Journal of the Society of Automotive Engineers Malaysia
Volume 7, Issue 1, pp 56-66, January 2023
e-ISSN 2550-2239

Vehicle Number Plate Detection and Verification Using YOLO Frameworks

S. Y. S. H. Hussein, M. H. M. Ariff and **M. A. A. Rahman***

Malaysia-Japan International Institute of Tech., Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia

*Corresponding author: azizi.kl@utm.my

ORIGINAL ARTICLE *Open Access*

Article History:

Received
2 Dec 2022

Accepted
20 Feb 2023

Available online
1 Mar 2023

ABSTRACT – *Vehicle number plate detection is a process of identifying and localizing license plates in images or videos, which is crucial in different applications such as toll collection, law enforcement, parking management, traffic monitoring, and access control systems. The number plate detection algorithm is a cultivated but imperfect technology. The traditional detection algorithm is easily affected by environmental factors such as light, shadow, and background complexity, resulting in a failure in accuracy and better efficient detection. With the development of deep learning, YOLO (you only look once) is an outstanding image-processing technique that leverages convolutional neural networks (CNNs) to identify and recognize license plates within images and videos. This project concerns the application of two YOLO frameworks (YOLOv3 and YOLOv4) for accurate number plate detection. The system aims to compare the effectiveness of different frameworks by outlining each one's unique advantages and disadvantages by determining the best performance in terms of confidence level and accuracy. The verification is applied to the Malaysian number plates following the regulation of the Malaysian transportation system. It differentiates between the standard and non-standard number plates using the YOLO frameworks. To accomplish the dataset is manually captured for training and evaluation on images and video. By training these frameworks separately, both frameworks can learn the detection of the plate. Based on the desired outcome, accuracy rates of 90–95% or higher must be attained to affirm a good performance for YOLOv3 and YOLOv4, which perform significantly better than the conventional detection techniques. Overall, the project will continue to develop the YOLO frameworks for providing recognition performance metrics using the classification results.*

KEYWORDS: Number plate detection, YOLO, Convolutional Neural Network (CNN), Optical Character Recognition (OCR)

1. INTRODUCTION

Nowadays, traffic monitoring and law enforcement use number plate detection more frequently. There have been several proposed methods for recognizing license plates, but each one has advantages and cons of its own. Therefore, classification and detection issues will be resolved by applying deep learning algorithms. It is utilized in several fields, including computer-aided diagnostics, natural language processing, object identification, and recognition. Various methods, including CNN, have been developed for object detection and recognition. However, since 2016, a new model was developed, namely YOLO (Kumar, 2021). Instead of conducting these two procedures in two scans, a real-time object detection system accomplishes object detection and object recognition in one scan.

Standard license plates are auto license plates with capital letters in a certain letter and digit form specified by the state code. As seen in Figure 1, it consists of three letters, four numerals, and an “em space” (word space). However, Figure 2 shows that after 2016, the new standard is three letters with four numerals, but one out of the three letters came after the numerals with an additional “em space” added between the letters and numbers (Jawi & Isa, 2017; Nair, 2018). The length of the LP number controls the number of distinct combinations and, thus, the total number of LPs that may be printed. For example, as MLP is made of four numbers and three letters, the number of unique combinations or possibilities of each position is 24 x 24 x 24 x 104 = 13,824,000. The value 24 represents the number of alphanumeric characters excluding I and O to prevent confusion with the numbers "1" (one) and "0" (zero), respectively. The value 10 is the number of digits (0-9), and 4 is the numerals in the plate.

FIGURE 1: Single “em space” in old number plate before 2016 [2]

FIGURE 2: Two “em space” in the new number plate after 2016 [2]

The significance of LP has never been more apparent because of the advancement of autonomous detection technology. Each jurisdiction creates and preserves its unique design for LPs to enhance their identification and recognition (Nair, 2021). Furthermore, non-standard license plates are number plates with various sizes, decorations, fonts, or arrangements of alphanumeric characters. It is not a complaint with JPJ regulations due to its variations. For instance, the size can be larger or smaller than the standard size, have a different background color, have fewer or more character numbers, and have a font style will be other than FE-Schrift. In this project, the dataset used for non-standard will be public or from a country near Malaysia, such as Thailand.

