



## Automatic Vehicle Classification for Electronic Toll Collection using YOLOv3

Mohammad Akid Bahrin<sup>1</sup>, Norhazwani Md Yunos<sup>2</sup>, Noor Fazilla Abd Yusof<sup>3</sup>

<sup>1</sup>Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia, mohdakid1410@gmail.com

<sup>2</sup>Center for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia, wanie.my@utem.edu.my

<sup>3</sup>Center for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia, elle@utem.edu.my

### ABSTRACT

This paper proposes the application of object detection and classification application in the area of intelligent transportation system, that is in Electronic Toll Collection (ETC) system. ETC is a cashless toll collection system, where there should be no human being involved at the toll booth anymore. Due to different toll rate applies to different type of vehicle, there is a need to have a system to detect and classify the type of vehicle passing through the toll booth. Thus, this paper proposed an automatic vehicle classification using an algorithm based on YOLOv3. The proposed model is tested and validated by extensive experiments. The result shows that the propose model manage to identify the type of vehicle in more than 92.12% accurate.

**Key words :** Automatic Vehicle Classification, Electronic Toll Collection System, Object Classification, YOLOv3.

### 1. INTRODUCTION

In the recent age, studies in intelligent transportation system, for short ITS, is progressing tremendously. The aim of ITS is to improve the transportation system, ranging from safety to environmental performance, and from productivity to mobility of the traffic planners and road users [1]. In the field of computer science, growing interest is focused on vehicle detection and classification, which aims to provide information such as counting number of vehicles, measuring a vehicle speed, predicting a traffic flow, identifying and classifying a vehicle and many more.

In real life application, vehicle detection and vehicle classification methods need an efficient system so that it can be operate in real-time condition, as well as in the change and different weather conditions [2]. Many studies have been conducted in vehicle detection and classification for the last

five years, some of them are [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

In this study, the object detection and classification is applied in the ITS being focus at Expressway network, in Malaysia context. The Malaysian Expressway network forms the first backbone network of national highways. The user of the expressway is charged upon exit with respect to the type of vehicle and the distance of travel in the expressway. Malaysia Expressway network uses Electronic Toll Collection (ETC) system for collecting the fees from users. The ETC system is a cashless toll collection system, where there should be no human being involved at the toll booth anymore. But, one problem arises when there is a different toll rate for different type of vehicle. Because of this problem, some toll booths still require a human to set the toll rate for each vehicle passing the booth.

To mitigate this problem, we propose an automated vehicle detection and classification at the toll booth. It uses one of the standard model architectures that commonly used by industry for object detection, which is known as YOLO, or in full sentence, it is known as You Only Look Once. There is also another option of model architecture that can be applied, but it needs a consideration in every aspect based on its requirements and its capabilities. Progressively, YOLO has come up with many versions. The latest of YOLO is named as YOLOv3. This version performs well especially in small object detection where it runs with the speed of 30 frames per second without losing the accuracy of detection [9].

The YOLOv3 uses regression problem where it works by predicts class probabilities and bounding box offsets directly from full images using a single forward convolution neural network. The proposed generation region and feature resampling will completely be eliminated during the process, and all stages in a single network will be encapsulated in order to form a true end-to-end detection system [13].

YOLOv3 is already applied to many applications of object detection. Recently, Luo et al. [9] used YOLOv3 in traffic, in detailed, they used YOLOv3 to assist unmanned vehicle system in classifying and detecting the object in traffic scene accurately. Yi et al. [10] also used YOLOv3 in traffic management, where they applied YOLOv3 to improve the accuracy of the existing real-time pedestrian detection. While Zhang et al. [3] used YOLOv3 to distinguish driver and passenger and form a one-to-one correspondence between vehicles and drivers.

Other than in traffic management system, YOLOv3 also has been applied in other areas, such as in underwater object detection, as reported by Yang et al [14], where they used YOLOv3 to locate and classified the images of sea cucumber, scallop and sea urchin. Besides, Jing et al. [15] used YOLOv3 to detect defect on fabric. While Tian et al. [16] used YOLOv3 to detect the differences in apple growth stages.

The remainder of this paper is organized as follows; Section 2 introduces the materials and methods involved in this study. We state our main result and discussion on our proposed model in Section 3. Next, Section 4 concludes the paper, and finally Section 5 gives the recommendation of future study.

