MEKATRONIKA
JOURNAL OF MECHATRONICS AND INTELLIGENT MANUFACTURING
E-ISSN: 2637-0883
VOLUME 7, ISSUE 1, JANUARY 2025, 44 - 53
DOI: https://doi.org/10.15282/mekatronika.v7i1.12064

RESEARCH ARTICLE

Plate Number Recognition for International Islamic University Malaysia Gate Security using Machine Learning

Irdina Hidayah Khairudin, Nurul Balqis Amyli Yaacob, Asmarani Ahmad Puzi* and Muhamad Aliff Imran Daud

Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia

ABSTRACT - The increasing number of vehicles at the International Islamic University Malaysia (IIUM) has led to challenges in traffic and parking management, posing significant enforcement difficulties for the Office of Security Management (OSeM). Therefore, a system needs to be designed to assist the university's policy enforcement to impose these rules and help OSeM to check for unregistered vehicles. To address these issues, this study proposes a robust and efficient vehicle license plate recognition system based on the state-of-the-art YOLOv5 object detection model. The proposed model was trained on a custom Malaysian vehicle dataset comprising 1,026 images. The proposed model detects and localizes the license plate, extracts the region of interest, and performs character recognition using EasyOCR. The YOLOv5 model achieved a mean average precision (mAP) of 99.5% for mAP@0.5-0.95, 87.7% at mAP@0.5, and a processing speed of 30 frames per second. The findings indicate that optimizing image-capturing techniques can further enhance detection accuracy, contributing to a more reliable real-time license plate recognition system for effective policy enforcement.

ARTICLE HISTORY

Received	:	23rd Feb. 2025
Revised	:	07th Mar. 2025
Accepted	:	30th Mar. 2025
Published	:	10th June 2025

KEYWORDS

Real-time
Number plate recognition
YOLOv5
EasyOCR
Computer vision
Object detection

1. INTRODUCTION

Automatic Number Plate Recognition (ANPR) is a vehicle surveillance system developed in the United Kingdom in 1976 that uses optical character recognition (OCR) to detect plate numbers from images or videos services [1], [2]. ANPR technology enables the real-time identification of vehicle plate numbers without requiring human intervention [3]. ANPR algorithms, which consist of four key steps: vehicle image capture, number plate detection, character segmentation, and character recognition, are widely applicable in public areas for purposes such as traffic safety enforcement, automatic toll collection, car park management, and automated vehicle parking systems [4]. In Malaysia, vehicles such as cars and motorcycles are widely used as the primary mode of transportation for commuting between locations, including homes, schools, and workplaces. At the International Islamic University Malaysia (IIUM), all lecturers, staff, and students are required to register their vehicles annually with the Office of Security Management (OSeM) to obtain parking privileges on campus. Failure to register may result in appropriate actions taken by OSeM. As the department responsible for overseeing security, traffic management, and related services, OSeM ensures the safety and well-being of the IIUM community.

However, some students choose not to register their vehicles, either to avoid paying the registration fee or due to the absence of a valid driving license. This has led to an additional issue of limited parking space, as only postgraduate students, off-campus students, and final-year students are eligible to register their cars under university regulations. Furthermore, OSeM must frequently conduct roadblocks within IIUM to identify and penalize unregistered vehicles, which requires significant time and manpower. To enhance vehicle registration enforcement, this system aims to automate the detection and recognition of vehicle plate numbers at IIUM's main gates. This will enable auxiliary police to efficiently monitor the registration status of vehicles entering the campus, reducing manual enforcement efforts and improving overall security.

2. RELATED WORKS

2.1 Plate Number Detection

Various statistical and methodological approaches have been explored for vehicle number plate detection, with significant advancements driven by deep learning frameworks. A review of existing literature identified key contributions in the development of real-time number plate recognition systems. Chan et al. (2024) [5] introduced a model integrating Single Shot Detector (SSD) with EfficientDet (SSD-ED) to enhance Malaysian number plate detection, addressing challenges such as language variations, font diversity, illumination conditions, occlusion, and adverse weather. Implemented using Python, TensorFlow, and Keras, the SSD-ED model was evaluated on the YouTube Malaysian number plate dataset, where SSD-ED7 achieved the highest accuracy of 94.6% for number plate detection and 93.4% for character recognition. Although SSD-ED7 demonstrated superior performance, longer processing times were observed.

*CORRESPONDING AUTHOR | M.Z. Khan | ✉ 200204006@hbtu.ac.in

In contrast, SSD-ED4, with slightly lower accuracy at 91.6% for number plate detection and 90.6% for character recognition, exhibited a better balance between speed and accuracy, making it suitable for real-time applications in law enforcement, toll collection, and parking management.

