ONLINE JOURNAL FOR TVET PRACTITIONERS
e-ISSN: 2289-7410

Vol. 9 No. 2 (2024) 224-235
https://publisher.uthm.edu.my/ojs/index.php/oj-tp

OJ-TP

Vehicle Detection and Counting for Traffic Congestion Estimation Using YOLOv5 and DeepSORT in Smart Traffic Light Application

Yew Zun Kam[1], Chessda Uttraphan[1], Mohd Norzali Haji Mohd[1]*, Fawad Salam Khan[1, 2], Chiok Chuan Lim[3]

[1] *Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA*
[2] *Department of Machine Learning Innovation, Country CONVSYS (Pvt) Ltd., Islamabad 45210, PAKISTAN*
[3] *Sena Traffic System Sdn. Bhd., Selangor, MALAYSIA*

*Corresponding Author: norzali@uthm.edu.my

DOI: https://doi.org/10.30880/ojtp.2024.09.02.021

Article Info

Received: 13 February 2024
Accepted: 30 September 2024
Available online: 30 September 2024

Keywords

Deep learning, YOLOv5, DeepSORT

Abstract

The escalating number of vehicles on the roads has exacerbated traffic congestion, presenting a significant and inevitable challenge for road users. Traffic congestion not only increases the travel time of road users; it also causes air pollution since more vehicles are stuck on the road. To address this issue, deep learning algorithms, including YOLOv5 and DeepSORT, have been leveraged to enable vehicle detection and estimate the number of vehicles through image processing. This study focuses on training a custom YOLOv5 dataset of vehicles and compares its performance with a pretrained YOLOv5 dataset in terms of feasibility for detecting and estimating the number of vehicles. In the proposed approach, YOLOv5 is employed to detect vehicles in video streams, DeepSORT is used for vehicle tracking and counting as they pass through the detection zone, and the number of vehicles in each lane is estimated with the aid of Supervision. The custom dataset's validation demonstrates promising results, with the precision curve indicating convergence for all classes at 97.4% precision and an 87% accuracy in predicting vehicle classes. Additionally, the precision-recall curve assesses the ability to detect individual vehicle categories, such as bus, car, motorcycle, and truck, with accuracies of 68.2%, 95.2%, 79.9%, and 70.1%, respectively. Furthermore, the overall accuracy for detecting all vehicle classes combined is 78.3%. The F1 curve indicates that the system achieves a confidence level of 30.9% F1 score, ensuring 78% confidence in accurately predicting all classes. Both the pretrained and custom datasets exhibit similar accuracy in counting the total number of vehicles passing through the detection zone as well as estimating the number of vehicles in each lane. However, it is evident that the custom dataset's performance can be further improved by incorporating more extensive and diverse datasets during the training process to enhance the accuracy of vehicle detection and estimation. In conclusion, the integration of deep

This is an open access article under the CC BY-NC-SA 4.0 license.

learning algorithms, specifically YOLOv5 and DeepSORT, offers a promising solution for addressing traffic congestion by efficiently detecting and estimating vehicle numbers, which allow traffic systems to identify which lane is congested and prioritise ensuring the congested lane has a smooth traffic flow. Continued research with larger datasets can lead to further advancements and refined results in the field of traffic management and optimization.

1. Introduction

Up to this day, traffic congestion has become a significant and inescapable predicament for drivers. The rising number of vehicles each year is directly proportional to the increase in population. In Malaysia, for example, the number of registered vehicles grew from 16,892,812 in December 2019 to 17,486,589 in December 2020, and 17,728,482 in December 2021, as reported by CEIC Data Global Database [1]. This escalating number of vehicles relative to the population will only exacerbate road congestion. The effect of the increasing number of vehicles can be seen in places with a large population, such as Kuala Lumpur in Malaysia. The congestion often occurs during peak hours, such as early in the morning when people are going to work or going back home from work.

To alleviate this issue, traffic lights are introduced to facilitate smoother traffic flow on the road, as poor traffic flow is one of the factors that lead to traffic congestion. The first traffic light with red and green lamps was set up in London, England in 1868, while the first traffic light consisting of the three lamps, red, yellow, and green, which is now commonly used, was installed in New York City in 1918 [2]. Traffic lights regulate traffic flow by indicating the lanes that are permitted to move in intersections, thus enabling vehicles from different lanes to take turns moving. Traffic lights not only help to reduce traffic congestion but also guarantee the safety of road users, including drivers and pedestrians.

