Journal of Informatics and Web Engineering

Vol. 3 No. 2 (June 2024) eISSN: 2821-370X

Assessing the Efficiency of Deep Learning Methods for Automated Vehicle Registration Recognition for University Entrance

Muhammad Syaqil Irsyad[1], Zarina Che Embi[1]* and Khairil Imran Bin Ghauth[1]

[1] Faculty of Computing and Informatics, Multimedia University, Malaysia

**corresponding author: (zarina.embi@mmu.edu.my, ORCiD: 0000-0001-9378-7380)*

Abstract - With the ever-increasing number of vehicles on the road, a faster reliable security system for university entry is needed. This paper presents an approach for Automatic Number Plate Recognition (ANPR) using deep learning and PP-OCRv3. The proposed approach utilizes a pre-trained object detection model to locate license plates, extracts a single frame of the license plate, performs license plate recognition, applies pre-processing techniques, and employs PP-OCRv3 for text extraction in real time. The system was tested with Malaysian vehicle plates, and its accuracy and speed of detection were evaluated. The results show the system's potential to be easily adapted to different camera systems, angles, and lighting conditions by retraining the deep learning model. The paper also explores various deep learning methods, such as CenterNet, EfficientDet, and Faster R-CNN, and their effectiveness in automated vehicle registration detection. The research methodology involves creating a dataset from Open Images Dataset V4, converting label text into XML files, and utilizing the TensorFlow model trained on the COCO dataset. The paper concludes with the synthetic evaluation of the trained models, comparing their performance based on precision, recall, and F1-score. Overall, the proposed approach highlights the potential of deep learning and PP-OCRv3 in achieving accurate and efficient ANPR systems.

Keywords— number plate detection, optical character recognition, license plate recognition, TensorFlow, CenterNet, EfficientDet, Faster R-CNN, PP-OCRv3

Received: 09 September 2023; Accepted: 18 December 2023; Published: 16 June 2024

I. INTRODUCTION

Automatic number plate recognition (ANPR) is a technology that uses sensors, such as cameras, to identify and segment a vehicle registration and uses optical character recognition (OCR) on the images to read out the vehicle registration number. In Malaysia, ANPR has been used in various applications such as automated toll booths, vehicle identification, and parking management. However, there is no available open-source system that works with Malaysian number plates. The ever-increasing vehicle count on our roads has hindered the smooth flow of traffic. It has also affected the traffic of vehicles entering a university campus. Thus, implementing the ANPR, in theory, can provide better and more reliable access control on the campus compared to traditional methods.

An approach for automatic number plate recognition (ANPR) using deep learning and PP-OCRv3 will be proposed. The objective is to develop a real-time ANPR system that can accurately locate and extract text from license plate images. The authors will use a pre-trained object detection model to identify license plates, capture a single frame of

Journal of Informatics and Web Engineering

https://doi.org/10.33093/jiwe.2024.3.2.4

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

the license plate, apply necessary preprocessing techniques, and utilize PP-OCRv3 for text extraction. The system will be tested using Malaysian vehicle plates, and the evaluation will focus on the accuracy and speed of detection. The goal is to create a user-friendly ANPR system that can be easily customized for different camera systems, angles, lighting conditions, and datasets, without requiring extensive technical knowledge or expertise. The findings of the paper will provide insights into the effectiveness of deep learning-based ANPR approaches and their potential for improving accuracy and efficiency in various conditions.

II. LITERATURE REVIEW

A. Related Technology

Traditional methods for license plate recognition such as edge detection and template matching have limitations in accuracy and robustness [4]. In recent years, deep learning has emerged as a powerful tool for computer vision tasks, including license plate recognition. Deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown remarkable effectiveness in extracting complex features from data [4]. [4] proposed a hybrid CNN-RNN architecture for moving vehicle number plate detection. CNN was utilized to extract features from the image, while RNN was used for the classification and recognition of number plate characters. Their method achieved an impressive accuracy of 95% on a dataset of 10,000 images, surpassing previous methods [4].

Recognizing moving vehicle number plates (MVNPs) is a challenging task that finds applications in traffic enforcement and law enforcement [3]. They introduced an approach using CNNs and Support Vector Machines (SVMs) for MVNP recognition. They enhanced video frames using image processing techniques and applied a CNN-based object detection model to detect MVNPs. An SVM-based classifier was used for recognition, resulting in high accuracy and robustness to noise and distortions [3]. As SVM is renowned for its adaptability and ability to handle complex data, it is a valuable tool for researchers across various domains [17].