To further address variations in plate number shape and size, Darji et al. (2020) [6] developed a lightweight Convolutional Neural Network (CNN) model utilizing a MobileNet-based Single Shot Multibox Detector (SSD), specifically designed for efficiency in mobile and embedded vision applications. This approach achieved 91.24% accuracy while maintaining real-time operability. Further enhancements in deep learning-based plate number detection were demonstrated by Raj et al. (2022) [7] and Tusar et al. (2022) [8], both employing the YOLOv5 object detection model. Raj et al. (2022) [7] trained the model for 100 epochs with a batch size of 32 while incorporating image processing techniques such as scaling, contrast enhancement, binarization, and blurring to refine outputs. Building on this approach, Tusar et al. (2022) [8] optimized detection by increasing the batch size to 64 and setting an object detection threshold of 0.5, leading to an overall detection accuracy of 98%.

Beyond detection, improvements in character recognition were examined by Setiyono et al. (2021) [9] through the YOLOv3 method with Darknet-53 as a feature extractor. The study analyzed the impact of data preprocessing on recognition accuracy by testing number plate images extracted from videos captured by mobile devices and cameras under two conditions: with and without preprocessing. Results indicated that preprocessing significantly improved recognition performance, increasing accuracy from 80% to 88% for number plate recognition and from 97.1% to 98.2% for character recognition. Further analysis explored the effect of different maximum batch sizes on accuracy, alongside the integration of a graphical user interface (GUI) to facilitate experimentation. In addition to improving recognition accuracy, the character recognition process successfully converted extracted number plate characters into digital text, optimizing the transition from physical to digital data.

Advancements in character recognition were also demonstrated by Kochale et al. (2021) [10], where a TensorFlow-based model incorporated edge detection for feature extraction and blob detection to locate regions with distinct brightness or color variations. Connected Component Analysis (CCA) was then applied to scan binary images and label pixels. Despite achieving significant improvements in recognition accuracy, lighting conditions remained a key limitation, affecting overall detection performance. Addressing language-specific challenges, Alam et al. (2021) [11] developed a CNN-based vehicle number plate recognition (VNPR) system tailored for Bangladeshi number plates, incorporating Bengali alphabets and numerals. The system integrated image preprocessing, detection, super-resolution, segmentation, and character recognition, with segmentation playing a crucial role in accuracy enhancement. Traditional methods, including morphological operators and Support Vector Machines (SVM), were found to be ineffective in handling blurred images and adverse weather conditions, limitations that deep learning techniques sought to overcome. The proposed approach extracted 4096 features per image using CNN, localized number plates via template matching, and enhanced low-resolution images through spatial super-resolution. A bounding box method was employed for character segmentation, effectively isolating individual characters. Implemented in MATLAB R2018a and validated on a dataset of 700 vehicle images, the system achieved 98.2% accuracy on validation data and 98.1% on testing data, demonstrating superior performance and reduced processing time compared to existing VNPR systems.

Further extending CNN applications in number plate recognition, Gani et al. (2021) [12] introduced a CNN-based approach for Android smartphones equipped with cameras. This method integrated computer vision algorithms for plate number detection and recognition, followed by verification against a database of registered users. Neural Architecture Search (NAS) was employed, demonstrating superior accuracy and latency compared to manually designed CNN models. The automation of hardware-CNN (HW-CNN) codesign was achieved through NAS, incorporating parameters from both the CNN model and the hardware accelerator. An AutoML deep learning system based on NAS was integrated into the application, where dataset training generated an optimized model. Performance degradation due to poor lighting conditions and extreme plate number angles was identified as a primary limitation, underscoring the need for further improvements in real-world implementation. Beyond deep learning-based methods, Agrawal et al. (2020) [13] proposed a Cognitive Number Plate Recognition (CNPR) system that enhances license plate detection using machine learning and data visualization techniques. Conventional Automatic Number Plate Recognition (ANPR) systems often struggle with inaccuracies in plate localization due to factors such as poor illumination, improper sizing, and disorientation. To mitigate these challenges, the CNPR system incorporated a structured approach, beginning with preprocessing techniques to enhance image quality by reducing noise and converting images to grayscale. License plate localization was then performed using bounding rectangle positioning, ensuring accurate segmentation. K-means clustering further refined plate localization and improved detection accuracy. Experimental results demonstrated a plate localization accuracy of 83.2%, significantly enhancing the system’s ability to detect number plates under challenging conditions.

