MALAYSIAN AUTOMATIC LICENSE PLATE RECOGNITION USING SINGLE-SHOT OBJECT DETECTION MODEL AT LOW VISIBILITY AND UNCONSTRAINED ENVIRONMENT

Ahmed Asaad, Hasan Firdaus Mohd Zaki*, Ahmed Rimaz Faizabadi

Department of Mechatronics Engineering
Kulliyyah of Engineering,
International Islamic University Malaysia

ahmsae.2015@hotmail.com, hasanzaki@iium.edu.my, ahmedrimaz@gmail.com

Article history

Received: 15 Nov 2022

Received in revised form: 20 Nov 2022

Accepted: 30 Nov 2022

Published online: 10 Dec 2022

*Corresponding author: hasanzaki@iium.edu.my

ABSTRACT

Automatic license plate recognition in the outdoor environment is challenging due to many factors that affect their performance. These include unconstrained environment conditions and variations in the license plate (LP) designs. In an open-world situation, the problem becomes more complicated due to non-compliance with the standard specifications and the existence of the special LP. To overcome this problem, we proposed a huge Malaysian license plate dataset and employed a high-accuracy single-shot object detection network YOLOv5 for the plate detection and character recognition tasks in a hierarchical manner. The dataset was collected by extracting video frames from cameras installed at different toll plaza booths day and night under various weather conditions. To capture large variations of license plate styles, camera viewpoints, lighting conditions as well as character font templates, we devised several augmentation strategies specifically tuned to address these challenges. The large dataset of 58,000 images was used to train the YOLOv5 model with 1200 epochs. The system performed very well on the challenging test dataset achieving a 95.96% accuracy rate.

***Keywords**: ALPR, Object detection, Dataset Augmentation*

1.0 INTRODUCTION

The automatic License Plate Recognition (ALPR) system is essential in many real-world applications. It is a crucial component of the intelligent transportation system (ITS), which is utilized to improve traffic operations, traffic monitoring, and law enforcement. ALPR is also used in automated vehicle parking and toll collection systems [1]. In addition, it is used for safety and security purposes, such as tracking stolen vehicles and vehicles used in criminal activity [2].

ALPR system ultimately aims to extract the alphanumeric characters on the LP of the vehicle and provide them as a text entry. ALPR output is primarily exploited in further tasks into which it is integrated. For instance, ALPR is used in a ticketless parking system to accurately estimate the parking charges based on the time of stay and to monitor the location of the parking bay where a specific vehicle has been parked. Similarly, ALPR output will be input into an RFID-based Electronic Toll Collection (ETC) system to determine the corresponding toll prices [3]. Through this automation, the efficiency of a system can be improved with increased revenue.

Plenty of computer vision techniques and approaches were employed to build ALPR. Generally, these techniques and approaches have the same pipeline, starting with using a camera to capture the scene, then applying a chosen technique to get the alphanumeric characters on the plate as an output through three main stages: license plate detection, character segmentation, and character recognition. However, some algorithms with a segmentation-free approach ignore the segmentation stage[4], and others even add addition step at first to detect the vehicle using dedicated algorithms [5] before the LP detection step. Figure 1 depicts the typical pipeline of the ALPR system.

Figure 1: General ALPR System Pipeline

License plate recognition (LPR) is considered the bottleneck of the system. It is complex and challenging since the objects which are alphanumeric characters on the LP are relatively small in the overall image and may appear with complex views due to the environmental effects such as occlusion, illumination, and camera used. In addition, the confusion caused by similarities in the characters complicates the task further. Moreover, in Malaysian LPs, this issue becomes more severe due to the existence of non-standard LPs, which are very popular in the country as well as the special LPs that are issued by authorities and have fancy fonts and designs, unlike the standard specifications. All the above factors must be taken into account when designing the LPR system. Failing to do so will lead to a non-robust system with suboptimal performance, and the system will fail to work in real-world unconstrained environments.