The integration of deep learning with automatic license plate recognition (ANPR) has gained attention due to its potential for accurate and efficient recognition in various conditions [13]. According to [13], CNNs have demonstrated effectiveness in spatial feature extraction, while RNNs excel in modelling sequential data. [13] proposed a hybrid deep learning architecture for license plate detection and recognition. Their approach utilized a CNN for spatial feature extraction and RNN for classification, achieving high accuracy and real-time performance [13].

While traditional ANPR systems have limitations in challenging conditions, deep learning-based approaches have shown promise in addressing these challenges [1, 7]. [1] proposed the use of YOLOv3 and YOLOv4 for character recognition. The first approach utilizes a YOLOv3 single character detector followed by a custom image classification network, while the second approach solely relies on a YOLOv4 character detector. The evaluation on a dataset of images shows that the second approach achieved higher accuracy in character and license plate recognition. The recognition of complete license plates is found to be less accurate than character recognition due to the dependency on correctly recognizing all characters. However, the matching procedure compensates for lower character recognition accuracy. In terms of inference speed, the second approach is faster as it involves fewer stages and computations. The study also assesses the system's performance on HD videos, demonstrating improved results with the inclusion of a multi-object tracker and the use of voting for sequence-based recognition. The importance of camera calibration, image resolution, and image quality is highlighted, along with the need to address challenges related to accuracy in unconstrained environments and security considerations in LPR systems [1].

B. Deep Learning Methods

Deep learning methods have revolutionized the field of computer vision and image recognition. Deep learning is a subfield of machine learning that uses neural networks with multiple layers to learn and extract high-level features from raw data. These methods have shown remarkable performance in various tasks, including object detection, image classification, and optical character recognition. Some common TensorFlow architecture and better comprehension of the reasons for choosing the said model are covered in the following sub-sections.

C. CenterNet

One prominent method is CenterNet, which employs a one-stage anchor-free architecture for precise localization and recognition of objects. [5] presented the application of CenterNet for traffic target detection and classification using radar data. The study simulated a large radar dataset with diverse types of traffic targets and evaluated the performance of CenterNet against conventional detection and classification algorithms. The results demonstrated that CenterNet achieved a detection rate (Pd) close to 1 at high signal-to-noise ratios (SNRs), outperforming the traditional methods. At lower SNRs, CenterNet maintained a higher Pd compared to CFAR, indicating its robustness to noise. Additionally, CenterNet exhibited a lower false alarm rate (Pf) than CFAR, showing its ability to distinguish between noise and target signals.

In a real-world application, [10] explored the usability of CenterNet for automated vehicle registration detection [10]. The study collected datasets containing various road users, such as trucks, cars, motorcycles, and pedestrians, in complex traffic scenarios. CenterNet was employed to detect and classify vehicle registration plates accurately. The results showed that CenterNet achieved a high detection rate (Pd) of 0.8014 and a low false alarm rate (Pf) of 0.0393, outperforming the traditional CFAR method. The overall accuracy (ACC) of CenterNet is 0.3954, indicating its effectiveness in classifying vehicle registration plates. The study demonstrated the potential of CenterNet as a deep learning-based method for improving the accuracy of automated vehicle registration detection systems [10].

D. EfficientDet

EfficientDet, a deep learning model developed by [14] proves to be highly effective and efficient for automated vehicle registration detection. The model achieves impressive results in terms of accuracy, surpassing the performance of existing architectures. According to the researchers, EfficientDet achieved an outstanding mean average precision (mAP) of 55.1% on the challenging COCO dataset. This highlights its ability to accurately detect and recognize vehicle registration plates. Furthermore, EfficientDet models demonstrated remarkable efficiency, making them suitable for real-time applications. On a Tesla V100 GPU, the models achieved an impressive frame rate of 40.1 frames per second (FPS) while maintaining high accuracy. This fast-processing speed enables EfficientDet to process a large volume of data in a short amount of time, facilitating real-time automated vehicle registration detection.

In a separate study conducted by [8], the performance of EfficientDet models was evaluated on a resource-constrained Raspberry Pi device. The experiments demonstrated that EfficientDet D0, the smallest variant of the model, excelled in terms of frame rate and size. Specifically, EfficientDet D0 outperformed other models in the EfficientDet family, achieving an average precision of 0.20 and an average recall of 0.27. These results further highlight the model's effectiveness in accurately identifying and localizing vehicle registration plates, even when deployed on resource-constrained devices. The size of the converted TensorFlow Lite model for EfficientDet D0 is also noteworthy, measuring only 4.2 MB. This compact size makes it highly suitable for deployment on devices with limited computational resources, ensuring efficient and accurate automated vehicle registration detection even in resource-constrained environments. Figure 1 illustrates the average precision and average recall of EfficientDet models, highlighting their impressive performance in automated vehicle registration detection tasks.