2.2 Character Recognition

Various approaches have been explored for license plate detection and character recognition to enhance recognition accuracy and robustness. Tusar et al. (2022) [8] applied the EasyOCR technique to detect images after cropping, utilizing the CRAFT algorithm for character detection. The recognition process was conducted using the CRNN method, which involves feature extraction, sequence labeling, and decoding. This approach achieved a recognition accuracy of 78% for Bangladeshi plate numbers. Similarly, Kochale et al. (2021) [10] implemented Optical Character Recognition (OCR) for character extraction after detecting the plate number. The input image was cropped to retain only the plate number region, where potential character regions were identified and extracted. The extracted characters were then converted into a string using the K-Nearest Neighbors (KNN) algorithm for final recognition. A multi-step approach for character recognition was introduced by Gnanaprakash et al. (2021) [14], where plate number images were first converted to grayscale, followed by the application of a smoothing filter and a bilateral filter. Canny's edge detection technique was employed to enhance image clarity and reduce noise before OCR-based character extraction using the Tesseract system. This method resulted in 90% recognition accuracy. Image preprocessing techniques, including grayscale conversion and image resizing, were further explored by Mushthofa et al. (2020) [15], where a Convolutional Neural Network (CNN) model with 36 classes corresponding to letters (A-Z) and numbers (0-9) was implemented for plate number recognition.

An improved approach for character segmentation and recognition was introduced by Agrawal et al. (2020) [13], where machine learning techniques were incorporated to enhance segmentation accuracy. The Cognitive Number Plate Recognition (CNPR) system demonstrated 80.5% accuracy in character segmentation and 73.4% accuracy in character recognition, outperforming traditional OCR methods. Data visualization techniques using JavaScript libraries such as D3.js and DC.js were further utilized to improve result interpretation. Despite these advancements, further improvements were suggested through the integration of deep learning models such as TensorFlow for real-time applications. A hybrid approach combining Optimal K-Means (OKM) clustering and CNN was developed by Pustokhina et al. (2020) [16] to address challenges posed by uneven lighting conditions. The OKM-CNN model implemented using Python and integrated with TensorFlow, Pillow, OpenCV, and PyTesseract, incorporated an enhanced Bernsen Algorithm (BA) for binary plate number extraction, achieving 98.1% overall accuracy across three datasets. Further advancements were introduced by Hawar Hussein Yaba & Latif (2022) [17], where a hybrid plate number recognition system was designed specifically for Arabic license plates in Iraq. The integration of machine learning, and image processing techniques enabled enhanced Arabic numeric character recognition through character segmentation and background color coding for vehicle classification. Character recognition performance was evaluated using Google Tesseract OCR and KNN, where KNN achieved 92.22% accuracy, surpassing the 45.56% obtained with OCR. Tested on 90 images, the system demonstrated 97.78% success in color detection and high accuracy in plate localization. These findings underscore the effectiveness of hybrid methodologies in plate number recognition and highlight their potential integration into intelligent transportation systems.

3. THE PROPOSED METHODOLOGY

3.1 Data Collection

A total of 484 images of vehicles with plate numbers were collected to develop a dataset for plate number detection and model training. The plate numbers consist of a subset of 36 characters, including 26 alphabetic characters (A-Z) and 10 numerical digits (0-9). Data collection was conducted with informed consent, where vehicle owners permitted the capturing of images of their cars. Additionally, a consent form was distributed within the IIUM community to encourage voluntary participation in data collection. For most vehicles, multiple images were captured from both the front and rear at different angles to ensure visibility of the plate numbers. Photographs were also taken under various lighting conditions to account for real-world variations.

Furthermore, diverse conditions were introduced to enhance the model's ability to recognize plate numbers under different scenarios, including variations in background, image angles, and ambient lighting. These efforts aimed to improve pattern recognition and enhance feature extraction from regions of interest. The primary objective of this custom dataset is to effectively train the model using Malaysian plate numbers, enabling accurate detection of plate numbers from vehicles entering IIUM from various angles and directions. Figure 1 illustrates sample images from the collected dataset.

Figure 1. Images from custom dataset

In the custom object detection dataset, each image was annotated by specifying the bounding box coordinates and classification. This project focuses on a single class, labeled as "LicensePlate". Roboflow was utilized to manually draw bounding boxes around plate numbers and export the annotated data in YOLO format. Preprocessing steps included resizing images to 640×640 pixels. To enhance dataset diversity and improve model generalization, data augmentation techniques such as zoom, and brightness adjustments were applied, expanding the dataset to a total of 1,026 images. The annotated dataset was then partitioned into training (80%), validation (10%), and test (10%) sets, resulting in 816 images for training, 105 for validation, and 103 for testing.

