Journal of Engineering Science and Technology
Vol. 20, No. 1 (2025) 250 - 270

INTEGRATION OF YOLOV8 AND PADDLEOCR FOR CAR PLATE RECOGNITION SYSTEM IN DIVERSE ENVIRONMENTS: MALAYSIA

LIM ZHENG YOU[1], BONG ZI JUN[1],
PANG YING HAN[1,*], OOI SHIH YIN[1], KHOH WEE HOW[1], HIEW FU SAN[2]

[1]Faculty of Information Science and Technology, Multimedia University,
Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia
[2]Infineon Technologies, Free Trade Zone, Batu Berendam, 75350 Melaka, Malaysia
*Corresponding Author: yhpang@mmu.edu.my

Abstract

An efficient car plate recognition system is significant for traffic management. However, diverse environmental factors such as poor illumination and different weather conditions, may pose challenges to car plate recognition. The efficiency of object detection and character recognition serves as a route map toward accomplishing desired outcomes within these constraints. This study devises a car plate recognition system by integrating the new variant of You-Only-Look-Once, i.e. YOLOv8, for car plate detection and PaddleOCR for extraction and recognition of the detected car plate. A self-collected database containing more than 600 car plates is collected and employed to assess the efficacy of the developed system. In this study, comprehensive experiments are carried out to assess the effectiveness of the developed system under different environmental conditions in Malaysia, including normal (day), nighttime, and rainy conditions. The empirical results exhibit the superior performance of the proposed system across diverse conditions. In conclusion, we can deduce that the proposed system is able to demonstrate robustness and reliability in car plate recognition, even under challenging environmental conditions. The integration of YOLOv8 and PaddleOCR in the car plate recognition system allows efficient detection and recognition of car plates, effectively dealing with occlusions and complex backgrounds. In future work, the system will be tested under a broader array of environmental conditions encountered in other countries, including foggy, dusty, snowfall environments, etc., to fortify its robustness in real-world scenarios.

Keywords: Artificial intelligence, Car plate recognition, Environmental conditions, Object detection, Optical character recognition, PaddleOCR, YOLOv8.

1. Introduction

The rise in the deployment of traffic monitoring cameras and the number of vehicles has resulted in the prevalent popularity of car plate recognition. The integration of vehicles into daily life underscores the importance of efficient car plate recognition systems, crucial for vehicle monitoring, law enforcement, traffic management, and automated toll collection [1]. It is very tough to check and monitor each vehicle on the road in a manual manner as it is time-consuming, labour-intensive, and erroneous. Developing intelligent transport systems is an active research area today. Intelligent transport system aims to provide users with smart traffic coordination and safe transport networking. A reliable car plate recognition system is crucial in the intelligent transport system for parking payment, vehicle tracking, road accident detection and controlling, over-speed vehicle detection, and others. However, conventional recognition methods often struggle with challenging illumination conditions such as poor lighting and adverse weather [2].

Rahman et al. highlighted that existing car plate recognition systems face difficulties in accurately detecting and recognizing plates under varying lighting conditions [3]. Similarly, Zhang et al. reiterated the influence of poor lighting, including dark environments and rainy conditions, on the effectiveness of traditional car plate recognition methods, leading to compromising performance and reliability [4].

Unfortunately, our country Malaysia's climate is equatorial, which is hot, humid, and rainy throughout the year. The rainfall is abundant throughout the year. From the research in 2015 [5], there was a drastic annual rainfall increase with a 95% confidence level along with Northeast monsoon season rainfall. Subsequent studies have continued to highlight this trend. There was a rise in precipitation in Malaysia from 3287.92 mm in 2021 to 3484.51 mm in 2022 [6]. With an average of 212 rainy days annually in Malaysia [7], shown in Fig. 1, car plate recognition operations encounter a great challenge in coping with such climatic conditions in Malaysia.

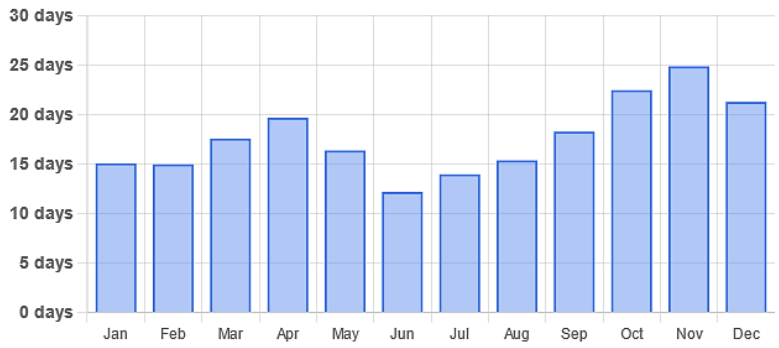


Fig. 1. Rainy days of each month in Kuala Lumpur, Malaysia [7].

Therefore, this work aims to develop a reliable car plate recognition system for the diverse environmental conditions prevalent in Malaysia, including poor lighting and rainy weather. The devised system integrates the new variant of You-Only-Look-Once, namely YOLOv8, a state-of-the-art object detection framework, with PaddleOCR, a robust optical character recognition tool. The pipeline of the proposed

car plate recognition system is illustrated in Fig. 2. The process is categorised into 3 phases: (1) data collection, (2) model training, and (3) model performance testing. The data collection phase involves collecting a diverse dataset of car plate images under different environmental conditions, such as normal illumination, poor lighting, and rainy conditions.

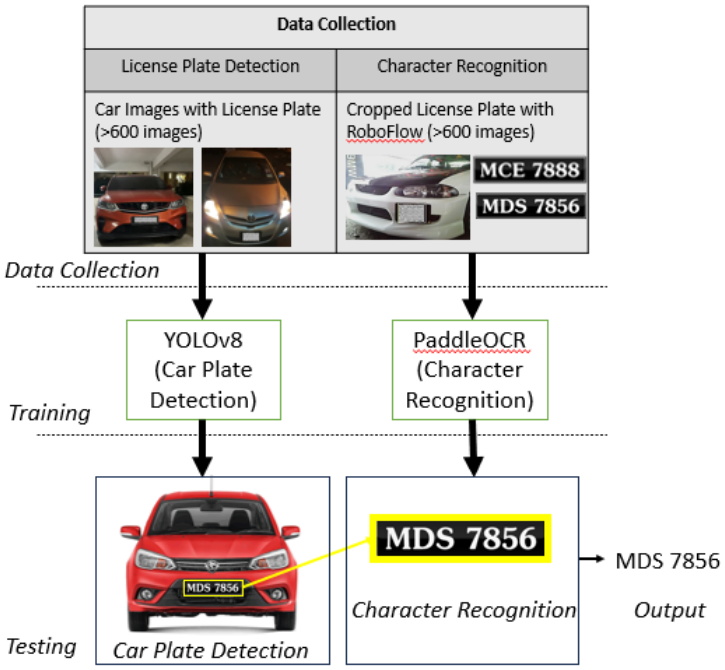


Fig. 2. The pipeline of the proposed car plate recognition system.

