OPEN ACCESS
International Journal of Innovations in Science & Technology

An Efficient and Robust Deep Learning Approach for Vehicle Recognition using Light-weight Deep Network

Fatima Shahzad[1], Muhammad Hassan Khan[1], Muhammad Shahid Farid[1], Nazish Ashfaq[1*]
[1]Department of Computer Science, University of the Punjab, Quid-e-Azam Campus, Lahore, 100190, Pakistan, Pakistan.
*** Correspondence.** nazishashfaq30@gmail.com.

Citation | Shahzad. F, Khan. M. H, Farid. M. S, Ashfaq. N, "An Efficient and Robust Deep Learning Approach for Vehicle Recognition using Light-weight Deep Network", IJIST, Vol. 7 Issue. 1 pp 506-522, March 2025
Received | Feb 11, 2025 **Revised |** March 4, 2025 **Accepted |** March 6, 2025 **Published |** March 8, 2025.

In the realm of intelligent transportation systems, automatic number plate detection has emerged as a crucial research topic due to its wide range of applications, including traffic violation monitoring, support for autonomous vehicles, vehicle speed tracking, automated toll collection, stolen vehicle identification, and overall traffic management. The goal of automatic number plate detection is to accurately identify vehicles based on their number plates. This study proposes a hierarchical approach for detecting number plates. In the initial phase, a lightweight deep learning model, Mobile Net-SSD, is employed to detect number plates. Subsequently, the alphanumeric characters from the detected number plates are extracted using an Optical Character Recognition (OCR) technique. The model is, built on a convolutional neural network, and efficiently uses depth wise and pointwise convolution layers, making it suitable for mobile and embedded systems. Additionally, we introduce a dataset of 30,613 vehicle number plate images to foster further research. Experimental evaluations show that the proposed method achieves 95% accuracy on this dataset, significantly enhancing real-time number plate detection and making it suitable for large-scale implementations in smart cities and intelligent transportation networks.

Keywords. Automatic number plate detection; Intelligent transportation system; Mobile Net-SSD; Optical character recognition.

Introduction.

Automatic Number Plate Recognition (ANPR) systems are designed to autonomously detect and extract vehicle license plate information without the need for human intervention. The rapid growth in the vehicular sector demands a quick and real-time advancement that can be used for traffic management and monitoring. The manual management and monitoring of traffic raises several difficulties including the significant amount of time, possibility of errors, and high expense. The ANPR plays a pivotal role in various important real-world applications such as traffic monitoring [1], identification of traffic violations, calculating vehicle speeds [2], toll payments [3], recognition of stolen vehicles [4], urban mobility, automatic parking where authentication is required, facilitating the use of autonomous vehicles, and surveillance [5]. It may also help in the intelligent transportation system of smart cities [6].

The ANPR framework typically involves two core components. image processing and pattern recognition. In particular, the vehicle's image is captured using a visual sensor (e.g., surveillance camera), and several features are extracted from the image to localize the information of the number plate. The localized image patch (number plate region) is then processed to extract readable text, enabling vehicle identification by cross-referencing it with a pre-existing database. An illustration of the ANPR system is depicted in Figure 1. Numerous approaches to ANPR have been developed over the past two decades, primarily categorized into two groups. handcrafted feature extraction methods and deep learning-based approaches. Handcrafted techniques involve manually extracting features from the image such as the Scale-Invariant Feature Transform (SIFT) [7], Histogram of Oriented Gradients (HOG) [8], and Haarlike [9] feature to represent the number plate. Later, these features are fed to machine learning algorithms (e.g., Support Vector Machine (SVM) [10] and AdaBoost [11][12] for recognition. Contrarily, the deep learning models are based on a neural network and extract the relevant features from the image directly. Usually, a deep network comprises several layers of neurons that perform linear and non-linear transformations on input data to get the high semantics of raw input data. For instance, a few deep learning models such as Alex-Net [13], Convolutional Neural Network (CNN) [14], Region-based CNN (R-CNN) [15], You Look Only Once (YOLO) [16] and their different variants produce convincing results to detect the number plate. Although deep learning models offer superior accuracy, they are computationally intensive and demand high-performance computing resources for training.

With the rapid advancement in hardware technology, deep learning models can now be deployed on mobile and embedded devices, enabling applications such as mobile policing and real-time traffic monitoring [17][18]. However, these low-power systems require models that are not only accurate but also computationally efficient to run. Traditional object detection models, while highly accurate, often demand substantial processing power, making them impractical for edge deployment. To address this, several lightweight deep learning models have been proposed [19], including ShuffleNet [20], SqueezeNet [21], and MobileNet [22], which are optimized for mobile and embedded environments. Among them, MobileNet-SSD stands out as an optimal choice for real-time ANPR because it balances efficiency and accuracy better than other lightweight models. MobileNet uses depthwise separable convolutions to reduce the computational cost, while SSD eliminates the need for a region proposal network, enabling real-time inference. Although some lightweight models tend to compromise accuracy compared to larger architectures like YOLO and Faster R-CNN, MobileNet-SSD offers a superior trade-off, making it well-suited for number plate recognition on resource-constrained devices.

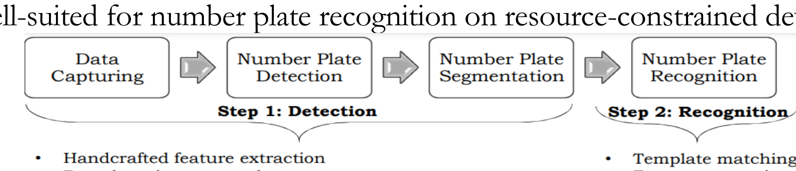

Figure 1. An illustration of the complete framework to automatically recognize the vehicles

using their number plates. The framework consists of two stages. In the first stage, the number plate is detected in the input image whereas, the second step emphasizes the recognition of characters in the detected number plate region.

This paper presents a lightweight deep-learning model to recognize vehicle number plates. The proposed approach works hierarchically. In the first step, we employed the concept of transfer learning, and a pre-trained MobileNet SSD architecture was used to detect the number plate region in an input image. The network was trained on the Microsoft COCO (Common Objects in Context) dataset [23], however, we fine-tuned the network with our proposed dataset. We explored a variety of augmentation techniques such as brightness alteration, shear transformation, rotation, color jittering, adding noise, scaling, etc. on our dataset to enhance the model's generalization power. We experimentally observed that the model's recognition ability is improved substantially by applying these augmentation techniques. This also implies the applicability of the proposed model in real-life scenarios. The experimental evaluation demonstrates that the proposed network is efficient and quite accurate in detecting small objects like number plates on a computing machine with limited resources. In the later step, the segmented number plate region is forwarded to the lightweight pre-trained deep network EasyOCR[1] to recognize the characters. To the best of our knowledge, no public dataset of number plates (with English alphabets) is freely available, therefore, this paper also proposes a large dataset of 30,613 real vehicle images to the research community to address the scarcity of publicly available datasets. The proposed dataset is freely accessible to the public and encompasses images conducive to real-time detection, thus contributing to the advancement of detection methodologies.