The Induction loop method, also known as vehicle detection loops or inductive-loop traffic detectors, is the most common way to detect the presence of vehicles at traffic lights. Induction loops are typically installed underground beneath the road surface enabling them to detect passing vehicles. Then, there is smart traffic light, which is a combination of traditional traffic light with sensors and artificial intelligence (AI) to remotely control the traffic flow according to the situation without human interference. In a situation where heavy traffic congestion occurs, human intervention such as traffic police officers will help to manage the traffic flow in place of the traffic light as normal traffic lights have fixed-preset timing. A smart traffic light on the other hand will be able to change and adjust the duration of the lights sequence according to the real time data collected around the traffic light.

AI is a computer program to mimic the human brain to solve problems and make decisions according to the problem at hand. Machine learning is a subfield of AI, which learns and improves itself like humans with data and algorithms without being explicitly programmed to improve its accuracy [3]. Deep learning, on the other hand, is a subset of machine learning, which learns to recognise objects using artificial neural networks like a human brain. One of the deep learning algorithms is called Convolutional Neural Network (CNN) which is good at interpreting, analysing, recognising and classifying images. In the training and learning process of CNN, an image is broken down into several layers with each layer containing a different set of characteristics [4]. Sharma et al. and Naranjo-Torres et al. have conducted reviews on the performance of CNN to identify and classify the objects in the images [5-6]. CNN learns to identify and classify objects through the characteristics, such that it identifies the object by using part of images it learned rather than using the whole image.

A smart traffic detects the vehicles around the traffic light using sensors, such as CCTV (closed circuit television), RFID (Radio Frequency Identification) reader, WSN (wireless sensor network) and Internet of Things (IoT) sensors, in order to obtain real time situation and data on the road around the traffic light and send them to traffic light controller to be implemented [7-10]. The approach in using CCTV or video captured by the camera has become possible with the advancement in the field of AI, specifically through the image processing using deep learning. Several research on using image processing to identify specific objects in the video can be found in [11-14].

Through the development of CNN, the YOLO (You Only Look Once) series is introduced, which received its name due to the image only passing through CNN algorithm only once in order to get the output, and it is popular due to its speed in identifying objects [15]. The first YOLO series was introduced by Joseph et al. and improved it into newer versions called YOLOv2 and YOLOv3 [16]. Later, there were several more versions by different authors, which are YOLOv4, YOLOv5, YOLOv6, YOLOv7 and YOLOv8. YOLOv5 which was introduced by Jocher et al. has become popular in 2020 due to its high flexibility in controlling model and its major competitor was YOLOv4 which has higher accuracy and faster result [17-18]. Apart from the YOLO series, the development of CNN also introduces DeepSORT. DeepSORT is the improvement of SORT (Simple Online Realtime Tracking) where it not only uses Kalman filters and Hungarian algorithms to track objects, but also uses CNN to extract and identify appearance features [19]. Several authors have conducted research using both YOLO and DeepSORT together to identify objects in the video [20-24]. Mohamed et al. and Kumari et al. have trained a custom YOLOv5 dataset with the dataset they prepared and used it to detect vehicles in the video, where YOLOv5 is used to detect and classify different classes of vehicles while DeepSORT is used to track vehicles across different frames in the video sequence [20-21]. Meanwhile, Dang et al. also uses DeepSORT and a YOLO series, but it is YOLOv3 instead of YOLOv5, to detect objects in video [22]. Bo Gao uses pretrained YOLOv5 and DeepSORT to track vehicles in the video [23]. Egi et al. has trained a custom YOLOv5 dataset to detect flower, green tomatoes and tomatoes so it can be together with DeepSORT so the algorithm will be able to track the detected object [24]. YOLOv5 comes with a pretrained dataset which is trained with Common Objects in Context (COCO) dataset and it is able to detect up to 80 types of objects. In order to detect specific objects using YOLOv5, a custom dataset has to be trained using a dataset of the desired objects.