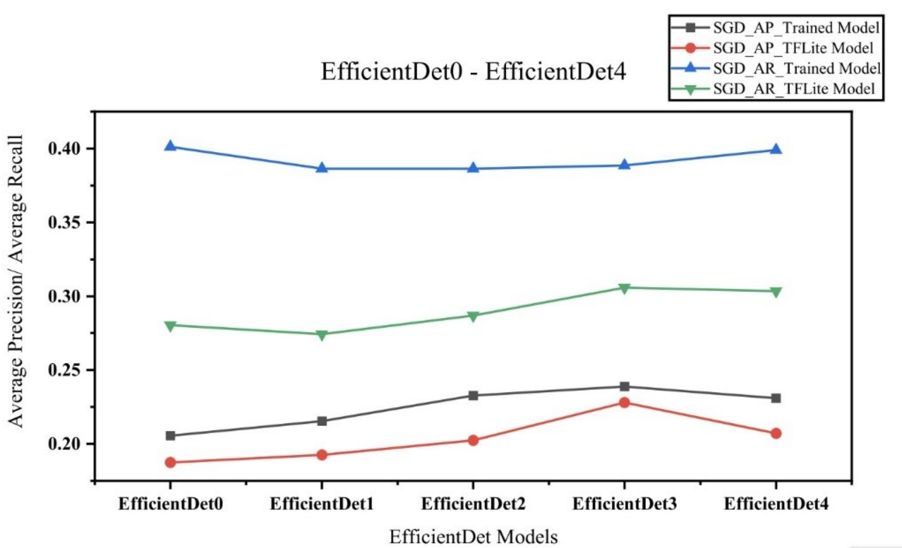

Figure 1. Block Diagram of YOLOv5 Model

E. Faster R-CNN

The [6] provided an overview of different R-CNN algorithms, focusing on R-CNN, Fast R-CNN, and Faster R-CNN. R-CNN proposes a set of boxes within an image and checks if any of these boxes contain objects. It utilizes selective search to extract regions from the image, followed by training a convolutional neural network (CNN) and using support vector machines (SVMs) for classification. However, R-CNN is computationally expensive and time-consuming. Fast R-CNN addresses this issue by running CNN only once per image and sharing the computation across regions of interest. It achieves faster object recognition by feeding the input image to the CNN, extracting convolutional feature maps, and reshaping proposed regions using ROI pooling. Fast R-CNN significantly improves speed by utilizing a single model for feature extraction, classification, and bounding box generation. Despite these improvements, Fast R-CNN still relies on selective search, which remains slow and inefficient for large datasets.

A region proposal network (RPN) was incorporated to enhance efficiency to overcome the limitations of Fast R-CNN, and Faster R-CNN [11]. The RPN takes image feature maps as input and generates object proposals with their abjectness scores. By using multiple CNNs for region suggestion and classification, Faster R-CNN surpasses Fast R-CNN in terms of speed and accuracy. RPN employs anchor boxes of different shapes and sizes, predicting the probability of an anchor being an object and adjusting the bounding boxes accordingly. This approach eliminates the need for selective search and significantly reduces the computational burden. The evaluation of different R-CNN algorithms on the COCO dataset reveals that both Fast R-CNN and Faster R-CNN outperform R-CNN in terms of mean average precision (mAP) [11]. Faster R-CNN demonstrates superior detection accuracy and speed compared to the other models, achieving a significant improvement over the previous state-of-the-art methods. Therefore, Faster R-CNN emerges as a powerful and efficient deep learning model for automated vehicle registration detection, offering improved accuracy and real-time performance.

F. PP-OCRv3

The growing demand for automated vehicle registration systems has driven the development of advanced optical character recognition (OCR) frameworks. Among these frameworks, PP-OCRv3 has gained considerable attention due to its remarkable performance in character detection and recognition tasks. As demonstrated in the research paper [9], PP-OCRv3 combines state-of-the-art deep learning models, such as Mask R-CNN and CRNN, to accurately extract and recognize characters from vehicle registration images. By employing a region-based approach, PP-OCRv3 achieves impressive results in terms of accuracy and efficiency.