Objectives and Novelty Statement.

The major contributions of this research are explained in the following.

- A large dataset of real vehicle images is collected from various online/offline sources to generate a new dataset and make it freely available to the research community.
- A lightweight MobileNet-SSD architecture is proposed to build a real-time ANPR system aiming to enable mobile devices to detect vehicle number plates accurately in an intelligent transportation system.
- Exploration of numerous data augmentation techniques to enhance the adaptability and accuracy of the model.
- Benchmark results are reported on the proposed dataset.

Extensive analysis of existing approaches and they are grouped based on their underlying methods and techniques.

Literature Review.

The working of the ANPR system can be divided into two main parts. (1) Number plate detection, and (2) recognition. Numerous techniques have been proposed in the last two decades and their brief overview is summarized in the following.

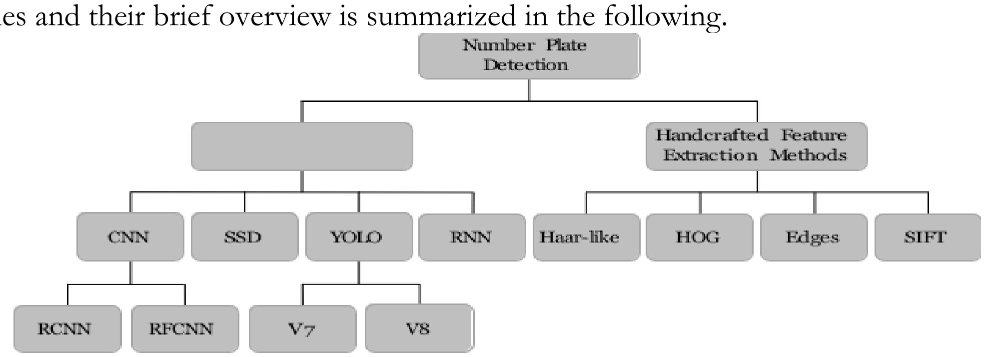

Figure 2. The categorization of existing number plate recognition methods into hand-

[1] https://github.com/JaidedAI/EasyOCR

crafted feature extraction techniques, which are further categorized into Haar-like, HOG, SIFT, and Edges, and deep learning approaches, which are further categorized into CNN, SSD, YOLO, and RNN.

Number Plate Detection.

The process of number plate detection involves the localization of the desired rectangular patch (i.e. number plate) in an input image. The existing approaches at this stage can be categorized into two broader groups (Figure 2) based on their underlying implementation, namely, handcrafted feature extraction methods and deep learning approaches.

Handcrafted Feature Extraction.

The set of techniques in this group exploits the researchers' expertise and domain knowledge to manually craft specific features that are extracted from raw data to represent the relevant patterns or characteristics of the data [24]. For instance, the authors in [25] employed a Haar-like feature extraction method to detect number plates using a template matching technique. Their proposed technique used a window-sliding approach over the vehicle input image, calculating Haar-like features in each window, and applying an Adaboost-trained classifier to determine if the region contains a number plate or not. However, the Haar-like feature extraction approach is computationally complex and sensitive to variation, such as deviation in lighting, rotation, and background noise. The authors in [26] employed the HOG feature extraction technique with an AdaBoost classifier to detect the number plates. Similarly, Bachchan et al. [27] explored the Local Binary Patterns (LBP) feature extraction approach to detect the number plate region in an input image. Specifically, they compared the pixel's intensity value to the pixels around it to produce a binary pattern and encoded the texture data needed to identify a number plate. However, LBP's performance is noticeably low due to bad lighting conditions, and a noisy and complicated background. The authors in [28] employed the SIFT technique to detect number plates. This process involves the extraction of stable key points in an input image and computing descriptors that are invariant to scale variation. These descriptors are later used to train/test a model for number plate discrimination. The authors reported that their method accurately detects license plate regions in images with varying scales and orientations, but it has high computational complexity.

The edge detection techniques use the information of the rectangular region by performing horizontal and vertical edge detection on the original image to localize the number plate. For example, the techniques proposed in [29][30] detect the information of vertical edges using the Sobel operator [29] and Gabor filter [30] to estimate the number plate region. The authors in [31] computed the magnitude of vertical edges to extract the desired region. Khan et al. [32] presented a performance comparison between multiple edge detection filters for number plate localization. Nonetheless, the detection accuracy of the edges is sensitive to the image quality and they are computationally intensive. The handcrafted feature extraction approaches have certain limitations including their accuracy is dependent on the domain knowledge of the researchers, application dependent, and they perform poorly when dealing with rotation, noisy images, and poor lighting conditions.

Deep Learning Approaches.

Deep learning utilizes artificial neural networks to automatically learn complex patterns and representations from raw data without any manual feature computation. A deep network is usually built by stacking several layers of neurons that perform linear and non-linear transformations on input data to get the high-level representation of raw input data [33].

For instance, the authors in [34] employed CNN for license plate detection. Their proposed networks are composed of convolutional layers for high-level feature extraction, pooling layers for lowering the parameter of the input tensor, and fully connected layers for non-linear feature learning. Nonetheless, their proposed structure implies an enormous computational cost. The technique proposed in [35] employed R-CNN to generate regional proposals, extract deep features using a pre-trained CNN, classify the regions as license plates or not, and apply post-processing to eliminate duplicates. Tu et al. [36] proposed a hierarchical approach to detect the number plate. In the first stage, a higher-level R-CNN is used to extract vehicle regions from the input image. Subsequently, the regions of the detected vehicle are passed to a lower-level (smaller) R-CNN to specifically detect the number plate. Lately, the region-based fully convolutional network (R-FCN) [37] has been proposed for object detection. They utilized position-sensitive score maps and region-based pooling to achieve precise object localization and classification within predefined regions of interest. To increase accuracy, several R-FCN and Mask R-CNN [38] extensions are suggested too to detect number plates.

The authors in [39] fine-tuned the YOLO deep network to detect number plates. During inference, the pre-trained YOLO model processed the input image in a single pass, dividing it into a grid and predicting the class probabilities for each grid cell (i.e. bounding box). The box with a high confidence score is selected as the detected number plate region. The YOLO deep network reported convincing recognition results in the domain of object detection, however, they do not perform well in the recognition of small objects when their location is erroneous [40]. The authors in [41] proposed a two-stage approach using YOLOv2 to detect number plates. In the first step of automatic detection of the number plate, the vehicle is localized in the input image and the second step detects the number plate region in the localized patch. Similarly, the recent variants of the YOLO deep network are explored too in [42][43]. The authors in [44] use YOLOv8, SSD, and Faster RCNN models to detect and identify license plates, focusing specifically on an Iranian motorcycle license plate dataset. The authors in [45] first detect the license plates in video frames using the SSD MobileNet model, followed by number plate recognition with EasyOCR. Subsequently, remove duplicate and inaccurate license plates before storing them in the database. The authors in [46] propose an efficient Automatic License Plate Recognition (ALPR) system using YOLOv8 for license plate detection and EfficientNet B7 for character recognition. The system achieved promising results, demonstrating its feasibility and effectiveness in real-world smart city applications. Similarly, [47] used EfficientDet as a robust object detection framework. Its efficient feature fusion and scalable architecture make it well-suited for license plate detection, addressing challenges such as varying image sizes, orientations, and lighting conditions while maintaining high accuracy and computational efficiency. Despite the promising recognition accuracies of the aforementioned deep networks, they require enormous amounts of training data and significant computational costs.