According to the Ministry of the Economy, it showed that since September 2021, the Malaysian population increased to 32.6 million, which leads to traffic congestion, difficulty in traffic monitoring, and law enforcement (DOSM, 2021). It proved that Malaysians drive for 269.8 million hours annually or 1,037,050 hours daily. That statistic emphasized the necessity of vehicle license plate detection and identification, particularly for toll-collecting systems (Khalifa et al., 2007). Considering the issues above, the following objectives must be accomplished:

- To identify the standard and non-standard Malaysian plates;
- To detect license plates on vehicles in an image using YOLO frameworks; and
- To validate a video stream and compare the performance metrics of the YOLO frameworks.

The project scope is to build and test a system using YOLO that can be implemented in the future for the cameras used in the multi-lane free flow system or gateless toll collection system that will help to collect tolls without the requirement of physical toll booths or barriers for vehicles to stop or slow down when passing through tolling points (Lee, 2018). Also, the training of the dataset for standard and non-standard will be applied to the algorithm for differentiation purposes. The research underscores the significance of image recognition in monitoring systems to advance their overall performance. However, the Malaysian License Plate Recognition (MLPR) system encounters constraints in gathering local license plate data, as the open-source related data does not surpass 30 images, which leads to collecting the data manually, which is time-consuming if the training is done on a vast dataset. The user of this system will need to install the software and libraries manually and collect their dataset for their country. One critical drawback is finding the standard and non-standard datasets because this term was not established or applied before. So, the data collected is manually captured by phone camera in Malaysia, and for non-standard, the dataset is from public sources. The difficulty in finding the dataset is due to the country's privacy policy. Another limitation is that YOLOv3 can only perform detection without recognition, whereas YOLOv4 does the detection and recognition of the characters, making it more practical to use in real-life situations than YOLOv3.

The traffic monitoring system is a significant application; used to collect data, analyze traffic patterns, and optimize traffic management. This system utilizes various technologies such as cameras, sensors, and data analytics to monitor traffic flow, detect congestion, and provide real-time information to commuters in the gateless toll collection system, which is crucial for transportation planning and improving traffic flow. This project aims to develop an object detection design approach for LPR systems. Two darknet models will be utilized, with the Malaysian license plate serving as the standard dataset and a public dataset as the non-standard dataset. It compares the performance metrics of

YOLO frameworks, an advanced object detection algorithm. That enables vehicles to pass through toll plazas or designated areas at regular driving speeds, ensuring a smoother flow of traffic and reducing congestion through more accessible payment methods. If recognition is standard, the toll amount will be deducted from the personal account, and if not standard, either the user will pay more, or the process will be rejected. By using parked automobile photos as a test set, high accuracy will be attained by modifying the object detection technique in MLPR. That will help develop a system that accurately detects and recognizes license plates within one click. Table 1 shows the comparison between the two YOLO versions.

TABLE 1: Comparison between YOLOv3 and YOLOv4

Model	YOLOv3	YOLOv4
Backbone	Darknet53	CSPDarkent53
Neck	FPN	PANet+SPP
Head	YOLO	YOLO
Activations	Leaky-Relu	Leaky-Relu + Mish
Bounding Box Regression Loss	MSE loss	CIoU-loss
Data Augmentation	Pixel-wise adjustments	Mosaic
Regularization	Dropout	DropBlock
Normalization	Batch Normalization (BN)	Cross mini- Batch Normalization (CmBN)
Attention Module	None	Spatial Attention Module (SAM)
Loss Function Trick	None	Class label smoothing, Grid sensitivity
Advantages	• Real-time detection • can effectively detect and localize license plates with good precision and recall rates. • detect multiple license plates within an image simultaneously. • Flexibility in hardware	• Improved accuracy because of the use of PANet and Mish activation function which helps in getting high recall and precision. • can effectively detect and localize license plates with different designs, variations, and orientations. • Faster inference speed which is more suitable for real-time detection. • Multi-scale predictions. It is beneficial in license plate recognition, as license plates can vary significantly in size depending on the distance and camera viewpoint.
Disadvantages	• Difficult to detect small-size license plates. • Fine-textured or complex plates may face issues in accuracy. • Variability in lighting conditions such as shadow, glare, or overexposure. Training data requirements need a large and diverse dataset	• Increased computational requirements such as high processing power and memory. • Larger model size • Training data requirements: Collecting and annotating such datasets for training can be time-consuming and resource intensive. • Complexity in implementation

2. EXPERIMENTAL SETUP

Figure 3 illustrates the workflow of the Malaysian License Plate Recognition (MLPR) used to measure the detection and recognition performance using the YOLO algorithm. It shows the process of applying the detection application on the license plate. Based on that, the accuracy levels can be measured in YOLOv3 and YOLOv4 in images and videos. The procedure of every step in the flowchart is explained in section 2.1.