In the model training phase, YOLOv8 is adopted for car plate detection and segmentation. YOLOv8 is a real-time object detection framework that could accurately detect and segment car plates from complex scenes, providing a robust foundation for subsequent analysis [8]. PaddleOCR is an integral component of the car plate recognition system, which was adopted for its excellent performance in character recognition. Its deep learning techniques facilitate the reliable extraction of text characters from the car plate images. This precise extraction of plate numbers empowers accurate recognition of vehicles [9]. By amalgamating these technologies, our proposed system achieves unprecedented levels of performance and resilience across a spectrum of lighting scenarios.

Additionally, this study contributes to the advancement of car plate recognition technology by conducting various tests under different environmental conditions prevalent in Malaysia. These include normal illumination, poor lighting, and rainy conditions. Through rigorous evaluation, the integrated model demonstrates its prediction efficacy and robustness. The results highlight its ability to outperform traditional approaches in challenging real-world environments. The developed model paves the way for enhanced applications in law enforcement, traffic management, and beyond, thereby addressing a critical need in modern society.

2. Related Work

Sasi et al. (2017) proposed an intelligent approach for automatic car number plate recognition by employing three algorithms: Ant Colony Optimization (ACO) for plate localization, character segmentation, and extraction [10]. The system is designed to enhance road mishap monitoring and management, it finds applications in toll collection, speed control, and traffic management. The dataset comprises noisy, blurred, night, and daylight license plate images, with preprocessing involving RGB to grayscale conversion, noise reduction, and adaptive histogram equalization for contrast enhancement. Feature extraction utilizes Kohonen neural networks for character segmentation and extraction.

The work compares the ACO algorithm's performance with the existing edge detection methods and presents the flow graph of the license plate detector and plate localization. The algorithm architecture adopts a hierarchical combined classification method based on inductive learning and Support Vector Machine (SVM) for individual character recognition. Performance evaluation metrics include Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE). The empirical results demonstrate that the proposed recognition system exhibits superior performance compared to the existing license plate recognition systems. The character recognition algorithm yields a success rate of 79.84%, with plans to enhance recognition accuracy through the utilization of two SVMs for digits and characters.

Yogheedha et al. [11] proposed an automatic vehicle license plate recognition system utilizing image processing techniques and OCR methods to bolster the efficiency and reliability of the security management system at UniMAP. The dataset comprises images of 14 cars captured using an iPhone 5s smartphone, saved in jpeg format. Preprocessing entails Otsu's thresholding, noise removal, and image cropping to extract the segmented area of interest in license plate images. Feature extraction employs bounding box features to isolate characters from the background, with techniques including colour conversion, image segmentation, noise removal, and template matching for optical character recognition. The algorithm architecture involves a process to identify data from segmented license plates using bounding boxes and template matching methods for character recognition. The encouraging performance, successfully recognizing license plates for 13 out of 14 cars, underscores its efficacy in real-world scenarios.

Lin et al. [12] elevated license plate recognition systems through the application of deep learning techniques. The authors employed the Coco 2017 dataset for vehicle detection and collected 1,779 positive plate samples along with 5,401 negative samples for license plate detection. The proposed preprocessing involves preparing labelled samples and training the LPRCNN model.

The proposed feature extraction utilizes Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT), while YOLOv2 is employed for vehicle detection and LPRCNN for character recognition. The algorithm architecture integrates CNN, classifier, and bounding box regression into a single model. The results reveal a 94.23% plate detection rate and 99.2% character recognition accuracy, with an average 93.47% accuracy in recognising the car plate number, which surpasses traditional systems in efficacy.

Ali et al. [13] proposed an efficient automatic vehicle identification solution specifically tailored for Indian conditions, leveraging an Android app and cost-

effective cameras. Engineered for real-time operation, the system ensures swift and precise results. The Machine Learning Model is trained on a dataset comprising vehicle license plate images, with preprocessing techniques including text-line construction, binarization, and vertical projection facilitating accurate character segmentation. The feature extraction proposed in this work entails deriving statistical features from the segmented characters, while a multilevel classification RBF neural network is utilized to recognize feature vectors. The algorithm architecture integrates frontend and backend functionalities using Python, OpenCV, and Machine Learning. The proposed method attains a high precision rate of 98.4% and a recall rate of 95.1%.

In the field of Automatic License Plate Recognition (ALPR), Silva and Jung [14] develop systems capable of accurately detecting and recognizing license plates across diverse scenarios and camera setups. This involves utilizing four datasets covering various viewing angles, illumination conditions, and license plate regions for comprehensive evaluation. Preprocessing techniques, including vehicle detection and license plate recognition modules, along with geometrical and photometric augmentation during training, enhance the system's adaptability to different conditions.

Feature extraction relies on the Warped Planar Object Detection Network (WPOD-NET) and deep Convolutional Neural Networks (CNNs), with modifications such as YOLO networks and recurrent layers improving character recognition. Evaluation metrics such as precision (PE), recall (RE), and Intersection over Union (IoU) provide quantitative insights into system performance, showcasing advancements like WPOD-NET that contribute to enhanced recognition performance. The proposed WPOD-NET achieves 97.99% accuracy on average. These efforts underscore ongoing progress in ALPR technology, promising broader practical applications.

YOLO (You-Only-Look-Once) is a well-known series of real-time object detection models, and YOLOv8 is the most recent version. YOLOv8 enhances both accuracy and speed, leveraging the advancements of its previous versions. The system utilizes an innovative structure that merges the capabilities of convolutional neural networks (CNNs) with attention mechanisms, allowing it to accurately and efficiently identify objects in images [15].