Number Plate Recognition.

The second stage of the ANPR framework is the recognition of characters in the detected number plate region. The set of recognition techniques can be broadly categorized into template matching [48] and feature extraction-based techniques [49]. In template matching, a pre-defined template of character is traversed over the segmented image patch of the number plate and the correlation is computed at each pixel [50]. The pixel location with a high correlation represents the possible match. For instance, Naito et al. [51] used several templates for each of the characters to capture their different orientations. The authors in [52] used normalized cross-correlation for character recognition. Usually, template-matching techniques are applied to edge images to eliminate the effects of illumination. However, their recognition may degrade if the character shape is broken, and does not have the same size and orientation as the pre-defined template [53].

Feature extraction-based techniques compute relevant features from the detected region and feed them to classifiers to recognize the characters such as OCR [54]. The authors in [55] used Adaptive boosting (AdaBoost) in conjunction with Haar-like features for training cascade classifiers to recognize the characters. Apart from these conventional machine learning techniques which are based on handcrafted features, various types of Artificial Neural Networks (ANNs) [56] have also been used for character recognition. Though ANN-based techniques reported good recognition results, however, they require a huge amount of training instances and expensive computational resources.

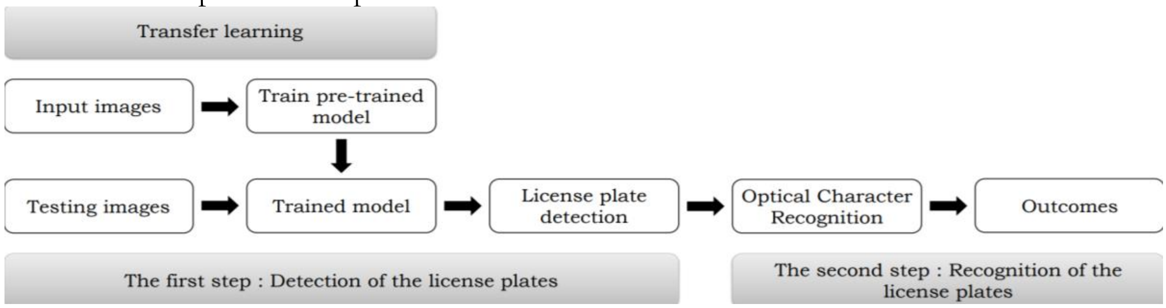

Figure 3. The diagram depicts the workflow of the proposed automatic license plate detection and recognition method, starting with license plate detection, followed by OCR-based character extraction, and ending with text output.

Material and Methods.

In recent, numerous research efforts have been made in the domain of deep learning to increase the recognition performance of the network by enhancing the architecture of the network which ultimately increases the computation complexity of the model [57]. However, in some real-time systems e.g. ANPR in visual surveillance, traffic monitoring, and mobile policing; a comparatively fast network is needed with less computational cost. To cope with this limitation, this study presents a lightweight deep-learning model based on VGG-SSD [19]. It combines the Single Shot Multi-Box Detector (SSD) framework with the Mobile-Net structure to efficiently detect the desired object. Mobile-Net SSD works well for real-time license plate identification on devices with limited resources because it uses lightweight depth-wise separable convolutions, that provide a balance between accuracy and computational cost [58]. The proposed automatic license plate detection system works hierarchically. In the initial phase, a vehicle image is passed to a pre-trained Mobile-net SSD deep model, and it isolates the area holding the number plate. The area of the retrieved license plate is submitted to an OCR system in the second stage, which recognizes the characters and shows the results as text. The complete process is depicted in Figure 3 and the detail of each step is outlined in the following subsections.

Detection of number plate.

In the initial phase of number plate detection, we employed transfer learning by utilizing a pre-trained MobileNet SSD architecture to identify the number plate region within an input image. Although the network was originally trained on the Microsoft COCO dataset[2], we fine-tuned it using our collected dataset. To improve the model's generalization capabilities, we applied various data augmentation techniques, including brightness adjustment, shear transformation, rotation, color jittering, noise addition, and scaling. MobileNet-SSD replaces the standard convolutional process with a depthwise separable convolution using two layers. (1) depthwise convolution and (2) pointwise convolution. Unlike VGG-SSD [19], MobileNet-SSD uses convolution masks of size only 3×3 and 1×1. In particular, the depthwise convolution takes a single 3 × 3 kernel and applies it to each input channel. The pointwise convolution uses a 1 × 1 kernel to combine the results of all the depthwise convolutions. That is, the network contains a separate layer for filtering and another layer for merging. Table 1 describes the difference between standard convolution and depthwise separable convolutions. Both of the layers used batch normalization and rectified linear unit (ReLU) non-linearities.

Based on these two-step convolutions, the computation cost is substantially reduced with no effect on recognition accuracy. The proposed deep network reduces the number of parameters and, in turn, lowers the computation cost, making it appropriate for devices with limited resources.

Table 1. Differences between standard convolution layers and depthwise separable convolution layers.

Standard Conv.	Depthwise Separable Conv.
conv 3 × 3	depthwise conv 3 × 3
Batch normalization	Batch normalization
ReLU activation function	ReLU activation function
-	pointwise conv 1 × 1
-	Batch normalization
-	ReLU activation function

Preprocessing.

A total of 2,800 vehicle images were collected from the internet, and various data augmentation techniques were applied, including adjustments to brightness, rotation at multiple angles, the addition of noise, random scaling, grayscale conversion, shearing, and modifications to hue, contrast, and saturation. The LabelImg software was used to annotate vehicle license plates, while all other objects in the images were considered as background. The applied augmentations enhanced model performance by increasing the diversity of the dataset, helping the model generalize better to real-world variations. They also contributed to balancing the dataset by artificially expanding it with variations in lighting, orientation, and noise, reducing biases toward specific conditions. The final dataset comprised 30,613 images, which were divided into training, validation, and testing subsets. 21,430 images (70%) for training, 3,061 images (10%) for validation, and 6,122 images (20%) for testing. All images were resized to dimensions of 300 × 300 pixels.