2.1 Extraction Process of License Plate Region

Firstly, YOLO divides the input image into a grid and predicts bounding boxes and their corresponding class probabilities for each grid cell. The grid cell responsible for detecting the license plate has a high probability associated with the 'license plate' class. Once the license plate class is identified, using regression, YOLO refines the initially predicted bounding box coordinates. This step helps improve the bounding box's accuracy and alignment around the license plate region.

Secondly, characters on the license plate are isolated after determining the location of the license plate in the previous process. This process is where the height and width will be calculated and cut using four specified angles, x-min, x-max, y-min, and y-max. The license plate image will be cut directly from the original RGB format input image, and the license plate will be converted to grayscale and then to binary for morphological operations. It is implemented to eliminate unwanted noise. In ensuring this process is successful, morphological techniques are performed license indirectly. A sample of the conversion method is shown in Figure 4.

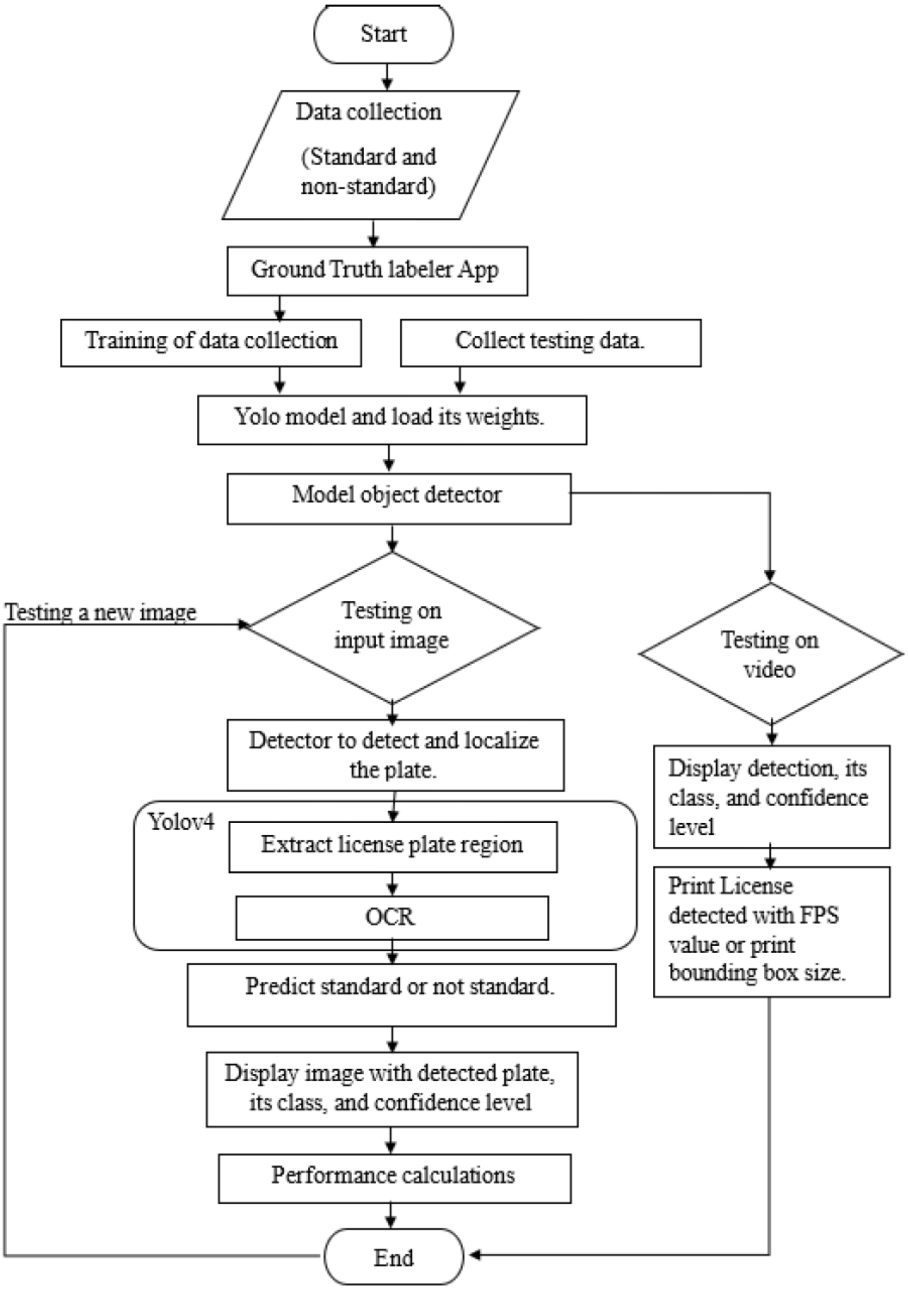

FIGURE 3: Overall flowchart of the MLPR

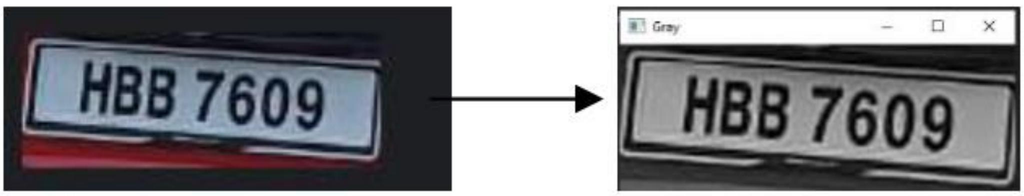

FIGURE 4: Cropped to grayscale

The third procedure is the morphological operations that execute the following processes:

(i) **Threshold** – The threshold is applied to pre-process the image and convert it from grayscale to a binary image, where the license plate regions can be easier to detect or segment. It involves setting a threshold value; every pixel in the image is compared to this threshold. For example, the value used for the threshold in yolov4 and v3 is 0.5, with an image input size of 416. The threshold is shown in Figure 5.

FIGURE 5: Otsu threshold