YOLOv8 is a versatile model specifically created to handle a diverse set of object detection tasks in many domains, such as traffic surveillance, security systems, and autonomous driving. The combination of its capacity to analyse photos at various scales, along with the utilization of sophisticated data augmentation techniques, enables it to maintain its effectiveness in a wide range of environmental conditions. The model's exceptional performance in both large-scale and small-object recognition tests underscores its versatility and efficacy in intricate circumstances.

PaddleOCR is an open-source OCR technology developed by Baidu that uses deep learning to extract and detect text from photos. The pipeline provided offers a thorough range of functions, such as text detection, text recognition, and layout analysis, which greatly enhances its efficiency for a variety of OCR applications. PaddleOCR is notable for its versatility and ability to handle many languages, typefaces, and character sets [16]. This makes it a desirable option for practical uses, including document processing, automated data entry, and license plate

recognition. The system is designed to be very efficient and operate well on both cloud and edge devices, allowing it to be deployed on many hardware setups. PaddleOCR's connection with deep learning frameworks such as PaddlePaddle enables convenient customization and precise adjustment, rendering it suitable for specific use cases and datasets.

To date, no study has integrated the YOLOv8 object detection algorithm with PaddleOCR for the purpose of recognizing car plate numbers. This integration utilizes the unique skills of YOLOv8 and PaddleOCR algorithms to produce a robust system for identifying and recognizing automobile plates in different situations and environments. YOLOv8 excels in accurately detecting objects, while PaddleOCR has strong text recognition capabilities. Thus, they might be integrated to form an efficient solution for car plate detection and recognition.

In summary, multiple studies propose solutions to perform car plate number recognition using deep learning and machine learning methods. It shows that deep learning and machine learning models can be applied in real-life applications that involve car plate number recognition. Among the multiple techniques, WPOD-NET has the highest accuracy of 97.99% compared to the other methods. However, there is still room for improvement in detection accuracy, and the limitation lies in the need for further improvement to enhance recognition accuracy and adaptability to diverse scenarios.

3. Proposed Method

To tackle the intricate challenge of license plate recognition under diverse environmental conditions prevalent in Malaysia, this work harnesses an integration of YOLOv8 and PaddleOCR. YOLOv8 is tasked with car plate detection, while PaddleOCR excels at accurately extracting characters from license plates. YOLOv8's real-time application makes it an adequate solution for the fast and accurate detection of car plates. After car plate region detection, PaddleOCR continues the process by extracting the text characters from the detected car plates. The advanced deep learning models of PaddleOCR allow precise character extraction, even under challenging environments, such as partial obscuration of car plate characters due to rain or poor lighting. Figure 3 depicts the overall process of the developed system. There are three vital phases: data collection and preprocessing, YOLOv8 model training, and optical character recognition (OCR).

3.1. Data collection and preprocessing

A total of 600 samples were collected in this study, and the details of the data collection are discussed in Section 4.1. All car plate images used in the research are obtained with the car owners' consent to ensure compliance with privacy and ethical standards. The dataset is randomly divided into training and testing sets, where 85% of the dataset is for training and the remaining 15% is for testing. The chosen split ratio of 85% for training and 15% for testing is justified as it strikes a balance between providing the model with enough data to learn effectively while reserving a sufficient portion for evaluating its performance. The larger training set allows the model to capture diverse patterns and features in the data, leading to better generalization.

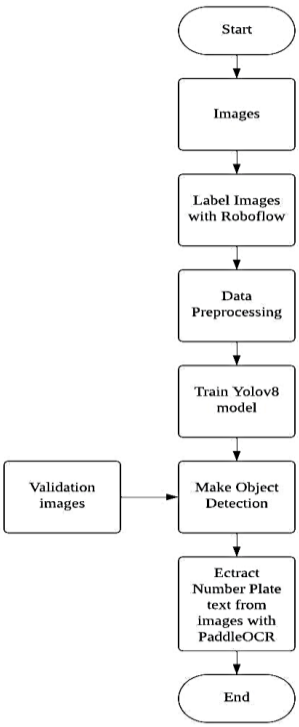


Fig. 3. Flowchart of the system development.

Meanwhile, the 15% testing set is substantial enough to provide a reliable assessment of the model's accuracy and robustness on unseen data, ensuring that the results are statistically significant without risking overfitting. For the initial step, data preprocessing plays a vital role in dataset preparation. This phase begins with data collection, image annotation, and data splitting. The captured images during the data collection process are first processed for image annotation. Roboflow Annotate is employed for efficient image annotation. Images are uploaded, and annotation tools are utilized to draw bounding boxes around objects, assigning the appropriate class [17]. The annotated data, including images and annotation files, are exported and organized into a dataset folder with "images" and "labels" subfolders. An example of annotated images using RoboFlow is shown in Fig. 4.

Fig. 4. Example of the annotated image using RoboFlow.

Then, a YAML descriptor file detailing the dataset structure is created. This YAML file is crucial for the training process, serving as input for the YOLOv8 model. Upon the completion of the data preprocessing, the organized dataset structure, annotated images, and descriptor file collectively form a comprehensive training dataset that is ready for YOLOv8 model training in the next phase.

3.2. YOLOv8 model for number plate detection

As aforementioned, YOLOv8 is a pivotal integral component in our car plate recognition system. It relies on a sophisticated network of convolutional layers and specialized blocks to achieve accurate and efficient detection. The architecture of YOLOv8 is illustrated in Fig. 5, presenting a comprehensive overview of its components and functionalities essential for license plate detection.

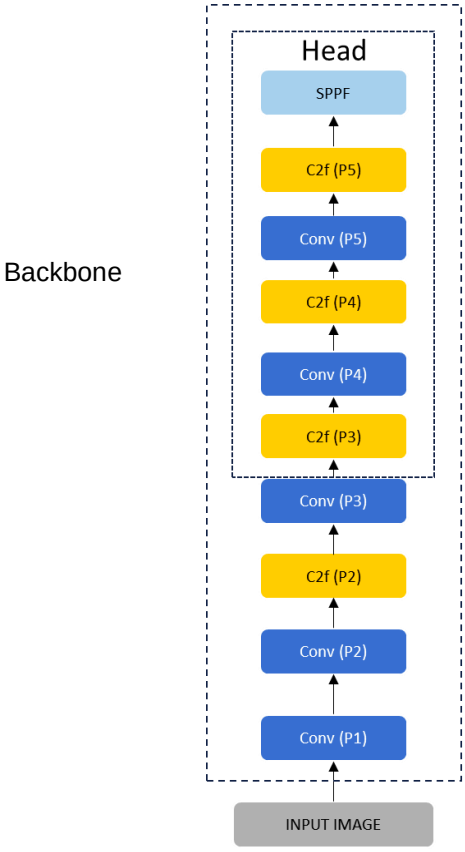


Fig. 5. Overall architecture of YOLOv8.