3.2 Plate Number Detection

The state-of-the-art YOLOv5 object detection model was utilized and trained using a Malaysian vehicle dataset comprising 1,026 images of varying dimensions, each containing a car with a visible plate number. You Only Look Once (YOLO) is a high-performance, CNN-based object detection algorithm capable of rapidly identifying objects in both images and video streams. The algorithm detects objects through a single forward propagation of a neural network, enabling simultaneous detection of all objects within a frame using a Convolutional Neural Network (CNN). To optimize computational efficiency, YOLO processes images by dividing them into a grid structure, where each grid cell detects objects through a single CNN pass rather than performing multiple passes per object. This approach significantly reduces computational complexity, making YOLO well-suited for real-time object detection. To enhance accuracy and processing speed, YOLO leverages an optimized neural network architecture. The primary objective of the YOLO algorithm in this study is to predict the bounding box that precisely locates the plate number within each input frame. A bounding box, which outlines the detected object, includes parameters such as width, height, center coordinates, and class label.

Model training was conducted using Google Colab, which offers free access to high-performance GPUs without the need for additional configuration. The YOLOv5 model was initialized with pre-trained weights from Ultralytics, trained on the Common Objects in Context (COCO) 2017 dataset. Custom layers with additional weights were incorporated during training to enhance model accuracy while maintaining efficient training time. The model was trained for 250 epochs with a batch size of 16, utilizing the YOLOv5-medium weight initialization. Figure 2 presents an example of plate number detection with 97% accuracy using the trained YOLOv5 model. To further evaluate the model's performance, an additional plate number dataset was obtained from a publicly available Kaggle dataset. The same training methodology applied to the custom dataset was implemented for this dataset to facilitate comparative analysis. A total of 644 annotated images were utilized, with the dataset partitioned into a training set (80%), a validation set (10%), and a test set (10%).

Figure 2. Output from the detection model with 97% accuracy

3.3 Character Recognition

Easy Optical Character Recognition (EasyOCR) is employed as a recognition method that processes an image as input and generates a string of characters as output. As a Python-based package implemented using the PyTorch library, EasyOCR enables efficient separation of individual characters within an image, facilitating accurate optical character recognition. By enhancing image clarity and contrast, the method improves text readability and contributes to higher model accuracy. Following the detection of the region of interest containing the plate number, the model crops the identified area before comparing the extracted image to a database template or character dictionary. Without requiring manual intervention, EasyOCR autonomously identifies and classifies the characters. The output consists of a list containing the coordinates of the cropped image, the predicted string, and the confidence score of the recognition. For number plate identification, EasyOCR offers a relatively simplified approach compared to other methods, as the characters on number plates typically follow standardized fonts. The overall accuracy of EasyOCR is influenced by the quality of the input image and the specific OCR settings applied. To achieve optimal accuracy, high-quality images with clear text should be used, and OCR parameters should be customized to align with the characteristics of the input text.

3.4 Intersection Over Union (IoU)

Intersection over Union (IoU) is utilized in YOLOv5 to generate an output box that accurately encloses the plate number object. This metric serves as a key performance indicator by measuring the area of overlap between two bounding boxes. A higher IoU value indicates better alignment between the predicted and ground-truth bounding boxes, ensuring precise object localization.The calculation of IoU is illustrated in Figure 3, where the blue box represents the predicted bounding box, while the orange box denotes the annotated ground-truth bounding box. In this study, the dataset consists of a single class label, specifically the plate number, making the model a simple binary classification system. A prediction is considered accurate if the IoU value is greater than or equal to 0.5; otherwise, the detection is deemed inaccurate and subsequently excluded from the final results.

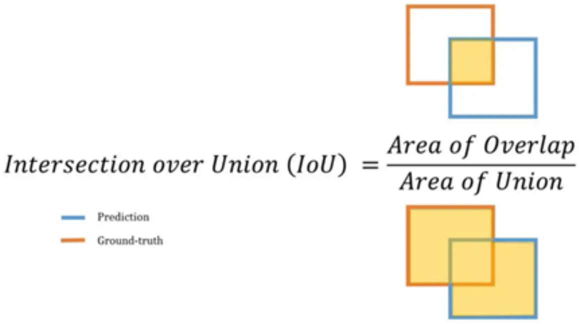

Figure 3. Intersection over Union Formula

3.5 Mean Average Precision

Precision and mean average precision (mAP) were employed as key evaluation metrics to assess the performance of the object detection model. As one of the most critical parameters in object detection, mAP provides a comprehensive measure of model accuracy by comparing predicted object detections with ground-truth annotations in the dataset. The calculation of mAP involves determining the Average Precision (AP) for each class and then computing the mean across all classes. This metric accounts for both false positives (FP) and false negatives (FN), effectively balancing precision and recall ensuring a robust evaluation of model performance. Due to its ability to provide an overall assessment of detection accuracy, mAP is widely regarded as the most suitable metric for object detection applications. The formula for mAP is presented in Equation (1) below.