This paper proposes a method to detect and count the number of vehicles on the road in the video to estimate the traffic congestion by using deep learning algorithms. The deep learning algorithms that were utilised in this work are the YOLOv5 and DeepSORT. YOLOv5 is used to detect and identify the vehicles while DeepSORT is used to track vehicles that are detected by YOLOv5. Since the pretrained YOLOv5 dataset has 80 different types of objects, a custom dataset is trained using YOLOv5 such that it will only detect different types of vehicles such as buses, cars, motorcycles and buses so it only focuses on these objects rather than having a broad range of detection unrelated to vehicles.

2. Methodology

The methodology flowchart is as shown in Figure 1.

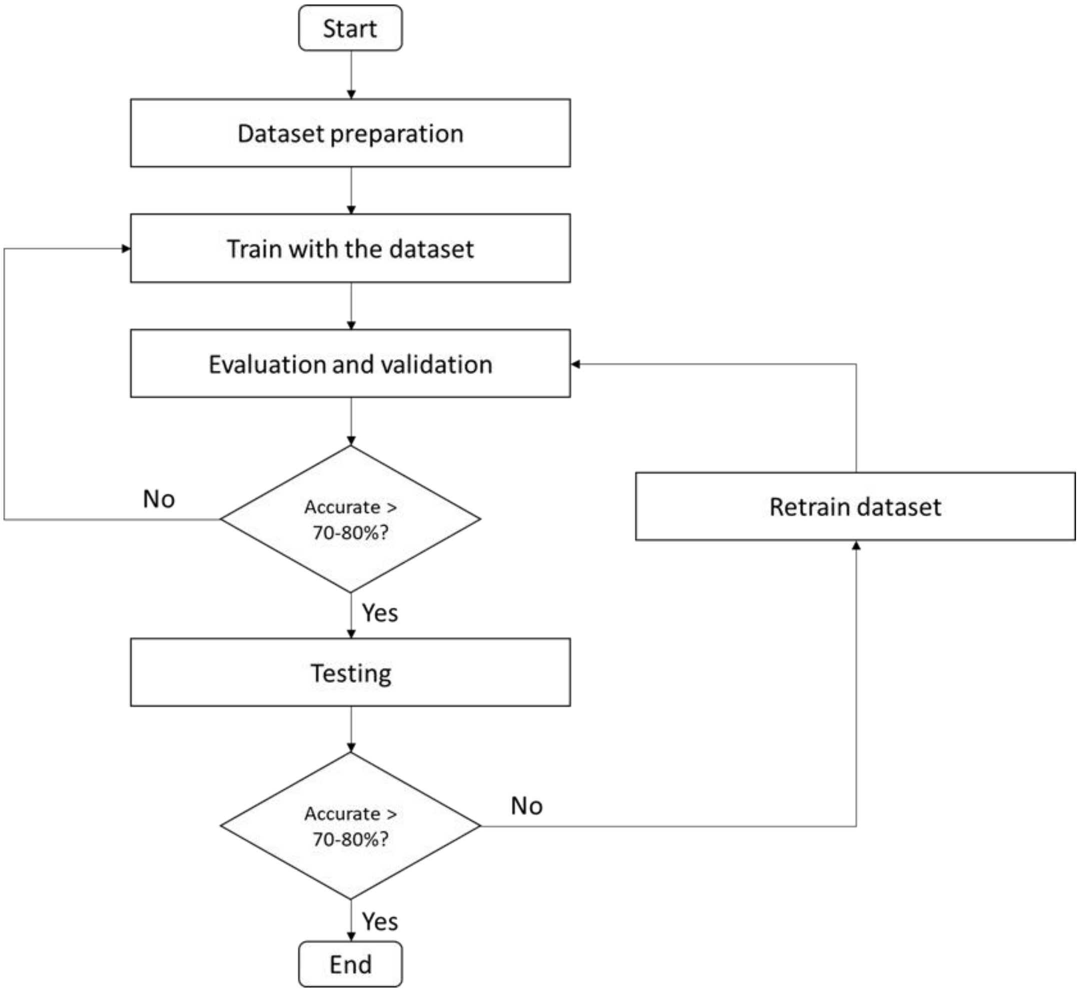

Fig. 1 *Methodology flowchart*

2.1 Dataset Collecting

In order to train a custom YOLOv5 dataset, a dataset of vehicles is prepared. Sena Traffic System Sdn. Bhd have provided 15 videos of road traffic. The video was then being converted into image frames that were used as the dataset and trained in YOLOv5. The video consists of daytime, and night-time videos as shown in Figure 2. In order to annotate and label each vehicle in each image, Roboflow Annotate is used to create a bounding box on each vehicle and identify its class as shown in Figure 3. The classes of vehicles needed for the dataset are bus, car, motorcycle and truck. The labelled images are then split into two categories, which are trained and valid. The labelled images in the trained category will undergo data augmentation by having the images flip horizontally and vertically to have more images with different positions. The dataset is then exported to YOLOv5 Pytorch TXT annotation format to be trained using YOLOv5.