Figure 3 presents an overall depiction of the YOLOv8 architecture, serving as a guide to understanding its intricacies. It can be observed that the entire architecture includes convolutional layers, a cross-stage partial bottleneck block with two convolutions - fast (c2f) block, and spatial pyramid pooling – fast (SPPF) block. The architecture can be separated into two parts: head and backbone. The backbone, composed of multiple convolutional layers, down samples pixel

resolutions and extracts distinct features critical for accurate detection. The head is part of the backbone and serves as the core element of the neural network, incorporating the C2f block positioned after the P3 convolutional layer until the SPPF layer [18]. Figure 6 provides the details of the C2f block.

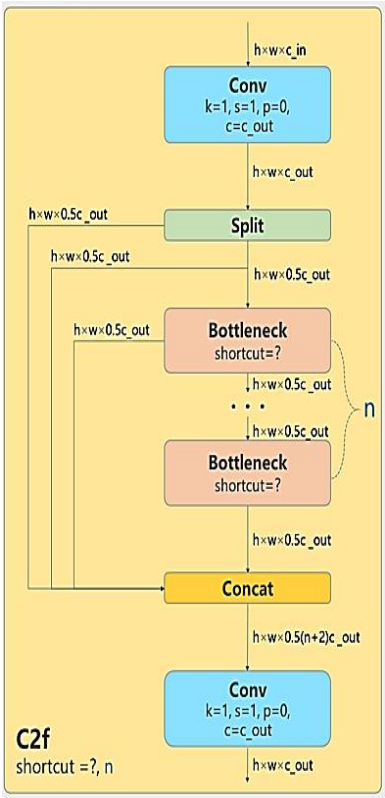


Fig. 6. C2f block architecture [14].

The C2f block is where feature maps from the convolutional block are processed. It consists of a split block, multiple bottleneck blocks, and a concat block. The bottleneck block, akin to the residual block in ResNet but featuring a bottleneck with optional shortcuts, consists of a series of convolutional blocks [18]. The C2f blocks in the backbone are C2f blocks with shortcuts. The bottleneck block architecture is shown in Fig. 7.

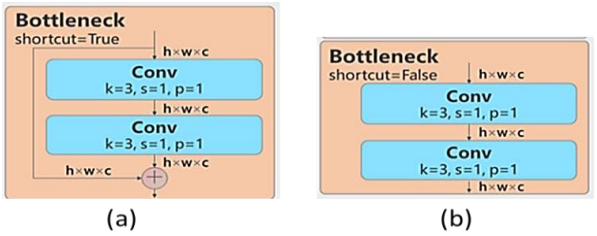


Fig. 7. Bottleneck block: (a) with shortcut (b) without shortcut [19].

The last layer of the YOLOv8 is the SPPF block. It introduces a modification of spatial pyramid pooling with heightened speed. Within SPPF, a convolutional block precedes a 32D max pooling layer, with each result concatenated just before the conclusion of SPPF, leading to a final convolutional block. The primary function of the SPPF is to generate a fixed feature representation of objects of various sizes in an image without resizing the image or introducing special information loss [20]. Fig. 8 shows the details of the SPPF block architecture.

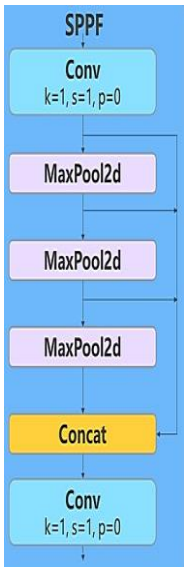


Fig. 8. SPPF block architecture [19].

The YOLOv8 head is a crucial part of the entire architecture, it is responsible for processing image features and making detections. Figure 9 illustrates the architecture of the YOLOv8 head part.

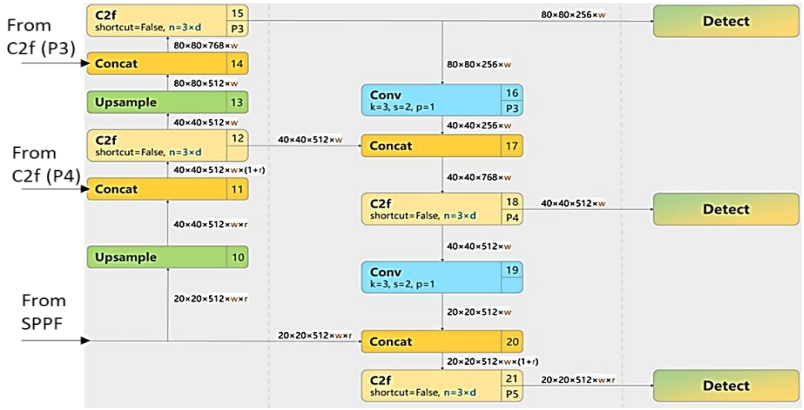


Fig. 9. YOLOv8 head block architecture [19].

Addressing the YOLOv8 head incorporates an up-sample layer to enhance the feature map resolution of the SPPF. This is to align it with the feature map resolution of the C2f block. The up-sampled feature map is then merged with the features from the C2f block using 'concatenation.' Through this concatenation process, the number of channels is aggregated while the resolution remains unaltered.

Subsequently, the subsequent C2f block does not imply a shortcut, and the value of 'N' equals 3 multiplied by the depth and the resolution. The C2f feature map undergoes up-sampling to match the resolution of the feature from the C2f block. By employing 'concatenation,' the up-sampled feature map is combined with the features from the C2f block. Following this, the C2f block diminishes the channel size of the feature map, serving as input for the detect block [21]. The architecture of the detect block is illustrated in Fig. 10.