Most ALPR solutions in Malaysia use shallow methods such as image processing, statistical classifiers, and machine learning. There have been few attempts to harness the power of deep learning. However, the state-of-the-art two-stage deep learning LPR method [6] is not used in the Malaysian context. Table 1 lists major Malaysian ALPR studies. To develop a robust ALPR, we need to explore the most significant parameters that can affect the system performance, such as the algorithm or method used, the size of the training dataset, and coverage for types of LPs (single and double row, standard and non-standard). Due consideration should be made to significant factors that affect the image quality: illumination variation with day and night in weather conditions such as sunny, foggy, and rainy. Other factors come from camera type and position, such as noise and blur due to camera focus and different view angles based on camera placements. This paper presents a large Malaysian LPR dataset and robust deep learning-based Malaysian LPR to achieve higher accuracy and real-time performance. The major contributions of this study are to:

- Develop a large and challenging Malaysian license plate detection and recognition dataset with various real-world conditions.
- Implement a novel dataset augmentation strategy to capture different environmental conditions.
- Develop a robust LPR system using cutting-edge deep learning-based Convolutional Neural Network (CNN) for object detection.

The rest of the paper is organized as follows. In the second Section, a review of the literature in the field of LPR is done. Section three elaborates on the methodology of the study. Section four presents an exploration of the results and a discussion of the performance.

2.0 BACKGROUND AND RELATED WORK

The necessity to implement the Automatic License Plate Recognition (ALPR) system in various applications has led to a significant attempt to design robust, accurate, and efficient ALPR systems. Table 1 explores the critical studies that addressed the Malaysian LPR.

Each of the studies presented in Table 1 has limitations in terms of the size and quality of the datasets. In addition, most approaches could not ensure high performance on large challenging datasets. The ALPR model must be robust for real-time applications such as Multi-Lane Free Flow (MLFF) system. From Table 1, it can be noted that the systems in [7], [8], [9] and [10] reported good recognition accuracy. However, the systems lack a comprehensive and representative dataset showing system robustness. The drop in accuracy can be seen in [11] when the special plates are included.

Similarly, authors in [12] added special plates and double-row LP and made many variations in their dataset. Consequently, the system could not achieve good accuracy due to the complexity of the dataset. Although the study [13] could reach good accuracy when including special plates, the dataset quality shown in the paper is ideal and straightforward without complicated real-world instances. Double-row LPs and special plates are not considered in [14] which makes this study not robust to this parameter. In order to achieve a robust ALPR system, all mentioned factors must be considered in the dataset. In addition, the size of the dataset plays a crucial role since it will probably cover all variations in LP instances. The deep learning-based methods have the advantage of extracting features from an image and classifying it. However, there is no significant deep learning-based attempt on the Malaysian LPR system.

Table 1: Key parameters and performance of Malaysian LPR proposed systems. The factors mentioned are 1. Illumination variation 2. Day and night 3. Noise 4. Blur 5. Different view angle 6. Rotation 7. Skewness 8. weather complexity: sunny, foggy, and rainy, and Issues are a) Shallow Method, b) Small train and test dataset) The numbers assigned to these factors will be used in the Table to indicate which study considered these factors. CCA: Connected Component Analysis, FNN: Feedforward Neural Network, TM: Template Matching, KNN: k-Nearest Neighbors, SA: Smearing Algorithm, BA: Bees Algorithm, SVM: Support Vector Machine, RNN: Recurrent Neural Network.

Ref.	Method	Size of Train dataset	Size of the Test dataset	Used special plates	Double-row LP	Factors included	Accuracy %	Issues
[7]	CCA-FNN	136 char.	50	No	No	3	90	a,b
[11]	Edge detection-TM	-	350	Yes	No	4	81.33	a,b
[8]	CCA-KNN	330 char.	100	No	No	None	95	a,b
[9]	CCA-Pearson Correlation	-	270	No	No	1,7	91.5	a,b
[12]	SA	-	150	Yes	Yes	1,3,4,5	76	a,b
[13]	BA-SVM	-	1216	Yes	No	None	95.43	a,b
[10]	CCA-CNN	700	528 char.	No	No	None	94.6	b
[14]	CNN-RNN	2304	409	No	No	1,3	95.1	b

CNN has been used in many studies to recognize the LP. However, most deep learning-based methods are studied in other contexts for non-Malaysian ALPR. Since we use the deep learning-based method in our study, we have focused on related works that use these approaches. For example, in [15], a customized CNN with four convolutional layers and two fully connected layers is used to predict the segmented characters from the LP image. With some modification in kernels and applying a dropout of 0.5 in the fully connected layers, they could reach test results up to 94.8% on the Caltech dataset and an average accuracy of 95.5% in the Application-Oriented License Plate (AOLP) dataset. However, this approach fails in some conditions, such as bad weather conditions.

