Investigation of Convolutional Neural Network Model for Vehicle Classification in Smart City

Ahsiah Ismail[1], Amelia Ritahani Ismail[2], Nur Azri Shaharuddin[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*—Smart city optimize efficiency by integrating advanced digital technologies, real-time data analytics, and intelligent automation. With the evolution of big data, smart cities enhance infrastructure and provide intelligent solutions for transportation with the integration of high-level adaptability of computer technologies including artificial intelligence (AI). The optimization can be achieved through predictive analytics in providing intelligent solutions for transportation. However, this requires reliable and accurate informative data as input for predictive analytics. Therefore, in this paper, five models of Convolutional Neural Network (CNN) deep learning method are investigated to determine the most accurate model for classification; namely Single Shot Detector (SSD) Resnet50, SSD Resnet152, SSD MobileNet, You Only Look Once (YOLO) YOLOv5 and YOLOv8. A total of 1324 vehicle images are collected to test these CNN models. The images consist of five different categories of vehicles, which are ambulance, car, motorcycle, bus and truck. The performances of all the models are compared. From the evaluation, the model YOLOv8 attained 0.956 of precision, 0.968 of recall and 0.968 of F1 score and outperformed the others. In terms of computational time, YOLOv5 is the fastest. However, a minimal computational time difference is observed between the YOLOv5 and YOLOv8, which were separated by only 20 minutes.**

***Keywords*—Vehicle classification; Convolutional Neural Network; SSD; YOLO; MobileNets**

I. Introduction

"Smart City" is a city that utilises technologies, data and digital solutions to improve the efficiency by equipping systems and services with additional cameras and sensors to collect data [1]. Then it integrates with various services such as transportation networks, public transit, utilities, and others to improve its operation and the quality of life. In a smart city, the huge volume of data collected using vision sensors can be used to improve services provided by the city, especially for the road traffic management system. Monitoring the road monitor traffic congestion is important as the number of vehicles continues to increase every year. Starting from 2021, over 71 million vehicles are sold globally [2]. This shows that there is a need for efficient traffic monitoring and management systems to avoid traffic congestion.

With the increasing number and volume of traffic data vehicles on the road, effective road traffic monitoring is crucial to improve the flow of traffic, minimizing accidents, and reducing traffic congestion [3]. Vehicle classification plays an essential role in optimizing the road traffic flow and enhancing road safety. Different vehicle types, such as cars, buses, trucks, and motorcycles, have distinct speed, space, and acceleration characteristics, affecting congestion patterns. Accurate classification helps in traffic signal control, lane management, and smart tolling systems to ease the congestion. Thus, this will also allow predictive solutions and decision making for better road planning and road traffic management system. With the integration of Artificial Intelligence (AI) for road traffic monitoring, traffic congestion may be reduced by offering a predictive solution for future planning and decision making. The predictive analytics in AI can predict congestion by using historical data, detect accidents and related events in a short time, and optimise public transportation schedules, which will reduce the occurrence of congestion. In delivering a high-quality predictive solution for future road traffic monitoring, a highly accurate method is needed as reliable and accurate information for effective traffic management. Therefore, in this paper, we investigate the reliability of artificial intelligence using Convolution Neural Network (CNN) of deep learning models to classify between types of vehicles for road traffic monitoring. The types of vehicles on the road, information and vehicle distribution can be further used as the input for analytics model for future prediction of road traffic conditions. The analytics synthesize, analyze the trends and identify the patterns based on the data for future planning, decision making and actions to improve the road traffic monitoring.

This research focuses on the detection of a suitable CNN deep learning model for vehicle classification. The CNN method is chosen since it is one of the most reliable deep learning models which can automatically extract meaningful patterns and features. CNN utilizes its convolutional and pooling layer to preserve spatial relationships within an image which allows it to recognize objects regardless of its position in an image. CNN uses shared weight through convolutional filters which may reduce the number of parameters compared to a fully connected network. This capability will reduce the time complexity as fewer computations are needed, making CNN method a fast-training method [3].

To test the proposed CNN model, datasets of vehicles that consist of 1324 vehicle images from five different categories of vehicles which are ambulance, bus car, motorcycle and truck are created. Then, the performances of the CNN models are compared between Single Shot Detector (SSD) Resnet50, SSD Resnet152, SSD MobileNet, You Only Look Once (YOLO) YOLOv5 and YOLOv8 in terms of precision, recall, F1 score and computational time. The rest of this paper is organized as

follows: Section II presents the related work on the vehicle classification methods. Section III described the details of the proposed method. Section IV presents the experiment setup. The performance of the proposed method is presented in section V, followed by the discussion in Section VI. Finally, the conclusions and future work are highlighted in Section VII.

