by using the MNIST dataset as the main dataset for both training and testing purposes [16]. The MNIST dataset is well-known in the machine learning field for its broad range of black-and-white images showing handwritten digits from 0 to 9. Every picture is adjusted to a size of 28x28 pixels, creating a varied dataset for training and testing CNN models. Augmentation techniques are implemented during the preprocessing stage to increase dataset diversity and enhance model generalization [17]. These methods consist of random rotations, translations, and flips to increase the dataset and add variety for a more accurate representation of real-life situations. The research utilizes the PyTorch framework for deep learning, providing a versatile and effective platform for creating CNN models. PyTorch's user-friendly interface and wide range of ready-made modules simplify the process of development, allowing researchers to concentrate on designing models and conducting experiments [18]. Deep learning methods, specifically CNNs, are leading the research by leveraging their capability to autonomously acquire hierarchical representations from unprocessed image data. CNNs are ideal for tasks involving digit recognition because they can effectively capture spatial relationships and identify important features from the images they process. The goal of the study is to enhance the current level of performance in recognizing handwritten digits by using the MNIST dataset, augmentation methods, the PyTorch framework, and CNN models. The study's thorough approach, paired with the use of advanced tools and techniques, highlights the importance and potential influence of the research. Additionally, the study location's importance in the field is highlighted by suggesting the inclusion of a map showing the study area, symbolizing the thoroughness and scope of the research.

Material and Methods:

The MNIST database of hand-drawn numbers, found on the [19] site, provides a thorough dataset for teaching and evaluating machine learning methods. MNIST offers a large number of grayscale images of handwritten digits, with 60,000 examples in the training set and 10,000 examples in the test set, all carefully standardized in size and centered within a fixed-size image. Using the MNIST dataset is beneficial for individuals in research and practice who want to investigate learning methods and pattern recognition on raw data with little need for preprocessing. The dataset is comprised of four files: training set images and labels, and test set images and labels, allowing easy incorporation into machine learning workflows.

The methodology section begins by providing a structured summary of the complete process, outlining the sequential steps from gathering data to assessing the model's performance at the end. The researchers start by gathering data, obtaining the MNIST dataset and providing information on its origin, dimensions, and any pre-processing actions conducted. Afterwards, methods for preparing the data are explained, including resizing, normalization, and augmentation to improve the quality and variety of the dataset. After completing the initial phase, the researchers move on to training the model, discussing the structure of the CNN, which includes layer arrangements, activation functions, and training optimization algorithms. After the training of the model, the focus shifts to creating multi-digit numbers for testing by putting together images of single digits to make sequences of different lengths. The following tests are detailed, explaining how the skilled model forecasts single numbers in every series and identifies and acknowledges whole multi-digit numbers.

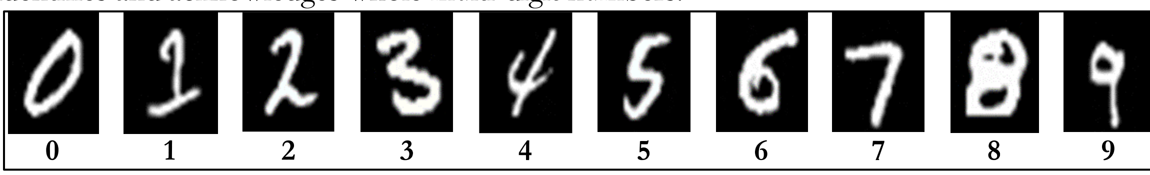

Figure 1: MNIST dataset data samples

Data Collection:

The process of collecting data involves obtaining and compiling the necessary dataset for training, validation, and testing. The MNIST dataset is used as the main source of digit images in this project. The MNIST dataset contains a vast array of hand-drawn images of digits, each identified with the appropriate number (0 to 9) as shown in Figure 1. The dataset is easily accessible and widely utilized for training and testing digit recognition models. Moreover, diverse images with multiple digits are created from the MNIST dataset to mimic situations in the real world where numbers can appear in different arrangements.