The proposed network architecture comprises 22 convolutional layers, which include 13 depthwise separable layers for feature extraction and 9 standard convolutional layers for object detection. The architecture of the proposed network is depicted in Figure 4. These detection layers predict class scores and localization offsets for anchor boxes on multiple feature maps, while predefined anchor boxes of different scales and aspect ratios are established for each spatial position. Subsequent non-maximum suppression is applied to refine the detection accuracy of the model. The model computes detected regions from six feature maps of different scales, and there are a series of fixed sizes on each cell of a feature map. This multi-scale feature extraction and detection technique enables the model to effectively capture objects of varying sizes and shapes, enhancing its robustness in handling complex and diverse scenarios, such as those encountered in real-world license plate detection. The combination of depthwise separable and standard convolutional layers also reduces computational complexity while maintaining high accuracy, making the network suitable for deployment in real-time applications. The network is built using TensorFlow [59] API from Google. Similar to Inception V3 [60], we used RMSprop with an asynchronous gradient descent algorithm [61] in the training phase. We retrained the model using the transfer learning principle on our proposed dataset. The training images are input to the network with different variations to enhance the generalization performance of the network. Specifically, we employed a variety of augmentation techniques such as brightness alteration, shear transformation, rotation, color jittering, adding noise, scaling, etc. on input images to enhance the model's generalization power. It is empirically concluded that the model's recognition ability is improved substantially by applying these augmentation techniques. This also implies the applicability of the proposed model in real-life scenarios. We used momentum optimizer and RMS Prop as optimization algorithms, and the initial learning rate was set to 0.08. A summary of the network's hyper-parameters is outlined in Table 2. After training, the test image is passed through several convolution layers of the Mobile-net SSD, which extract appropriate features at various scales. The proposed approach first uses a Mobile-net model to extract the features and then utilizes SSD layers to determine the bounding box coordinates and class probabilities for number plate detection.

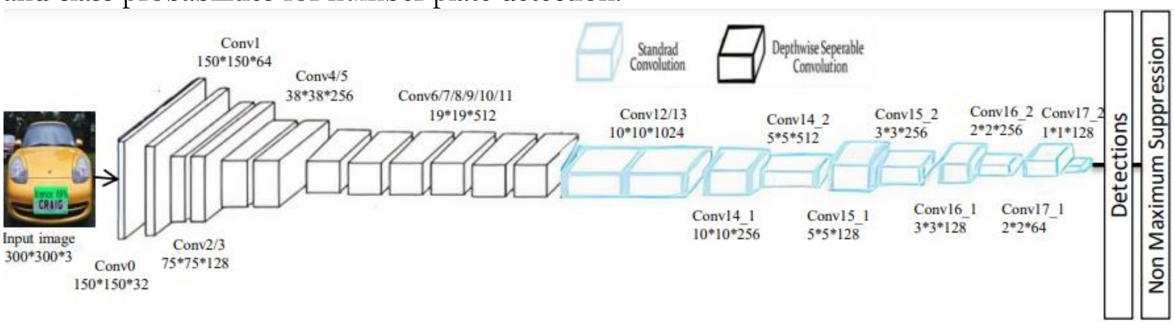

Figure 4. The architecture of the deep Mobile Net-SSD network comprises 22 convolutional layers, specifically designed to enhance detection capabilities. It includes 13 depthwise separable layers that optimize feature extraction efficiency and 9 standard convolutional layers utilized in object detection.

Table 2. The key parameters of the Mobile Net SSD model architecture used during the raining, validation, and testing phases for number plate detection, such as the loss function, activation function, learning rate, and optimization algorithm, play a crucial role in optimizing model performance and ensuring robust detection accuracy.

Hyperparameters	**Specification**
Optimization algorithm	RMSProp and Momentum optimizer
Activation function	ReLU
Loss function	Focal loss function
Number of layers	22 convolution layers
Batch size	32
Total iterations	94000

The proposed SSD is a stage object identification technique that can efficiently detect license plates in a single forward pass. The feature representation of objects at different sizes is enhanced by the feature pyramid network which although increases the detection accuracy, however, it generates several overlapping predictions of the number plate in an input image with their respective detection scores. To this end, first, we dropped all the predictions (most probable they are false positives) having a score less than a pre-defined threshold. The threshold value is empirically set to 0.5. Subsequently, we apply a non-maximum suppression technique, where the detection with the highest score is greedily selected as the final number plate. The network's output is the detected and localized number plate region in the input image.

Number plate recognition.

After the detection of the number plate region, the next step is the segmentation and recognition of the characters in the detected number plate using OCR. This process aims to extract and recognize the alphanumeric characters from digital images by examining the visual aspects and patterns of the letters and numbers before converting them into text that can be read by computers [62]. The extracted license plate region serves as an input for the number plate identification process. Contrary to other existing methods e.g., template matching [63], OCR is more effective and produces accurate results [64]. The template matching process involves choosing a template image and comparing it to various areas of the target image. A matching score is then determined by calculating the similarity between the template and each sub-region. If the score surpasses a predefined threshold, it is considered a match, indicating the presence of the template in that location. Although this technique is effective (to some extent) for identifying patterns, it has several limitations, such as not performing well in images with varying scale factors, rotations, and lighting conditions [65]. We used a pre-trained model EasyOCR[3] for character recognition which facilitates both text detection and recognition. For precise text region detection, it utilizes the Character Region Awareness for Text (CRAFT) model [66], which generates detailed character and affinity score maps to accurately identify and localize text regions. For text recognition, EasyOCR employs a Convolutional Recurrent Neural Network (CRNN) specifically designed for alphanumeric recognition on license plates. The convolutional layers within the CRNN extract spatial features from the license plate images, focusing on individual characters. These features are subsequently processed by recurrent layers, which include Long Short-Term Memory (LSTM) units or Gated Recurrent Units (GRUs), capturing the sequential dependencies between characters. This dual-stage approach leveraging CRAFT for precise text localization and CRNN for accurate text recognition ensures effective handling of complex text extraction tasks, providing a robust solution for license plate recognition.

Implementation Details.

The proposed deep network is implemented using TensorFlow [59] and trained with the RMSprop variant of the gradient descent algorithm utilizing its default parameters (such as an initial learning rate of 0.001). We used the ReLU (Rectified Linear Unit) activation function, which introduces non-linearity and helps in learning complex patterns while preventing the vanishing gradient problem [61]. The ReLU activation function is defined in equation (1) where x represents the input to the function. If the input is greater than zero, ReLU returns the input value; it returns zero, otherwise.

$$F_x = \max(0,x) \tag{1}$$

The epoch size is set to 500 and weights are updated with the mini-batch of size 128. We used the Focal Loss (FL) function [67] defined in equation (2) with its default values of balancing factor alpha and modulating factor gamma. That is, the default values are set at 0.25 and 2.0 respectively. The focal loss function has proven to be excellent in the domain of object detection when the imbalance between the background class and other classes is large. It down-weights well-classified examples and focuses on hard examples. In particular, it generates a much higher loss value for the miss-classified instance as compared to a well-classified instance. To limit the computational time during the training, validation, and testing phase, all the images are re-sized to 300×300 dimensions.

$$FL\ (p_t) = -a_t \cdot (1 - p_t)^{\gamma} \cdot \log(p_t), \tag{2}$$

where p_t is the predicted probability of the true class, a_t is a balancing factor to handle class imbalance, and γ is the focusing parameter. The experimental evaluation is carried out on a simple DELL machine with a CORE i7 processor and 8 GB of RAM, and the CPU does all of the training and testing; no extra GPU is needed.

Performance Evaluation on the Proposed Dataset.

