Vision-Based Vehicle Classification Using Deep Learning Model

Ahsiah Ismail[1*], Amelia Ritahani Ismail[2], Muhammad Afiq Mohd Ara[3], Asmarani Ahmad Puzi[4], Suryanti Awang[5]
Department of Computer Science-Kulliyyah of Information and Communication Technology, International Islamic University Malaysia (IIUM), 53100 Kuala Lumpur, Malaysia[1, 2, 3, 4]
Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia[5]

***Abstract*—Vehicle classification offers intelligent solutions for road traffic monitoring by enabling future prediction planning and decision making. Predictive analytics can be used to predict traffic congestion based on the types of vehicles on the road. In this research, the reliability of deep learning based models for vision-based vehicle classification is investigated. Four models of You Only Look Once (YOLO) are investigated, namely YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n. These models were trained and evaluated on a vehicle dataset comprising five vehicle classes, which are Ambulance, Bus, Car, Motorcycle, and Truck, with a total number of 1103 images. From the experiment conducted, YOLOv10n achieved the highest performance measure of mAP@0.5 with 0.859 across all vehicle classes, including per-class evaluation, demonstrating superior detection compared to the other models. Finally, the results indicate that the YOLOv10n model can be used in vision-based vehicle classification.**

Keywords*—YOLO; vehicle classification; deep learning; traffic monitoring

I. Introduction

Vision-based vehicle classification offers an intelligent solution for transportation systems, thus contributing to road traffic monitoring for the development of smart cities [1],[2],[3]. With the growing number of vehicles on the road, efficient and accurate vehicle classification systems are essential for managing urban mobility, enhancing public safety, and reducing traffic congestion [4],[5],[6]. Vehicle detection technologies involve the use of cameras and vision sensors to capture road footage, which is then analyzed to detect, classify, and track the movement of vehicles [7],[6]. Using these technologies may improve traffic control for future planning and decision making based on the current vehicle data on the road [8], [9]. Further data analytics can also be performed based on the current vehicle data on the road for traffic monitoring [10]. The analytics process identifies the patterns and learns for future planning and prediction, which helps to improve road traffic monitoring.

For this reason, a vision-based vehicle classification is proposed. The vision-based vehicle classifications are designed based on a deep learning method. The model of the deep learning methods are evaluated to determine the best vision-based classification for detecting types of vehicle. The deep learning method is chosen in this research due to its ability to detect objects in more complex environments under varying environmental conditions. In vehicle classification, the detection of vehicles especially on this condition is important. Failure to detect these may reduce the classification accuracy.

This research focuses on the detection of suitable methods that are able to increase the aforementioned classification accuracy in small sample vehicle dataset images. A small sample dataset of vehicles (1103 images) was utilized in this research as a proof-of-concept to classify vehicle images. To suite the problem stated previously, we investigate the deep learning based method, particularly Convolutional Neural Networks (CNNs). The CNN is chosen as it is able to provide a more robust alternative by learning discriminative features directly from images, enabling more accurate detection and classification. The CNN model has emerged as a powerful alternative due to its ability to learn spatial hierarchies of features directly from image data [11]. Among the CNN deep learning based object detection frameworks, the model You Only Look Once (YOLO) is the most suitable method for real-time vehicle detection tasks as it is able to balance between speed and accuracy, making it highly suitable [12]. This research investigates deep learning YOLO models, namely, YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n, for vehicle classification. Each of these models are trained using pretrained weights to reduce training time while maintaining accuracy during the transfer learning. These models are tested using a vehicle dataset consisting of five types of vehicles, which are Ambulance, Bus, Car, Motorcycle, and Truck.

The rest of this study is organized as follows: Section II presents the related works. Section III describes the materials and methods used. The results are presented in Section IV, followed by the discussion in Section V. Finally, the conclusion and future work are presented in Section VI.

II. Related Works

Vehicle detection plays a vital role in intelligent transportation systems to enable intelligent solution for road traffic monitoring [13]. The computer vision method is used to detect the vehicle on a road traffic. In early research, the commonly used methods for vehicle detection are background subtraction, edge detection, and Haar Cascades [5,6,14]. Although these methods are computationally efficient, they were less effective in detecting the images under varying environmental conditions, such as changes in illumination, shadows, and occlusion, thus limiting their effectiveness in real-world scenarios [15].