II. Related Work

The accurate prediction of road traffic patterns and vehicle types able to improve road traffic management. Early detection of traffic congestion and vehicle distribution is essential for prediction of road traffic conditions, optimizing traffic flow, and enhancing future planning for road transportation. Object detection methods play a crucial role in identifying and classifying the objects in images [4]. With computer vision technology and deep learning object recognition methods, smart traffic systems able to offer the automated detection and classification of various types of vehicles on the road. This enables more efficient traffic control monitoring. However, in vehicle detection and classification, achieving an accurate classification and consequently prediction remains a challenge due to factors such as varying lighting conditions, occlusions, and diverse vehicle appearances [5, 6].

In recent years, the deep learning-based models show the best performance in detecting objects with high classification accuracy [7]. In the intelligent transportation system, deep learning models able to automatically extract important features in order to classify vehicles such as motorcycles, buses, cars, ambulance and trucks into their own category. This will enhance transportation systems, especially road traffic monitoring and road safety to reduce traffic congestion. The deep learning methods focus on the useful features which are extracted automatically. The methods analyse extracted features with similar logical structures as the human brain which enable to obtain more accurate results compared to the other methods such as statistical, morphology and model-based methods. The deep learning method, able to improve object detection and classification [8].

In deep learning, the CNN method is the most promising method that is able to effectively extract the significant features of the object and able to achieve high classification accuracy in most of the application [9] . In CNN method, there are the two object detection which have gained popularity, namely Single Shot Detector (SSD) and You Only Look Once (YOLO). These methods have gained popularity due to their accuracy performance [10].

The SSD is an object detection framework which predicts a bounding box in a single forward pass through a deep neural network. To perform detection on objects of various sizes, it uses multiple feature maps at different scales. SSD have gained popularity as they balance well between speed and accuracy. SSD combined with different types of ResNet architecture will result in different SSD ResNet models such as SSDResNet50, SSD ResNet152 and SSDMobileNet. SSD ResNet50, combines SSD with ResNet-50 as the backbone, which is a deep convolutional neural network consisting of 50 layers. It is suitable for real time applications with moderate computational power. The SSD Resnet152 on the other hand uses a deeper neural network with 152 layers. The increased number of layers in the SSD Resnet152 able to improve the classification accuracy of the model. However, this increases the time complexity. The model that able to reduces the time complexity while maintaining a reasonable amount of accuracy is the SSD MobileNet [11]. The SSD MobileNet model is design and optimized for the mobile.

Apart from the SSD model, YOLO model is also widely used for object detection. It frames object detection in images as a regression problem for separated bounding boxes in YOLO. It also provides an image with captions and the object is highlighted with the probability of correct detection. YOLO models are also seen to be the most suitable model in many detections and classification tasks as it shows promising results and able to obtain high classification accuracy in object detection and classification [7]. There are two advanced versions of YOLO, namely YOLOv5 and YOLOv8. The key differences between these two versions are its architecture. YOLOv5 utilizes anchor-based architecture where the anchor box is needed to be predefined with different size and aspect ratios according to the object used in the image. When predicting, it adjusts the predefined anchor box to better match the actual object. On the other hand, YOLOv8 uses anchor-free architecture which directly predicts the object location by identifying key features from feature maps, rather than relying on an anchor box like YOLOv5.

Due to YOLO are among the most efficient method for object detection, Ghoreyshi et al. proposed vehicle classification based on YOLO of CNN model. In their work, they able to achieve a high classification accuracy with 91% accuracy. This shows that YOLOv3 models have potential in effectively recognizing and classifying vehicles. However, in their research, less detection is observed for the images with occlusion which is a common scenario in real world traffic environments [3].

Similarly, Gao et al. also use YOLO model for the detection of vehicles. However, in their research, the YOLOv5 is selected to classify multi-class vehicle detection. The YOLOv5 models used in their work able to obtain accurate results with 96% accuracy results. The results obtain in their work shows that YOLOv5 methods are effective in real time applications and can be used in traffic monitoring [12].

The performance of YOLOv5 is further explored by Kumar et al. They proposed YOLOv5 with DeepSORT algorithm in their work to address both detection and classification of vehicles in dynamic traffic scenarios. Their research focuses on real-time performance efficiency. The research offers a practical solution in applications which require quick monitoring and decision making in traffic environments [13].