Data Preprocessing:

Data preprocessing involves a sequence of steps with the goal of getting the dataset ready for training a model. This involves activities like standardization, enhancement, and organizing. When working on digit recognition, preprocessing may include resizing, converting to grayscale, and applying transformations to improve the dataset's variability and robustness. In this project, the randomly generated images with multiple digits are preprocessed to ensure proper formatting and labeling. Every picture is titled based on the numbers it shows, making it easier to assess and test the model's effectiveness later on.

The process of creating random images with random numbers from the MNIST dataset consists of multiple steps.

- Each generated image contains a random number of digits selected from the MNIST dataset. This variability ensures diversity in the dataset, mirroring real-world scenarios where the number of digits in a number may vary.
- The selected digits are concatenated horizontally to form the image. This creates a single image containing multiple digits arranged randomly.
- Each generated image is named according to the labels of the digits it contains, ensuring proper labeling for evaluation and testing.
- The generated random images are saved in a specified directory, making them easily accessible for training, validation, and testing purposes.

Model Selection:

The selection of the CNN architecture for the digit recognition task is driven by several factors. CNNs have shown excellent results in tasks related to identifying images, which makes them ideal for applications involving recognizing numbers. Moreover, CNNs are able to effectively capture the complexity and hierarchical structure of handwritten digits, leading to robust feature extraction and recognition. Additionally, the CNN model's architecture demonstrates outstanding accuracy, reaching an accuracy rate of 99.28%. This remarkable display highlights the appropriateness of the selected model for the task of digit recognition. By utilizing a pre-trained model of exceptional accuracy, the project is able to establish a strong base, reducing the necessity for in-depth testing of different architectures.

The project utilizes a carefully crafted CNN model architecture shown in Figure 2, Convolution block 1 contain 3 convolution layers, convolution block 2 contain 2 convolution blocks and the Convloution Block 3 contain 1 convolution layer. Fully connected layer contains Linear and then SoftMax is used in output layer for multi-class classification. The CNN is used to extract important features from hand-written digit images in order to achieve precise classification. The structure includes three sets of convolutional blocks, with each having several convolutional layers, batch normalization, ReLU activation, max pooling, and dropout layers. This model is designed to learn hierarchical representations of input images by first focusing on basic features like edges and corners, and then moving on to more complex features such as digit shapes and textures.

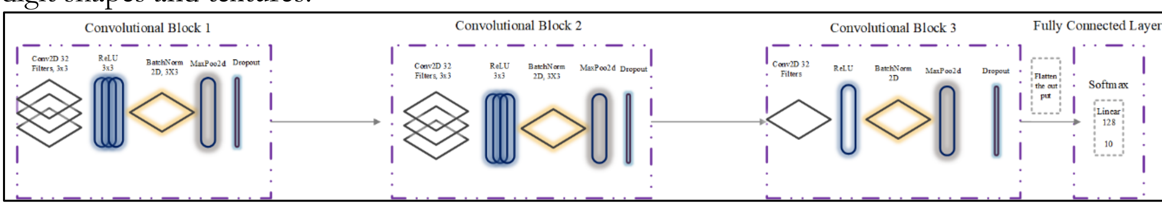

Figure 1: Model architecture

The model architecture is tailored for the MNIST dataset, extracting complex features through three convolutional blocks. Industry-standard frameworks like TensorFlow and PyTorch are used to streamline model development and deployment. Each layer is carefully configured for optimal information flow and feature extraction. The model's adaptability is enhanced by variable kernel sizes and batch normalization, ensuring stable training and robust learning. Through iterative refinement of hyperparameters and configurations, the CNN model excels in accurate digit classification. Overall, the meticulously crafted architecture and implementation lead to efficient and effective multi-digit number recognition.

Model Training:

The methodology for training a CNN for multi-digit recognition begins with data loading, where the training dataset comprising images containing multiple digits and their corresponding labels is loaded into memory. Data augmentation techniques are then applied to diversify the dataset and enhance the model's generalization capabilities, encompassing random affine transformations, color jittering, and horizontal flipping. Next, the CNN model is initialized with random weights for training, followed by a forward pass where input images undergo sequential processing through convolutional and pooling operations to extract pertinent features. Subsequently, the loss function, such as cross-entropy loss, is calculated to assess the model's performance by quantifying the disparity between predicted and actual labels. Backpropagation is employed to propagate gradients backward through the network, facilitating parameter updates based on computed losses. These parameters, including weights and biases, are adjusted by the optimizer, such as Adam optimizer, in an iterative process aimed at minimizing loss and improving accuracy in recognizing multiple digits within images. The training iterates over multiple epochs, allowing the model to gradually refine its parameters and enhance its capability to discern the complexities inherent in multi-digit images. The MNIST dataset, which consists of a large assortment of hand-drawn numbers, acts as the basis for training the model. Prior to starting the training process, the dataset is divided into two separate parts: one for training and one for validation.