(ii) **Dilation** – It is used to expand or enlarge regions of interest in a binary image. It is applied to the binary image obtained from thresholding. It helps to make the license plate regions more prominent and connected, making it easier to apply contouring for the next step to identify and extract the complete license plate region. This result is shown in Figure 6.

FIGURE 6: Otsu threshold

(iii) **Character Segmentation** – The license plate images are segmented. As well as the characters in the image by identifying the connected pixels. To make it simpler to recognize each character, each character is divided into an independent image, ROI. The projected result of the character segmentation method is displayed in Figure 7. It uses contour analysis to get the ROI.

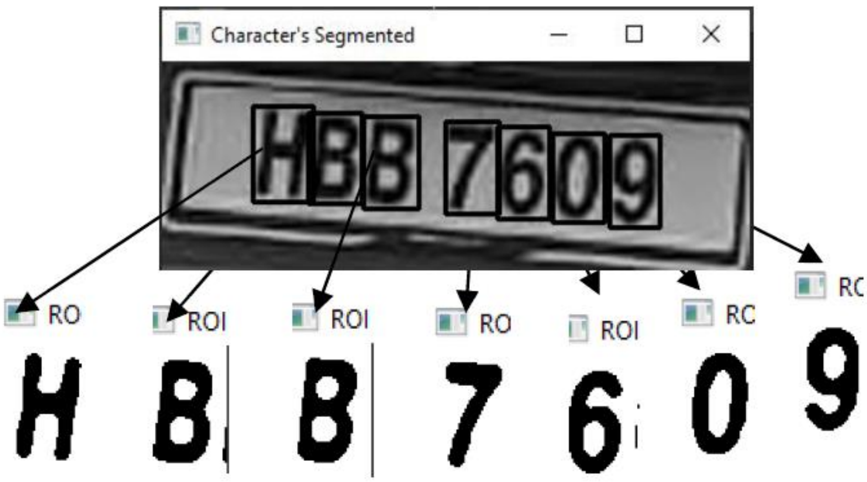

FIGURE 7: Character segmentation

(iv) **Normalization** – Normalization techniques, such as mean subtraction and division by scale factor, can be applied as a pre-processing step to normalize the input images. These normalization techniques to the input images ensure that the license plate detection and recognition model trained with YOLO receives consistent and standardized input data. This technique enables the model to focus on relevant features and patterns in the license plate images, making it more robust and accurate in detecting and recognizing license plates in different scenarios (Google, 2022). Moreover, YOLO also incorporates batch normalization. During the training process, batch normalization helps to normalize the activations within each mini-batch of license plate images, reducing the internal covariate shift. This action improves the stability and convergence of the YOLO model during training, enabling it to better detect and recognize license plates by effectively learning from the normalized activations within each mini-batch.