In [16] and [17], Bi-directional Recurrent Neural Network (BRNN) and Long Short-Term Memory (LSTM) are used to recognize the detected LP. First, features are extracted from the preprocessed cropped LP and fed to the BRNN for sequence recognition. In the sequence labeling, BRNN predicts the next character while carrying the current and previous ones as inputs. The network keeps processing the whole sequence in the same manner until the end, and predictions are given as probability values. As RNN cannot hold information in its memory for a long time, LSTM is used, which can remember information for a longer time. The last step to obtain the characters is done by using Connectionist Temporal Classification (CTC), which converts probability values to character strings.

Some studies implemented object detection-based methods to recognize the LP characters without segmenting the LP [18] [4]. In [18] and [4], YOLO with a CNN model combined (CR-Net) to recognize the LP. The proposed method improved the accuracy while preserving the real-time performance with layout independent method. A modified YOLOv3 (YOLOv3-SPP) network is used in [19] to recognize the LP. This network includes spatial pyramid pooling (SPP) block to recognize small and multiscale objects. YOLOv3 is also used in [6] to recognize the Persian characters. Images are resized to 240*240, and the threshold was defined based on the smallest characters that appear in the Iranian LP. The system records 95.05% for an end-to-end test. However, the training and testing dataset is not sufficiently large, and there are no such studies for Malaysian ALPR. The primary objective of this paper is to develop a robust deep learning-based approach for the Malaysian LPR with real-time performance.

3.0 PROPOSED METHOD

The proposed ALPR system is intended to be used in the toll payment collection system. Therefore, it must be accurate in real-time. Therefore, we have proposed an approach utilizing the single-shot object detector for real-time performance. The block diagram in Figure 2 depicts the proposed methodology. The process starts with the dataset acquisition, which will be used to train and test the model. Finally, the model is used in real-world inference.

The property of the single-shot detector is an advantage where the localization and classification are carried out in one stage. Thus, as soon as the input is given to the network, the result is presented. We extensively studied the single shot-detectors models such as R-CNN, Fast-RCNN, and YOLO. We found the YOLO is most promising in inference time and accuracy tradeoff [6]. The different YOLO networks v3[20] and v5[21] are developed with different hyperparameters chosen and compared for Mean Average Precision (mAP) and accuracy performance in a real-world application.

Figure 2 shows that a large raw dataset needs to be collected from the field in challenging weather and real-time conditions to develop a robust model. This raw data is cropped, cleaned, and labeled. Further, it is augmented to introduce different challenges for noise and blur related to camera focus and small rotations for camera position. Then the training of an ALPR system is performed using a large training dataset. Subsequently, the model is evaluated on the test dataset. The inference uses the detected license plate and performs the inference. In MLFF, the RFID tag should be matched with more than 90% accuracy from extracted text from LPR. The details of how the license plate dataset is collected and augmented are described in subsection 3.1, and training with selected deep learning models is described in subsection 3.2.