## 2. MATERIAL AND METHOD

In Malaysia, there are five category of vehicles that has been classed for the highway users for all expressway except for the Penang Bridge expressway and the SMART motorway tunnel. The toll rate is different for each class. The five classes are named as Class 1, Class 2, Class 3, Class 4 and Class 5. Private cars or vehicles with two axles and either three or four wheels, excluding taxis, fall under Class 1. Van and other small good vehicles with two axles and six wheels, exclude busses, are categorized in Class 2. Large trucks or vehicles with three or more axles, excluding busses, are assembled in Class 3. All types of taxis are labelled in Class 4, while all types of busses are labelled in Class 5. Figure 1 shows the description of vehicles for each class.




Class	Icon	Description
Class 1		Vehicles with 2 axles and 3 or 4 wheels excluding taxis
Class 2		Vehicles with 2 axles and 5 or 6 wheels excluding busses
Class 3		Vehicles with 3 or more axles
Class 4		Taxis
Class 5		Buses

Figure 1: Classes of Vehicles for the Malaysia Expressway users

As for model training and implementation, we use one of the standard model architectures that commonly used by industry for object detection, namely YOLOv3. YOLOv3 is the latest variant among other popular real-time object detection system. YOLOv3 is reported to be extremely fast and accurate [17]. It works by applying a single neural network to the full image, and the network divides the image into regions and predicts bounding boxes and probabilities.

### 2.1 Data Collection and Preparation

Before the model can be trained, a sufficient image related to the object need to be collected. The images were collected from open online dataset resources, such as BIT-vehicle dataset. Other than that, the images were also gathered by utilizing Microsoft’s Bing Image Search API, which is part of Microsoft cognitive services used to bring Artificial Intelligence (AI) to vision, speech, text and many more ranging from apps to software.

A total of 2,268 images of vehicles, where approximately about 400 images of different required classes were prepared for this study.

For training in YOLO, all images must be labelled, and the annotation needs to be in .txt format. The annotation must contain the object class number coordination for each object class, such as <object-class> <x-center> <y-center> <width> <height>.

Since all the images data were downloaded from the open source online database, then all the images have not annotated with any label. For easier process, Labelling tool was used because of several reasons. One of them, because it is an open-source graphical annotation tool, hence it is easy to install. Another reason, because the annotations were saved as XML files in PASCAL VOC format, which is supported in YOLOv3. The process of labelling the annotation in annotation tool is shown in Figure 2.

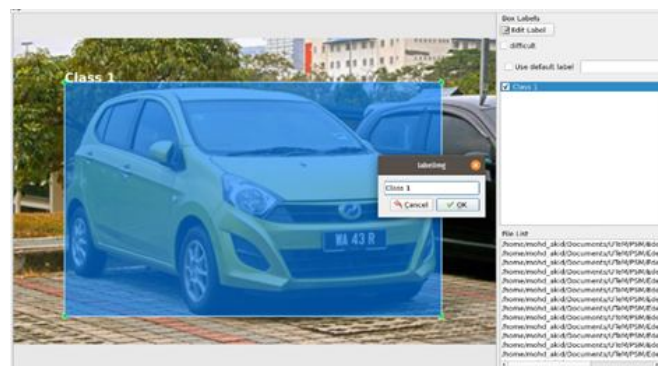


Figure 2: Labelling Annotation Tool

## 2.2 Model Training

In training the model, we used an open source neural network framework that support CPU and GPU computation, that is Darknet [18]. All of the prepared images were copied into the Darknet directory. Next, a suitable adjustment of the parameters need to be conducted to suit with the data and the study. There are batch hyper-parameter, subdivision configuration, width, height and channels, momentum and decay, learning rate, steps, scales and burn in, and number of iterations.

Batch hyper-parameter specifies a batch size used throughout training. The training process involves an iterative update of weights of the neural network based on the mistake made on the training of the dataset. It is not practical to use all images at once in updating the weights during training. Hence, a small group of images was used per iteration, and this group is called the batch size. When the batch size was set equal to 24, this means a set of 24 images were used in one iteration to update the network parameters.

In Darknet, a parameter called subdivision is used to help the GPU in order to process the batch size fraction by fraction at one time. This parameter should be used to avoid issues of GPU out of memory to process the batch size at once. The parameter value may start with subdivision = 1, if the error of out of memory turns up, then the value can be changed to multiples of 2, such that 2, 4, 6, 8, 16, until the training process were successful. The GPU will process the batch or subdivision of images at any pace, but the full batch or iteration is considered complete after all batch of images were processed.