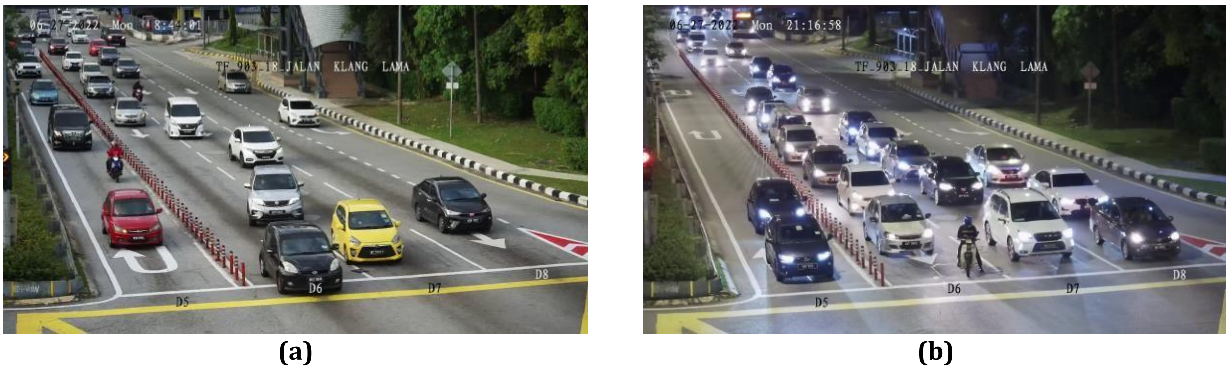

Fig. 2 *Road situation (a) Daytime; (b) Nighttime*

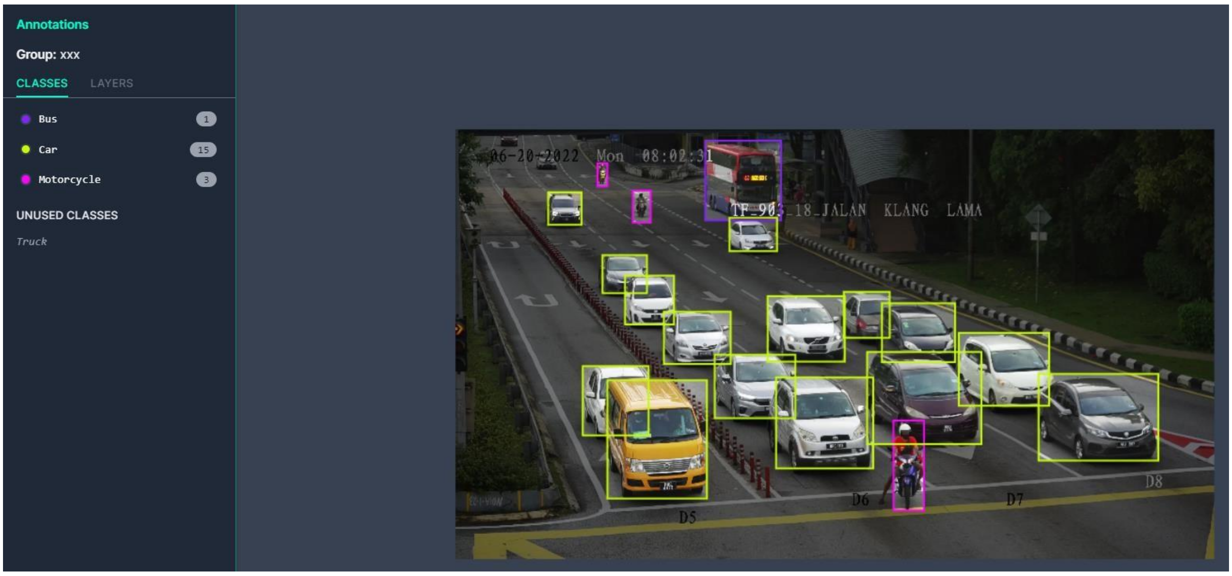

Fig. 3 *Labelling vehicles on robotflow annotate*

2.2 Custom Data Training and Validation

Figure 4 shows the architecture of YOLOv5 which consists of three main parts called Backbone, Neck and Head. The Backbone will extract the key features and characteristics from an image such that it can train the dataset with it. The Neck will create feature pyramids, which are basic components in recognition systems for detecting objects at different scales. The Head will determine the result of the detection.