2.2 Compilation and Optimization

The stochastic gradient descent method 'tf.keras.optimizers.Adam' utilizes an adaptive estimation of first and second-order moments. In this context, the second-order epochs are set to 30, while the first-order epochs are set to 20. Adam is a commonly used adaptive optimizer, especially in image recognition tasks. It performs well with sparse data, making it suitable for datasets with varying learning rates. The method is particularly effective for handling large-scale data and parameter problems while remaining computationally efficient. It also requires less memory and is unaffected by the diagonal rescaling of gradients (Lakshmanan et al., 2020). The initial learning rate is set at 1e-3 in the specific model being discussed, while the final learning rate is set at 1e-6. The higher learning rate enables faster training and testing of the data, but there is a trade-off as it may result in suboptimal weights and lower accuracy. Figure 8 displays a partial summary of the TensorFlow Keras model.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 416, 416, 3  0           []
                                )]

 conv2d (Conv2D)                (None, 416, 416, 32  864         ['input_1[0][0]']
                                )

 batch_normalization (BatchNorm  (None, 416, 416, 32  128        ['conv2d[0][0]']
 alization)                     )

 tf.math.softplus (TFOpLambda)  (None, 416, 416, 32  0           ['batch_normalization[0][0]']
                                )

 tf.math.tanh (TFOpLambda)      (None, 416, 416, 32  0           ['tf.math.softplus[0][0]']
                                )

 tf.math.multiply (TFOpLambda)  (None, 416, 416, 32  0           ['batch_normalization[0][0]',
                                )                                 'tf.math.tanh[0][0]']

 zero_padding2d (ZeroPadding2D)  (None, 417, 417, 32  0          ['tf.math.multiply[0][0]']
                                )

 tf.concat_15 (TFOpLambda)      (None, None, 4)      0           ['tf.__operators__.getitem_9[0][0
                                                                 ]',
                                                                  'tf.__operators__.getitem_10[0][
                                                                 0]',
                                                                  'tf.__operators__.getitem_11[0][