Another parameter to be considered is configuration parameter, namely weight, height and channels. These configuration parameters specify the size of the input image and the number of channels. The input training images were first being resized to width times height. The default value is 416 x 416. The result might be improved if the larger image size were used, but it would take longer time to train. Channel parameter is set to be 3, where 3 indicates that the training will process 3-channel RGB input images only.

Next, momentum and decay parameters. These parameters control on how the weight is updated for every batch. Since neural network behavior is updated based on a small batch of images and not the entire dataset, hence, momentum parameter was used to penalize large weight changes between iterations, while decay parameter will control the penalty term.

The learning rate parameter controls the level of aggressiveness of the learning process based on the current

batch. Typically, the value lies between 0.01 and 0.0001. At the beginning of the training process, the network starts with no information, the learning rate need to go higher, while the weights need to adapt and become less aggressive. Throughout the learning process, the learning rate will be decreased over time. In configuration file, the decrease in learning rate can be carried out by specifying the learning rate dropping policy, namely steps. Referring to the configuration setting, the learning rate will start from 0.001, and remains constant for 5400 iterations, before it is being multiply by a new scale to get a new learning rate.

## 2.3 Model Evaluation

The standard accuracy metric for object detection is mean average precision, for short MAP. MAP is the mean value of the average precision, AP, of every class. Generally, MAP value will increase at the beginning of training processes and slowly decreases due to overfitting of the deep learning model at a certain number of iterations. In this case, the size of the dataset is a bit smaller compared to other accessible deep learning datasets which seem to affect the training process.

To get MAP value, intersection over union, for short IoU, must be computed for every prediction of each image. If a predicted bounding box overlaps with a ground truth bounding box, the area of intersection and area of the union can be computed. The IoU can be computed by (1), and detail illustration is shown in Figure 3.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (1)$$

Depends on the value of IoU, the prediction is either a True Positive (TP), a False Positive (FP), a True Negative (TN) or a False Negative (FN). With IoU threshold value 0.5, a true positive is when  $IoU > 0.5$  and a false positive is when  $IoU < 0.5$  or the bounding box is a duplicated one. A false negative is when  $IoU \geq 0.5$ , but the prediction is the wrong class.

**Precision.** Precision measures how accurate the predictions, and it can be computed using (2):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

**Recall.** Recall measures how good all the positive results, and it can be computed using (3):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

## 2.4 Model Testing

Testing can be done using three types of input, namely image file, webcam feed and video file. Figure 4 shows the tested

data on classification of vehicle from Class 1 using image file. As we can see to the text appears in the detected green box, it shows the information of type of Class versus MAP. For example, Class 1:0.97 means the detected vehicle is from Class 1 and its confidence score is 0.97 which is equal to 97%.

Figure 5 shows the tested data on classification of vehicle from Class 2 using image file, Figure 6 shows the tested data on classification of vehicle from Class 3, Figure 7 shows the tested data on classification of vehicle from Class 4 and Figure 8 shows the tested data on classification of vehicle from Class 5.