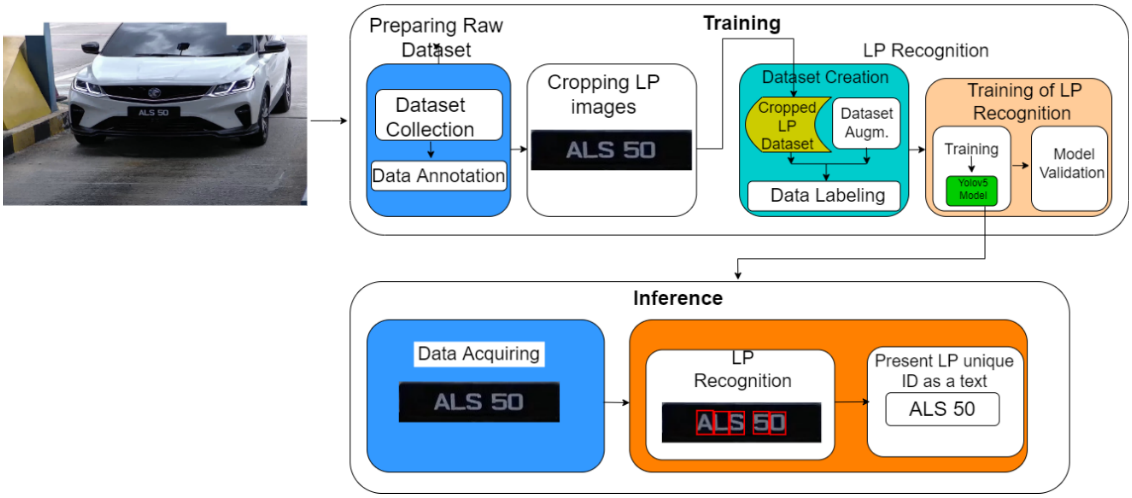


Figure 2: Proposed Methodology for robust real-time ALPR

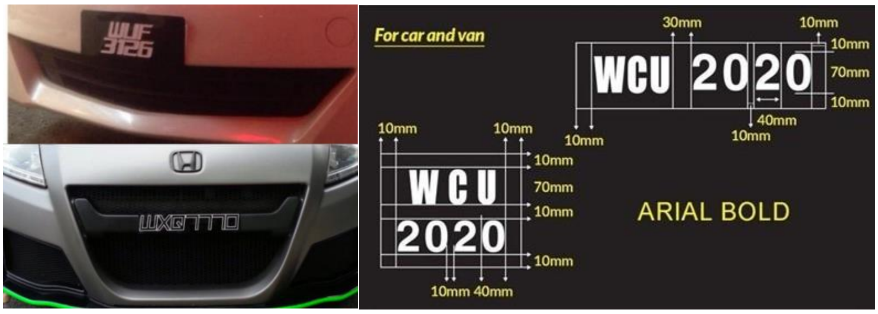
3.1 Malaysian License Plate (MLP) Dataset

During the dataset collection, we were keen to acquire as many variations as in the real-world situation, such as the change in illumination, day and night-time, different camera mounting, and different weather conditions. We also worked on the dataset augmentation technique to cover all possible situations which might not be adequately represented in the collected dataset.

It is worth mentioning the Malaysian License plate (MLP) standardization assigned by authorities and the problem of non-compliance to the standard layout of MLP. The Road Transport Department Malaysia (JPJ) clearly states the standards of MLP to the public. All specifications like size, layout, fonts, distances between characters, and others are provided in detail for the public as in Figure (3. a). The purpose is to ensure accurate identification and visibility. However, these standards are not strictly enforced, which makes it challenging to recognize the LP in many cases. Figure (3. b) shows some samples of the non-standard MLP. The non-predicted variations in the MLP make the task of the LPR harder, and accuracy is degraded for variations in the LP image. In this regard, the required solution must bring robustness to the system by covering many variations of license plate designs. Thus, we found that the solution is to acquire a sizeable dataset to cover all variations in the MLP. We collected a large challenging dataset in several batches as we conducted experiments and observed the system performance with a model trained with every batch of data.

3.11 Dataset Acquisition

As with any deep learning task, the dataset plays a significant role in the performance of the model. Looking at the studies on Malaysian LPs, we can see that there is no benchmarked dataset for Malaysian LPs. The raw dataset has been collected across four toll plaza booths during different daytimes and weather conditions from mounted cameras. Videos were taken in almost all environmental conditions, daytime, night-time, sunny weather, cloudy weather, rainy weather, noise, and illumination variations. Figure 4 shows the wide variations in our collected dataset.



(a) (b)

Figure 3: (a). Non-standard LP (b). Standard Specifications for Malaysian LP

Then, we started the preprocessing work for a real-world dataset for the training process by extracting video frames. For that, thousands of video frames were extracted. The LPR dataset is acquired from the raw dataset containing the LPs on vehicles. The LP region is annotated on the source image and points coordinates saved in XML files, and then we used a python code to extract and crop the region of interest (LP). The cropped images are provided to the LPR network for the training. We end up with fifty-eight thousand images.

3.12 Dataset Augmentation

Augmentation operations were conducted on one-third part of the dataset. We have taken 8000 images and divided them into eight groups and applied different augmentation strategies to ensure that augmentation is not applied to the same images. Hence the variation in the augmented dataset is guaranteed.