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP_i} \tag{1}$$

To further evaluate detection performance, additional indicators such as precision, recall, and the F1 score are employed. These metrics offer a detailed assessment of the model’s capability to correctly identify objects while minimizing incorrect detections. The calculation methods for these indicators are shown in Equations (2), (3), and (4).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{precision * recall}{precision + recall} \tag{4}$$

In these equations, True Positives (TP) represent correctly detected plate numbers, False Positives (FP) correspond to incorrectly detected plate numbers, and False Negatives (FN) indicate instances where plate numbers were present but not detected. By incorporating these evaluation metrics, a comprehensive analysis of the model’s performance can be achieved.

4. SYSTEM DESIGN

The proposed system comprises a structured process that can be divided into five key stages: plate detection, character recognition, cross-referencing with the registered vehicle database, data storage, and result display, as illustrated in Figure 4. The process begins with the system capturing individual frames, which serve as input for the model. A real-time output is then generated, displaying the vehicles entering through the monitored gate. During the plate number detection stage, the primary task involves identifying the bounding box of the license plate within each frame. For character recognition, EasyOCR is utilized to process the cropped image of the license plate, converting it into a machine-readable string format. Following this, the extracted plate number undergoes a cross-referencing procedure against the database of registered vehicles at IIUM to verify its registration status. The database used in this verification process consists of the most recent records obtained from the QSEM database, containing 4,460 raw data entries with four attributes: count, name, ID number, and license plate information. Upon verification, the system stores and maintains a database containing extracted plate numbers, check-in timestamps, and vehicle status for all entries at IIUM's security gate. Vehicles with registered plate numbers are assigned the status variable "Registered," while the plate number that is not found in the database is classified under the "Visitor" status. Subsequently, the compiled information is relayed to the QSEM auxiliary police, who oversee vehicle entry at the IIUM Security Gate.

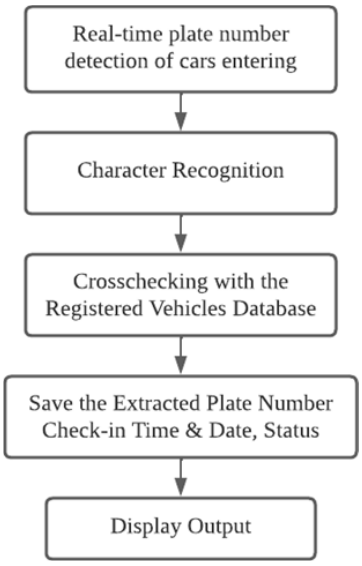

Figure 4. Flow diagram of the system

5. SYSTEM TESTING

The effectiveness of the proposed method was verified through a series of experiments and system tests. Testing was conducted at the International Islamic University Malaysia (IIUM) Main Security Gate to evaluate the system's functional reliability and robustness under real-world conditions. The test environment was set up outdoors using an external webcam positioned at the gate barrier to capture the front view of approaching vehicles. The camera was configured to focus on the front license plate to facilitate accurate detection and recognition. For this evaluation, the detection system was deployed in a local computing environment running Windows 10 Pro 64-bit. The hardware specifications included an 8th Generation Intel (R) Core (TM) i5-8250U processor and an NVIDIA GeForce MX110 GPU. The software environment was supported by CUDA 11.2 and OpenCV 4.5.4.60, with PyTorch v1.13.0 as the deep learning framework and Python 3.8.0 as the primary programming language.

A registered vehicle, as depicted in Figure 5, was utilized as the test subject to assess the system's ability to detect and recognize plate numbers accurately. The system was expected to extract and process the plate number characters, subsequently displaying "REGISTERED" as the vehicle status output. As the vehicle moved at a speed of 10 to 20 km/h, the algorithm successfully detected the license plate location with an accuracy range of 92% to 95%. Plate numbers were displayed in the output only if the character recognition confidence level exceeded the 70% threshold. However, due to factors such as vehicle distance from the camera and camera resolution limitations, two instances of inaccurate character recognition were observed in the initial data entries, as shown in Figure 5. As the vehicle approached closer to the camera, detection accuracy improved to 94%, and EasyOCR successfully recognized all characters of the plate number, 'W9327B,' with the status output correctly displayed as 'REGISTERED.'