PP-OCRv3 introduces several key features that contribute to its success in character recognition tasks. Firstly, the framework adopts the Mask R-CNN architecture, which effectively detects regions of interest (ROIs) containing characters within vehicle registration images. By incorporating both instance segmentation and bounding box regression, Mask R-CNN ensures precise localization of characters, enabling robust recognition [9]. Second, PP-OCRv3 uses the CRNN model, which combines recurrent neural networks (RNNs) and convolutional neural networks (CNNs), to recognize text. This integration enhances accuracy and robustness by enabling the framework to recognize both local and global dependencies in text sequences [9].

The integration of PP-OCRv3 into the automated vehicle registration process for a university entrance holds significant potential. By leveraging the high accuracy and efficiency of PP-OCRv3, the system can automatically extract and recognize vehicle registration numbers from images captured at entrance checkpoints. Moreover, the framework's ability to handle various font styles and image quality variations makes it suitable for real-world scenarios.

III. RESEARCH METHODOLOGY

A. Dataset

The dataset used in this study comprises 5000 images of vehicle license plates obtained from the Open Images Dataset V4. The Open Images Dataset is a large collection of approximately 9 million images, annotated with image-level labels, object bounding boxes, object segmentation masks, and visual relationships [12]. To reduce the computational load, we utilized the OIDv4 GitHub repository, which allowed us to focus on a specific class within the dataset. Specifically, we selected the object detection category with the class "Vehicle registration plate" from the training dataset. This choice was motivated by the randomized lighting conditions, image quality, and multiple angles present in the dataset, which are expected to enhance the performance of our TensorFlow model.

However, it is important to note that the dataset predominantly consists of overseas license plate numbers with a white background. We made the decision to exclude Malaysian license plate numbers due to the unavailability of a sufficiently large and diverse dataset of Malaysian license plate images with proper annotations. While this limitation should be acknowledged, we firmly believe that the overall diversity and quality of the dataset will still provide valuable insights and contribute to improvements in license plate recognition systems. Moreover, it is worth highlighting that the trained model can be potentially used for reading Malaysian license plates in the future, as indicated in the empirical evaluation section.

Figure 2. Sample Pictures Retrieved from Images Dataset V4

Each image in the dataset is accompanied by a label text that includes the bounding box coordinates of the license plate. For instance, the label text "Vehicle registration plate 344.96 493.348658 483.2 534.354612" represents the coordinates of the bounding box. In this example, the bounding box's top-left corner is located at (344.96, 493.348658), and the bottom-right corner is located at (483.2, 534.354612) in the X and Y axes, respectively. However, to facilitate the use of the TensorFlow model, the label text needs to be converted into Extensible Markup Language (XML) files. These XML files should maintain the picture file name and file path. Additionally, all images in the dataset are resized to a fixed size of 640 x 640 pixels.

To facilitate the training and evaluation process, the dataset is divided into two separate folders: a training folder and a testing folder. The training folder contains 3500 pictures, while the testing folder contains 1500 pictures. This division ensures that there is a sufficient number of images for both training and evaluating the TensorFlow model. Subsequently, a TFRecord file is created, which is a binary file format suitable for storing a sequence of records that can be efficiently utilized by TensorFlow models [16]. The TFRecord format will be instrumental in configuring the pipeline for training and evaluation purposes.

B. TensorFlow Model

The pre-trained models in the TensorFlow 2 detection model zoo were referred [15]. These models were trained on the COCO 2017. The models are open source and available to train from scratch. Each model has different pros and cons, which are listed through the processing time, COCO mAP accuracy, and output type.

Table 1. Sample Models

Model name	Speed (ms)	COCO mAP	Output
CenterNet MobileNetV2 FPN 512x512	6	23.4	Boxes
EfficientDet D0 512x512	39	33.6	Boxes
Faster R-CNN ResNet50 V1 640x640	53	29.3	Boxes

Table 1 shows some sample models that will be used in the model training. The chosen models are based on the research that has been done on the models. Each model has different pros and cons that will be analyzed after the training. From Table 1, it can be inferred that CentertNet would be the most efficient due to its speed, but accuracy is sacrificed. Meanwhile, the Faster R-CNN model is the most taxing to the system which takes the longest amount of time yet only produces the second-best accuracy. Therefore, EfficientDet is the balanced model between the three models, but further evaluation will be done to prove its effectiveness.

C. Model Training

In model training, hyperparameters are often selected through a combination of ablation studies and common practices. Ablation studies involve systematically varying and analyzing hyperparameter values to understand their impact on model performance. Common settings may emerge from established best practices, literature, or prior successful experiments, guiding researchers in choosing initial hyperparameter values before conducting specific ablation studies to fine-tune them for a given task.