Figure 5: Classification of Vehicle Class 2



Figure 6: Classification of Vehicle Class 3



Figure 7: Classification of Vehicle Class 4

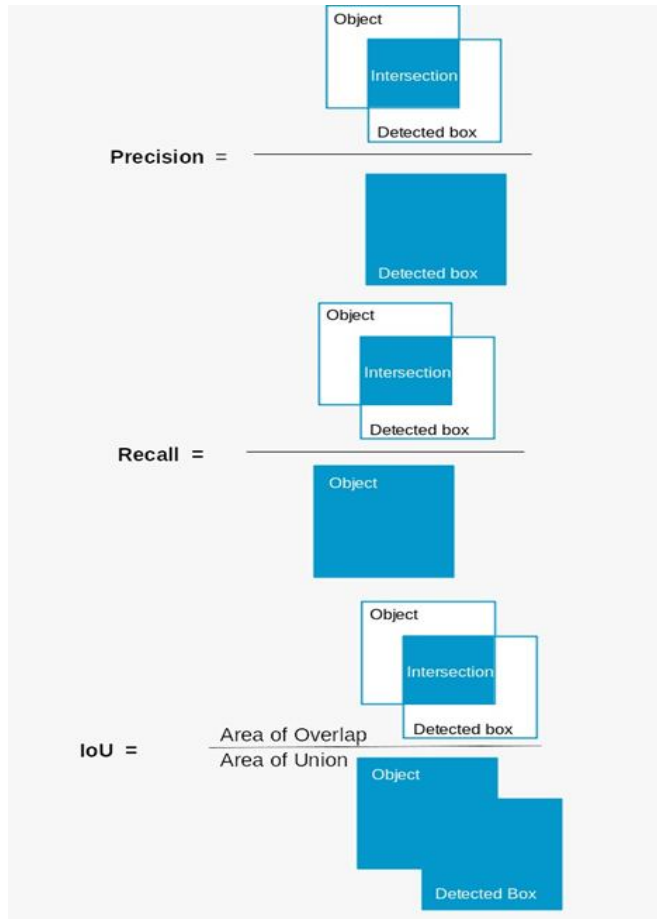


Figure 3: Precision, Recall and IoU for detection



Figure 4: Classification of Vehicle Class 1



Figure 8: Classification of Class 5

### 3. RESULT AND DISCUSSION

As mentioned earlier, the standard accuracy metric for object detection is MAP, and MAP is the mean value of the average precision, AP, of every class. Generally, MAP value increases at the beginning of the training process, and slowly decreases due to overfitting of the deep learning model at certain iterations. In this case, the size of the dataset is a bit smaller compared to other accessible deep learning datasets which seem to affect the training process. Figure 9 show that at the training process, learning rate reading is quite high, and it is drastically decreases in the first 600 iterations, and become less aggressive throughout the process. While the MAP reading is getting higher and more accurate throughout the process.

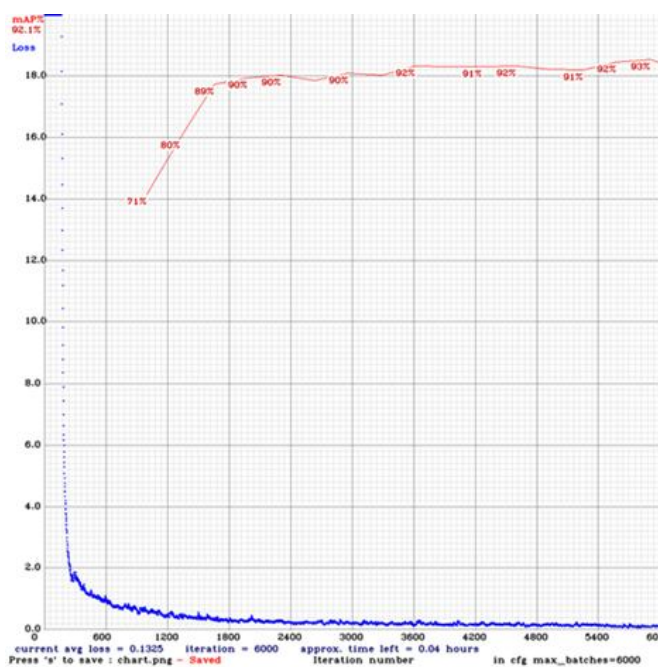


Figure 9: Learning Rate versus MAP

Table 1 shows the performance results after finish training the model. The average precision for each class is above 80%, and the calculated MAP of the model with a 0.5 IoU threshold value is 0.921222 or 92.12%.

Table 1: Average MAP Reading for each Class

Class	Average Precision AP (%)	Mean Average Precision, MAP @ 0.50	Precision	Recall
1	85.69	0.921222 (92.12%)	0.91	0.91
2	95.82			
3	97.60			
4	92.14			
5	89.36			

### 4. CONCLUSION

This paper has introduced a method for automatically detect and classify the type of vehicle purposely for Electronic Toll Collection (ETC) system based on YOLOv3. The result shows that the propose model manage to identify the type of vehicle in more than 92.12% accurate.

### 5. FUTURE RECOMMENDATION

This study has some limitations on a few things that can be done before and after training for the model's result improvement. Most of the steps are related to the dataset. In pre-training procedure, in order to increase the precision, try to increase a network resolution in the configuration file, that is height equal to 608 and width equal to 608, or it can be any value in multiple of 32. Other than that, dataset also requires more actual data of vehicles to increase the accuracy of vehicle classification as well as to increase speed of detection. The data should be in more variants such as scale of images, brightness of the images as well as the angle of the images so that the model has more informative input.