The robustness of the proposed algorithm is evaluated using various metrics, including accuracy, recall, and mean average precision. Recall measures the proportion of true positive results relative to all actual positive instances, while precision reflects the correctness of the model's results. Given the difficulty in achieving high recall and high precision simultaneously, the F_1 score is used to provide a balanced assessment by combining both metrics. The evaluation involves key indicators such as True Positive (TP), False Negative (FN), and False Positive (FP).

[3]https://github.com/JaidedAI/EasyOCR

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F_1\ score = \frac{2*Precision*Recall}{Precision+Recall} \quad (5)$$

The effectiveness of the proposed approach is evaluated using our proposed dataset. The suggested model's training continues until it achieves an acceptable level of per-performance, which in this case is 94000 iterations, yielding a mean average accuracy of 0.95 at 0.5 intersections over union (IOU). IOU is a criterion used in object detection models to assess the model's accuracy. The value of mean average precision changes automatically as the IOU value changes. Calculating IOU requires dividing the intersection and union areas of two bordering boxes.

$$IOU = \frac{Area\ of\ Interstction}{Area\ of\ Union} \quad (6)$$

The mean average precision is calculated by averaging all the classes that have an IOU (Intersection over union) criteria. The performance of the model is impacted by changing IOU values.

The authors in employ the TensorFlow framework for the detection and recognition of license plates. Using a dataset of approximately 500 images, the model achieves a detection accuracy of around 85%. The process is organized into five stages. image acquisition, preprocessing, number plate localization, character segmentation, and character recognition. Following the detection of the number plate, OCR is applied to extract and interpret the alphanumeric characters.

Result and Discussion.

To the best of our knowledge, there is no extensive public license plate dataset freely available in the English language, therefore, we collected our dataset. The detailed description of the dataset is outlined in the following.

We collected 2,800 vehicle images from the internet for this research. To enhance the training and testing phases and improve the model's generalization capabilities, various data augmentation techniques were applied. These techniques included adjustments in brightness, rotation at multiple angles, addition of noise, random scaling, grayscale conversion, shearing, and modifications to hue, contrast, and saturation. The dataset is made available for automatic license plate detection via the provided [Images-Dataset-ANPR - Google Drive]. We used LabelImg software[4] software to label the license plate of a vehicle and other objects in images are considered as background.

To augment the training and testing data, we applied a series of data augmentation techniques to each image collected from the internet, as detailed in Table 3. This approach was intended to enhance the model's generalizability, reduce overfitting, and increase resilience. The dataset was then divided as shown in Table 4, with 70% allocated for model training, 20% reserved for testing, and 10% assigned for validation.

Table 3. The data augmentation techniques optimize training, validation, and testing, enhancing dataset diversity to improve the model's robustness and accuracy in number plate detection.

Techniques	Set of Possible Values
Shear	-15 to +15
Hue	-10 to +10
Noise	Add up to 25% noise
Scaling	0.5 to 1.5
Brightness	30%
Rotation	5-,7-and 8-

[4]https://github.com/HumanSignal/labelImg

Saturation	0.5 to 1.5
Contrast	0.5 to 1.5
Grayscale	Conversion of all images into grayscale

Table 4. The description of the dataset, including training, validation, and testing images for number plate detection, after applying data augmentation techniques to the original images

Images	Count
Total	30,613
Training	21,430
Validation	3,061
Testing	6,122

The evaluation of the proposed approach measures on the proposed dataset yielded impressive performance results, with a mean average precision (mAP) of 0.9418, indicating its high accuracy and effectiveness in object detection tasks. When compared to the prior mechanism, our suggested one performs better at detecting license plates.

The performance of the network is also measured in terms of execution time. The model is tested on the entire test dataset for execution time and the average computation time is noted. It took an average of 0.31 seconds on an image for the detection and recognition of the number plate, as depicted in Table 5. We observed that the speedup of the algorithm is mainly due to the usage of depthwise separable convolution and search area optimization which is achieved using the resizing of the original images. The computation time required for the detection and recognition of a number plate on a single image using the proposed model is 0.31 seconds.

The model demonstrates effective and reliable results for number plate detection in a variety of lighting and brightness scenarios, with different resolutions and shooting angles. The mean average precision evaluates how accurate and precise the models are in locating and detecting objects. The higher mAP values show that the model performs better than other models, according to the data. It can correctly detect and categorize objects within the dataset. Specifically, the recognition results of different matrices e.g., Precision, Recall, F_1-score, and Accuracy are presented in Table 6.

The loss function continuously decreases during the testing process ultimately converging to a final value of 0.3 percent, demonstrating the model's effectiveness in precisely identifying the number plate in the testing data. The model's internal parameters have been successfully optimized, leading to a high degree of performance that is characterized by precise and accurate predictions, as indicated by the low value of loss.

The model performed effectively in the experiment, obtaining 95% accuracy at IOU 0.5, as shown in Figure 5. It indicates that when detecting number plates on our proposed dataset, the model produces accurate and effective results. By taking into account both the accuracy value and recall value of the predicted bounding boxes, the mapped value in object detection reveals the overall performance of the proposed algorithm. Smaller values of IOU offer greater mAP value but it gives numerous miss classifications. When the IOU value varies, the mean average precision value also changes. Figure 6 shows the results of the proposed method for number plate recognition on the proposed dataset, detailing the detection outcomes along with their corresponding confidence scores. The evaluation is based on an IoU threshold of 0.5.

Table 5. Performance of the proposed method on the proposed dataset, evaluated using metrics such as accuracy, precision, recall, and F_1 score. These metrics provide a comprehensive analysis of the method's accuracy and overall effectiveness in license plate detection.

Metrics	Score

Accuracy	0.95
Precision	0.95
Recall	0.90
F_1 Score	0.92

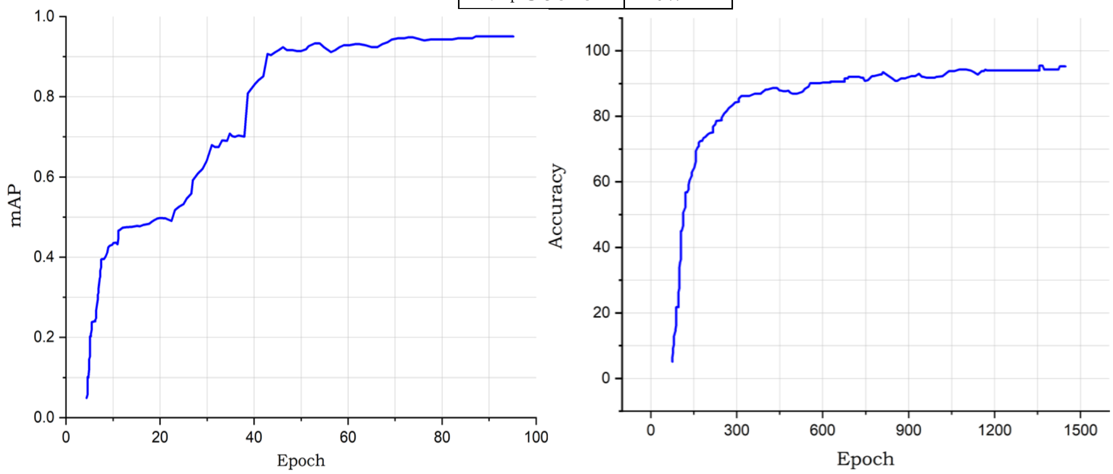

Figure 5. The performance metrics of the proposed algorithm. (a) shows the mean average precision (mAP) of 0.9418 on the testing dataset, demonstrating the model's performance in number plate detection and recognition, and (b) shows the model's accuracy of 95%, emphasizing its effectiveness in correctly detecting number plates.