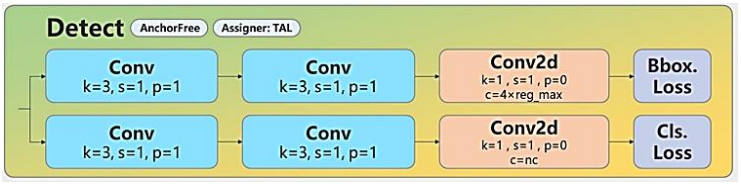


Fig. 10. Detect block architecture [19].

Detect block is designed to detect small objects. The output of this block is further utilized as input for the convolutional block. This iterative process, as elucidated above, continues to relay input to the detect block size of the feature map, serving as input for the detect block in detecting small objects [22]. The output of this block is further utilized as input for the convolutional block. Following this, the head consolidates features and executes detections based on loss metrics, encompassing box loss, class loss, and object loss. The output of the YOLOv8 generates the predictions based on the X-Y coordinates of the anchor box of the detected object [23]. Figure 11 depicts the anchor box predicted by YOLOv8, and Fig. 12 depicts the outcome of YOLOv8, highlighting the successful extraction of the license plate from the car image.

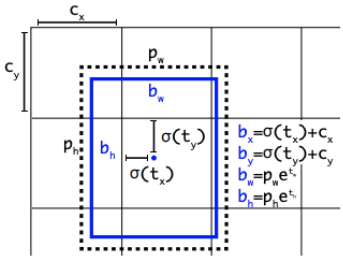


Fig. 11. Visualization of anchor box in YOLOv8 [24].

Fig. 12. Result of detected car plate by YOLOv8.

In this study, the hyperparameter settings include the initial learning rate (lr0) of 0.01, momentum of 0.937 for stabilizing training, and a weight decay of 0.0005 to prevent overfitting. The batch size is set to 16, with an image size of 640 pixels for input. The model is trained with 300 epochs, while the Intersection over Union (IoU) threshold is set at 0.2 for filtering overlapping bounding boxes, and a confidence threshold of 0.25 determines valid detections. The YOLOv8 is also set to utilize the automatic anchor box calculation.

3.3. PaddleOCR for optical character recognition (OCR)

Optical Character Recognition (OCR) allows the computing device to extract and recognize the text characters from the captured image. In this system, PaddleOCR, an advanced OCR toolkit with a deep neural network framework, is employed to recognize the alphanumerical character [25] from the number plate image extracted by the YOLOv8. Figure 13 illustrates the overall working procedure of the PaddleOCR algorithm.

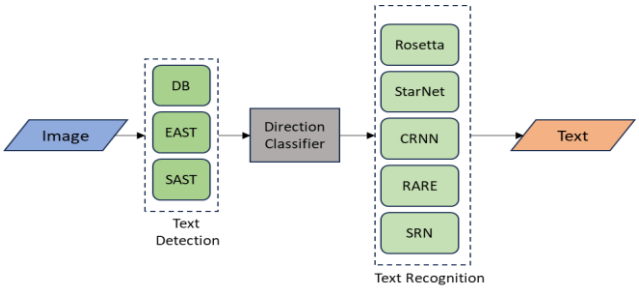


Fig. 13. Overall working procedure of PaddleOCR algorithm.

PaddleOCR meticulously follows a systematic approach for OCR, commencing with the input of an image containing text. The PaddleOCT starts with detecting the text with several text detection models such as Differentiable Binarization (DB), Efficient and Accurate Scene Text Detector (EAST), and Single-Shot Arbitrarily-Shaped Text Detector (SAST) [26]. PaddleOCR accurately identifies and locates text regions within the image by delineating their positions through bounding boxes. Next, a direction classifier algorithm is applied to determine the

direction of the text. Subsequently, the system extracts text from these identified regions and processes it by utilizing a text recognition model.

The model includes several deep neural networks, namely Rosetta, StarNet, Convolutional Recurrent Neural Network (CRNN), Robust text recognizer with Automatic Rectification (RARE), and semantic reasoning network (SRN), thereby converting it into recognizable text [26]. Post-processing techniques refine the results by handling overlapping text regions and enhancing overall accuracy. The final output comprises recognized text alongside corresponding bounding boxes, providing a comprehensive insight into the location and content of text within the input image.

Upon the completion of these processes, text extraction from the cropped car plate ensues. PaddleOCR facilitates this extraction by converting the letters and numbers from the car plate images into recognizable text through optical character recognition (OCR). The output from PaddleOCR manifests as a string of numbers, thereby revealing the alphanumerical characters from the car number plate.

4. Experimental Setup and Results

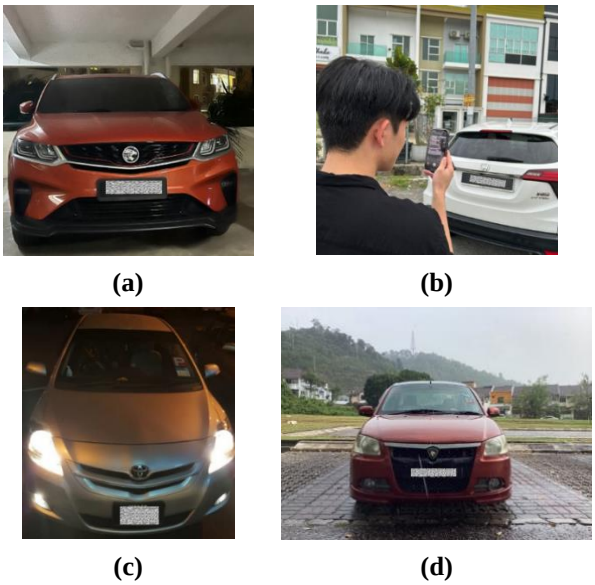
4.1. Data images

A testing procedure is designed to assess the model's performance across various environmental conditions, aiming to emulate real-world scenarios and ensure its robustness in diverse situations. Thus, the data images are collected from different weather conditions to evaluate the developed system's resilience under adverse environmental factors and to better simulate real-world conditions. Besides, the car plate images are captured from various random angles (upward/downward/from left/from right) rather than just a fixed angle directly in front of the plate.

Hence, the car images used to assess the developed system are captured under four conditions: (a) normal lighting conditions in indoor areas, (b) normal lighting conditions in outdoor areas, (c) low ambient light conditions in outdoor areas, and (d) rainy condition in outdoor areas. Figure 14 shows examples of car images captured (a) in a normal lighting condition in an indoor area, (b) under a normal lighting condition in an outdoor area (c) under a low ambient light condition in an outdoor area, and (d) under a raining condition in an outdoor area. The images are captured using two devices: a mobile phone and a laptop. The mobile phone is an iPhone 10, and the camera resolution is a dual camera with 12 megapixels. The laptop employed in this research is the Dell Latitude E7440 with a front camera of 2MP.