Moreover, applying augmentation operations on different image groups is better to avoid overrepresented classes when the augmentation is done on the same dataset. In addition, each augmentation group will contain a specific augmentation operation which will help control the conducted augmentation strategies to increase the system robustness.

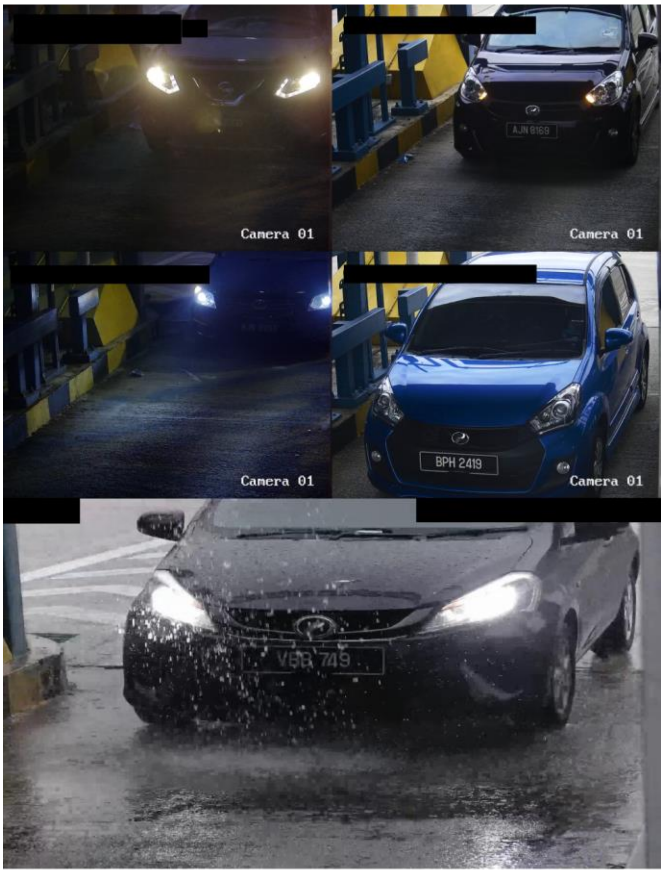


Figure 4: Variation in the dataset with low visibility and unconstrained environment

Augmentation actions include rotation, shear, brightness, exposure, blur, and noise. Every action contributes to getting a solution for each real-world problem. Augmentation operations on the eight thousand images resulted in 24 thousand images with wide variations in scenes. This new augmented dataset is added to the original dataset (34,000) to end up with (58,000) images. Figure 5 shows the different augmentations applied. We split the train, validate, and test dataset from this dataset to form balance classes. By doing so, we can ensure diversity in the dataset in terms of image quality. We also focused on capturing a non-standard LP and a sufficient number of instances of special LP. Finally, the total number of classes in the LPR dataset is 37 classes: 24 Latin letters (letter I is labeled as ‘1’ and letter O is labeled as ‘0’), ten digits (0-9), and three special plates, namely MALAYSIA, PUTRAJAYA, and PATRIOT. We named this dataset MYNO. This dataset is used for further training of ALPR models. A separate challenging dataset is created for testing the models. By fulfilling the terms and conditions, we will make the dataset MYNO version-v1 publicly available to the research community upon request.

PERINTIS eJournal

3.2 Training

The LPR process is complicated and challenging. It is due to the more complex features in the objects (characters) to be detected and the wide variations in the Malaysian LP designs. The MYNO, a large Malaysian LP dataset of 58K images, is used for training. After labeling is done for all the images in the dataset, the training process is conducted using the pre-trained model version YOLOv5[21]. Network parameters were set with a learning rate of 0.1, the intersection of union (IoU) threshold of 0.20, and an anchor threshold of 4. The input image size is set to 640*640. The models are trained on different batch inputs and the number of epochs and compared.

Figure 5: Data Augmentation samples with different effects
(Blur – noise – skewness – rotation)

3.3 Inference

The preprocessed LP image is fed to the second stage-trained model of LPR. The image is resized to 640*640 before the recognition step. Here, the task is to locate characters on LP with possible 37 classes comprising of ten digits, 24 Latin letters, and special plates. We propose a majority vote system for improved recognition. Since the system receives a series of video frames for the same vehicle in the region of interest (ROI) set for each toll booth, this ROI becomes an advantage for the LPR stage. In this regard, all captured frames are recognized and compared for the same vehicle. Since all frames here have the same characters, every character is compared in all frames, and the majority prediction is taken as the final prediction. This strategy is applied to all characters in the LP. This technique, called temporal redundancy [19], ensures higher system accuracy.