Around 80% of the MNIST dataset is used for training purposes, with the remaining 20% set aside for evaluating the model's performance. This division guarantees that the model is trained on various digit samples and tested on unseen data to evaluate its generalization ability. During the training process, data augmentation methods are used to improve the model's resilience and ability to generalize. These methods add changes and disturbances to the input information, leading to an increased variety of examples observed by the model. Arbitrary geometric changes like rotation, translation, scaling, and shearing are used on the training images to simulate situations in the real world where numbers may be seen in various angles and viewpoints. Furthermore, color jittering methods are used to add slight changes in brightness, contrast, and hue, increasing the diversity of the training data and helping the model adjust to different lighting conditions and image features.

Regularization techniques like dropout layers and max pooling are used in the CNN model to prevent overfitting and improve generalization. The fully connected layer maps the extracted features to 10 output classes for MNIST digit recognition. The log SoftMax activation function normalizes outputs into probability distributions for accurate class predictions. The training process involves data preprocessing, model optimization, and performance evaluation to minimize the error between predicted and actual labels. The Adam optimizer updates model parameters based on gradients from the loss function to converge towards optimal values. Overall, this model combines architectural innovation, computational power, and algorithmic finesse to excel in multi-digit number recognition, promising advancements in computer vision and pattern recognition.

The negative log likelihood (NLL) loss function minimizes the disparity between predicted class probabilities and true labels, optimizing model performance in digit recognition. Early stopping prevents overfitting by halting training when validation loss stagnates, ensuring the model's ability to generalize accurately. The training process combines data augmentation, model optimization, and performance monitoring to enhance the model's ability to discern multi-digit numbers with precision. Through iterative refinement, the model gains the cognitive skills needed for effective digit recognition, driving innovation and excellence in computer vision and pattern recognition.

Generating Multi-Digits:

Creating images with multiple digits is added to the MNIST dataset to increase the amount of training and testing data available. This replicates real-life situations in which numbers can be found grouped together in different ways, like in addresses or item serial numbers. Several random numbers from the MNIST dataset are chosen and combined side by side to create composite images with multiple digits. The quantity and organization of numbers in each picture are changed to generate different sets for training and testing.

Every produced image is correctly tagged based on the numbers it includes. This labeling procedure guarantees that the true information is accessible to assess the model's performance in both training and testing stages. Extra augmentation methods can be used on the produced images to add diversity and improve the model's capacity to adapt to new data. This might involve changes like turning, resizing, and moving to imitate the diverse ways digits appear in the real world. The research guarantees a varied dataset for training, validation, and testing of the multi-digit recognition system by merging the MNIST dataset with randomly generated images showing multiple digits, as illustrated in Figure 3. This method allows the model to incorporate lessons from typical digit samples as well as intricate, true-to-life situations, enhancing its strength and precision in real-world uses.

The key to randomly generating images is to carefully combine randomly chosen digit images to create coherent multi-digit compositions. In order to imitate situations found in the real world where numbers come together to create intricate numerical patterns, the chosen digits are placed next to each other horizontally, blending smoothly to create a cohesive visual unit.