In Table 7, we present a comparative analysis between our model and the existing model to provide a comprehensive assessment of our approach.

Failure Case Analysis.

The model performs well on most images, effectively detecting license plates in various conditions. However, certain challenging scenarios lead to reduced accuracy. Specifically, for images that are rotated above 45° or excessively tilted, the model still detects the license plate but with a lower confidence score, as shown in the figure below. Additionally, in cases where the image is blurry, has low lighting, or has poor focus, the model struggles to accurately recognize the text, leading to reduced detection accuracy, misclassification, or partial recognition of the license plate characters. These factors impact the overall performance and reliability of the model in real-world scenarios.

Table 6. Comparison with existing model based on accuracy to evaluate the effectiveness of the proposed model.

Method	Recognition score
Tote et al.	85%
Proposed method	95%

In future work, we aim to deploy our proposed ANPR model on low-power edge devices and integrate it with IoT-based traffic monitoring systems to enhance real-time vehicle identification. Given the lightweight nature of MobileNet-SSD, it is well-suited for resource-constrained environments such as embedded AI cameras, Raspberry Pi, and other edge computing platforms. Our future research will focus on optimizing the model further to reduce inference time and power consumption while maintaining high accuracy. Additionally, we plan to develop an IoT-based framework where multiple edge devices can communicate with a centralized cloud system, enabling real-time traffic surveillance, automated toll collection, and law enforcement applications. This integration will allow seamless data transmission, adaptive model updates, and improved system scalability, making intelligent transportation systems more efficient and responsive.

Conclusion. This paper presents a lightweight deep-learning algorithm to recognize a vehicle's number plate. The proposed framework employs a pre-trained MobileNetv2 SSD model, fine-tuned on a dataset we collected from the internet and augmented with different data enhancement methods. The proposed algorithm effectively identifies the letters and numbers on number plates using an OCR technique and provides accurate results in real-world scenarios. At 0.5 IoU, the model achieves a mean average accuracy of 0.95%. The experimental results on our proposed dataset demonstrate the effectiveness and high accuracy of our approach. In the future, we plan to improve the detection accuracy of small objects using super-pixel techniques to deal with real-world scenarios.

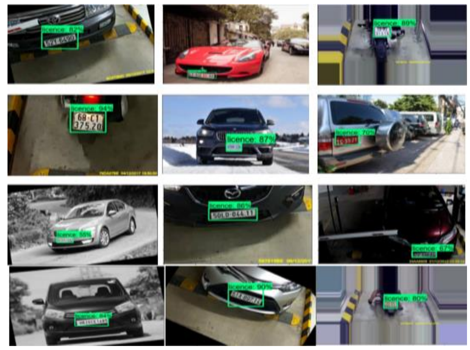

Figure 6. Results of the proposed method for number plate recognition on the proposed dataset, detailing the detection outcomes along with their corresponding confidence scores. The evaluation is based on an IoU threshold of 0.5.

Acknowledgment. This research was partially supported by the Higher Education Commission Pakistan within the project "HumCare. Human Activity Analysis in Healthcare" (Grant Number. 15041).

Author's Contribution. Conceptualization, F.S., N.A., M.H.K. and M.S.F.; methodology, F.S., and M.H.K.; software, F.S., and M.S.F.; validation, F.S., N.A.; investigation, N.A., M.H.K., and M.S.F.; writing—original draft preparation, F.S.; writing—review and editing, N.A., M.H.K., and M.S.F.; supervision, M.H.K. and M.S.F.; project administration, M.H.K. and M.S.F. All authors have read and agreed to the published version of the manuscript.

Conflict of interest. The authors declare no conflicts of interest.

Project details. This research was conducted under the project titled "HumCare. Human Activity Analysis in Health Care" under grant number. 15041.

References.

[1] T. S. Muhammad Attique Khan, Muhammad Sharif, Muhammad Younus Javed, Tallha Akram, Mussarat Yasmin, "License number plate recognition system using entropy-based features selection approach with SVM," *Inst. Eng. Technol.*, vol. 12, no. 2, pp. 200–209, 2018, doi. https.//doi.org/10.1049/iet-ipr.2017.0368.

[2] D. C. Luvizon, B. T. Nassu, and R. Minetto, "Vehicle speed estimation by license plate detection and tracking," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 6563–6567, 2014, doi. 10.1109/ICASSP.2014.6854869.

[3] S. Mahalakshmi and R. Sendhil Kumar, "Smart Toll Collection Using Automatic License Plate Recognition Techniques," *Commun. Comput. Inf. Sci.*, vol. 805, pp. 34–41, 2018, doi. 10.1007/978-981-13-0755-3_3.

[4] P. Saini, K. Bidhan, and S. Malhotra, "A Detection System for Stolen Vehicles Using Vehicle Attributes with Deep Learning," *Proc. IEEE Int. Conf. Signal Process. Control*, vol. 2019-October, pp. 251–254, Oct. 2019, doi. 10.1109/ISPCC48220.2019.8988389.

[5] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR). A state-of-the-art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, 2013, doi. 10.1109/TCSVT.2012.2203741.

[6] S. W. Junqing Tang, Li Wan, Jennifer Schooling, Pengjun Zhao, Jun Chen, "Automatic number plate recognition (ANPR) in smart cities. A systematic review on technological