A total of 600 car images were collected in this research. 150 images are captured in normal indoor lighting conditions. The indoor environment includes the indoor parking of a multi-level building and under the car porch of the landed house. 150 images are captured under normal outdoor sunlight conditions. 150 images are captured under low ambient light conditions outdoors at night. 150 images were captured under rainy conditions outdoors, 75 images were captured under heavy rain, and 75 images were captured under light/drizzling conditions. 74 of the total datasets are used as the testing set to assess the developed system in this research. 20 images are randomly selected from images under normal lighting conditions in indoor areas, another 20 images are randomly selected from images under normal lighting conditions in outdoor areas, 13 images are randomly selected

from images under low ambient light conditions in outdoor areas, and 21 images are randomly selected from images under rainy conditions in outdoor areas.



(a) **(b)**

(c) **(d)**

Fig. 14. Examples of car images captured (a) in a normal lighting condition (indoor), (b) under a normal lighting condition (outdoor), (c) under a low ambient light condition (outdoor), and (d) under a raining condition (outdoor).

4.2. Results and Discussion

Prediction outcomes of the proposed model's performance evaluation across different environmental conditions are presented. Figure 15 shows the car plate recognition interface for prediction. The results detail the accuracy of car plate recognition, the number of predicted car plates, and average confidence levels are recorded in Tables 1 to 5.

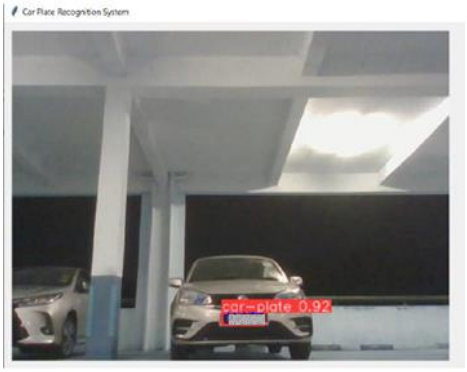


Fig. 15. Car plate recognition interface for prediction.

Table 1. Predictions under normal lighting conditions in indoor areas.

Total Number of Car Plate	Total Number of Car Plate Predicted Correctly	Accuracy (%)	Average Confidence Level
20	19	95	0.9723

Table 2. Predictions under normal lighting conditions in outdoor areas.

Total Number of Car Plate	Total Number of Car Plate Predicted Correctly	Accuracy (%)	Average Confidence Level
20	20	100	0.9826

Table 3. Predictions under rainy conditions in outdoor areas.

Total Number of Car Plate	Total Number of Car Plate Predicted Correctly	Accuracy (%)	Average Confidence Level
13	13	100	0.9893

Table 4. Predictions under low ambient light conditions in outdoor areas.

Total Number of Car Plate	Total Number of Car Plate Predicted Correctly	Accuracy (%)	Average Confidence Level
21	21	100	0.9800

The compiled results are visualized in Fig. 16. The details of the predicted car plate information can be found in Tables A-1 to A-4 in Appendix A.

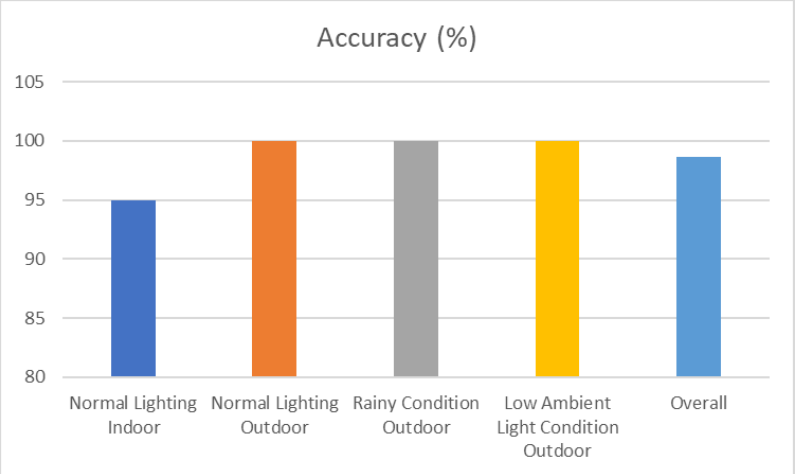


Fig. 16. Visualized bar chart for car plate prediction accuracy under different circumstances.

As observed from Table 1, the proposed model could accurately recognize car plates in 19 out of the 20 images. Despite the high accuracy, there are instances of incorrect classifications, such as "KW91**" being misclassified as "JKW9166." (Refer to Table A-1 in Appendix). The classification result for this image is still incorrect, although it has a high confidence level of 0.9594. This might be due to the mistake of the YOLOv8 as the car plate detection that used to detect and crop out the car plate from the images. The cropped car plate image might have missed out on the first character of the car plate. Thus, the PaddleOCR recognises the car plate number with such a high confidence level. Hence, incorrect result with a high confidence level reflects the detection flaw of the YOLOv8 model for car plate detection.

On the other hand, an incorrect result with low confidence levels reflects the optical character recognition flaw of the PaddleOCR. Table 2 shows the prediction results of the car plate number under normal conditions in outdoors. The empirical results show a high accuracy of 100% without error in recognizing the car plate number under normal conditions outdoors. Table 3 shows that even under nighttime conditions with low ambient lighting, the model achieves a perfect recognition rate, correctly identifying all 13 car plates in the images. This performance underscores the model's robustness even in low-light scenarios.

Lastly, Table 4 showcases the model's performance in recognizing car plates under rainy conditions. Across 21 images, the model demonstrates consistent accuracy, correctly identifying all car plates despite the challenges posed by adverse weather conditions. Figure 16 shows that the model is able to perform 100% correct prediction under the circumstances of normal lighting outdoors, rainy conditions outdoors, and low ambient light conditions outdoors. Hence, the overall accuracy of the developed model is 98.64%.

The performance of the developed car plate recognition model is also compared with other state-of-the-art works using its own datasets. Table 5 compiles the accuracy performance of the existing model with the related works.

Table 5. Performance comparison of different recognition models.