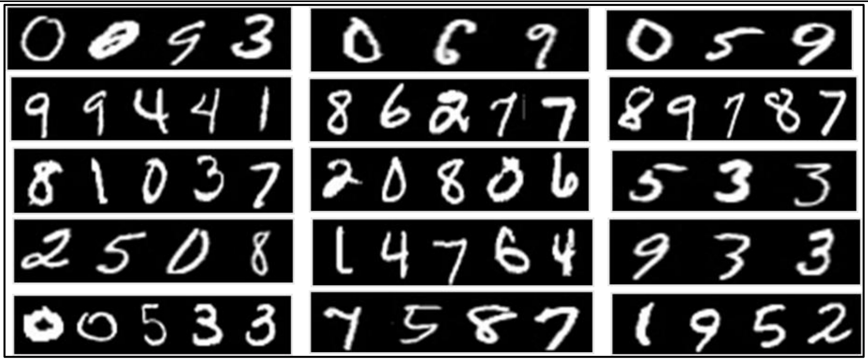

Figure 2: Samples of multi-digit images

By changing the quantity and layout of linked numbers, a wide range of multi-digit patterns is created, including many different combinations and placements of digits. In order to make it easier to evaluate and test the multi-digit recognition model, every created image is carefully labeled based on the individual digits it contains. By carefully annotating, the actual labels for each digit in the generated images are clearly marked, allowing for accurate assessment of the model's recognition abilities and efficiency. These markers are essential markers that help direct the model in achieving unmatched accuracy and precision in understanding the complex numerical patterns found in the synthetic images. Basically, the random image creation part is a crucial element in the structure of recognizing multi-digit numbers, providing the model with diverse training and assessment data needed to develop skills and adaptability in real-life situations.

Test Process:

The peak of the effort to recognize multi-digit numbers is represented in the thorough testing phase, during which the trained model is carefully examined to evaluate its effectiveness and strength in interpreting multi-digit number combinations. The testing process involves using a wide range of randomly generated images designed to replicate real-world situations in optical character recognition and digital image analysis. Before evaluating the model, a set of preprocessing methods, such as Gaussian blur, bilateral filtering, and Canny edge detection, are used to improve feature extraction and reveal important attributes within the images.

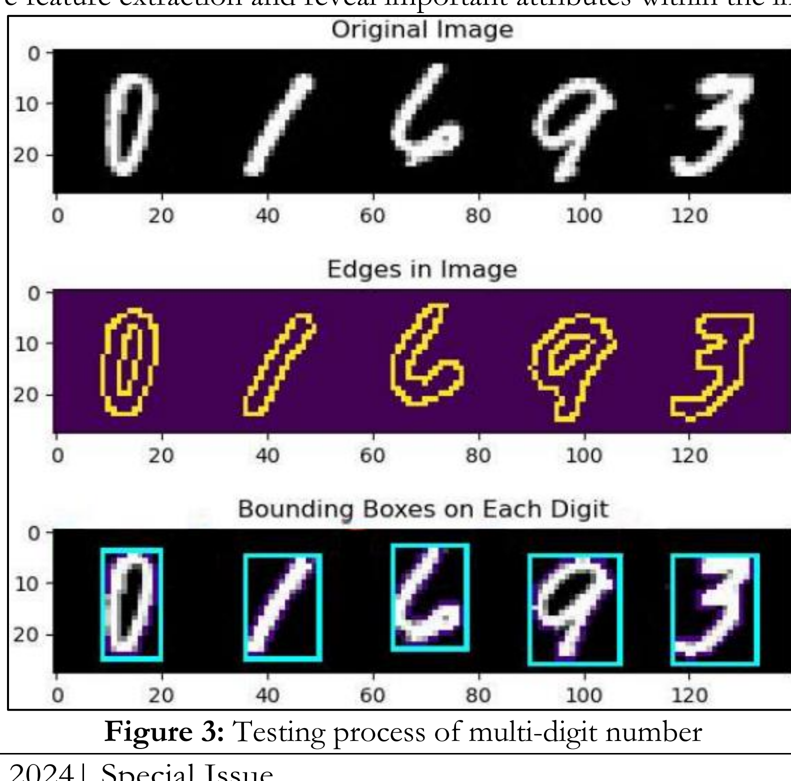

Figure 3: Testing process of multi-digit number

Equipped with these improvements, the model then goes on to identify outlines in the test pictures, exposing the hidden numerical values as shown in Figure 4. Utilizing its cognitive ability, developed through intense training, the model strives to understand and interpret the numerical importance contained in every shape, based on subtle details and unique characteristics visible to its perceptive gaze. This thorough testing method guarantees that the model is prepared to correctly and dependably identify multi-digit sequences in practical scenarios.