==================================================================================================
Total params: 64,003,990
Trainable params: 63,937,686
Non-trainable params: 66,304
__________________________________________________________________________________________________
```

FIGURE 8: Model summary of YOLOv4

2.3 Optical Character Recognition (OCR)

Character recognition is the last stage in modeling. Template matching is one of the procedures used in optical character recognition. After the ROI is detected for each character on the plate, the Tesseract OCR will read it, send the result to the terminal, and extract the characters in the license detected. The recognition of characters depends on –psm 8, which treats the image as a single word. This mode instructs Tesseract to process the input image as a single word, assuming that the text in the image is a single word or a few closely spaced words. Later, when the image is displayed. The number of objects detected, standard or not standard, confidence level, and license plate detection and recognition will be shown in Figures 9 and 10. Then the loop will be continued while running the code for a new image. This process only applies to YOLOv4.

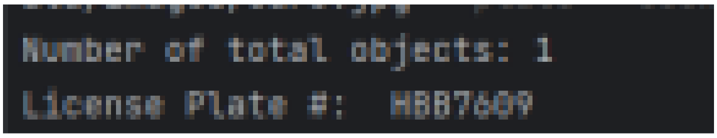

FIGURE 9: Output in terminal

FIGURE 10: Image displayed

2.4 Performance Metrics

To calculate the performance, the cumulative value of TP, FP, and FN should be calculated manually, and then the prediction formulas and recall get the AP and mAP (Zakaria et al., 2020). TP is the correct character prediction, FP is the incorrect character prediction, and FN is the ignored character means segmentation ignores these characters. These metrics can be observed from the output of license recognition. Moreover, the average confidence level will help get the overall accuracy of Yolo frameworks and FPS to be detected from the validation of the video.