From the above review, SSD and YOLO were among the methods that had been considered in existing vehicle classification. The SSD and YOLO are seen as the most suitable candidate for vehicle classification. Therefore, in this research, a CNN deep learning based method of SSD and YOLO are chosen for vehicle detection and classification. The SSD is chosen due to its ability to balance well between speed and accuracy. On the other hand, the YOLO is selected as it able to obtain high classification accuracy in many object detection and classification tasks. Based on the review, four models which are

YOLOv5, YOLOv8, SSD ResNet50, SSD ResNet152 were among the models that had been considered in existing vehicle classification recognition due to their strength in detection and classification. YOLOv5 shows high accuracy and rapid inference, ideal for real-time applications. YOLOv8 also shows high accuracy and robustness against challenging conditions. The SSD ResNet50 on the other hand able to improve the detection on moderate complex scenarios and better handling the details of the objects than the lightweight. The SSD ResNet152 is also one of the chosen models due to the robust detection in complex traffic scenarios. The SSD is seen to be an efficient model for real-time applications with faster processing speeds [14].

Based on the review, five models which are SSD Resnet50, SSD Resnet152, SSD MobileNet, YOLOv5 and YOLOv8 are seen to be the most suitable model for vehicle classification tasks. The four models which are SSD Resnet50, SSD Resnet152, YOLOv5 and YOLOv8 are selected due to their accuracy and performance in detection and classification. While the SSD MobileNet model is selected due to it ability to reduce the time complexity while maintaining a reasonable amount of accuracy. To determine the best model for vehicle classification in smart city, the performance of these five models namely, SSD Resnet50, SSD Resnet152, SSD MobileNet, YOLOv5 and YOLOv8 are evaluated and compared.

III. Proposed Model

The proposed model for vehicle classification can be divided into two phases, training and testing. In this paper, generally, two CNN deep learning models are evaluated which are the SSD and YOLO. The overall scheme for the vehicle detection is shown in Fig. 1. Each of these methods will be discussed in the following subsections. To obtain the most suitable model for vehicle classification from these two main models, five different models are tested, and their performance are compared. The five different models are the SSD Resnet50, SSD Resnet152, SSD MobileNet, YOLOv5 and YOLOv8. Each of these models will be discussed in the following subsections.

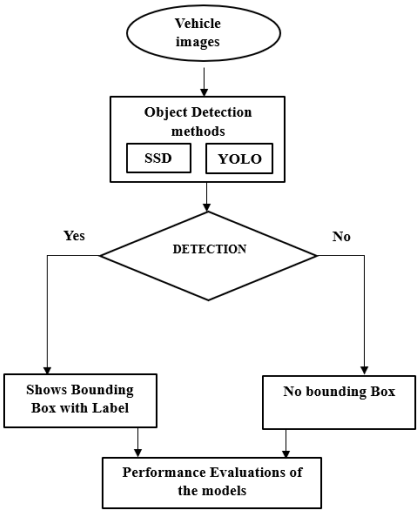


Fig. 1. Overall scheme with two different object detection methods.

A. Single Shot Detector (SSD)

The SSD is an extension of Convolutional Neural Network (CNN) model architecture which is designed for object detection. The SSD method uses deep CNN for the detection and classification of the object location in an image [15]. SSD utilizes multiple feature maps and anchor boxes to detect objects from various sizes and predicts its location and class score of the objects within the images [16]. Three SSD models evaluated are SSD ResNet50, SSD ResNet152 and SSD MobileNet. The basic SSD architecture is shown in Fig. 2.

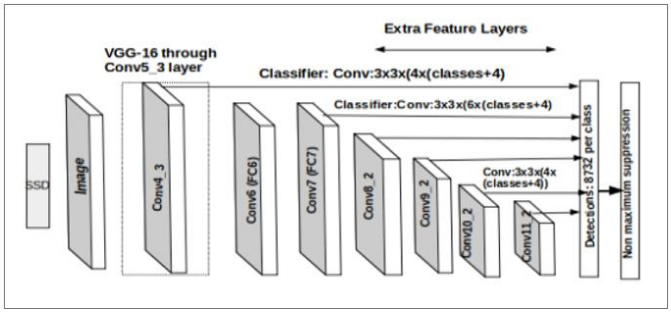


Fig. 2. SSD Architecture [15].