Result and Discussion:

The process of evaluating of the trained multi-digit recognition model provided numerous insights into the effectiveness and robustness, revealing its performance and ability to distinguish complex numerical patterns in synthetic images. Allows to explore the depth of the findings discovered through careful analysis and examination as shown in the Figure 5.

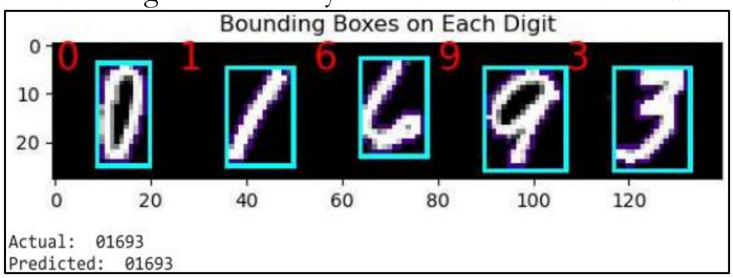

Figure 4: Multi digit detection result

The trained model has shown impressive precision in identifying and understanding multi-digit number sequences, with an astute ability to unravel the complexities of numerical arrangements hidden in artificial images. By combining preprocessing, contour detection, and cognitive inference effectively, the model has shown accuracy and precision in its predictions, demonstrating a natural grasp of the intricate details and unique characteristics present in numerical sequences. The confusion matrix of the test data is illustrated in Figure 6, indicating a training accuracy of 99.60% for the final model and a validation accuracy of 99.28%, surpassing the previous work done on the MNIST dataset.

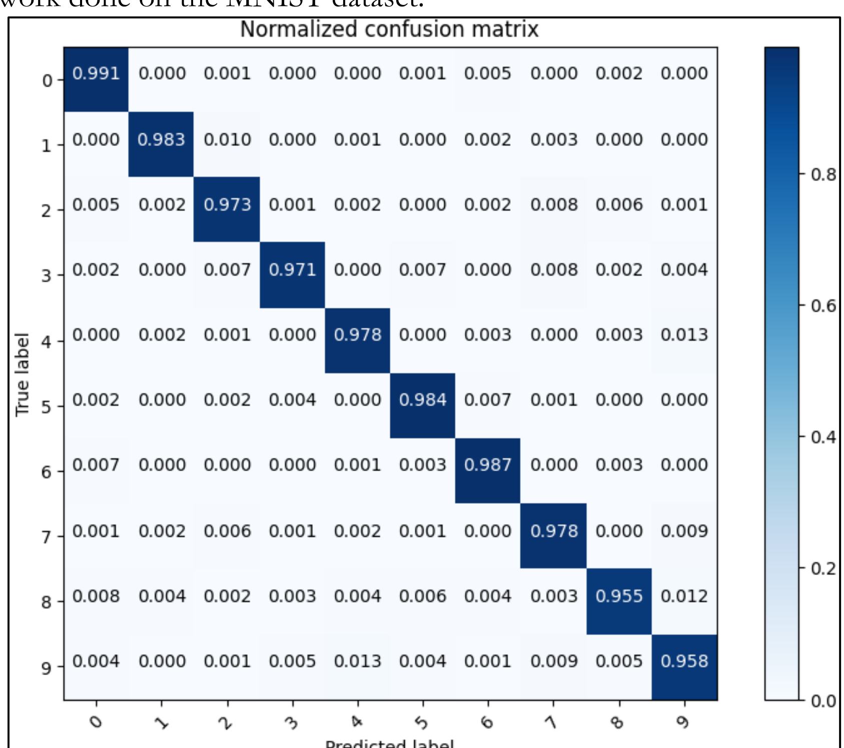

Figure 5: Confusion matrix

The results represent the performance metrics of the model across different classes, where each class corresponds to a specific digit from 0 to 9. Precision indicates the accuracy of the model in correctly identifying instances of each digit class, while recall measures the model's ability to capture all instances of a particular digit from the dataset. The F1-score is a combined metric that balances both precision and recall, providing an overall measure of the model's performance for each class. Overall, the model demonstrates high precision and recall values across most digit classes, with scores ranging from approximately 0.985 to 0.996. This indicates that the model is proficient in accurately identifying individual digits across the board. Notably, classes 0, 1, 4, 5, and 8 exhibit particularly high precision and recall values, exceeding 0.99 for both metrics. These digits are likely well-represented in the dataset and are easily distinguishable by the model. However, there are slight variations in performance across different digits. For example, classes 2, 3, 6, and 9 have slightly lower precision and recall scores compared to other digits, although they still remain above 0.98 for most metrics. This could be attributed to inherent similarities or ambiguities in the visual representation of these digits, posing challenges for the model in accurately distinguishing them from others.