In more recent object recognition method, the deep learning method have gain attention due to its ability to detect object in most classification task. Compared to other methods such as edge detection and background subtraction method, the CNN

method is one of the most reliable deep learning based method which provides more accurate detection and classification. The other methods often shown to be less effective in detecting the object under varying environmental conditions and in real-time, dynamic environments due to their sensitivity to lighting conditions and noise [11,16]. In the CNN deep learning based object detection frameworks, the YOLO models are widely used for object detection due to their ability to balance between speed and accuracy. It is also suitable for real-time detection [12]. Although the other models of the CNN method, such as R-CNN, Fast R-CNN, and Faster R-CNN, demonstrated significant improvements in accuracy detection by learning hierarchical features from the data, these models rely on region proposals that led to slower inference speeds, making them less suitable for real-time detection tasks [8,9,17]. On the other hand, the YOLO models enable fast and accurate detection in a single pass over the image. YOLO is a single-stage object detection framework that performs both object localization and classification in one forward pass through the network [18]. This design makes the YOLO model efficient for real-time applications such as traffic surveillance and vehicle monitoring. Unlike two-stage detectors like R-CNN, which separate the tasks of region proposal and classification, YOLO treats detection as a single regression problem, mapping image pixels directly to bounding box coordinates and class probabilities. This architecture offers faster inference times and smaller model sizes, making YOLO ideal for the deployment of edge devices and in time-sensitive environments. Different YOLO variants offer different unique strengths and limitations. The challenge is in selecting the most suitable model that is able to balance accuracy with computational time performance while maintaining a high classification.

The YOLO variants, such as YOLOv3 and YOLOv4 models, strike a balance between speed and precision [19]. While YOLOv5 series models improve the limitation by architectural optimizations and improved training strategies [11,12]. Among the YOLOv5 series, YOLOv5s and YOLOv5x provide trade-offs between computational efficiency and accuracy, making them ideal for both edge and cloud-based deployments. The YOLOv5s model is a lightweight YOLO architecture and one of the smallest models in the YOLOv5 variant. YOLOv5s model offers fast training and inference with minimal resource consumption. This makes the model an excellent starting point for experimentation and prototyping, especially when assessing trade-offs between speed and accuracy. On the other hand, YOLOv5x is designed to maximize detection accuracy and mean Average Precision (mAP). It represents the upper bound of the YOLOv5 variants, with deeper architecture and more parameters. However, the YOLOv5x model requires a higher computational cost and longer inference time. Recent variants such as YOLOv10 and YOLOv12 continue to improve the YOLO performance by optimizing depth, parameter count, and inference speed, especially to be implemented in resource-constrained environments [20]. YOLOv10 model is a lightweight model but is able to offer high performance [20]. The model is optimized for low-latency, high-efficiency deployment, particularly on edge devices. It offers better performance-per-FLOP than earlier nano-scale models such as YOLOv5s. The YOLOv10 is built to deliver real-time performance without reducing its accuracy performance. Compared with the other earlier YOLO model such as YOLOv5s, YOLOv10 model improve the model architecture and enhance the efficiency and accuracy even in a compact model. The latest generation nano-scale model in YOLO variant is the YOLOv12n model. This model been designed as a compact real-time model that able to offer high efficiency and competitive accuracy. The YOLOv12n model incorporating attention-centric modules for smarter and more selective feature extraction [21].

Despite all these improvements, object recognition remains challenging, particularly in scenarios involving large and complex object images. The detection accuracy often drops significantly for large objects under poor lighting conditions, especially in large object images taken during night conditions, due to increased noise and reduced contrast in large object images [22].

From the above review, the YOLO model is seen to be the most suitable model for vision-based vehicle classification due to its capability to perform both object localization and classification in one forward pass through the network. This design makes YOLO model suitable and efficient for real-time applications such as traffic surveillance and vehicle monitoring. Therefore, in this research, the YOLO variants are investigated to detect and classify the vehicle, particularly for large vehicle images.