The SSD architecture consists of VGG-16 acts as the backbone of the architecture. The model extracts features from the input image and produces various feature maps which capture different levels of detail in the image. The extra feature layers that include various sizes of layers from bigger to smaller. The feature maps from the VGG-16 produced are then being processed further to create additional feature maps. These additional feature maps are much smaller in spatial size which enables the detection of objects from various sizes. The detection layer feeds on all the feature maps produced by each of the layers and applies a small convolutional filter to predict the class and localization of the objects for each of them. Then, the non-maximum suppression refines the prediction by removing heavily overlapping bounding boxes to finalize the bounding box and class label. To obtain the optimum classification accuracy in this research, three SSD variants are produced by replacing the VGG with three different model namely; ResNet50, ResNet152 and MobileNet. Each of these models will be discussed in the following subsections.

1) SSD ResNet 50: The SSD ResNet50 model consists of SSD as a framework with ResNet50 as its backbone. The SSD ResNet50 is an object detection model which has been pretrained with COCO 2017 dataset. This model localizes detected objects by drawing bounding boxes around the object. The model utilizes Feature Pyramid Network (FPN) for multi-level feature maps generation to feed into the SSD framework as the input. During the training phase, momentum optimizer with 0.04 learning rate is used and it is reduced when a plateau in the performance is detected. The SSD ResNet50 model architecture used is shown in Fig. 3.

2) SSD ResNet-152: The SSD ResNet-152 model used in this research is the combination SSD architecture with ResNet-152 as its backbone for feature extraction and SoftMax classifier for the class prediction. It is a deep convolutional neural network (DCNN) that consists of 152 layers which

include convolution layers, down sampling layers and fully connected layers. The SSD ResNet-152 model uses deep residual connections to overcome the vanishing gradient issue during the deep network training. The SSD ResNet-152 model architecture is shown in Fig. 4.

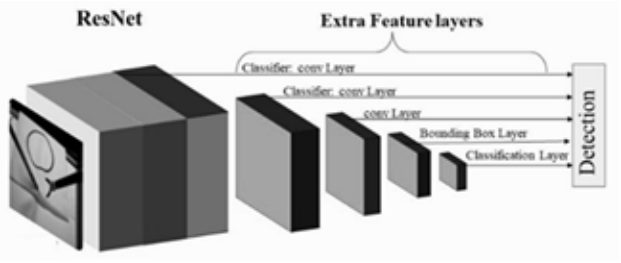


Fig. 3. SSD-ResNet50 [17].

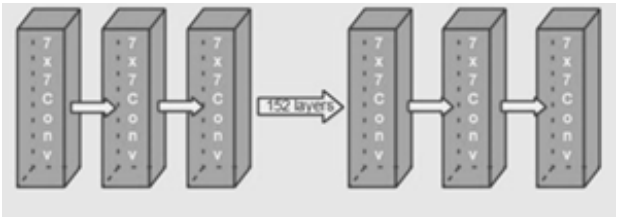


Fig. 4. Architectural diagram of SSD ResNet-152 [18].

3) SSD-MobileNet: The SSD-MobileNet is an object detection model which consists of SSD architecture with MobileNet as its backbone. This model is suitable for real time applications as it able to balance between speed and accuracy performance. The SSD-MobileNet is an efficient model where it preserves the important information while processing an image [19]. The model architecture of the SSD-MobileNet is shown in Fig. 5.

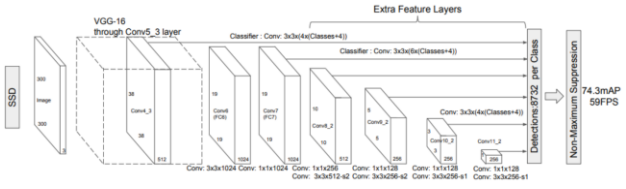


Fig. 5. The architecture of SSD-MobileNet [19].

B. You Only Look Once (YOLO)

The general structure of YOLO for the classification is shown in Fig. 6. In YOLO architecture, the detection of objects is treated as a regression problem in which the image is divided into grids and the prediction is done by predicting the bounding box and class probability for each grid cell in a single pass, making it exceptionally fast [20, 21]. In YOLO architecture, the vehicle images are resized and standardized to ensure consistency in grid structure and to simplify the process. Then, the images pass through the Convolutional layers which extract basic features of the vehicle images. In the early stage, the convolutional layers are paired with max pooling for down sampling to reduce the spatial dimension of the feature maps produced by the convolutional layer. In this research, the process of reducing the spatial dimension is applied to help the model focus on more abstract features of the vehicle images. This process is repeated several times until it is sufficiently small. In the middle stage, the vehicle images are then pass through a convolutional layer without max pooling. These layers deepen the feature extraction process to capture more detailed patterns. Lastly, the compressed feature maps produced by the previous layers are processed by fully connected layers to output the final prediction in the form of bounding box, confidence score and class probabilities.