3. RESULTS AND DISCUSSION

The preliminary results of this project are the license plate image pre-processing and the classification results of the plate, whether it is a standard or not, which are illustrated and discussed in the subsequent sections.

3.1 Main Stages of the Proposed MLPR Approach – YOLOv4 & YOLOv3

As shown in Figures 11 and 12, the process of the Yolo framework has three stages: data collection, training, and testing. The difference between the process of YOLOv4 and YOLOv3 is clearly shown. In Figure 11, data collected were imported into the model for training, then testing was applied with OCR for character recognition; and YOLOv4 shows the result, including detection, confidence level, and text extracted in one step, whereas Figure 12 shows that YOLOv3 follows same stages as v4 excluding the OCR technique so it only can show the detection and confident level in one step. These stages were explained in the methodology.

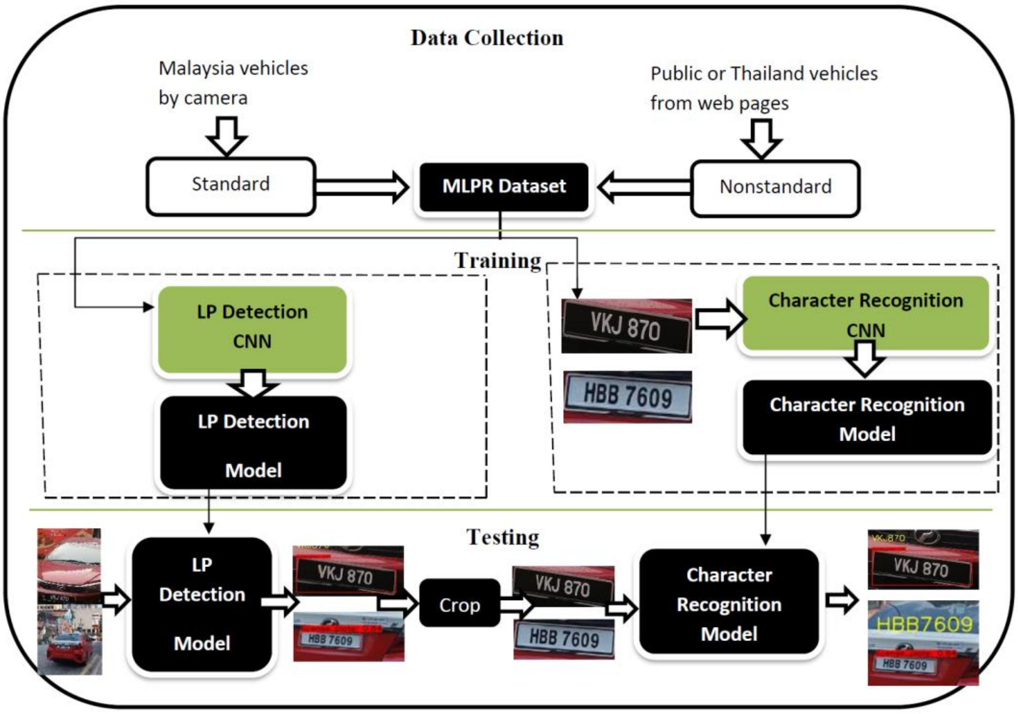

FIGURE 11: The process of YOLOv4 with the results in each stage

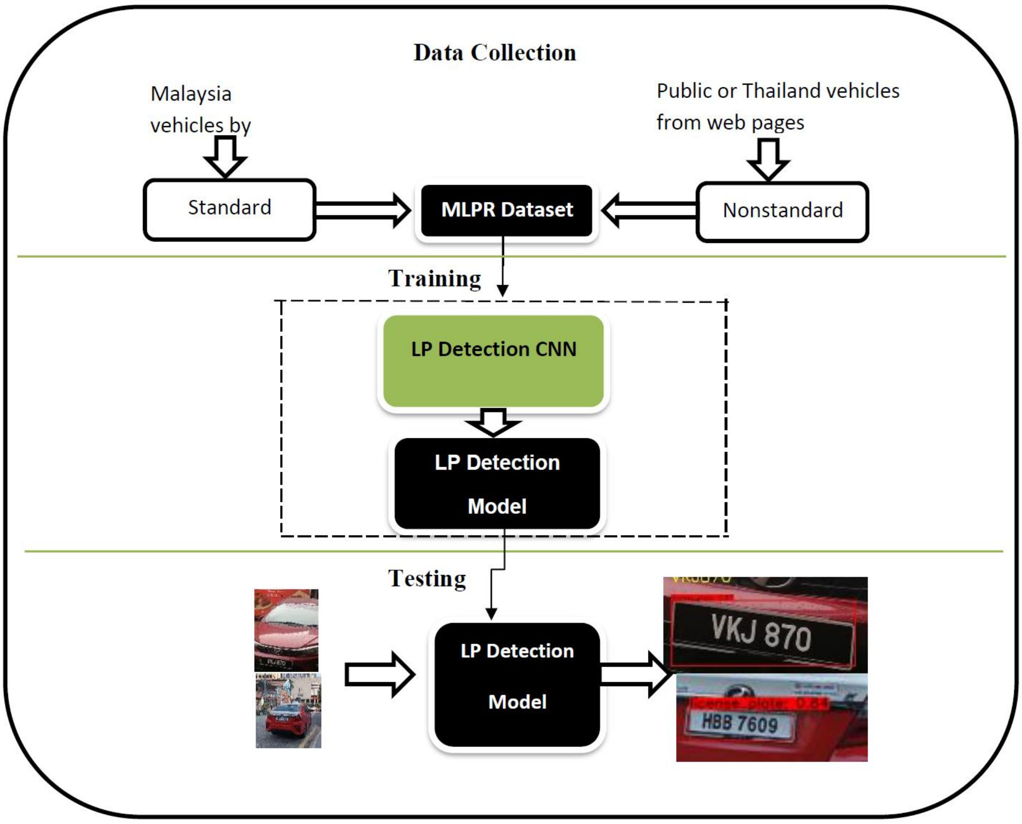

FIGURE 12: The process of YOLOv3 with the results in each stage

3.2 Pre-processing Results in YOLOv4

After running the command, the result shows the extraction from Tesseract OCR, confidence level, and label of the standard license plate, as shown in Figure 13.

FIGURE 13: Output of the image

As shown in Figure 14, the recognition could not read the letter “I” because the system usually excludes the letters “I” and “o” due to the confusion with the numbers “1” and “0”. In this case, the letter “I” is an FN.