III. Materials And Methods

Generally, the proposed work can be divided into two phases: detection and classification. In this research, four YOLO models are evaluated, which are YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n. The overall scheme for the vehicle classification models evaluated is shown in Fig. 1. Each of these models is selected due to its popularity and stability for the object classification task. The YOLOv5s is selected as it offers fast training and inference with minimal resource consumption. The YOLOv5s is the smallest model in the YOLOv5 variant. This makes an excellent starting point for experimentation and prototyping, especially when assessing trade-offs between speed and accuracy. While the YOLOv5x model requires higher computational, and longer inference time. The YOLOv5x model represents the upper bound with deeper architecture and more parameters. Although it requires higher computational resources, YOLOv5x is designed to maximize detection accuracy and mean Average Precision (mAP). On the other hand, YOLOv10n and YOLOv12n models are chosen due to these models are the recent iterations of YOLO variants that optimize depth, parameter count, and inference speed in the YOLO architecture.

The model architecture of vehicle classification using YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n are shown in Fig. 2(a), Fig. 2(b), Fig. 2(c) and Fig. 2(d). As shown in Fig. 2, each of the models consist three main parts which are backbone network, neck network and detection head. The backbone network is the first step in YOLO model architecture, where the features extracted from the vehicle input images. Then, the feature fusion in the neck part, where feature maps from different scales of the backbone network are fused to a neck network to process the detection. The head part completes the final prediction which includes the bounding boxes and the

associated class labels to obtain the detection result. Three different scales of detection heads are used to detect small, medium and large vehicles.

To demonstrate the reliability of the selected model for vision-based vehicle classification, a series of comprehensive experiments is conducted. All the YOLO models are trained using Google Colab with GPU acceleration, and the dataset tested is annotated using CVAT.ai and hosted on Roboflow for streamlined training and validation. The training environment includes dependencies of torch, numpy, opencv, and the official YOLO repositories for each YOLO version.

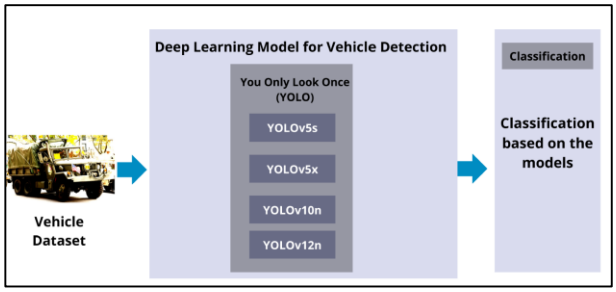
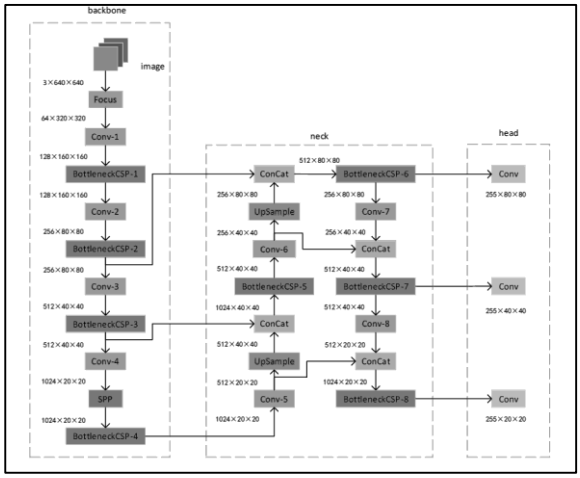
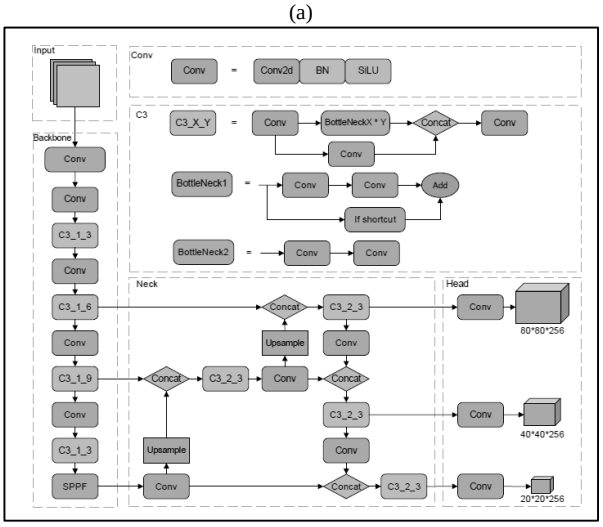