4.0 RESULTS

4.1 System performance based on the evaluation metrics

To see how the proposed Method of LPR performs, precision and recall measures are used. Precision refers to the number of correct detections (True Positives) to the number of all detections (either true or false) or:

$$Precision\ (P) = TP/(TP + FP) \qquad (1)$$

, where TP is the true positive, FP is the false positive, while Recall represents the number of the correct detections (True Positives) to the whole number of ground truths (objects labeled initially in training dataset) or:

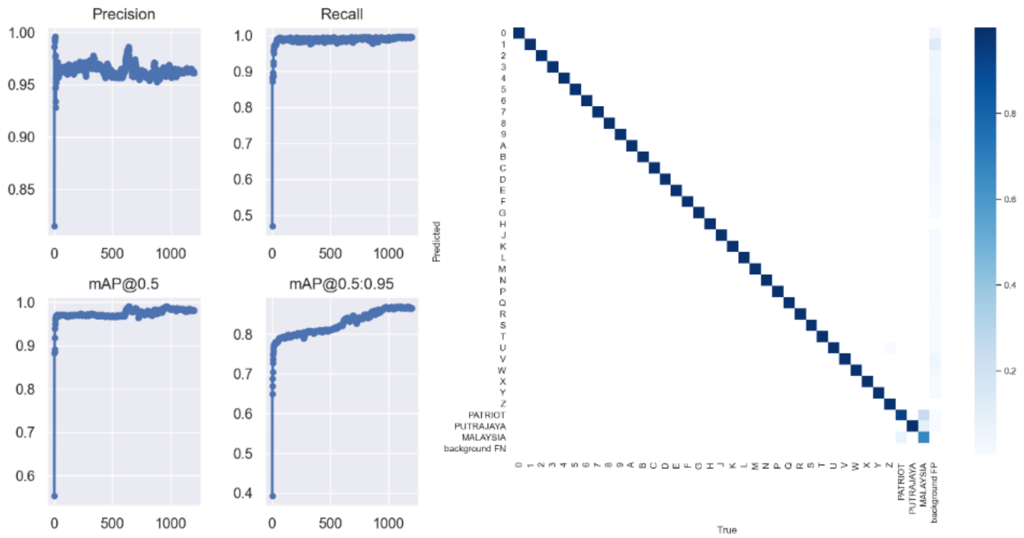
$$Recall(R) = TP/(TP + FN) \qquad (2)$$

, where FN is a false negative. Precision (P) and recall (R) results are also seen in Table. 2. With the recorded precision, 0.961 of the detections by the system are correct, while Recall with 0.996 indicates the percentage of the detected objects from the whole actual objects.

One of these metrics does not give a clear picture of the system's performance. Hence, combining both metrics in one evaluation metric called mean Average Precision (mAP) is the solution. The mAP is the plotting of precision values against all recall values in one graph. The bigger the area under the curve, the better the system performance. The mAP records 0.981at 0.5 IoU and 0.866 at 0.05 to 0.95 IoU. The result is depicted in Figure 6. it can be seen that the area under the curve is almost covered. Figure 6(b) also shows the confusion matrix where the FP and TN can be easily noticed. After 1200 epochs of training, we could achieve high performance with a mean average precision ***mAP (0.5) of 98.1%.***

Table 2: System Performance based on the evaluation metrics

Class	Number of samples	**labels**	P	R	mAP 0.5	mAP 0.5:0.95
Overall	58240	390149	0.961	0.996	0.981	0.866



(a) (b)

Figure 6: System Performance Results. a) Evaluation metrics, b) Confusion matrix

4.2 End-to-End LPR Performance Result

For the end-to-end performance evaluation, we looked at the accuracy instead, which means that if the final LP recognition result is the same as the actual LP, it is considered correct detection. Otherwise, misclassification in the characters will lead to false detection for the whole plate. That means 100% of characters are correctly matched in LP is the ultimate goal in result evaluation in the ALPR.