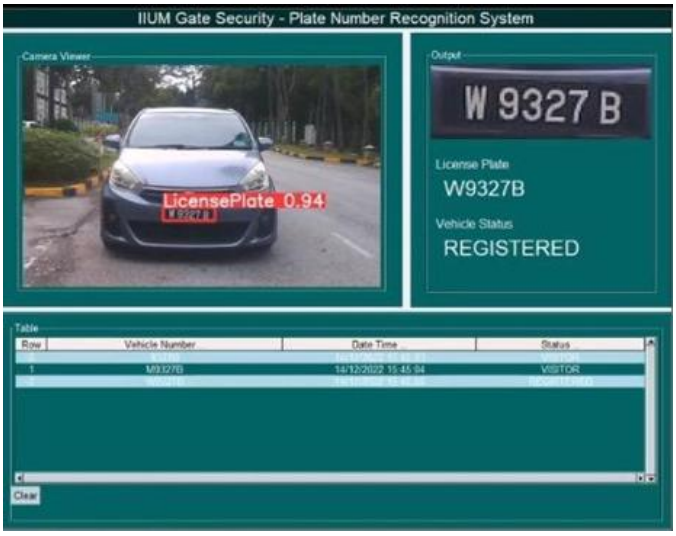

Figure 5. System testing at IIUM's main gate

6. RESULT AND DISCUSSION

6.1 Plate Number Detection using YOLOv5

The rapid improvement of the model is evident in the graphs presented in Figure 6, which illustrate various performance metrics for both the training and validation datasets. The plots depict box loss, objectness loss, precision, recall, and mean average precision (mAP) over 250 training epochs. Two distinct types of loss values are analyzed: box loss and objectness loss. These loss values represent the overall error accumulation within the model. Box loss quantifies the accuracy of object localization by assessing how precisely the algorithm identifies the center coordinates of an object and the extent to which the predicted bounding box aligns with the actual object. Meanwhile, objectness loss measures the probability of the plate number being present within a designated region of interest. A higher objectness score indicates a greater likelihood that the proposed region contains the target object. The results demonstrate a significant reduction in both loss values throughout the training process, indicating that the model achieves high accuracy with minimal errors.

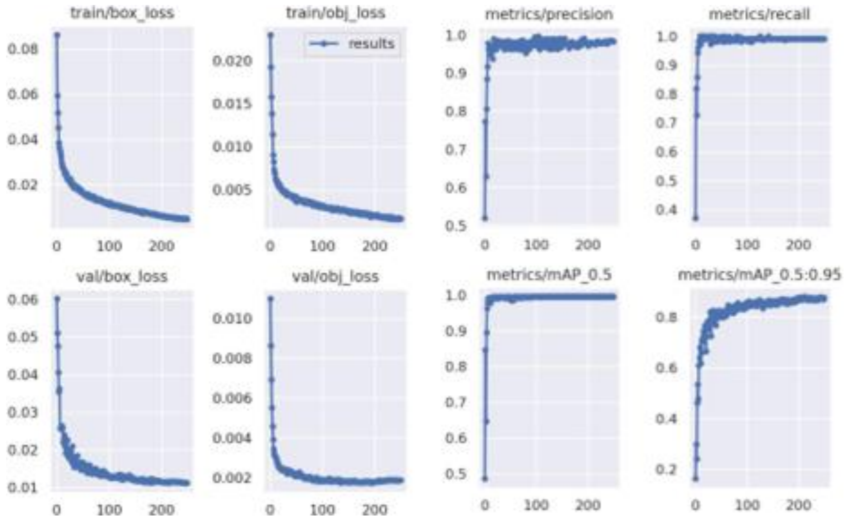

Figure 6. Performance metrics of the model

Precision is defined as the ratio of true positive predictions to the total number of positive predictions, while recall represents the ratio of true positives to the total number of actual instances. In the performance graphs, both precision and recall show a rapid increase after a few training epochs. Mean Average Precision (mAP) is calculated as the mean of all Average Precision values across different classes within the model. Significant improvements are observed in precision, recall, and mAP, demonstrating the effectiveness of the proposed approach. Additionally, a rapid decline in box and objectness losses for the validation data further indicates enhanced model performance. To prevent overfitting, early stopping was applied to select the optimal model weights.

To further validate the method, evaluations were conducted using two datasets: a custom Malaysian dataset and a publicly available Kaggle dataset. This comparison was essential to verify the integrity and quality of the collected data. The YOLOv5 model achieved high performance on the custom dataset, with mAP@0.5, mAP@0.5-0.95, precision, recall, and F1-score values of 99.5%, 87.7%, 97.6%, 99.0%, and 98.3%, respectively. A detailed evaluation of the detection model's performance is presented in Table 1.

Table 1. Performance metrics for different datasets

Metric	Datasets	
	Our Custom Dataset	Kaggle's Dataset
mAP@0.5	0.995	0.987
mAP@0.5-0.95	0.877	0.738
Precision	0.976	0.926
Recall	0.99	0.989
F1 Score	0.983	0.956

The mAP@0.5-0.95 metric evaluates the average precision of an object detection algorithm across multiple Intersection over Union (IoU) thresholds, ranging from 0.5 to 0.95, with a step size of 0.05. This metric is computed by first determining the average precision for each class at different IoU thresholds and then calculating the mean of these values. Consequently, all detections with an IoU within this range are considered positive detections in the proposed model. In contrast, mAP@0.5 measures the average precision at a fixed IoU threshold of 0.5. Since lowering the IoU threshold increases the number of detections counted as true positives, mAP@0.5 typically yields the highest value due to the greater number of positive detections.