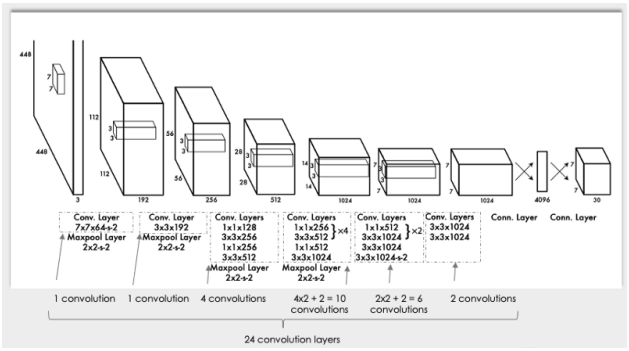


Fig. 6. YOLO Architecture [22].

In this research, there are two YOLO models used for vehicle classification which are YOLOv5 and YOLOv8. These models are chosen due to their speed, accuracy and efficiency in real time object detection. YOLOv5 models are highly accurate with rapid inference time, making it ideal for real time applications while YOLOv8 models are highly accurate and robust against challenging conditions. Each of these models will be discussed in the following subsections.

1) YOLOv5: The YOLOv5 model that used in this research consists of four parts which are input, backbone, neck and head as shown in Fig. 7. It utilizes CSP-Darknet53 with the Cross Stage Partial (CSP) strategy as its backbone to improve information flow and gradient issues [23]. YOLOv5 incorporates with a Spatial Pyramid Pooling (SPP) variant and uses BottleNeckCSP within the Path Aggregation Network (PANet) for the neck structure to enhance the receptive field and contextual feature extraction. The head of YOLOv5 models consists of three convolutional layers that predict bounding boxes, scores, and object classes.

2) YOLOv8: YOLOv8 is the improvised version of YOLOv5. Unlike YOLOv5 which uses anchor boxes, YOLOv8 is the improvised version of YOLOv5 where it uses different approach to detect the object by directly predicting the object centres. This approach helps to improved generalization and irregular shapes challenges. The YOLOv8 used the Spatial Pyramid Pooling Feature (SPPF) as a training technique to improve multi scale object handling. The YOLOv8 model also improve efficiency and flexibility while maintaining the performance by swapping the original larger Kernal size (6x6) convolution with a 3x3 in the stem. It also updates the core by swapping the C3 blocks with C2f. By concatenating features in the neck without the needs of identical channel dimension, the overall parameter counts and tensor size is reduced [23]. The SSD YOLOv8 model architecture is shown Fig. 8.

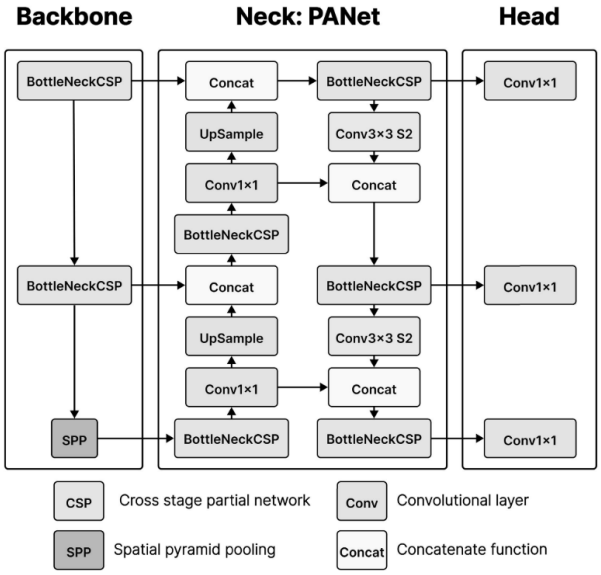


Fig. 7. Structure of YOLOv5 [24].

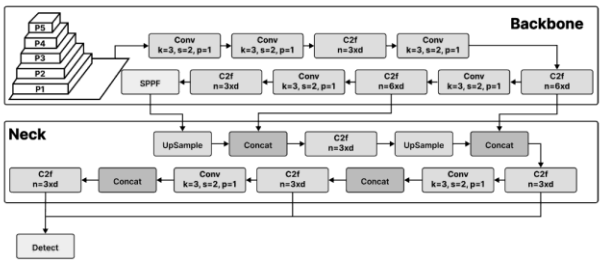


Fig. 8. YOLOv8 architecture structure [23].