### ACKNOWLEDGEMENT

The authors would like to acknowledge the Center for Advanced Computing Technology (C-ACT), Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka for providing research facilities.

### REFERENCES

1. Z. Yang and L. S. Pun-Cheng. **Vehicle detection in intelligent transportation systems and its applications under varying environments: A review**, *Image and Vision Computing*, Vol. 69, pp. 143-154, 2018. <https://doi.org/10.1016/j.imavis.2017.09.008>
2. S. Gupte, O. Masoud, R. F. Martin, and N. P. Papanikolopoulos. **Detection and classification of vehicles**, *IEEE Transactions on intelligent transportation systems*, Vol. 3, No. 1, pp. 37-47, 2002.
3. Y. Zhang, C. Hu, and X. Lu. **Improved YOLOv3 object classification in Intelligent Transportation System**, arXiv preprint arXiv:2004.03948, 2020.
4. M. N. Roecker, Y. M. Costa, A. S. Britto, L. E. Oliveira, and D. Bertolini. **Vehicle detection and classification in traffic images using ConvNets with constrained resources**, in *2019 International Conference on Systems, Signals and Image Processing*, June 2019, pp. 83-88, IEEE. <https://doi.org/10.1109/IWSSIP.2019.8787310>
5. S. Chen and W. Lin. **Embedded system real-time vehicle detection based on improved YOLO network**, in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control*

- Conference, October 2019, pp. 1400-1403, IEEE.
6. D. Zhao, Y. Chen, and L. Lv. **Deep reinforcement learning with visual attention for vehicle classification**, *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 9, No. 4, pp. 356-367, 2016.
  7. Y. Zhou, L. Liu, L. Shao, and M. Mellor. **DAVE: A unified framework for fast vehicle detection and annotation**, in *European Conference on Computer Vision*, October 2016, pp. 278-293, Springer, Cham. [https://doi.org/10.1007/978-3-319-46475-6\\_18](https://doi.org/10.1007/978-3-319-46475-6_18)
  8. W. Ji, L. Tang, D. Li, W. Yang, and Q. Liao. **Video-based construction vehicles detection and its application in intelligent monitoring system**, *CAAI Transactions on Intelligence Technology*, Vol. 1, No. 2, pp. 162-172, 2016.
  9. S. Luo, C. Xu, and H. Li. **An application of object detection based on YOLOv3 in traffic**, in *Proceedings of the 2019 International Conference on Image, Video and Signal Processing*, February 2019, pp. 68-72.
  10. Z. Yi, S. Yongliang, and Z. Jun. **An improved tiny-yolov3 pedestrian detection algorithm**, *Optik*, Vol. 183, pp. 17-23, 2019.
  11. V. Nayak, S. P. Holla, K. M. AkshayaKumar, and C. Gururaj. **Automatic Number Plate Recognition**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 3, pp. 3783-3787, 2020. <https://doi.org/10.30534/ijatcse/2020/195932020>
  12. M. B. U. Chowdhury, P. Dhar, and S. Guha. **Detection and Recognition of Bangladeshi License Plate**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 3, pp. 3734-3740, 2020. <https://doi.org/10.30534/ijatcse/2020/187932020>
  13. L. Zhao, and S. Li. **Object detection algorithm based on improved YOLOv3**, *Electronics*, Vol. 9, No. 3, pp. 537, 2020.
  14. H. Yang, P. Liu, Y. Hu, and J. Fu. **Research on underwater object recognition based on YOLOv3**, *Microsystem Technologies*, pp. 1-8, 2020.
  15. J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng. **Fabric defect detection using the improved YOLOv3 model**, *Journal of Engineered Fibers and Fabrics*, Vol. 15, 1558925020908268, 2020.
  16. Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang. **Apple detection during different growth stages in orchards using the improved YOLO-V3 model**. *Computers and Electronics in Agriculture*, Vol. 157, pp. 417-426, 2019. <https://doi.org/10.1016/j.compag.2019.01.012>
  17. J. Redmon and A. Farhadi. (2018). **Yolov3: An incremental improvement**, arXiv preprint arXiv:1804.02767, 2018.
  18. J. Redmon. **Darknet: Open source Neural Networks in C**. Available at <http://pjreddie.com/darknet/>, 2013-2016.