Overall, the high precision and recall values across most digit classes indicate that the model is robust and reliable in recognizing handwritten digits. However, further analysis and potential adjustments may be warranted to address any discrepancies in performance observed across specific digit classes. The model is evaluated with 20% test data and the classification report is shown in Table 1 for the test data.

Table 1: Classification report of the model

Classes	Precision	Recall	F1-Score
0	0.9967	0.9983	0.9975
1	0.9963	0.9941	0.9952
2	0.9906	0.9948	0.9927
3	0.9935	0.9902	0.9918
4	0.9941	0.9941	0.9941
5	0.9945	0.9926	0.9935
6	0.9929	0.9947	0.9938
7	0.9852	0.9898	0.9875
8	0.9949	0.9873	0.9911
9	0.9899	0.9924	0.9912

At the end of the training, the model showed exceptional performance with a training loss of 0.013 and accuracy of 99.6%, along with a validation loss of 0.024 and accuracy of 99.3%. These numbers demonstrated their commitment to perfection and represented a significant achievement in their model's development. Beside these accomplishments, the charts depicting loss and accuracy illustrated a narrative of strength and creativity the accuracy and loss plots is shown in Figure 7.

Researchers embarked on the challenge of creating a system that can identify multi-digit numbers, facing a journey like navigating a complex maze filled with both discoveries and obstacles. The conversation shifts to the interesting specifics about how the model works, the challenges encountered, and the hopeful future directions. The combined efforts led to the development of a CNN that is similar to an experienced detective with exceptional accuracy in detecting multi-digit numbers. By undergoing intensive training and meticulous adjustments, the model became skilled at decoding complex numerical patterns, accurately identifying the intricacies present in images. Through meticulous training and optimization, the model has been imbued with a discerning eye capable of unraveling the intricacies of numerical compositions embedded within synthetic images, showcasing fidelity and precision in its predictions.

Despite the model's commendable performance, several challenges and limitations have been encountered along the journey, warranting careful consideration and mitigation strategies.

One notable challenge lies in the variability and complexity inherent in real-world scenarios, which may pose challenges for the model's generalization and adaptability. Additionally, the efficacy of the model may be contingent upon the quality and diversity of the training data, necessitating careful curation and augmentation to mitigate bias and enhance robustness. The application of preprocessing techniques, including Gaussian blur, bilateral filtering, and Canny edge detection, has been instrumental in enhancing feature extraction and facilitating the model's discernment of numerical compositions. By augmenting the raw input data with processed features, the model has been endowed with a richer representation of numerical compositions, thereby enhancing its efficacy and robustness in discerning intricate patterns.