IV. EXPERIMENT

To demonstrate the reliability of the proposed model, a series of comprehensive experiments is conducted using Google Colab Notebook. Colab based on the Jupyter Notebook is used for machine learning computational operations and Python 3 with T4 GPU hardware accelerator in the runtime is also used in the experiment. The parameter settings for each of the models SSD and YOLO are configured with specific parameters as summarized in Table I. The file path for label map, training and testing data and TF records are set based on the storage location while the other parameters are kept unchanged. The number of classes is set to five classes. These five classes are set up based on the five vehicle types used to test the model which are ambulance, bus, car, motorcycle and truck.

Generally, the dataset consists of 1324 vehicles classified into five categories namely, ambulance, bus, car, motorcycle and truck are created. These images are then manually labelled to train the model. The PBXT files containing the respective class name are created for the purposes of training. Then the vehicle images and its label are converted into TF Record files which are in a sequence of binary format. After the process of labelling, the images are then split into the training and validation sets with a ratio of 80% images used as training set and the remaining 20% for the validation set. The aforementioned ratio is considered in this research since most of the work related to the deep learning models from literature use these ratios to split the data in their work [25]. Following these ratios, the training set consists of 1059 images while the remaining 265 images are used in the validation set. The details categories of the vehicle dataset used in this experiment are shown in Table II.

TABLE I. PARAMETERS SETTING OF SSD AND YOLO

Model/ Parameter	SSD ResNet 50	SSD ResNet 152	SSD Mobile Net	YOLOv5	YOLOv8
Learning Rate	0.04	0.04	0.04	0.01	0.01
Weight	0.0004	0.0004	0.0004	0.0005	0.001
IoU threshold	0.6	0.6	0.6	0.7	0.1
Batch size	4	4	4	16	16

TABLE II. VEHICLE DATASET CATEGORIES

Vehicle Class	Number of Images
Ambulance	158
Bus	312
Car	320
Motorcycle	248
Truck	286
Total	1324

V. RESULTS

In this experiment, the classification performance is measured in terms of precision, recall and F1 Score. The computational time for each experiment is also recorded. Precision and recall are calculated by true positive, true negative, false positive and false negative value. True positive is the ability of the model to predict positive class correctly while, a true negative in which the model predicts the negative class correctly. On the other hand, false positive and false negative are the opposite from the true positive and true negative respectively. False positive occurs when the model incorrectly predicts the positive class while false negative occurs when the model incorrectly predicts the negative class. The formula for each of the performance measures is defined as follows:

$$\text{Precision} = TP/(TP+FP)$$

$$\text{Recall} = TP/(TP+FN)$$

$$\text{F1 score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$$

TP = True Positive, TN = True Negative

FP = False Positive, FN = False Negative

To evaluate the models, to find the most optimum result, the SSD models are trained with three numbers of training steps which are 10,000, 15,000 and 25,000. The YOLO models are trained with 25 epochs which are equivalent to 1,654 training steps. The training steps for YOLO models are calculated as in (1).

$$Training\ Steps = (Epoch \times Train\_Dataset)/Batch\ Size$$
$$= (25 \times 1059)/16$$
$$= 1654 \quad (1)$$

The results of SSD ResNet50, SSD ResNet152, SSD MobileNet, YOLOv5, and YOLOv8 model for the evaluation of each training step is shown in Table III.

TABLE III. COMPARISON RESULTS FOR SSD RESNET 50, SSD RESNET152, SSD MOBILENET, YOLOV5 AND YOLOV8 MODELS

Model	Training Steps	Precision	Recall	F1 Score	Computational Time
SSD Resnet 50	10,000	0.257	0.370	0.3033	1h 37m 30s
SSD Resnet 152	10,000	0.408	0.473	0.4381	2h 25m 56s
SSD Resnet 50	15,000	0.325	0.433	0.371	4h 30m
SSD Mobile Net	15,000	0.204	0.292	0.240	2h 59m 43S
SSD ResNet 50	25,000	0.264	0.363	0.305	1h
YOLOv 5	1645	0.909	0.944	0.926	10 mins
YOLOv 8	1645	0.956	0.968	0.968	30mins