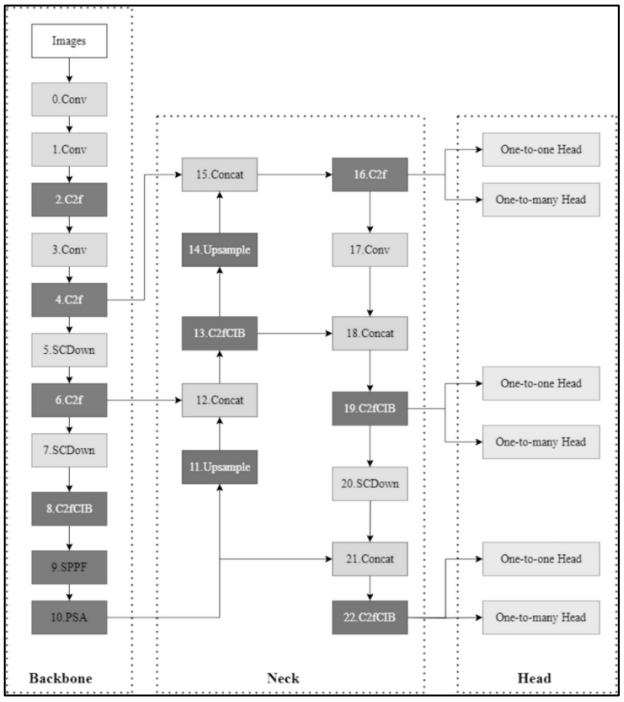
Fig. 1. Overall scheme of the proposed vision-based vehicle classification with four different YOLO models (YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n).



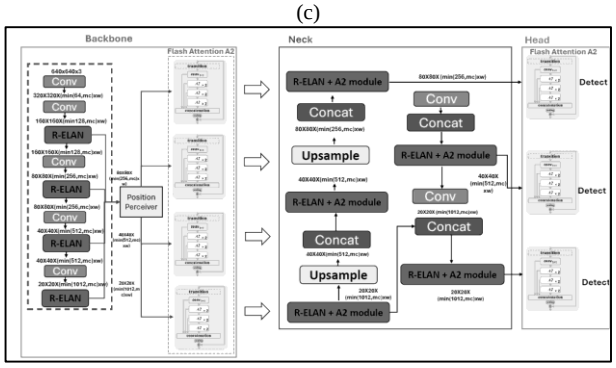
(a)



(b)



(c)



(d)

Fig. 2. YOLO model architecture for: (a) YOLOv5s [23], (b) YOLOv5x [24] (c) YOLOv10n [25] and (d) YOLOv12n [26].

The dataset used in this research consists of 1103 annotated images categorized into five vehicle types which are Ambulance (161 images), Bus (84), Car (321), Motorcycle (247), and Truck (290). These images were manually curated to ensure balanced class representation and relevance to real-world conditions. Each image was manually labelled with bounding boxes and associated class labels using CVAT.ai (Computer Vision Annotation Tool), an open-source tool suitable for precise object detection annotation. Bounding boxes were drawn around each vehicle, and class names were assigned to each types of vehicle accordingly. The properties of the dataset used are shown in Table I.

To train and evaluate the models effectively, the dataset was divided into training, validation and test set with a ratio of 70%, 20% and 10% as shown in Table II.