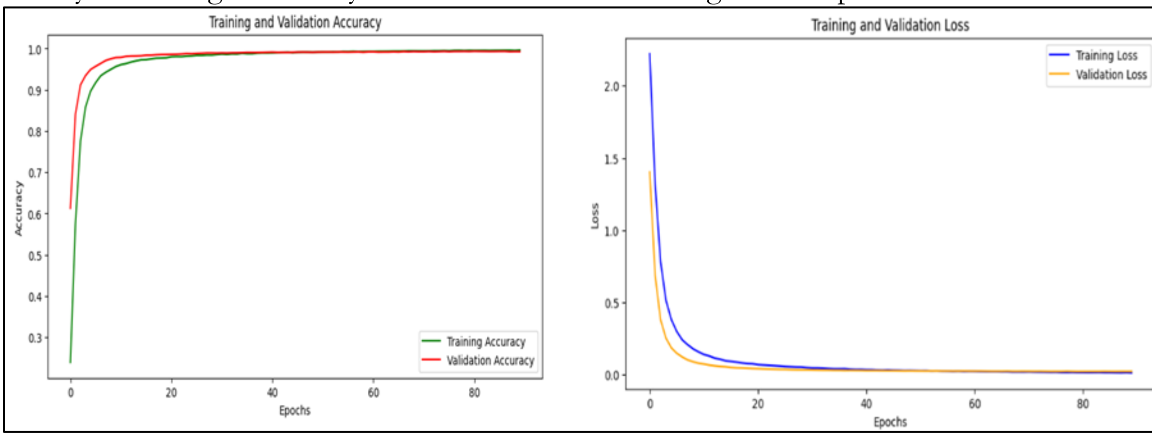

Figure 6: Accuracy and loss plots

Despite its overall robustness, the model exhibits variations in performance across different digit classes, highlighting potential challenges in accurately distinguishing certain digits from others. These discrepancies may stem from inherent similarities or ambiguities in the visual representation of specific digits, posing hurdles for the model in classification. However, such variations provide valuable insights for future refinement and optimization, guiding efforts to enhance the model's discriminative capabilities and address class-specific challenges. Through targeted analysis and adjustment, the model can potentially overcome these hurdles and achieve greater uniformity in performance across all digit classes. One of the model's notable strengths lies in its high precision and recall values across most digit classes, indicating its proficiency in accurately identifying individual digits. This high precision not only minimizes errors in digit recognition but also instills confidence in the model's predictions, bolstering its utility in real-world scenarios. Moreover, the model's robustness and reliability underscore its potential for widespread adoption in applications requiring precise and consistent digit recognition, such as optical character recognition (OCR) systems and automated data entry processes.

Contour detection has emerged as a pivotal technique for delineating and segmenting numerical compositions within synthetic images, facilitating the model's cognitive inference and prediction. Through the judicious identification and extraction of contours, the model has been empowered to discern and decipher numerical sequences with fidelity and precision, traversing the labyrinthine corridors of synthetic images with acumen and aplomb. The transformative breakthroughs unearthed through the crucible of model development and testing portend a myriad of future directions and explorations, beckoning towards new vistas of innovation and discovery. Future research endeavors may seek to explore novel architectures and methodologies for enhancing the model's efficacy and robustness in real-world scenarios, encompassing diverse domains such as automated document processing, optical character recognition, and digital image analysis. One of the model's notable strengths lies in its high precision and recall values across most digit classes, indicating its proficiency in accurately identifying individual digits. This high precision not only minimizes errors in digit recognition but also instills confidence in the model's predictions, bolstering its utility in real-world scenarios.

Moreover, the model's robustness and reliability underscore its potential for widespread adoption in applications requiring precise and consistent digit recognition, such as optical character recognition (OCR) systems and automated data entry processes. Additionally, the integration of advanced techniques, including transfer learning, ensemble methods, and reinforcement learning, may offer promising avenues for further enhancing the model's performance and adaptability across diverse and dynamic environments.

Conclusion:

In conclusion, this research marks a major advancement in the field of recognizing multi-digit numbers. Through careful creation, testing, and assessment of models, important findings have been discovered, setting the stage for future progress in computer vision and pattern recognition. The results emphasize the effectiveness of convolutional neural networks in understanding complex numerical structures, while also pointing out the upcoming obstacles. Even though it was trained with single-digit images, the model demonstrated its strength and flexibility by achieving high accuracy rates of 99.60% in training and 99.28% in testing on multi-digit numbers. Researchers are advised to investigate new designs and techniques to improve the model's effectiveness in real-life situations, promoting creativity and advancing the field.

Author's Contribution:

Both the authors have exactly same contribution in this paper.

Conflict of Interest:

The authors declare that there are no conflicts of interest associated with the publication of this manuscript in IJIST. The research was conducted with integrity and impartiality, devoid of any external influences that could compromise the objectivity of the findings.

References:

[1] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," Sensors 2020, Vol. 20, Page 3344, vol. 20, no. 12, p. 3344, Jun. 2020, doi: 10.3390/S20123344.

[2] A. M. Crawford, D. M. Ommen, and A. L. Carriquiry, "A statistical approach to aid examiners in the forensic analysis of handwriting," J. Forensic Sci., vol. 68, no. 5, pp. 1768–1779, Sep. 2023, doi: 10.1111/1556-4029.15337.