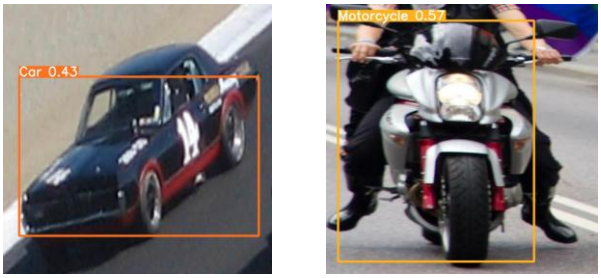
The SSD Resnet50 models which trained for all the three training steps of 10,000, 15,000 and 25,000 steps show low performance for all measure of precision, recall and f1 score. The high computational time is also observed for SSD Resnet50 in all training steps, which are 1h 37m 30s for 10,000 steps, 2h 25m 56s for 15,000 steps and 4h 30m for 25,000 training steps respectively. The SSD Resnet152 which trained for 10,000 steps also shows lower performance for all performance measure of precision, recall, f1 score including computational time compared to SSD Resnet50. For the training steps of 15,000 steps, SSD Resnet50 able to achieve high performance measure for all of precision, recall, f1 score and computational time compared to the SSD MobileNet. The less computational time is also observed in SSD Resnet50 compared to SSD MobileNet. On the other hand, both YOLO models which are YOLOv5 and YOLOv8 are trained with 1,654 training steps. From the experiment conducted, Both YOLO models able to obtain high performance measures for all precision, recall and accuracy with less computational time compared to all of the SDD models. The computational time of YOLOv5 model takes only 10 minutes while slightly minimal longer time taken is observed in YOLOv8 with only 20 minutes difference.

Among all the models tested, the YOLOv8 model achieved the highest performance measure for all precision, recall and F1 score with 0.956 precision, and 0.968 both recall and F1 score. This is followed by the YOLOv5 model with 0.909 precision, 0.944 recall and 0.926 F1 Score.

VI. DISCUSSION

Overall, the YOLOv8 model outperformed others in detecting vehicle classification. Both of the YOLO models YOLOv5 and YOLOv8 outperform all of the SSD models which are SSD Resnet 50, SSD Resnet152, SSD MobileNet for all performance measure of precision, recall and f1 score including the computational time. The lower result obtained in all the SSD models is due to the drawbacks of the SSD approach that having difficulty to accurately detect on the small or far away vehicle. The SSD model relies on the lower-resolution feature maps for detecting small or far away vehicles, which sometimes may lack sufficient semantic information. Thus, the small or far away vehicle can be missed or misclassified. On the other hand, the YOLO model enhanced detection accuracy and robustness against challenging conditions including small object [26].

Among all the models tested, YOLOv8 able to achieve the highest performance measure for all precision, recall and F1 score with more than 0.956. This is followed by YOLOv5 with 0.909 of precision,0.944 recall and 0.926 F1score. Despite YOLOv8 able to detect vehicles in various conditions including vehicles in close proximity, far away and blur images, the low confidence score detection can still be observed on classic racing car and occluded motorcycle images as shown in Fig. 9.



(a) (b)

Fig. 9. Examples of the low confidence score detection (a) Classic racing car (b) Occluded motorcycle images.

VII. CONCLUSION

The YOLOv8 model is proposed for vehicle classification to classify between types of vehicles in smart city with a goal to provide intelligent solutions for road traffic monitoring. This can be achieved using accurate predictive data analytics for future action planning and decision making. The result shows that the model YOLOv8 able to effectively classify types of vehicles including vehicles in close proximity, far away and blur images. Therefore, it can be used to provide intelligent solutions to improve road traffic system for smart city. Though the YOLOv8 model is superior compared to other models and achieves more accurate classification, the method was shown to be less effective in detecting some of the vehicle images as shown in VI. In the future, we will concentrate on improving these drawbacks by increasing the diversity of vehicle images dataset by using data augmentation techniques for more advanced deep learning methods.

ACKNOWLEDGMENT

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

REFERENCES

[1] [1] B. Kidmose, "A review of smart vehicles in smart cities: Dangers, impacts, and the threat landscape," Veh. Commun., vol. 51, no. July 2024, 2025, doi: 10.1016/j.vehcom.2024.100871.

[2] Statista, "Global car sales 2010–2021 | Statista." Accessed: Nov. 06, 2021. [Online]. Available: https://www.statista.com/statistics/200002/international-car-sales-since-1990

[3] A. M. Ghoreyshi, A. AkhavanPour, and A. Bossaghzadeh, "Simultaneous Vehicle Detection and Classification Model based on Deep YOLO Networks," in 2020 International Conference on Machine Vision and Image Processing (MVIP), IEEE, Feb. 2020, pp. 1–6. doi: 10.1109/MVIP49855.2020.9116922.

[4] M. Hnewa and H. Radha, "Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques," IEEE Signal Process. Mag., vol. 38, no. 1, pp. 53–67, Jan. 2021, doi: 10.1109/MSP.2020.2984801.

[5] M. Y. Mamilla, R. Al-haddad, and S. Chowdhury, "Resampling Imbalanced Healthcare Data for Predictive Modelling," vol. 16, no. 2, 2025.