A test dataset of 1040 images collected over 16 months, including comprehensive environment variations (day and night time, sunny and rainy weather, varying in illumination and blurry), are used to get test accuracy. Different experiments have been conducted utilizing different training dataset sizes and data quality. Model 1 experiment contains 18K images that are clean from bad instances and augmentation. Model 2 has a dataset size of 40K, including the raw dataset with bad image instances (broken, extremely blurred) and random augmentation. In model 3, the final dataset (described in sec. 3.1) - the cleaned dataset from model 1 and the systematic augmented dataset - is introduced with 700 epochs. The proposed model has the same dataset as model 3 but with a more significant number of epochs (1200). Table 3 shows the results. The proposed system can detect and recognize LP with end-to-end recognition of 95.96% accuracy outperforming other models and showing that our strategy in preparing the training dataset in terms of the quality and augmentation process is very efficient.

4.3 Comparison of the proposed system with other methods

The performance of the proposed method is compared with other object detection (OD) methods for ALPR. A previous study [6] compared object detection networks and showed the performance of YOLOv3 over other OD methods like Fast R-CNN, Faster R-CNN, and YOLOv2. To our knowledge, no ALPR studies have compared YOLOv3 and YOLOv5. We performed a study on both models. To make a fair comparison, the networks were trained on the same dataset and number of epochs (400 epochs) and tested on the same test dataset, as shown in Table 4.

Eventually, the results show a clear conclusion that our proposed method (YOLOv5) provides the best performance. This improvement is due to the replacement of the first three layers of YOLOv3 with a single layer (Focus layer) [22] and the Auto-learning bounding box anchors in YOLOv5 [23].

Table 3: End-to-end Accuracy of LP Recognition on Different Models

Model	Training Dataset	No. Epochs	Accuracy
Model 1	18K	800	58.46
Model 2	40K	1000	94.04
Model 3	58K	700	95.77
Proposed	58K	1200	95.96

Table 4: The comparison of the Proposed system (YOLOv5) with YOLOv3

Method	Performance Evaluation		
	P	R	mAP 0.5
YOLOv3	0.979	0.989	0.991
YOLOv5 (Proposed)	0.978	0.972	0.993

5.0 CONCLUSION

This study proposed a robust license plate recognition system for Malaysian LP with an mAP (0.5) of 98.1%. The deep learning single-shot detection approach with YOLOv5 is utilized. The system was trained on a real-world dataset collected from different toll plazas around Malaysia containing comprehensive environment distinctions. The proposed MYNO dataset is a challenging Malaysian license plate dataset with over 50 thousand labeled images. We also introduced a systematic augmentation strategy to represent the possible real variations. The system was tested on a challenging test dataset with low visibility and an unconstrained environment, resulting in 95.96% end-to-end accuracy.

Acknowledgment

This work was supported by the Ministry of Higher Education of Malaysia under Fundamental Research Grant Scheme (project ID: FRGS/1/2019/ICT02/UIAM/02/7). We extend a special thanks to Mr. Vinod J. Nair to provide the digital fonts Myno Extended (© 2018) and JPJ 1 (© 2021).

REFERENCES

[1] Janowski, L., Kozłowski, P., Baran, R., Romaniak, P., Glowacz, A., & Rusc, T. (2014). Quality assessment for a visual and automatic license plate recognition. *Multimedia Tools and Applications*, *68*(1), 23–40.

[2] Saini, P., Bidhan, K., & Malhotra, S. (2019). A Detection System for Stolen Vehicles Using Vehicle Attributes with Deep Learning. *Proceedings of IEEE International Conference on Signal Processing,Computing and Control*, *2019-October*, 251–254.

[3] Mohandes, M., Deriche, M., Ahmadi, H., Kousa, M., & Balghonaim, A. (2016). An Intelligent System for Vehicle Access Control using RFID and ALPR Technologies. *Arabian Journal for Science and Engineering 2016 41:9*, *41*(9), 3521–3530.

[4] Silva, S. M., & Jung, C. R. (2017). Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks. *Proceedings - 30th Conference on Graphics, Patterns and Images, SIBGRAPI 2017*, 55–62.