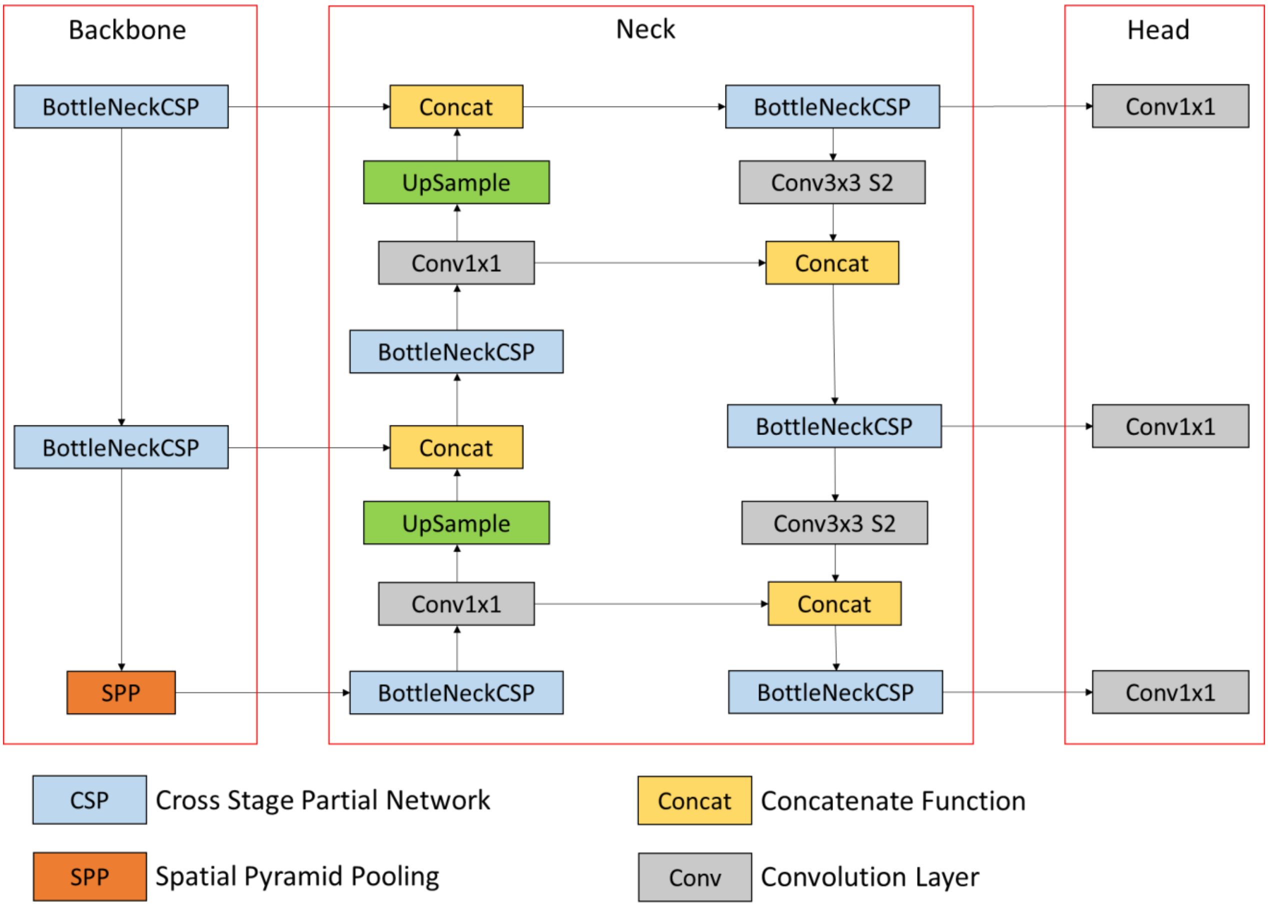

Fig. 4 *Architecture of YOLOv5*

The two categories of labelled trained and valid images are used for training and validation respectively. There are 2,800 images in the trained category while there are 700 images in the valid category. Results of the evaluation of the custom dataset through validation are given in the form of precision curve, recall curve, precision-recall curve and F1 curve. The equation of precision, recall and F1 score are by Eq. (1), Eq. (2) and Eq. (3), respectively.

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

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

$$F1 = 2\frac{(Precision)(Recall)}{Precision + Recall} \quad (3)$$

2.3 Vehicle Detection, Counting and Congestion Estimation

2.3.1 Total number of vehicles

After training and validation of the dataset, the weight of custom dataset is obtained. Weight is the file contains the output after the YOLO model is trained. Weight of both pretrained dataset and custom dataset are used to detect and count the number of vehicles in the video using YOLOv5 and DeepSORT. The function of YOLOv5 is to detect the vehicles in the video while DeepSORT will assign the detected vehicles with an ID and track them. The performance accuracy of DeepSORT is determined by MOTA (Multiple Object Tracking Accuracy) as given in Eq. (4). When the detected vehicle passes through the detection zone, the counter, which indicates the number ofvehicles in the video, will increase. The percentage error of the number of vehicles counted in the video is calculated using Eq. (5).