[6] S. Emmons-Bell, C. Johnson, and G. Roth, "Prevalence, incidence and survival of heart failure: a systematic review," Heart, vol. 108, no. 17, pp. 1351–1360, 2022.

[7] M. A. Feroz, M. Sultana, M. R. Hasan, A. Sarker, P. Chakraborty, and T. Choudhury, "Object Detection and Classification from a Real-Time Video Using SSD and YOLO Models," 2022, pp. 37–47. doi: 10.1007/978-981-16-2543-5_4.

[8] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," Neurocomputing, vol. 396, pp. 39–64, Jul. 2020, doi: 10.1016/j.neucom.2020.01.085.

[9] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," Mar. 01, 2023, MDPI. doi: 10.3390/computation11030052.

[10] G. Chandan, A. Jain, H. Jain, and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," in 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, Jul. 2018, pp. 1305–1308. doi: 10.1109/ICIRCA.2018.8597266.

[11] S. Bouraya and A. Belangour, "Object Detectors' Convolutional Neural Networks backbones : a review and a comparative study," Int. J. Emerg. Trends Eng. Res., vol. 9, no. 11, pp. 1379–1386, Nov. 2021, doi: 10.30534/ijeter/2021/039112021.

[12] X. Gao, J. Xu, C. Luo, J. Zhou, P. Huang, and J. Deng, "Detection of Lower Body for AGV Based on SSD Algorithm with ResNet," Sensors, vol. 22, no. 5, p. 2008, Mar. 2022, doi: 10.3390/s22052008.

[13] S. Kumar et al., "Fusion of Deep Sort and Yolov5 for Effective Vehicle Detection and Tracking Scheme in Real-Time Traffic Management Sustainable System," Sustainability, vol. 15, no. 24, p. 16869, Dec. 2023, doi: 10.3390/su152416869.

[14] K. Beckman, "Pruning a Single-Shot Detector for Faster Inference: A Comparison of Two Pruning Approaches," 2022.

[15] K. Wadhwa and J. Kumar Behera, "Accurate Real-Time Object Detection using SSD," Int. Res. J. Eng. Technol., 2020, [Online]. Available: www.irjet.net

[16] L. Fang, X. Zhao, and S. Zhang, "Small-objectness sensitive detection based on shifted single shot detector," Multimed. Tools Appl., vol. 78, no. 10, pp. 13227–13245, May 2019, doi: 10.1007/s11042-018-6227-7.

[17] F. R. Fathabadi, J. L. Grantner, I. Abdel-Qader, and S. A. Shebrain, "Box-trainer assessment system with real-time multi-class detection and tracking of laparoscopic instruments, using cnn," Acta Polytech. Hungarica, vol. 19, no. 2, pp. 7–27, 2022, doi: 10.12700/aph.19.2.2022.2.1.

[18] S. Athisayamani, R. S. Antonyswamy, V. Sarveshwaran, M. Almeshari, Y. Alzamil, and V. Ravi, "Feature Extraction Using a Residual Deep Convolutional Neural Network (ResNet-152) and Optimized Feature Dimension Reduction for MRI Brain Tumor Classification," Diagnostics, vol. 13, no. 4, 2023, doi: 10.3390/diagnostics13040668.

[19] M. Ulaszewski, R. Janowski, and A. Janowski, "Application of computer vision to egg detection on a production line in real time.," Electron. Lett. Comput. Vis. Image Anal., vol. 20, no. 2, pp. 113–143, 2021, doi: 10.5565/rev/elcvia.1390.

[20] P. Soviany and R. T. Ionescu, "Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction," in 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, Sep. 2018, pp. 209–214. doi: 10.1109/SYNASC.2018.00041.

[21] M. A. Zuraimi and F. H. Kamaru Zaman, "Vehicle Detection and Tracking using YOLO and DeepSORT," in 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, Apr. 2021, pp. 23–29. doi: 10.1109/ISCAIE51753.2021.9431784.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02640

[23] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverria, "YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios," J. Image Graph., vol. 12, no. 2, pp. 127–136, 2024, doi: 10.18178/joig.12.2.127-136.

[24] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverria, "Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection," IEEE Access, vol. 11, no. September, pp. 96554–96583, 2023, doi: 10.1109/ACCESS.2023.3312217.

[25] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation."

[26] M. Zhao, Y. Zhong, D. Sun, and Y. Chen, "Accurate and efficient vehicle detection framework based on SSD algorithm," IET Image Process., vol. 15, no. 13, pp. 3094–3104, 2021.