This dataset is split to ensure all classes are proportionally represented in each set, for effective training and accurate performance evaluation. There are two processes applied on all the images used in the experiment, which are auto-orientation

and resize. The auto-orientation is applied to ensure that all images are correctly aligned based on EXIF metadata, to avoid misalignment issues during the training phase. All the images are also resized to fit into the YOLO models, which requires fixed size of input. All images were resized to 640×640 pixels to fit into the YOLO model input layer while maintaining acceptable aspect ratios for object preservation. These pre-processed datasets allowed uniform input across all models to ensure a fair comparison. To ensure a fair comparison in the model configuration, the same training parameters settings used in all YOLO models tested as listed in Table III.

TABLE I. DATASET DISTRIBUTION

Classes	Number of Images
Ambulance	161
Bus	84
Car	321
Motorcycle	247
Truck	290
Total	1103

TABLE II. DATASET SPLIT

Dataset Split	Training Set (70%)	Validation Set (20%)	Test Set (10%)
Images	772	221	110

TABLE III. PARAMETER SETTING FOR YOLO MODELS TRAINING

Model / Parameter	YOLOv5s	YOLOv5x	YOLOv10n (Selected Model)	YOLOv 12n
Image Size	640×640 pixels	640×640 pixels	640×640 pixels	640×64 0 pixels
Epoch Size	50	50	50	50
Batch Size	15	15	15	15
Pretrained Weights	Enable	Enabled	Enabled	Enabled

In this research, Roboflow was used to provide seamless dataset integration into the YOLO training scripts. Each model was saved as best.pt, representing the model weights with the highest validation performance during training.

IV. RESULTS

In all the experiments, the performance of each YOLO models on the vehicle classification are measured in terms of Precision, Recall, Mean Average Precision (mAP), and Training Time. Precision measures the proportion of true positive detections among all positive predictions and evaluates how many vehicles were correctly identified. On the other hand, recall evaluates the correctly identify all actual vehicle instances in the dataset. It is the proportion of true positive detections out of all actual positive instances. While the Mean Average Precision (mAP) is also considered in this research since most of the work related with object detection from the literature use mAP to evaluate performance measure in their work [27,28]. In all the experiment conducted, the Training Time had also been recorded to measure the suitability of each of the model for the deployment under limited time or processing power environments. Models with a less Training Time and able to maintain high detection accuracy are more practical to be implemented in real-time or resource-sensitive applications. The classification performance of all models are defined as follows:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (3)$$

$$AP_k = the\ AP\ of\ class\ k, n = the\ number\ of\ class$$

The models are trained and evaluated on a vehicle dataset image that consists of five vehicle classes. The performance of each model, namely YOLOv5s**,** YOLOv5x, YOLOv10n, and YOLOv12n, is presented in Table IV.

TABLE IV. COMPARATIVE RESULTS (PRECISION, RECALL, TIME, MAP@0.5) FOR YOLOV5S, YOLOV5X, YOLOV10N AND YOLOV12N

Models	Precision	Recall	Training Time (hrs)	mAP@0.5
YOLOv5s	0.823	0.882	0.263	0.763
YOLOv5x	0.921	0.909	1.222	0.855
YOLOv10n (Selected Model)	0.912	0.886	0.296	0.859
YOLOv12n	0.925	0.883	0.350	0.832

As shown in Table IV, the YOLOv10n model demonstrates superior performance among all the models tested with 0.859 of mAP@0.5. This is followed by the YOLOv5x model with 0.855 of mAP@0.5 and the YOLOv12n with 0.832 of mAP@0.5. The lowest mAP@0.5 is observed in the YOLOv5s model with 0.763 mAP@0.5. YOLOv10n model has shown to be the most optimum model with highest maAp@0.5 of 0.859 obtain in a less Training Time requires to train the models. The lowest Training Time among all the models is the YOLOv5s model, with 0.263hours. This is followed by the YOLOv10n model with 0.296 hours of Training Time. Only a minimal difference is observed between the YOLOv10n model with YOLOv5s model which less than 0.033 hours Training Time between the YOLOv10n model. Despite the YOLOv10n model being smaller than the YOLOv5 model variant, the YOLOv10n model is able to achieve the highest maAp@0.5 with only 0.263 hours of Training Time taken. The YOLOv10n model is superior compared to the other models in terms of both maAp@0.5 and Training Time. The results show that the YOLOv10n model is suitable for use in real-time applications and edge deployments. On the other hand, YOLOv12n outperformed the mAP@0.5 of YOLOv5s models while maintaining a much smaller size. Although YOLOv5s is the fastest model, it shows the lowest mAP@0.5 with 0.763. The YOLOv5s model is able to obtain faster training time due to its lightweight baseline model, making it able to train the model faster. However, the lowest mAP@0.5 is observed in the model.