```
eo ./data/video/license_plate.mp4 --output ./detections/recognition.avi --plate
License Plate #:  5EBC
FPS: 0.24
License Plate #:  S5EBC
FPS: 0.26
License Plate #:  LS5EBC
FPS: 0.25
License Plate #:  S5EBC
FPS: 0.39
License Plate #:  LS5EBC
FPS: 0.37
License Plate #:  S5EBC
FPS: 0.34
License Plate #:  S5EBC
FPS: 0.40
License Plate #:  LS53BC
FPS: 0.38
License Plate #:  LS5E8C
FPS: 0.40
License Plate #:  LS5EBC
FPS: 0.36
License Plate #:  LS5EBC
FPS: 0.39
```

FIGURE 14: Top – Testing of FPS reading with license plate extraction from video; Bottom – Testing of validation results on a public video

3.3 Pre-processing Results in YOLOv3

As shown in Figure 15, the first four numbers (0.41222674, 0.60945445, 0.17895153, 0.02561579) represent the coordinates and dimensions of the bounding box. Specifically, they indicate the normalized values of the box's top-left corner (x, y) coordinate, width, and height.

The fifth number (0.9999758) represents the confidence or probability score associated with the detected bounding box. It indicates the algorithm's confidence that the object within the bounding box is a license plate. Higher scores indicate a higher confidence level.

The sixth number (0.99991333) is also a confidence score but represents the algorithm's confidence that the detected bounding box contains an object from any class. This score is independent of the specific class (license plate) and represents the overall confidence of object detection.

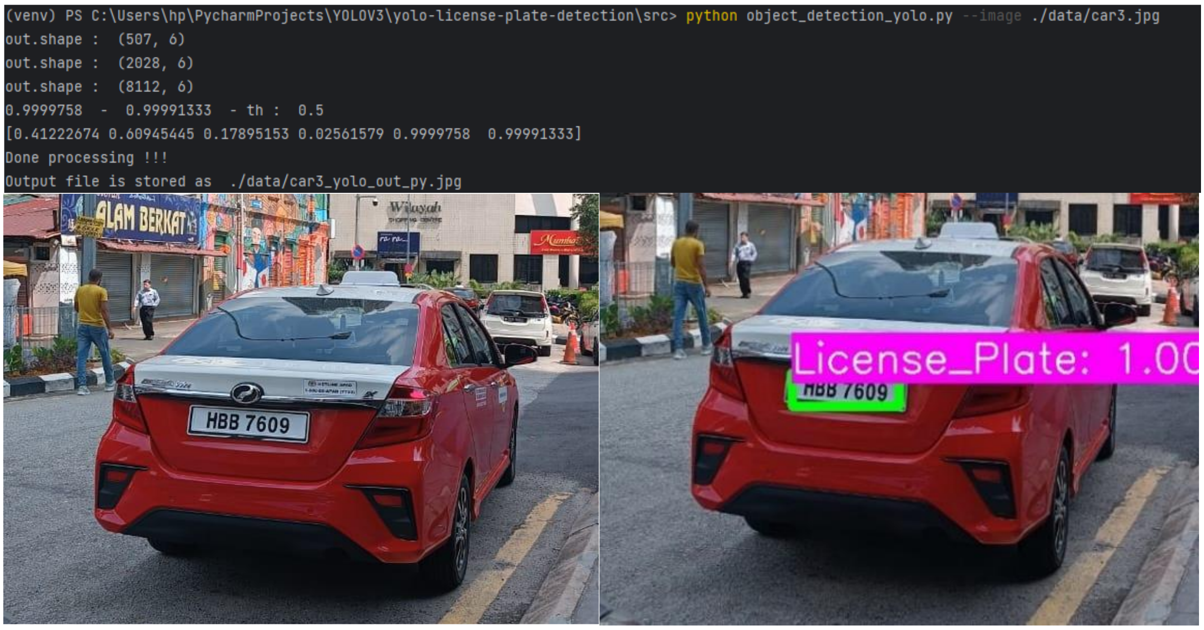

FIGURE 15: Output in the terminal for YOLOv3; Loaded original image; and the output of the image for YOLOv3

As shown for YOLOv4 in Figure 13, it gives the confidence level as 0.84, which is dependent on bounding box accuracy and character recognition, whereas YOLOv3 in Figure 15 only shows the confidence as 1, depending on the bounding box accuracy. That is why the value of confidence in YOLOv3 is higher than the value of confidence in YOLOv4. Theoretically, YOLOv4 should be more accurate in results as it includes the accuracy of characters with detection, which makes more sense and makes it easier to get a better analytical result than YOLOv3.

4. CONCLUSION

This article explained the background theory for deep learning and YOLO. A comparison in terms of advantages and disadvantages was made to get a clear understanding of both frameworks. Also, it clearly explains the process of detection and recognition in YOLOv4 and the process of YOLOv3, proven by some testing results on both frameworks. These outcomes will be used to have better results with the differentiation of standard and not the standard of plates. Based on the testing results obtained, YOLOv3 can be improved by adding OCR to get a better comparison with YOLOv4 in terms of accuracy and confidence level. Moreover, by adding the concept of standard and not standard, the algorithm can learn a new dataset detection.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support from the Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under the UTM Fundamental Research grant (22H27) and partial contribution from the ASEAN NCAP Collaborative Holistic Research (ANCHOR) under grant number A3-C21.

REFERENCES

DOSM (2021). Quarterly Demographic Statistics. Department of Statistics Malaysia. Retrieved from https://www.dosm.gov.my/portal-main/release-content/6f55f6d9-8b7c-11ed-96a6-1866daa77ef9

Google (2022). Normalization – Machine Learning. Google for Developers. Retrieved from https://developers.google.com/machine-learning/data-prep/transform/normalization

Jawi, Z. M., & Isa, M. M. (2017). Paying big bucks for cool number plates. Journal of the Society of Automotive Engineers Malaysia, 1(2), 82-85.

Khalifa, O. O., Khan, S., Islam, M. R., & Suleiman, A. (2007). Malaysian vehicle license plate recognition. Int. Arab J. Inf. Technol., 4(4), 359-364.

Kumar, V. (2021). It's all YOLO. Medium. Retrieved from https://vikmr.medium.com/its-all-yolo-58752e449683

Lakshmanan, V., Robinson, S., & Munn, M. (2020). Machine learning design patterns. O'Reilly Media.

Lee, J. (2018). Touch 'n Go releases more details of the RFID toll system. Paul Tan's Automotive News. Retrieved from https://paultan.org/2018/09/14/touch-n-go-releases-more-details-of-rfid-toll-system/

Nair, S. (2021). Car plates: Malaysia rules and regulations according to JPJ. DS&F. Retrieved from https://www.dsf.my/2021/07/jpj-malaysia-car-plates-specifications/

Nair, V. (2018). A standardised number plate typeface for Malaysia. Kreatif Beats. Retrieved from https://kreatifbeats.com/2018/08/01/a-standardized-number-plate-typeface-for-malaysia-proposal/

Zakaria, N. J., Shapiai, M. I., Rahman, M. A. A., & Yahya, W. J. (2020). Lane line detection via deep learning based-approach applying two types of input into network model. Journal of the Society of Automotive Engineers Malaysia, 4(2), 208-220.