advancements and application cases," *Cities*, vol. 129, p. 103833, 2022, doi. https.//doi.org/10.1016/j.cities.2022.103833.
[7] T. T. & L. V. G. Herbert Bay, "SURF. Speeded Up Robust Features," *Comput. Vis. – ECCV 2006*, pp. 404–417, 2006, doi. https.//doi.org/10.1007/11744023_32.
[8] T. D. and J. M. R. Girshick, J. Donahue, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA. Accessed. Mar. 09, 2025. [Online]. Available. https.//ieeexplore.ieee.org/document/6909475
[9] A. Panning, A. K. Al-Hamadi, R. Niese, and B. Michaelis, "Facial expression recognition based on Haar-like feature detection," *Pattern Recognit. Image Anal.*, vol. 18, no. 3, pp. 447–452, Sep. 2008, doi. 10.1134/S1054661808030139/METRICS.
[10] E. M. Rosario Capparuccia, Renato De Leone, "Integrating support vector machines and neural networks," *Neural Networks*, vol. 20, no. 5, pp. 590–597, 2007, doi. https.//doi.org/10.1016/j.neunet.2006.12.003.
[11] Y.-H. Wen, X.-Z., Fang, W., Zheng, "An algorithm based on Haar-like features and improved AdaBoost classifier for vehicle recognition," ACTA ELECTONICA SINICA. Accessed. Mar. 09, 2025. [Online]. Available. https.//www.researchgate.net/publication/290547288_An_algorithm_based_on_Haar-like_features_and_improved_AdaBoost_classifier_for_vehicle_recognition
[12] M. M. Dehshibi and R. Allahverdi, "Persian V ehicle License Plate Recognition Using Multiclass Adaboost," *Int. J. Comput. Electr. Eng.*, pp. 355–358, 2012, doi. 10.7763/IJCEE.2012.V4.511.
[13] G. E. H. Alex Krizhevsky, Ilya Sutskever, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi. https.//doi.org/10.1145/306538.
[14] S. L. Chen, C. Yang, J. W. Ma, F. Chen, and X. C. Yin, "Simultaneous End-to-End Vehicle and License Plate Detection with Multi-Branch Attention Neural Network," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3686–3695, Sep. 2020, doi. 10.1109/TITS.2019.2931791.
[15] C. H. Lin and Y. Li, "A License Plate Recognition System for Severe Tilt Angles Using Mask R-CNN," *Int. Conf. Adv. Mechatron. Syst. ICAMechS*, vol. 2019-August, pp. 229–234, Aug. 2019, doi. 10.1109/ICAMECHS.2019.8861691.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once. Unified, Real-Time Object Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 779–788, Jun. 2015, doi. 10.48550/arxiv.1506.02640.
[17] R. Agrawal, N. Gupta, and A. S. Jalal, "CACBL-Net. a lightweight skin cancer detection system for portable diagnostic devices using deep learning based channel attention and adaptive class balanced focal loss function," *Multimed. Tools Appl.*, pp. 1–24, Jun. 2024, doi. 10.1007/S11042-024-19485-1/METRICS.
[18] J. C. M.-S. & E. P. Nicolás Hernández-Díaz, Yersica C. Peñaloza, Y. Yuliana Rios, "A computer vision system for detecting motorcycle violations in pedestrian zones," *Multimed. Tools Appl.*, 2024, doi. https.//doi.org/10.1007/s11042-024-19356-9.
[19] C.-Y. F. & A. C. B. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, "SSD. Single Shot MultiBox Detector," *Comput. Vis. – ECCV 2016*, pp. 21–37, 2016, doi. https.//doi.org/10.1007/978-3-319-46448-0_2.
[20] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet. An Extremely Efficient Convolutional Neural Network for Mobile Devices," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 6848–6856, Dec. 2018, doi. 10.1109/CVPR.2018.00716.
[21] K. K. Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, "Squeezenet. Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model

size," *arXiv.1602.07360*, 2016, doi. https.//doi.org/10.48550/arXiv.1602.07360.
[22] A. G. Howard *et al.*, "MobileNets. Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, doi. 10.48550/arxiv.1704.04861.
[23] T. Y. Lin *et al.*, "Microsoft COCO. Common Objects in Context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi. 10.1007/978-3-319-10602-1_48.
[24] Amjad, F., Khan, M. H., Nisar, M. A., Farid, M. S., & Grzegorzek, M. (2021). A comparative study of feature selection approaches for human activity recognition using multimodal sensory data. Sensors, 21(7), 2368.
[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2001, doi. 10.1109/CVPR.2001.990517.
[26] N. O. Yaseen, S. G. S. Al-Ali, and A. Sengur, "An Efficient Model for Automatic Number Plate Detection using HOG Feature from New North Iraq Vehicle Images Dataset," *1st Int. Informatics Softw. Eng. Conf. Innov. Technol. Digit. Transform. IISEC 2019 - Proc.*, Nov. 2019, doi. 10.1109/UBMYK48245.2019.8965573.
[27] A. K. Bachchan, A. Gorai, and P. Gupta, "Automatic License Plate Recognition Using Local Binary Pattern and Histogram Matching," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10362 LNCS, pp. 22–34, 2017, doi. 10.1007/978-3-319-63312-1_3.
[28] S. M. S. Zahedi Morteza, "License plate recognition system based on SIFT features," *Procedia Comput. Sci.*, vol. 3, pp. 998–1002, 2011, doi. https.//doi.org/10.1016/j.procs.2010.12.164.
[29] I. S. Ahmad, B. Boufama, P. Habashi, W. Anderson, and T. Elamsy, "Automatic license plate recognition. A comparative study," *2015 IEEE Int. Symp. Signal Process. Inf. Technol. ISSPIT 2015*, pp. 635–640, Jan. 2016, doi. 10.1109/ISSPIT.2015.7394415.
[30] B. Pechiammal and J. A. Renjith, "An efficient approach for automatic license plate recognition system," *ICONSTEM 2017 - Proc. 3rd IEEE Int. Conf. Sci. Technol. Eng. Manag.*, vol. 2018-January, pp. 121–129, Jun. 2017, doi. 10.1109/ICONSTEM.2017.8261267.
[31] J. W. Zheng, Danian, Zhao, Yannan, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, 2005, doi. https.//doi.org/10.1016/j.patrec.2005.04.014.
[32] Lubna, M. F. Khan, and N. Mufti, "Comparison of various edge detection filters for ANPR," *2016 6th Int. Conf. Innov. Comput. Technol. INTECH 2016*, pp. 306–309, Feb. 2017, doi. 10.1109/INTECH.2016.7845061.
[33] Ashfaq, N., Sadiq, I., Junaid, M., Khan, M. H., & Farid, M. S. Optimizing Human Activity Recognition with Ensemble Deep Learning on Wearable Sensor Data.
[34] Z. Zhao, S. Yang, and X. Ma, "Chinese license plate recognition using a convolutional neural network," *Proc. - 2008 Pacific-Asia Work. Comput. Intell. Ind. Appl. PACIIA 2008*, vol. 1, pp. 27–30, 2008, doi. 10.1109/PACIIA.2008.196.
[35] N. P. Ap, T. Vigneshwaran, M. S. Arappradhan, and R. Madhanraj, "Automatic Number Plate Detection in Vehicles using Faster R-CNN," *2020 Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2020*, Jul. 2020, doi. 10.1109/ICSCAN49426.2020.9262400.
[36] S. Du Chunling Tu, "A hierarchical RCNN for vehicle and vehicle license plate detection and recognition," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, p. 731, 2022, doi. http.//doi.org/10.11591/ijece.v12i1.pp731-737.
[37] Y. L. Jifeng Dai, "R-FCN. Object Detection via Region-based Fully Convolutional Networks," Neural Information Processing Systems. Accessed. Mar. 09, 2025. [Online]. Available. https.//www.semanticscholar.org/paper/R-FCN%3A-Object-Detection-via-Region-based-Fully-Dai-Li/b724c3f7ff395235b62537203ddeb710f0eb27bb