[5] Zulkhairi, M. A., Mustafah, Y. M., Abidin, Z. Z., Zaki, H. F. M., & Rahman, H. A. (2019, October 1). Car Detection Using Cascade Classifier on Embedded Platform. *2019 7th International Conference on Mechatronics Engineering, ICOM 2019*.

[6] Tourani, A., Shahbahrami, A., Soroori, S., Khazaee, S., & Suen, C. Y. (2020). A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems. *IEEE Access*, *8*, 201317–201330.

[7] Angeline, L., Lau, H. K., Ghosh, B. K., Goh, H. H., & Kin Teo, K. T. (2012). Development of a License Plate Recognition System for a Non-ideal Environment. *International Journal of Simulation: Systems, Science and Technology*, *13*(3 C), 26–33

[8] Suandi, S. A., Soon, C. K., Lin, K. C., & Jeng, C. Y. (2012). Malaysian Car Number Plate Detection and Recognition System. *Australian Journal of Basic and Applied Sciences*, *6*(3), 49–59.

[9] Weng Keong, W., & Iranmanesh, V. (2016). Malaysian Automatic Number Plate Recognition System using Pearson Correlation. *In 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),* 40-45. IEEE

[10] Marzuki, P., Syafeeza, A. R., Wong, Y. C., Hamid, N. A., Nur Alisa, A., & Ibrahim, M. M. (2019). A design of license plate recognition system using convolutional neural network. *International Journal of Electrical and Computer Engineering*, *9*(3), 2196–2204.

[11] Ng, H. S., Tay, Y. H., Liang, K. M., Mokayed, H., & Hon, H. W. (2015). Detection and Recognition of Malaysian Special License Plate Based On SIFT Features. *arXiv preprint arXiv:1504.06921*.

[12] Abdullah, M., Ali, M., Bakhtan, H., & Mokhtar, S. A. (2017). Number Plate Recognition of Malaysia Vehicles Using Smearing Algorithm. *Sci.Int.(Lahore)*, *29*(4), 823–827.

[13] Shatnawi, N. M. (2018). Bees algorithm and support vector machine for Malaysian license plate recognition. *International Journal of Business Information Systems,* 28(3), 284-298.

[14] Koon Cheang, T., Shean Chong, Y., & Haur Tay, Y. (2017). Segmentation-free Vehicle License Plate Recognition using ConvNet-RNN. *arXiv preprint arXiv:1701.06439.*

[15] Selmi, Z., ben Halima, M., & Alimi, A. M. (2018). Deep Learning System for Automatic License Plate Detection and Recognition. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *1*, 1132–1138.

[16] Anson, A., & Mathew, T. (2020). Detection and Recognition of License Plate Using CNN and LSTM. In Jayakumari J., Karagiannidis G., & H. S. Ma M. (Eds.), *Advances in Communication Systems and Networks. Lecture Notes in Electrical Engineering*, 701–721. Springer.

[17] Wang, W., Yang, J., Chen, M., & Wang, P. (2019). A Light CNN for End-to-End Car License Plates Detection and Recognition. *IEEE Access*, *7*, 173875–173883.

[18] Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., & Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems*, *15*(4), 483–503.

[19] Henry, C., Ahn, S. Y., & Lee, S. W. (2020). Multinational License Plate Recognition Using Generalized Character Sequence Detection. *IEEE Access*, *8*, 35185–35199.

[20] J Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767.*

[21] Jocher, G., Stoken, A., Chaurasia, A., Borovec, J., NanoCode012, TaoXie, Kwon, Y., Michael, K., Changyu, L., Fang, J., V, A., Laughing, tkianai, yxNONG, Skalski, P., Hogan, A., Nadar, J., imyhxy, Mammana, L., … wanghaoyang0106. (2021). Ultralytics/yolov5: v6.0 - YOLOv5n "Nano" models, Roboflow integration, TensorFlow export, OpenCV DNN support. https://github.com/ultralytics/yolov5. (accessed Nov. 04, 2021).

[22] Jocher, G. (2021). YOLOv5 Focus Layer**.** Ultralytics (yolov5)**.** https://github.com/ultralytics/yolov5/discussions/3181m1**.** (accessed June. 28, 2022).

[23] Souidene Mseddi, W., Sedrine, M. A., & Attia, R. (2021). YOLOv5 Based Visual Localization For Autonomous Vehicles. *European Signal Processing Conference*, *2021-August*, 746–750.