The models were trained on a local machine using an NVIDIA 3060 TI GPU with 8GB of VRAM. Each model's pipeline configuration was tailored to its specific learning capabilities, allowing for a faster training setup. After conducting several experiments, a balanced challenge was achieved among the three models by training them for 50,000 steps with the default learning rate. Due to limited GPU memory, a batch size of two was chosen, as it determines the number of training examples processed together in each forward/backward pass. A larger batch size would have resulted in faster training, but the GPU's memory constraints necessitated this choice. The models were trained with a single label map id, which corresponds to "license." The training and test paths were configured using the respective folders from the previous separation.

D. Process Flow

The process flow of the image processing, deep learning model, LPR, and result extraction steps is depicted in Figure 3. The process starts with the "Input image," which then goes through "Image processing" to obtain a "Processed image". The "Processed image" is then input into the "Deep learning model," resulting in the "Detected license plate". The "Detected license plate" is further processed by the "LPR system," leading to the extraction of "License plate characters". Finally, the recognized "License plate characters" are presented as the "Results".

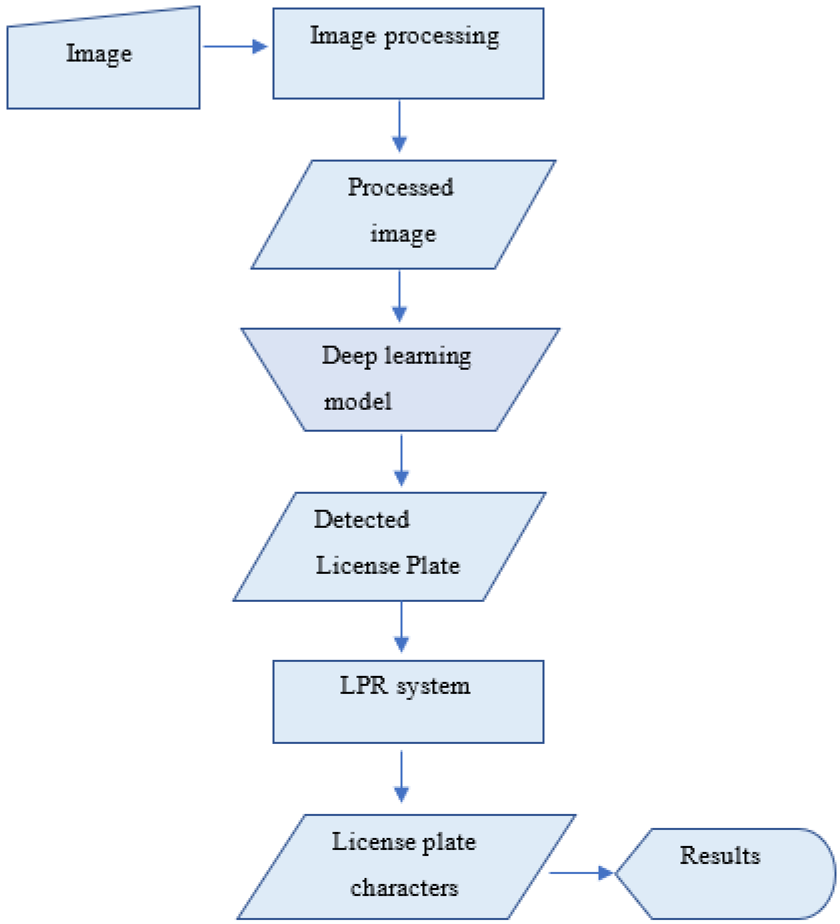

Figure 3. Process Flow

IV. RESULTS

A. Synthetic Evaluation

All three models were then compared after the completion of the training process. The performance of the models was evaluated with precision, recall, and F1-score. When working with class-imbalanced datasets, it is believed that the F1-score is the best evaluation to be used. F1-score provides a single measure that considers both precision and recall, which are two important evaluation metrics in binary classification. This is the first step in evaluating the three models.

The F1-Score is based on the precision and recall of the models. Recall is a metric used to represent the model's sensitivity towards correctly identifying the object out of all actual positive instances. Thus, it measures the model's ability to identify positive instances which is a key factor in a vehicle registration detection system.

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \tag{1}$$

Precision, however, is the model's ability to correctly predict positives out of all positive predictions made by the model. A higher precision value is key as it needs to be precise for a real-time detection system.