[38] S. Poojary, M. N. Gaunkar, B. Dessai, M. Ayalli, V. K. N. Pawar, and S. Aswale, "Deep Learning Methods for Automatic Number Plate Recognition System. A Review," *Proc. 3rd Int. Conf. Intell. Eng. Manag. ICIEM 2022*, pp. 19–25, 2022, doi. 10.1109/ICIEM54221.2022.9853092.
[39] R. Laroca *et al.*, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018-July, Oct. 2018, doi. 10.1109/IJCNN.2018.8489629.
[40] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, and Q. Liu, "A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 82–95, Jun. 2019, doi. 10.1109/MITS.2019.2903518.
[41] S. Yonetsu, Y. Iwamoto, and Y. W. Chen, "Two-Stage YOLOv2 for Accurate License-Plate Detection in Complex Scenes," *2019 IEEE Int. Conf. Consum. Electron. ICCE 2019*, Mar. 2019, doi. 10.1109/ICCE.2019.8661944.
[42] Z. A. H. and M. N. A. W. Y. Y. Lee, "License Plate Detection Using Convolutional Neural Network–Back to the Basic With Design of Experiments," *IEEE Access*, vol. 10, pp. 22577–22585, 2022, doi. 10.1109/ACCESS.2022.3153340.
[43] J. Y. Sung, S. B. Yu, and S. H. P. Korea, "Real-time Automatic License Plate Recognition System using YOLOv4," *2020 IEEE Int. Conf. Consum. Electron. - Asia, ICCE-Asia 2020*, Nov. 2020, doi. 10.1109/ICCE-ASIA49877.2020.9277050.
[44] A. Fathi, B. Moradi, I. Zarei, and A. Shirbandi, "Deep learning based System for automatic motorcycle license plates detection and recognition," *Signal, Image Video Process.*, vol. 18, no. 12, pp. 8869–8879, Dec. 2024, doi. 10.1007/S11760-024-03514-5/METRICS.
[45] P. C. Janya Sainui, Chaichan Thepporn, "Thai License Plate Recognition using SSD MobileNet and EasyOCR," *IPMV '24 Proc. 2024 6th Int. Conf. Image Process. Mach. Vis.*, pp. 36–41, 2024, doi. https.//doi.org/10.1145/3645259.364526.
[46] B. Rohan Reddy, G. Swathi, E. Himesh, and S. Southarrajan, "Encompassing YOLOv8 and EfficientNet B7 for Automatic License Plate Identification," *Proc. - 4th IEEE 2023 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2023*, pp. 694–700, 2023, doi. 10.1109/ICCCIS60361.2023.10425775.
[47] K. W. Leong, H. Nisar, V. V. Yap, K. H. Yeap, and P. K. Lo, "License Plate Detection Using Deep Learning Object Detection Models," *Proc. 9th Int. Conf. Comput. Commun. Eng. ICCCE 2023*, pp. 377–382, 2023, doi. 10.1109/ICCCE58854.2023.10246050.
[48] I. Slimani, A. Zaarane, A. Hamdoun, and I. Atouf, "Vehicle license plate localization and recognition system for intelligent transportation applications," *2019 6th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2019*, pp. 1592–1597, Apr. 2019, doi. 10.1109/CODIT.2019.8820446.
[49] M. Li, T. Sun, and H. Liu, "Image recognition of steel plate based on an improved support vector machine," *2018 IEEE Int. Conf. Inf. Autom. ICIA 2018*, pp. 1411–1415, Aug. 2018, doi. 10.1109/ICINFA.2018.8812381.
[50] M. H. Khan, K. Shirahama, M. S. Farid, and M. Grzegorzek, "Multiple human detection in depth images," *2016 IEEE 18th Int. Work. Multimed. Signal Process. MMSP 2016*, Jan. 2017, doi. 10.1109/MMSP.2016.7813385.
[51] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2309–2319, Nov. 2000, doi. 10.1109/25.901900.
[52] X. Lu, X. Ling, and W. Huang, "Vehicle license plate character recognition," *Proc. 2003 Int. Conf. Neural Networks Signal Process. ICNNSP'03*, vol. 2, pp. 1066–1069, 2003, doi. 10.1109/ICNNSP.2003.1281053.
[53] M. G. Muhammad Hassan Khan , Julien Helsper, Muhammad Shahid Farid, "A computer vision-based system for monitoring Vojta therapy," *Int. J. Med. Inform.*, vol. 113, pp. 85–95, 2018, doi. https.//doi.org/10.1016/j.ijmedinf.2018.02.010.

[54] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic License Plate Recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004, doi. 10.1109/TITS.2004.825086.
[55] Q. H. Jiao, Jianbin, Ye, Qixiang, "A configurable method for multi-style license plate recognition," *Pattern Recognit.*, vol. 42, no. 3, pp. 358–369, 2009, doi. https.//doi.org/10.1016/j.patcog.2008.08.016.
[56] K. Vijayalakshmi, M. Dhanamalar, V. A. Lepakshi, and S. Jamtsho, "Smart Checkpoint Management System for Automatic Number Plate Recognition in Bhutan Vehicles Using OCR Technique," *SN Comput. Sci.*, vol. 5, no. 5, pp. 1–8, Jun. 2024, doi. 10.1007/S42979-024-02905-2/METRICS.
[57] Ashfaq, N., Khan, M. H., & Nisar, M. A. (2024). Identification of optimal data augmentation techniques for multimodal time-series sensory data. A framework. Information, 15(6), 343.
[58] M. Abadi *et al.*, "TensorFlow. Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, Accessed. Jun. 16, 2024. [Online]. Available. https.//arxiv.org/abs/1603.04467v2
[59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, Dec. 2016, doi. 10.1109/CVPR.2016.308.
[60] H. Seya and D. Shiroi, "Lecture 6.5-rmsprop. Divide the Gradient by a Running Average of Its Recent Magnitude," *COURSERA Neural Networks Mach. Learn.*, vol. 4, no. 2, pp. 26-, Apr. 2012, doi. 10.1111/GEAN.12283.
[61] Batool, S., Khan, M. H., & Farid, M. S. (2024). An ensemble deep learning model for human activity analysis using wearable sensory data. *Applied Soft Computing*, 111599.
[62] N. Klingler, "Automatic Number Plate Recognition (ANPR)," viso.ai. Accessed. Mar. 09, 2025. [Online]. Available. https.//viso.ai/computer-vision/automatic-number-plate-recognition-anpr/
[63] V. . Kaur, E.K., Banga, "Number plate recognition using ocr technique," *Int. J. Res. Eng. Technol.*, vol. 2, no. 9, 2013, [Online]. Available. https.//ijret.org/volumes/2013v02/i09/IJRET20130209042.pdf
[64] M. T. Qadri and M. Asif, "Automatic number plate recognition system for vehicle identification using optical character recognition," *2009 Int. Conf. Educ. Technol. Comput. ICETC 2009*, pp. 335–338, 2009, doi. 10.1109/ICETC.2009.54.
[65] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 9357–9366, Jun. 2019, doi. 10.1109/CVPR.2019.00959.
[66] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2999–3007, Dec. 2017, doi. 10.1109/ICCV.2017.324.
[67] A. D. P. Tote, Anuj S., Pardeshi, Sujit S., "Automatic number plate detection using TensorFlow in Indian scenario. An optical character recognition approach," *Mater. Today Proc.*, vol. 72, no. 3, pp. 1073–1078, 2023, doi. https.//doi.org/10.1016/j.matpr.2022.09.165.