To further investigate the performance of each models, the models are further evaluated on the specific class performance for each of the five vehicle classes which are Ambulance, Bus, Car, Motorcycle and Truck. Their results of precision, recall and map@0.5 are presented in Table V.

TABLE V. COMPARATIVE RESULTS PER-CLASS PERFORMANCE (PRECISION, RECALL AND MAP@0.5) FOR YOLOV5S, YOLOV5X, YOLOV10N AND YOLOV12N

Class	Models	Precision	Recall	mAP@0.5
Ambulance	YOLOv5s	0.771	0.882	0.916
	YOLOv5x	0.908	0.971	0.987
	YOLOv10n (Selected Model)	1.00	0.932	0.979
	YOLOv12n	0.967	0.864	0.933
Bus	YOLOv5s	0.835	0.954	0.929
	YOLOv5x	0.928	0.954	0.984
	YOLOv10n (Selected Model)	0.890	0.969	0.979
	YOLOv12n	0.914	0.985	0.986
Car	YOLOv5s	0.765	0.938	0.910
	YOLOv5x	0.859	0.837	0.927
	YOLOv10n (Selected Model)	0.902	0.822	0.928
	YOLOv12n	0.922	0.849	0.920
Motorcycle	YOLOv5s	0.904	0.846	0.947
	YOLOv5x	1.00	0.911	0.984
	YOLOv10n (Selected Model)	0.937	0.846	0.966
	YOLOv12n	0.958	0.873	0.971
Truck	YOLOv5s	0.840	0.789	0.897
	YOLOv5x	0.909	0.873	0.929
	YOLOv10n (Selected Model)	0.829	0.860	0.952
	YOLOv12n	0.863	0.842	0.917

From Table V, the YOLOv5x model give the best performance in Ambulance class with 0.987 of mAP@0.5. This is followed by YOLOv10n model with 0.979 of mAP@0.5, YOLOv12n model with 0.933 of mAP@0.5. The lowest mAP@0.5 for Ambulance class is observed in YOLOv5s model with 0.916. While in Bus class, YOLOv12n model is outperformed the other models with 0.986 of mAP@0.5. This is followed by YOLOv5x model with 0.984 of mAP@0.5, YOLOv10n model with 0.979 of mAP@0.5. The lowest mAP@0.5 for Bus class is observed in YOLOv5s model with 0.929 of mAP@0.5. For Car class, YOLOv10n models is outperformed the other models with 0.928 of mAP@0.5. This is followed by YOLOv5x model with 0.927 of mAP@0.5, YOLOv12n model with 0.920 of mAP@0.5. The lowest mAP@0.5 for Car class is observed in YOLOv5s model with 0.910 of mAP@0.5. On the other hand, for Motorcycle class, YOLOv5x models is outperformed the other models with 0.984 of mAP@0.5. This is followed by YOLOv12n model with 0.971 of mAP@0.5, YOLOv10n model with 0.966 of mAP@0.5. The lowest mAP@0.5 for Motorcycle class is also observed in YOLOv5s model with 0.947 of mAP@0.5. Lastly, for the Truck class, the highest performance of mAP@0.5 is YOLOv10n model. This is followed by YOLOv5x with 0.929 of mAP@0.5, YOLOv12n model with 0.917 of mAP@0.5. The lowest mAP@0.5 for Truck class is also observed in YOLOv5s model with 0.897 of mAP@0.5. Among all the models tested across all five vehicle classes, YOLOv10n model is seen to be the most optimum models as it is able to achieve the highest mAP@0.5 for the two classes which are Car and Truck classes. While the lowest model which can be observed for all the five tested vehicle class is the YOLOv5s model.