The performance of the model trained on the Kaggle dataset resulted in mAP@0.5 and mAP@0.5-0.95 values of 98.5% and 73.0%, respectively. The lower performance at mAP@0.5-0.95 can be attributed to the poor quality and high variability of the Kaggle dataset, which contains plate numbers from diverse regions with differing formats, backgrounds, and arrangements. In contrast, the custom dataset was collected in high resolution and under controlled environmental conditions, contributing to improved model performance and higher detection accuracy.

6.2 Character Recognition using EasyOCR

Table 2 presents the performance evaluation of character recognition using the EasyOCR model on eight different test samples. For instance, test sample 1 features a car image with an actual plate number of BHJ 9520, as illustrated in Figure 7. Following the detection of the region of interest containing the plate number, the extracted region is cropped and processed using EasyOCR for character recognition. The recognition output for this sample is BHJ 9520, which matches the actual plate number with an accuracy of 98.4%. In addition to recognition accuracy and the predicted plate number, EasyOCR also provides the bounding box coordinates for the detected plate. The x and y coordinates for test sample 1 are represented in a 2D array format: [[13, 7], [177, 7], [177, 49], [13, 49]], corresponding to the bottom-left, bottom-right, top-right, and top-left points, respectively. This information further enhances the localization and verification of the detected plate number.

Table 2. Character recognition result

Test Sample	Actual Plate Number	Predicted Plate Number	Prediction Accuracy
Sample 1	BHJ9520	BHJ9520	0.984
Sample 2	CEE467	CEE467	0.943
Sample 3	VEM4950	VEM4950	0.998
Sample 4	WB3347H	WB3347H	0.968
Sample 5	MBW7203	MBW7203	0.985
Sample 6	BQW9996	BQW9996	0.830
Sample 7	TAK4004	TAK4O04	0.412
Sample 8	MCR8986	MCR8986	0.999

Figure 7. Performance metrics of the model

Overall, the results demonstrate that EasyOCR achieves high accuracy in character recognition for vehicle plate numbers across the eight test samples. However, challenges remain in distinguishing visually similar characters, such as the digit ‘0’ and the letter ‘O.’ This issue is evident in Sample 7 of Table 2, where the actual plate number ‘TAK 4004’ was misrecognized as ‘TAK 4O09,’ resulting in an accuracy of only 41.2%. To mitigate the impact of such misclassifications, the system is designed to display recognized plate numbers only when the character recognition accuracy exceeds a predefined confidence threshold of 70%. This approach ensures that only reliable recognition results are considered for further processing, enhancing the overall robustness of the system.

6.3 Limitations, Challenges, and Future Directions

Extensive research has been conducted on Automatic Number Plate Recognition (ANPR) systems, employing various methods and techniques tailored to different project requirements. Each approach offers distinct advantages and disadvantages, influenced by factors such as plate number placement, numbering system, background, size, colors, and character language, which vary across countries. The development of an improved image-capturing technique is crucial in enhancing image quality, minimizing noise, and subsequently increasing the accuracy of character detection and extraction.

The proposed system aims to automate the detection and recognition of vehicle plate numbers at IIUM’s main gates, enabling auxiliary police to efficiently monitor the registration status of vehicles entering the campus. This implementation reduces the need for manual enforcement while enhancing overall security measures. By refining image-processing techniques and optimizing character recognition accuracy, the system addresses key challenges associated with plate detection in varying environmental conditions. Despite its effectiveness, the system encounters limitations, particularly in character recognition accuracy when the plate number is detected from a distance. Since the system is designed to detect moving objects, variations in lighting, shadows, distance, and angles influence the ability of EasyOCR to accurately recognize characters. Although the region of interest is cropped before being processed, misclassification of visually similar characters, such as ‘O’ and ‘0’ or ‘I’ and ‘1,’ remains a challenge. To address this issue, future work should incorporate pre-processing techniques, including skew correction, noise removal, and binarization, to enhance the clarity of plate number characters, thereby improving recognition accuracy.

For future work, the proposed model and methodology can be further explored for vehicle model identification, traffic control, speed regulation, and location tracking, contributing to enhanced traffic safety enforcement. Integrating additional features would strengthen security measures in specific locations. Moreover, the system can be adapted to detect plate numbers with different formats, extending beyond the Malaysian dataset used for training and evaluation. Expanding the system's capability to recognize plate numbers from various countries would significantly improve its versatility and applicability in diverse environments. The current accuracy of plate number detection stands at 87.7%, which may be insufficient for certain real-world applications and necessitates further enhancement. Additionally, system integration with a barrier sensor at the security gate can be implemented to enable automated access for registered vehicles. When a recognized plate number matches a registered entry in OSeM, the barrier would automatically open, eliminating the need for manual operation by security personnel and streamlining the entry process.