[3] R. A. Hicklin et al., "Accuracy and reliability of forensic handwriting comparisons," Proc. Natl. Acad. Sci. U. S. A., vol. 119, no. 32, p. e2119944119, Aug. 2022, doi: 10.1073/PNAS.2119944119/SUPPL_FILE/PNAS.2119944119.SD05.XLSX.

[4] A. Sharma and A. Sharma, "A combined static and dynamic feature extraction technique to recognize handwritten digits," Vietnam J. Comput. Sci. 2015 23, vol. 2, no. 3, pp. 133–142, Jan. 2015, doi: 10.1007/S40595-014-0038-1.

[5] M. Asif, M. Bin Ahmad, S. Mushtaq, K. Masood, T. Mahmood, and A. Ali Nagra, "Long Multi-digit Number Recognition from Images Empowered by Deep Convolutional Neural Networks," Comput. J., vol. 65, no. 10, pp. 2815–2827, Oct. 2022, doi: 10.1093/COMJNL/BXAB117.

[6] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks," 2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc., Dec. 2013, Accessed: May 25, 2024. [Online]. Available: https://arxiv.org/abs/1312.6082v4

[7] C. L. Tan and A. Jantan, "Digit Recognition Using Neural Networks," Malaysian J. Comput. Sci., vol. 17, no. 2, pp. 40–54, Dec. 2004, Accessed: May 25, 2024. [Online]. Available: https://ejournal.um.edu.my/index.php/MJCS/article/view/6216

[8] F. Alimoglu and E. Alpaydin, "Combining multiple representations and classifiers for pen-based handwritten digit recognition," Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, vol. 2, pp. 637–640, 1997, doi: 10.1109/ICDAR.1997.620583.

[9] X. Liu, Y. Deng, Y. Sun, and Y. Zhou, "Multi-digit recognition with convolutional neural

network and long short-term memory," ICNC-FSKD 2018 - 14th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov., pp. 1187–1192, Jul. 2018, doi: 10.1109/FSKD.2018.8686963.

[10] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," Procedia Comput. Sci., vol. 167, pp. 2554–2560, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.309.

[11] H. huang Zhao and H. Liu, "Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition," Granul. Comput., vol. 5, no. 3, pp. 411–418, Jul. 2020, doi: 10.1007/S41066-019-00158-6/TABLES/1.

[12] J. Li, G. Sun, L. Yi, Q. Cao, F. Liang, and Y. Sun, "Handwritten Digit Recognition System Based on Convolutional Neural Network," Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020, pp. 739–742, Aug. 2020, doi: 10.1109/AEECA49918.2020.9213619.

[13] R. Karakaya, S. Kazan, and S. Cakar, "Handwritten Digit Recognition Using Machine Learning," Sak. Univ. J. Sci., vol. 25, no. 1, pp. 65–71, Feb. 2021, doi: 10.16984/SAUFENBILDER.801684.

[14] M. Jain, G. Kaur, M. P. Quamar, and H. Gupta, "Handwritten digit recognition using CNN," Proc. Int. Conf. Innov. Pract. Technol. Manag. ICIPTM 2021, pp. 211–215, Feb. 2021, doi: 10.1109/ICIPTM52218.2021.9388351.

[15] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, "An improved faster-RCNN model for handwritten character recognition," Arab. J. Sci. Eng., vol. 46, no. 9, pp. 8509–8523, Sep. 2021, doi: 10.1007/S13369-021-05471-4/METRICS.

[16] A. Baldominos, Y. Saez, and P. Isasi, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," Appl. Sci. 2019, Vol. 9, Page 3169, vol. 9, no. 15, p. 3169, Aug. 2019, doi: 10.3390/APP9153169.

[17] R. Raj, J. Mathew, S. K. Kannath, and J. Rajan, "Crossover based technique for data augmentation," Comput. Methods Programs Biomed., vol. 218, p. 106716, May 2022, doi: 10.1016/J.CMPB.2022.106716.

[18] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Adv. Neural Inf. Process. Syst., vol. 32, Dec. 2019, Accessed: May 25, 2024. [Online]. Available: https://arxiv.org/abs/1912.01703v1

[19] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." Accessed: May 25, 2024. [Online]. Available: http://yann.lecun.com/exdb/mnist/