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \tag{2}$$

F1-Score is the harmonic mean of the two values, precision, and recall. It provides a single value that summarizes the model's performance in terms of both precision and recall. A higher F1-score indicates a better balance between precision and recall, representing a more accurate and reliable model.

$$\mathrm{F1-score} = \frac{2\ \mathrm{X}\ (\mathrm{Precision}\ \mathrm{X}\ \mathrm{Recall})}{\mathrm{Precision} + \mathrm{Recall}} \tag{3}$$

Table 2. Performance Analysis of the Model

Model	CenterNet	EfficientDet	Faster R-CNN
Precision	0.830	0.960	0.927
Recall	0.870	0.940	0.963
F1-Score	0.850	0.949	0.945
Training Time	0.98	2.86	2.57

Table 2 shows the results after training the models. Comparing the results, CenterNet was the fastest to finish with only 0.98 hours which is 61.54% and 62.05% faster compared to the training time for EfficientDet and Faster R-CNN, respectively. However, the speed of training would come at a cost of accuracy as can be seen from Table 2. EfficientDet, with the longest training time for 50,000 steps achieved 11.65% higher F1-score compared to CenterNet performance and 0.42% higher compared to the Faster R-CNN model.

Subsequently, CenterNet would be able to be trained on additional steps with the time taken by EfficientDet which is approximately 145,161 steps. This, in theory, could provide better precision, recall, and F1-Score due to a higher learning step. However, this is only the evaluation phase. Better results of the models can be seen in an empirical evaluation test case.

B. Empirical Evaluation

In this empirical evaluation, we selected a set of 112 unique Malaysian cars with distinct styles and fonts on their registration plates. These cars were recorded at the entrance of Multimedia University Cyberjaya (MMU) and were exclusively used for testing purposes. They were not included in the training of the model. The objective of this evaluation is to determine the most suitable model for the vehicle registration detection system, utilizing PP-OCRv3 for the OCR process. By analyzing the number of detections made by each model on the 112 cars, we can identify the model that achieves the highest accuracy. Figure 4 showcases a few sample cars captured in the video.

Figure 4. Sample Cars in the Video

Before applying the scanner for the OCR process, the captured images undergo several pre-processing steps to improve the accuracy and reliability of the PP-OCRv3 results. These steps include:

- Brightness Adjustment

  The brightness adjustment factor (alpha) is multiplied by the image pixel values, and the bias value (beta) is added to the result. This step helps enhance the overall brightness of the image, making it easier for subsequent operations to detect and process the characters.

- Grayscale Conversion

  Grayscale images contain only shades of gray, representing the intensity or brightness of each pixel. This simplifies the image and reduces the complexity of subsequent operations. For character recognition, the color information is not necessary since it primarily relies on the contrast between characters and the background.

- Image Inversion

  Inverting the image reverses the pixel values, transforming dark regions into light and vice versa. This step is often used to enhance the contrast between characters and the background, improving the legibility of the text.

- Image Resizing

  Resizing the image can help standardize the input size for the character recognition model. It may also enhance the clarity and sharpness of the characters by interpolating pixel values based on the neighboring pixels.

- Image Denoising

  Applying a median filter helps remove small unwanted details and noise from the image. It preserves the edges and key features while effectively reducing the impact of noise on character recognition.

- Image Sharpening

  The sharpening operation accentuates the edges and minute details in the image. By adjusting the weights of the original and blurred images, the algorithm enhances the contrast between adjacent pixels, which can help make the characters more distinct.

- Thresholding

  Thresholding separates the image into two regions based on a predefined threshold value. Pixels below the threshold are set to one value (e.g., black), and pixels above the threshold are set to another value (e.g., white). In the case of character recognition, thresholding helps isolate the characters from the background and enhances their visibility.

Pre-processing is crucial to improve the OCR engine's ability to segment and detect the letters on the registration plate. Given the variations in lighting conditions, camera angles, and noise from camera placement, the optimal values for each pre-processing step must be determined experimentally. Future implementations should explore testing different values to find the optimum adjustments. Figure 5 illustrates an example of a sample image before and after pre-processing.

Figure 5. Before Pre-processing (Left) and After Pre-processing (Right)

C. Discussions

Table 3 shows the result of the video analysis. EfficientDet performed well in detecting vehicle registration from the video. Meanwhile, CenterNet and Faster R-CNN faced some difficulties in detecting two-row license plates. Malaysian cars have two types of license plates, single-row, and two-row. Thus, the models trained on the Open Images dataset, which predominantly includes single-row license plates, encountered challenges in detecting the two-row license plates. This finding highlights the potential of the trained model to be utilized in reading Malaysian license plates in the future.

Table 3. Performance Analysis of the Model

Model	CenterNet	EfficientDet	Faster R-CNN
Missed Vehicle Registration	11%	2%	8%
Correct OCR per characters	87%	94%	91%
FPS	12.83	8.92	8.51

Figure 6 shows a double-row license plate that was successfully detected by EfficientDet.