7. CONCLUSION

In conclusion, this study presents a Plate Number Recognition system for IIUM's Security Gate using deep learning techniques, leveraging the YOLO framework for plate number detection. A Malaysian vehicle dataset comprising 1,026 images was employed in the YOLOv5 model training process. EasyOCR was utilized for character recognition to extract digits and letters from detected plate numbers. The system demonstrated an accuracy of 87.7% in detecting plate numbers, and real-time testing at IIUM’s main security gate confirmed its functionality, effectively capturing and identifying plate numbers from approaching vehicles. The findings highlight the system's potential for automated security applications, with further improvements necessary to enhance recognition accuracy and overall system performance.

ACKNOWLEDGEMENT

Author gratefully acknowledge Department of Computer Science, International Islamic University Malaysia for providing the necessary resources and facilities.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] O. Shobayo, A. Olajube, N. Ohere, M. Odusami, and O. Okoyeigbo, “Development of smart plate number recognition system for fast cars with web application,” *Applied Computational Intelligence and Soft Computing*, vol. 2020, p. 8535861, 2020.

[2] S. Joshi, P. Jejure, C. Jadhav, and V. Jankar, “Automatic number plate recognition,” *International Journal of Scientific Research in Science and Technology*, vol. 11, no. 5, pp. 439–448, 2024.

[3] Lubna, N. Mufti, and S. A. A. Shah, “Automatic number plate recognition: A detailed survey of relevant algorithms,” *Sensors*, vol. 21, no. 9, p. 3028, 2021.

[4] S. Tripathi, S. Jain, S. Shetty, and V. Sharma, “Automatic Number Plate Recognition System (ANPR): The Implementation,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 10, no. 8S, pp. 580-585, 2021.

[5] A. A. S. Chan, S. M. Mustam, F. C. Seman, M. F. L. Abdullah, F. A. Po'ad, A. Joret, et al., “Single shot detector-efficientdet (SSD-ED) model for real-time Malaysian number plate detection and recognition,” *Journal of Advanced Manufacturing Technology (JAMT),* vol. *18, no.* 2, pp. 125-138, 2024.

[6] M. Darji, J. Dave, N. Asif, C. Godawat, V. Chudasama, and K. Upla, “Licence plate identification and recognition for non-helmeted motorcyclists using light-weight convolution neural network,” in *2020 International Conference for Emerging Technology (INCET),* pp. 1-6, 2020.

[7] S. Raj, Y. Gupta, and R. Malhotra, “License plate recognition system using yolov5 and cnn,” in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 372-377, 2022.

[8] M. H. Tusar, M. T. Bhuiya, M. S. Hossain, A. Tabassum, and R. Khan, "Real time bangla license plate recognition with deep learning techniques," in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pp. 1-6, 2022.

[9] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, "Number plate recognition on vehicle using YOLO – Darknet," *Journal of Physics: Conference Series*, vol. 1821, no. 1, p. 012049, 2021.

[10] M. Kochale, A. Khemariya, and M. A. Tiwari, "Real-time automatic vehicle (license) recognition identification system with the Help of Opencv & Easyocr model," *International Journal of Research, Science, Technology, and Management*, vol. 23, no. 3, pp. 11-15, 2021.

[11] N. A. Alam, M. Ahsan, M. A. Based, and J. Haider, "Intelligent system for vehicles number plate detection and recognition using convolutional neural networks," *Technologies*, vol. 9, no. 1, p. 9, 2021.

[12] S. F. Abd Gani, M. F. Miskon, R. A. Hamzah, N. Mohamood, Z. Manap, M. F. Zulkifli, et al., "A live-video automatic number plate recognition (ANPR) system using convolutional neural network (CNN) with data labelling on an android smartphone," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 10, pp. 88–95,

[13] R. Agrawal, M. Agarwal, and R. Krishnamurthi, "Cognitive number plate recognition using machine learning and data visualization techniques," in *2020 6th International Conference on Signal Processing and Communication (ICSC)*, pp. 101-107, 2020.

[14] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012027, 2021.

[15] A. Mushthofa, A. Bejo, and R. Hidayat, "The improvement of character recognition on ANPR algorithm using CNN method with efficient grid size reduction," *2019 5th International Conference on Science and Technology (ICST)*, pp. 1–4, 2020.

[16] I. V. Pustokhina, D. A. Pustokhin, J. J. P. C. Rodrigues, D. Gupta, A. Khanna, K. Shankar, et al., "Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020.

[17] H. H. Yaba and H. O. Latif, "Plate number recognition based on hybrid techniques," *UHD Journal of Science and Technology*, vol. 6, no. 2, pp. 39–48, 2022.