V. DISCUSSION

Overall, YOLOv10n model demonstrate superior performance of mAP@0.5 among all the models for all evaluation including the per-class performance evaluation from the experiments conducted. While the lowest mAP@0.5 results are observed in YOLOv5s for all performance evaluations. The highest performance can be seen on YOLOv10n model due to its improvement in the model architecture that enhance both efficiency and accuracy. The model is able to balance trade-off between speed and accuracy while retaining its compact versions. Although the YOLOv10 model is a lightweight, the model able to obtain high-performance due to the model architecture that is optimized for low-latency, better performance-per-FLOP than earlier nano-scale models like a YOLOv5s. The YOLOv10 model is built to deliver real-time performance without reducing its accuracy performance [20]. From the experiment conducted, the lowest performance is observed in YOLOv5s models due to its lightweight, simplicity and it is the smallest YOLO architecture, thus reducing its recognition performance. The YOLOv5s and YOLOv5x model is shown to be less effective especially on detecting the large vehicle images such as Truck vehicle. YOLOv5x model obtain low confidence score detection on truck that is clearly present in the image. On the other hand, the YOLOv10n model able to obtain high confidence score detection and able to detect most of the truck. This is followed by the YOLOv12n model. The comparison models for the confidence score detection on Truck is shown in Fig. 3.

Fig. 3. Comparison models for confidence score detection on Truck: (a) YOLOv5s, (b) YOLOv5x, (c) YOLOv10n (d) YOLOv12n.

Despite YOLOv10n able to detect most of the vehicles including large vehicle and various conditions, the low confidence score detection still can be observed on some of the dark colour truck, occluded motorcycle images and bus are shown in Fig. 4.

Fig. 4. Examples of the low confidence score detection on YOLOv10n.

VI. Conclusion and Future Work

A vision-based vehicle classification is proposed as it is able to offer intelligent solution for transportation systems, thus contribute to road traffic monitoring for the development of smart cities. Four models of YOLO are investigated, namely YOLOv5s, YOLOv5x, YOLOv10n, and YOLOv12n are evaluated on a multi-class vehicle dataset. From the experiment conducted, the YOLOv10 able to obtain the highest detection performance of mAP@0.5 with 0.859 across all vehicle classes compared to the other model. Although, YOLOv10n is able to achieve high performance, low confidence can be observed on the images as shown in Fig. 4. It can be seen that the models are less effective in detecting the dark colour truck, occluded motorcycle and bus images. The lighting variations, especially in the detection of large, visually complex vehicles like trucks or bus may affected the detection and recognition. Thus, reducing the performance of the models. In the future, we will concentrate on improving these drawbacks by optimized the anchor box and modified the loss function to improve the detection especially on a large vehicle such as bus and truck. Increasing the diversity of the vehicle images dataset will also be considered by using data augmentation techniques for more advanced deep learning models to optimize the performance.

Acknowledgment

This research was funded by UMP-IIUM Sustainable Research Collaboration 2022 grant (IUMP-SRCG22-015-0015).

References

[1] M. Balfaqih, S. A. Alharbi, M. Alzain, F. Alqurashi, and S. Almilad, “An accident detection and classification system using internet of things and machine learning towards smart city,” Sustain., vol. 14, no. 1, pp. 1–13, 2022.

[2] S. Awang, N. M. A. N. Azmi, and M. A. Rahman, “Vehicle Type Classification Using an Enhanced Sparse-Filtered Convolutional Neural Network with Layer-Skipping Strategy,” IEEE Access, vol. 8, pp. 14265–14277, 2020.

[3] B. Kidmose, “A review of smart vehicles in smart cities: Dangers, impacts, and the threat landscape,” Veh. Commun., p. 100871, 2024.

[4] S. S. Harsha and K. R. Anne, “Gaussian Mixture Model and Deep Neural Network based Vehicle Detection and Classification,” vol. 7, no. 9, pp. 17–25, 2016.