Figure 6. Double Row Registration Plate Detected by EfficientDet

CenterNet exhibited the highest efficiency in terms of frames per second (FPS), indicating a faster processing speed for the ANPR system. However, this efficiency resulted in the worst performance out of the three models. There were false positives in the detection, such as identifying the car's lights as license plates, observed in both CenterNet and Faster R-CNN. These false positives were not present in the EfficientDet model, likely due to its higher confidence level. Thus, considering the overall performance, EfficientDet is chosen as the model for the automated vehicle plate recognition implementation in the future.

As shown in Table 3, all three models achieved satisfactory accuracy in optical character recognition (OCR). PP-OCRv3 successfully read almost all the segmented registration plates correctly. However, there were some cases of discrepancy in the readings. For example, in one instance, PP-OCRv3 couldn't detect properly due to the angle of the camera placed.

Figure 7 shows two pictures of vehicles entering the same MMU Gate B. But this entrance has two different pathways for incoming vehicles to go through. As a result, the cars stopping at the security guard can be facing two different angles when stopped for security checks. Surprisingly, in the left picture of Figure 7, EfficientDet still manages to capture in the frame that a registration plate exists in the frame but PP-OCRv3 would sometimes produce the wrong reading due to the angle of the letters. This shows that the dataset used from Images Dataset V4 is so robust that EfficientDet manages to detect with multiple different angles. Thus, we can conclude that it is important to make sure that the camera placement is correct and consider all possible stopping positions of the vehicle.

Figure 7. A Picture of a Vehicle Entering from The Left Angle (Left). A Picture of a Vehicle Entering from The Right Angle (Right)

Besides that, EfficientDet and PP-OCRv3 also manage to surprise in a different way. Malaysian vehicles sometimes have special registration plates. These special registration plates are released in limited quantities. Figure 8 shows the sample of the special registration plates type on the road in Malaysia.

Figure 8. Special Registration Plate Released by the Land Transport Bureau Malaysia

This special registration plate would be challenging to the system as the characters can sometimes be different. As we can see in Figure 8, the "Putrajaya" uses a cursive font compared to other vehicles registration. But during the recording of vehicles entering the MMU campus, one of the cars was using that registration plate.

Figure 9. Result of the EfficientDet and PP-OCRv3 on a Special Registration Plate

It can be seen from Figure 9 that EfficientDet and PP-OCRv3 are not only able to detect the license plate but they are also able to OCR the right characters from the license plate. This is due to the robustness of the models and their ability to handle variations in fonts and styles. This is a significant advantage, as it shows that the system can effectively handle special registration plates and produce accurate OCR results.

Based on these results, EfficientDet and PP-OCRv3 have demonstrated strong performance in detecting and recognizing vehicle registration plates. EfficientDet has shown high accuracy in detecting license plates, especially in the case of two-row license plates, while PP-OCRv3 has achieved satisfactory accuracy in reading the segmented registration plates. Both models have shown resilience in handling variations in angles and fonts, which are common challenges in license plate recognition systems.

Therefore, considering the overall performance and robustness, EfficientDet and PP-OCRv3 are recommended as models for automated vehicle plate recognition implementation in the future. Further optimization can be done by refining camera placement and exploring additional training with local datasets to improve accuracy and ensure effective recognition of special registration plates.

V. CONCLUSION

In conclusion, this research demonstrates the potential of deep learning-based Automatic Number Plate Recognition (ANPR) approaches in enhancing the accuracy and efficiency of automated vehicle registration detection systems. The findings indicated that EfficientDet performed well in detecting vehicle registrations, while CenterNet and Faster R-CNN encountered challenges in detecting two-row license plates commonly found in Malaysian cars. The results also revealed that PP-OCRv3, an OCR engine, achieved satisfactory performance in recognizing characters on the segmented registration plates.

Despite CenterNet's higher frame per second (FPS) processing speed, it exhibited the poorest performance among the three models, often producing false positives by incorrectly identifying car lights as license plates. In contrast, EfficientDet demonstrated superior accuracy and reliability by avoiding such false positives. Therefore, EfficientDet emerges as the preferred model for future implementation of automated vehicle plate recognition systems.

Overall, this research highlights the potential of deep learning, particularly the combination of EfficientDet and PP-OCRv3, in enhancing license plate recognition systems. The integration of deep learning techniques, such as those offered by TensorFlow, offers exciting possibilities for future advancements in ANPR systems. The proposed methodology serves as a foundation for developing efficient and reliable ANPR systems that can contribute to enhanced traffic management and access control in various settings, including university campuses and areas with high vehicle traffic. Future research can focus on further optimization of the models and exploring additional enhancements to enhance the system's accuracy and performance across different scenarios.