$$MOTA = 1 - \frac{\sum [False\ Negative + False\ Positive + Number\ of\ ID\ switched)]}{\sum Ground\ Truth} \tag{4}$$

$$Percentage\ Error = \frac{|Measured\ Value - Exact\ Value|}{Exact\ Value} \times 100\% \tag{5}$$

2.3.2 Congestion Estimation On Each Lane

In order to estimate the congestion on each lane, the coordinates of each lane in the video are determined using Robotflow PolygonZone. By using an algorithm called Supervision, it will estimate the number of vehicles within the coordinate of each lane. The vehicle density in the predetermined area will determine the congestion levelon the road. Supervision requires the YOLOv5 weight to detect objects, therefore, weight of both pretrained dataset and custom dataset were used.

3. Result and Analysis

The results obtained from validation are presented as below, where precision curve, recall curve, precision- recall curve and F1 curve are presented in Figure 5, 6, 7 and 8 respectively. The precision curve shows that all classes start to converge at 97.4% precision. Meanwhile, the recall curve shows an 87% accuracy to predict the class of each vehicle. The precision-recall curve in Figure 6 shows the accuracy of the model in detecting bus, car, motorcycle and truck individually and also all the vehicles together. The accuracies in detecting buses, cars, motorcycles, and trucks are 68.2%, 95.2%, 79.9% and 70.1%, respectively. Meanwhile, the accuracy to detect all vehicles is 78.3%. The F1 curve in Figure 8 shows that the system becomes confident at 30.9% F1 score and has 78% confidence to accurately predict all classes.