[5] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, “Sensor technologies for intelligent transportation systems,” Sensors, vol. 18, no. 4, p. 1212, 2018.

[6] P. Premaratne, I. J. Kadhim, R. Blacklidge, and M. Lee, “Comprehensive review on vehicle detection, classification and counting on highways,” Neurocomputing, p. 126627, 2023.

[7] B. Neupane, T. Horanont, and J. Aryal, “Real-time vehicle classification and tracking using a transfer learning-improved deep learning network,” Sensors, vol. 22, no. 10, p. 3813, 2022.

[8] A. B. Ahmad and T. Tsuji, “Traffic monitoring system based on deep learning and seismometer data,” Appl. Sci., vol. 11, no. 10, p. 4590, 2021.

[9] H. Zhang, M. Liptrott, N. Bessis, and J. Cheng, “Real-Time Traffic Analysis using Deep Learning Techniques and UAV based Video,” 2019.

[10] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, “Deep learning-based vehicle behavior prediction for autonomous driving applications: A review,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 1, pp. 33–47, 2020.

[11] N. O’Mahony et al., “Deep learning vs. traditional computer vision,” in Advances in computer vision: proceedings of the 2019 computer vision conference (CVC), volume 1 1, Springer, 2020, pp. 128–144.

[12] R. Ayachi, Y. Said, M. Afif, A. Alshammari, M. Hleili, and A. Ben Abdelali, “Assessing YOLO models for real-time object detection in urban environments for advanced driver-assistance systems (ADAS),” Alexandria Eng. J., vol. 123, pp. 530–549, 2025.

[13] A. Shabbir, A. N. Cheema, I. Ullah, I. M. Almanjahie, and F. Alshahrani, “Smart city traffic management: Acoustic-based vehicle detection using stacking-based ensemble deep learning approach,” IEEE access, vol. 12, pp. 35947–35956, 2024.

[14] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149), IEEE, 1999, pp. 246–252.

[15] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” Video-based Surveill. Syst. Comput. Vis. Distrib. Process., pp. 135–144, 2002.

[16] A. Gholamhosseinian and J. Seitz, “Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective,” IEEE Open J. Intell. Transp. Syst., vol. 2, pp. 173–194, Jan. 2021, doi: 10.1109/OJITS.2021.3096756.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2016.

[18] N. Wu, “Application and evaluation of deep learning based image recognition techniques in agriculture,” Appl. Comput. Eng., vol. 48, no. 1, pp. 141–147, 2024.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv Prepr. arXiv2004.10934, 2020.

[20] A. Wang et al., “YOLOv10: Real-Time End-to-End Object Detection,” May 2024, [Online]. Available: http://arxiv.org/abs/2405.14458

[21] R. Sapkota et al., “YOLOv12 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series,” 2025. [Online]. Available: https://www.researchgate.net/publication/381650388

[22] Y. Zhang, B. Hu, X. Yuan, and Y. Li, “Road object recognition method based on improved YOLOv3,” Acad. J. Comput. Inf. Sci, vol. 5, pp. 1–9, 2022.

[23] A. Mechanism, “An Improved YOLOv5s Algorithm for Object Detection with an Attention Mechanism,” 2022.

[24] F. Tang and Q. Zhang, “Real-Time Detection of Drill Pipe Joints Using Improved YOLOv5x Model Applied to Drilling Operation Images,” 2024.

[25] P. Yan, H. Sun, Y. Zhao, P. Wang, and Z. Wu, “Multispectral imaging and enhanced YOLOv10n for efficient coal gangue detection in complex mining environments,” J. Real-Time Image Process., vol. 22, no. 3, pp. 1–14, 2025.

[26] R. Sapkota and M. Flores-calero, “YOLOv12 to Its Genesis : A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series,” no. February 2025, 2024.

[27] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," Sustain., vol. 14, no. 19, Oct. 2022.

[28] D. Li, E. Wang, Z. Li, Y. Yin, L. Zhang, and C. Zhao, "STE-YOLO: A Surface Defect Detection Algorithm for Steel Strips," Electron., vol. 14, no. 1, pp. 1–21, 2025.