Model	Overall Accuracy(%)
Inductive learning + Support Vector Machine (SVM) [10]	79.84
Automatic Vehicle License Plate Recognition System [11]	92.85
LPRCNN +YOLOv2 [12]	93.47
IWPOD-NET [14]	97.99
YOLOv8 + PaddleOCR	98.64

The results show that our developed model with the integration of YOLOv8 and PaddleOCR yields the highest recognition accuracy of 98.64% and outperforms the other existing models. The high performance of our car plate recognition system, achieving a recognition accuracy of 98.64%, can be attributed to the advanced technologies and methodologies employed. YOLOv8, a new variant of the You-Only-Look-Once family, is highly efficient in real-time object detection, enabling the system to accurately detect car plates under various environmental conditions, including day, night, and rain. This ensures that the system can consistently locate plates regardless of the lighting or weather.

Additionally, PaddleOCR, a robust Optical Character Recognition tool, plays a crucial role in accurately extracting and recognizing the text from the detected plates, handling diverse fonts, sizes, and challenging conditions with precision. The seamless integration of YOLOv8 for detection and PaddleOCR for recognition ensures that the entire process is optimized for both accuracy and efficiency, contributing significantly to the system's overall success.

Overall, the model has exhibited commendable performance across varying environmental conditions, demonstrating its potential for real-world applications where weather conditions can significantly impact image quality and recognition accuracy. One potential limitation of the current car plate recognition system is its dependency on specific device settings and image resolutions, which may affect its accuracy and consistency across different scenarios.

Additionally, the model might be less effective when encountering car plates from regions not represented in the current dataset, such as neighbouring countries with different plate designs. To address these limitations, future enhancements could include taking photos with various devices and camera resolution settings to increase the model's adaptability. Expanding the dataset to include images of car plates from neighbouring countries like Singapore and Thailand would improve the model's ability to generalize across diverse plate settings. Furthermore, developing a lightweight version of the model could enable its deployment on a broader range of devices, enhancing accessibility and usability.

5. Conclusions

In conclusion, this project devises a robust car plate recognition system integrating YOLOv8 and PaddleOCR. The proposed model is capable of accurately identifying car plates under challenging environmental conditions. Through rigorous testing, the system exhibits commendable performance, achieving high accuracy rates of 98.64% across different lighting scenarios. This highlights its potential for real-world deployment in scenarios where weather conditions may impact traditional recognition systems. Future efforts will focus on further refining the system's capabilities, expanding testing to include additional environmental factors such as fog and snow environments, as well as with a broader array of samples. Expanding the dataset to include car plates from neighbouring countries such as Singapore and Thailand, which have distinct plate designs, along with the application of data augmentation methods, would enhance the model's generalization. Additionally, developing a lightweight version of the model could enable its deployment on a wider range of devices, making the technology more accessible and versatile.

References

1. Shashirangana, J. et al. (2020). Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9, 11203-11225.
2. Zhang, W. et al. (2022). Research on the algorithm of license plate recognition based on MPGAN haze weather. *IEICE Transactions on Information and Systems*, 105(5), 1085-1093.
3. Rahman, M.J.; Beauchemin, S.S.; and Bauer, M.A. (2020). License plate detection and recognition: An empirical study. In Arai, K.; and Kapoor, S.

(Eds.), *Advances in Computer Vision. CVC 2019. Advances in Intelligent Systems and Computing*, Springer, Cham.

4. Zhang, Y. et al. (2020). A novel deep learning based number plate detect algorithm under dark lighting conditions. *Proceedings of the IEEE 20th International Conference on Communication Technology* (*ICCT*), Nanning, China, 1412-1417.
5. Mayowa, O.O. et al. (2015). Trends in rainfall and rainfall-related extremes in the east coast of peninsular Malaysia. *Journal of Earth System Science*, 124, 1609-1622.
6. Trading Economics. Malaysia average precipitation. Retrieved April 25, 2024, from https://tradingeconomics.com/malaysia/precipitation.
7. Weather and Climate. Monthly rainy days in Kuala Lumpur. Retrieved April 25, 2024, from https://weather-and-climate.com/average-monthly-Rainy-days,Kuala-Lumpur,Malaysia.
8. Jiang, P. et. al. (2022). A Review of YOLO algorithm developments. *Procedia Computer Science*, 199, 1066-1073.
9. Peng, Q.; and Tu, L. (2024). Paddle-OCR-based real-time online recognition system for steel plate slab spray marking characters. *Journal of Control, Automation and Electrical Systems*, 35 (1), 221-233.
10. Sasi, A.; Sharma, S.; and Cheeran, A.N. (2017). Automatic car number plate recognition. *Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems* (*ICIIECS*), Coimbatore, India, 2017, 1-6.
11. Yogheedha, K. et al. (2018). Automatic vehicle license plate recognition system based on image processing and template matching approach. *Proceedings of the International Conference on Computational Approach in Smart Systems Design and Applications* (*ICASSDA*). Kuching, Malaysia. 1-8.
12. Lin, C.; Lin, Y.; and Liu, W. (2018). An efficient license plate recognition system using convolution neural networks. *Proceedings of the IEEE International Conference on Applied System Invention* (*ICASI*), Chiba, Japan, 224-227.
13. Ali, F.; Rathor, H.; and Akram, W. (2021). License plate recognition system. *Proceedings of the International Conference on Advance Computing and Innovative Technologies in Engineering* (*ICACITE*), Greater Noida, India, 1053-1055.
14. Silva, S.M.; and Jung, C.R. (2022). A flexible approach for automatic license plate recognition in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 5693-5703.
15. Terven, J.; Córdova-Esparza, D.M.; and Romero-González, J.A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
16. Mukherjee, S. et al. (2023). OCR using Python and its application. *Journal of Advanced Zoology*, 44, 1083.
17. Ciaglia, F. et al. (2022). Roboflow 100: A rich, multi-domain object detection benchmark. arXiv preprint arXiv:2211.13523.

18. Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7), 677.
19. RangeKing. (2023). Brief summary of YOLOv8 model structure #189. Retrieved April 25, 2024, from https://github.com/ultralytics/ultralytics/issues/189.
20. Huang, Z. et al. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Information Sciences*, 522, 241-258.
21. Lou, H. et al. (2023). DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics*, 12(10), 2323.
22. Terven, J.; Córdova-Esparza, D.M.; and Romero-González, J.A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
23. Yang, G. et al. (2023). A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. *Agronomy*, 13(7), 1824.
24. Solawetz, J.; and Francesco. (2023). What is YOLOv8? The Ultimate Guide. Retrieved April 25, 2024, from https://blog.roboflow.com/whats-new-in-yolov8/.
25. Feng, L.; Ke, Z.; and Wu, N. (2022). ModelsKG: A design and research on knowledge graph of multimodal curriculum based on PaddleOCR and DeepKE. *Proceedings of the 14th International Conference on Advanced Computational Intelligence* (*ICACI*), Wuhan, China, 2022, 186-192.
26. Hu, S. et al. (2021). Efficient scene text recognition model built with PaddlePaddle framework. *Proceedings of the 7th International Conference on Big Data and Information Analytics* (*BigDIA*), Chongqing, China, 139-142.

Appendix A

Table A-1 Predictions under normal lighting conditions in indoor areas.

Accurate Car Plate	Predicted Car Plate	Confidence Level	Correct /Incorrect
JKW91**	KW9166	0.9594537615776062	Incorrect
VES65**	VES65**	0.9781081080436707	Correct
JUT89**	JUT89**	0.9974735379219055	Correct
JKG30*	JKG30*	0.989314615726471	Correct
JNU72**	JNU 72**	0.9577354192733765	Correct
WYS68**	WYS68**	0.9757224917411804	Correct
WB75***	WB 75***	0.9021086096763611	Correct
VFL12**	VFL12**	0.997146487236023	Correct
WSQ51**	WSQ51**	0.9967860579490662	Correct
JKJ82**	JKJ82**	0.9967371821403503	Correct
JSS79**	JSS79**	0.9998505711555481	Correct
WC21***	WC2I***	0.8845201134681702	Correct
JRH96**	JRH96**	0.9990937113761902	Correct
JWA92**	JWA92**	0.9950596690177917	Correct
ALV3**	ALV 3**	0.9930489659309387	Correct
AFP72**	AFP72**	0.9953505396842957	Correct
MDA52**	MDA 52**	0.9516216516494751	Correct
AKY30**	AKY30**	0.9258216619491577	Correct
BQC41**	BQC 41**	0.9575700759887695	Correct
PHR22**	PHR22**	0.9952625632286072	Correct

*Car plate numbers are partially masked for privacy concerns

Table A-2. Predictions under normal lighting conditions in outdoor areas.

Accurate Car Plate	Predicted Car Plate	Confidence Level	Correct /Incorrect
WWW98**	WWW98**	0.9402579665184021	Correct
JVY12**	JVY12**	0.995975732803344	Correct
JQM58**	JQM58**	0.9700007438659668	Correct
PQU68**	PQU68**	0.9861013293266296	Correct
MBL98**	MBL98**	0.9961497783660889	Correct
WTS53**	WTS53**	0.9900046586990356	Correct
VDK3**	VDK3**	0.9973015785217285	Correct
VET99**	VET99**	0.9902146458625793	Correct
PKX99**	PKX99**	0.9900819659233093	Correct
BMV82**	BMV82**	0.9825574159622192	Correct
JHL90**	JHL90**	0.9845503568649292	Correct
JUB72**	JUB72**	0.9505385160446167	Correct
VAG69**	VAG69**	0.9935787320137024	Correct
JVS82**	JVS82**	0.9932316541671753	Correct
JSB33**	JSB33**	0.9553326368331909	Correct
JRN90**	JRN90**	0.9854446053504944	Correct
JPQ89**	JPQ89**	0.9943268895149231	Correct
WST85**	WST85**	0.9876059293746948	Correct
AKA98**	AKA98**	0.9960055947303772	Correct
VLM60**	VLM60**	0.9738370180130005	Correct

*Car plate numbers are partially masked for privacy concerns

Table A-3. Predictions under rainy conditions in outdoor areas.

Accurate Car Plate	Predicted Car Plate	Confidence Level	Correct /Incorrect
NDJ65**	NDJ65**	0.9909552931785583	Correct
MCX84**	MCX84**	0.9927701950073242	Correct
VFR10**	VFR10**	0.9963904619216919	Correct
NDW67**	NDW67**	0.9928526878356934	Correct
JEC81**	JEC81**	0.9896259903907776	Correct
WVA71**	WVA71**	0.9993683695793152	Correct
WFH58**	WFH 58**	0.9292913675308228	Correct
AMR87**	AMR87**	0.9965880513191223	Correct
NCJ60**	NCJ60**	0.9914274215698242	Correct
MDR89**	MDR89**	0.9943579435348511	Correct
MCU62**	MCU62**	0.9952625632286072	Correct
WTR89**	WTR89**	0.9955345392227173	Correct
MCQ62**	MCQ62**	0.9966668486595154	Correct

*Car plate numbers are partially masked for privacy concerns

Table A-4. Predictions under low ambient light conditions in outdoor areas.

Accurate Car Plate	Predicted Car Plate	Confidence Level	Correct /Incorrect
MD**6	MD**6	0.9964555501937866	Correct
NCC55**	NCC55**	0.9850406646728516	Correct
MCS4**	MCS4**	0.9955493807792664	Correct
TAS12**	TAS 12**	0.956983745098114	Correct
WSW80**	WSW80**	0.9224337339401245	Correct
JBS29**	JBS 29**	0.9462471604347229	Correct
MCE93**	MCE93**	0.9971750974655151	Correct
TAY88**	TAY88**	0.9588430523872375	Correct
CAE89**	CAE89**	0.9983441829681396	Correct
MC**8	MC**8	0.9968608617782593	Correct
VFN76**	VFN76**	0.9327377080917358	Correct
MDQ60**	MDQ60**	0.9950875639915466	Correct
MDB35**	MDB35**	0.9951518177986145	Correct
JU**22	JU**22	0.9948901534080505	Correct
WSL16**	WSL16**	0.9955191612243652	Correct
MDR66**	MDR66**	0.995988667011261	Correct
MCB36**	MCB36**	0.9969854354858398	Correct
BNG88**	BNG88**	0.9957936406135559	Correct
MBC8**	MBC8**	0.9621796011924744	Correct
JKG82**	JKG82**	0.9978259801864624	Correct
MDL73**	MDL 73**	0.9504014253616333	Correct

*Car plate numbers are partially masked for privacy concerns