ACKNOWLEDGEMENT

A version of this paper was presented at the third International Conference on Computer, Information Technology and Intelligent Computing, CITIC 2023, held in Malaysia on 26th-28th July 2023.

AUTHOR CONTRIBUTIONS

MSI: Data Curation, Formal Analysis, Investigation, Validation, Visualization, Writing – Original Draft Preparation; ZCE: Conceptualization, Methodology, Project Administration, Resources, Supervision, Writing – Review & Editing; KIG: Conceptualization, Methodology, Writing – Review & Editing

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

REFERENCES

[1] A. Ammar, A. Koubaa, W. Boulila, B. Benjdira, & Y. Alhabashi (2023). A Multi-Stage Deep-Learning-Based Vehicle and License Plate Recognition System with Real-Time Edge Inference. Sensors (Basel, Switzerland), 23(4), 2120. https://doi.org/10.3390/s23042120

[2] EscVM. (2019). GitHub - EscVM/OIDv4_ToolKit: Download and visualize single or multiple classes from the huge Open Images v4 dataset. GitHub. https://github.com/EscVM/OIDv4_ToolKit

[3] R. Fernandes, K.M. Rai, A.P. Rodrigues, B.A. Mohan, N. Sreenivasa & N. Megha (2021). Recognition of Moving Vehicle Number Plates using Convolutional Neural Network and Support Vector Machine Techniques. In 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 93-98, IEEE.

[4] P.T. Gayathri, A. Bhavana, M. Anuhya, M. Kavitha, N.K. Chakravarthy & D.M. Reddy (2022). Moving Vehicle Number Plate Detection using Hybrid Deep Convolutional and Recurrent Neural Network Algorithm. In 2022 IEEE 2nd International Conference on Sustainable Energy and Future Electric Transportation (SeFeT), pp. 1-5, IEEE.

[5] Z. Guo, W. Yi, Y. Wu & T. Luo (2021). Robust radar detection and classification of traffic vehicles based on anchor-free CenterNet. In 2021 International Conference on UK-China Emerging Technologies (UCET), pp. 252-257, IEEE.

[6] O. Hmidani & E.I. Alaoui (2022). A comprehensive survey of the R-CNN family for object detection. In 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), pp. 1-6, IEEE.

[7] J. J. Ng, K. O. M. Goh, and C. Tee, "Traffic Impact Assessment System using Yolov5 and ByteTrack," J. Informatics Web Eng., vol. 2, no. 2, pp. 168–188, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.13.

[8] V. Kamath & A. Renuka (2021). Performance Analysis of the Pretrained EfficientDet for Real-time Object Detection on Raspberry Pi. In 2021 International Conference on Circuits, Controls and Communications (CCUBE), pp. 1-6, IEEE.

[9] C. Li, W. Liu, R. Guo, X. Yin, K. Jiang, Y. Du, Y. Du, L. Zhu, B. Lai, X. Hu, D. Yu, Y. Ma (2022). PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2206.03001

[10] M. Li, H. Ge & H. Wang (2022). IMG-centernet: an optimized algorithm based on CenterNet for pedestrian detection. In 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Vol. 6, pp. 203-208, IEEE.

[11] Y. Li, S. Zhang & W.Q. Wang (2020). A lightweight faster R-CNN for ship detection in SAR images. IEEE Geoscience and Remote Sensing Letters, 19, pp. 1-5.

[12] Open Images V7 – Description. (2023, May 15). Retrieved from Open Images V7 - Description (storage.googleapis.com)

[13] T.A. Pham (2023). Effective deep neural networks for license plate detection and recognition, Visual Computer, 39(3), pp. 927-941.

[14] M. Tan, R. Pang & Q.V. Le (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10781-10790.

[15] Tensorflow. (2021, May 7). models/research/object_detection/g3doc/tf2_detection_zoo.md at master · tensorflow/models. GitHub. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

[16] TFRecord and tf.train Example, TensorFlow. (2023, May 15). Retrieved from https://www.tensorflow.org/tutorials/load_data/tfrecord

[17] T. A. Khan, R. Sadiq, Z. Shahid, M.M. Alam, & M. B. M. Su'ud, "Sentiment Analysis using Support Vector Machine and Random Forest," J. Informatics Web Eng., vol. 3, no. 1, pp. 67–75, Feb. 2024, doi: 10.33093/jiwe.2024.3.1.5.