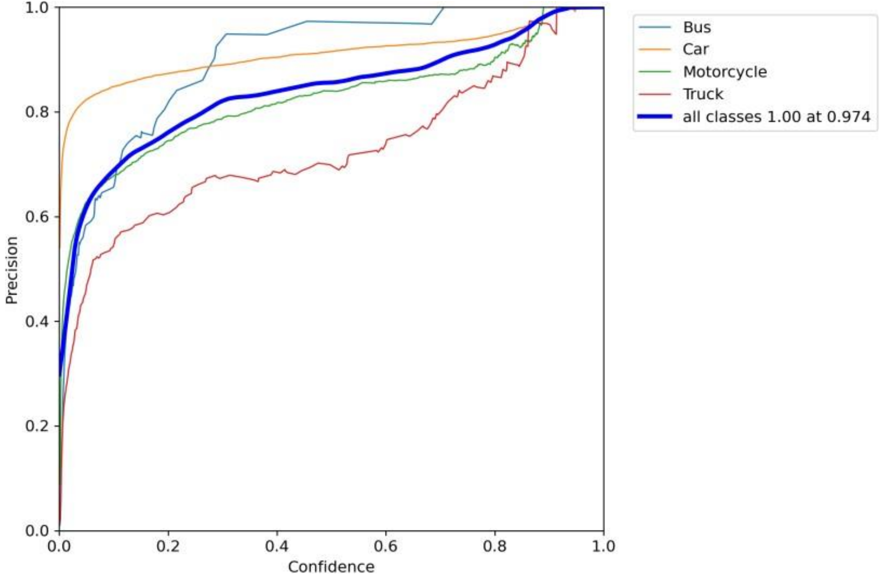

Fig. 5 *Precision curve*

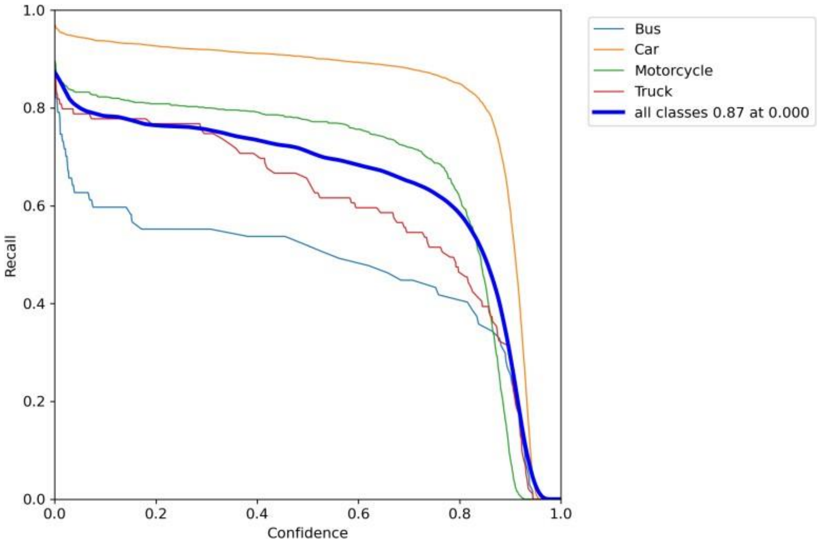

Fig. 6 *Recall curve*

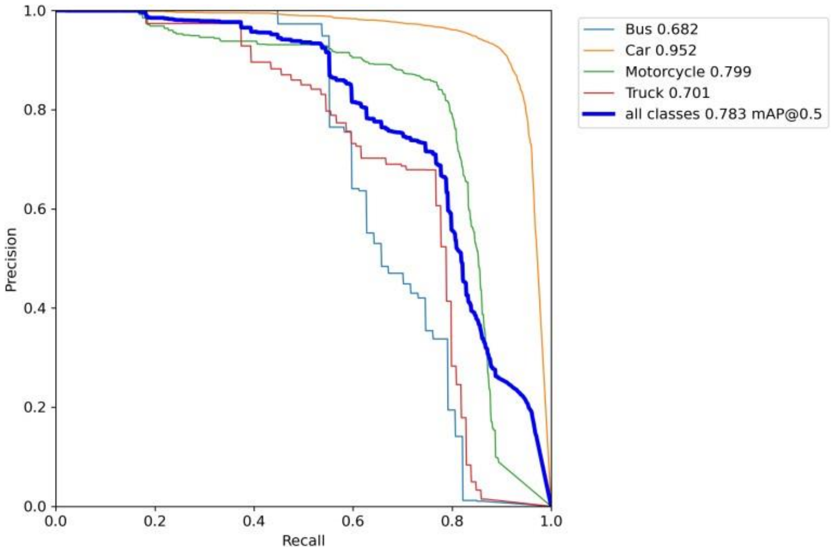

Fig. 7 *Precision-recall curve*

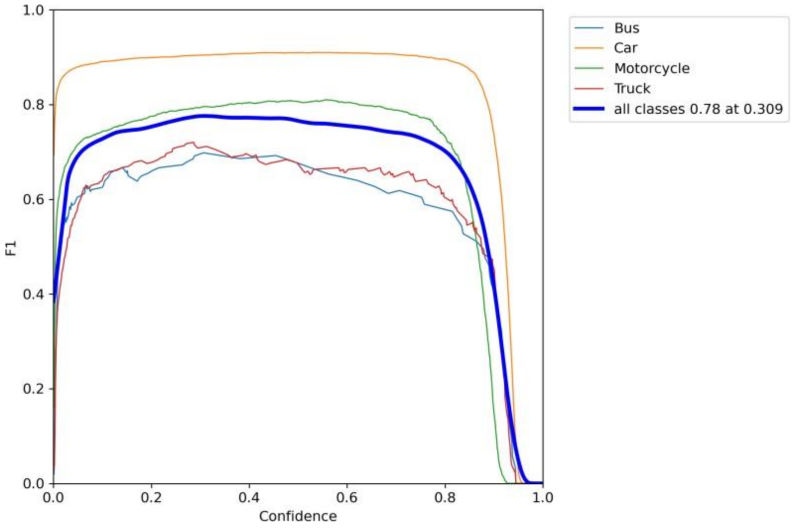

Fig. 8 *F1 curve*

Figure 9 shows the vehicles detected and classification into its respective classes by YOLOv5 during the validation process. The vehicles detected are classified as either bus, car, motorcycle, or truck.

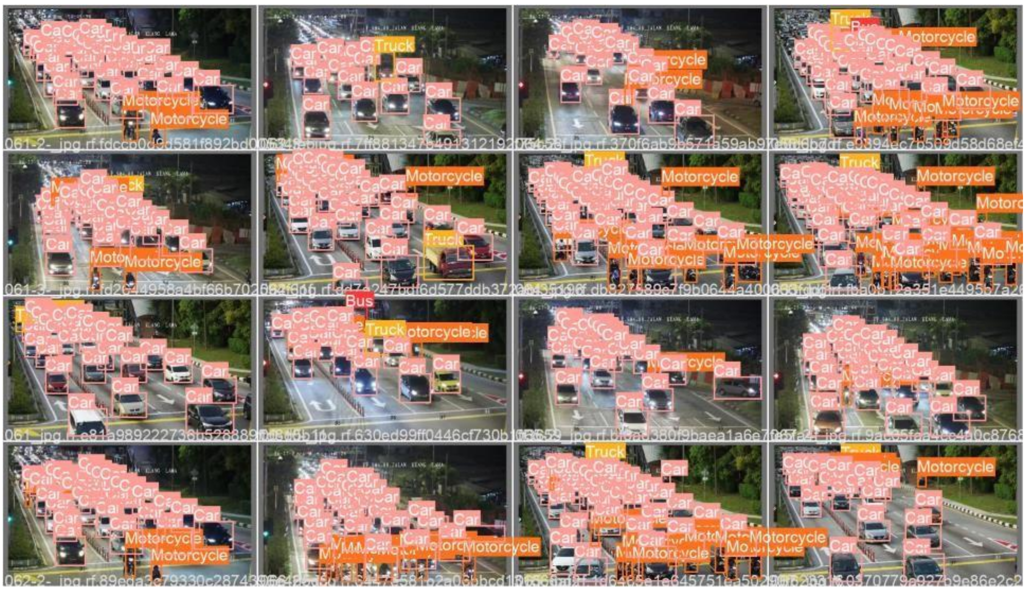

Fig. 9 *Vehicles detected and classified*

The total number of vehicles in the video is determined by estimating the number of vehicles passing through the detection zone. The total number of vehicles detected by the pretrained dataset are shown in Figure 10 while the result of the custom dataset is shown in Figure 11. Both pretrained dataset and custom dataset detected the same total number of vehicles passing through the detection zone, which is 76. When calculated manually, there are 62 vehicles passing through the detection zone in the 3 minutes 43 seconds long video. The percentage error of the number of vehicles detected passing through the detection zone is 22.58%. Thus, the accuracy of the number of vehicles detected is 77.42%. The error in counting the total number of vehicles in the video is caused by the overlapping class detection of a vehicle. For example, a truck might be identified as both car and truck by YOLOv5 due to their almost similar features and characteristics. When the truck with two overlapping classes passes through the detection zone, the counter will increase by two rather than one. To reduce the error of misidentifying the vehicles, the number of images in the dataset should be increased to improve the accuracy of the detection.

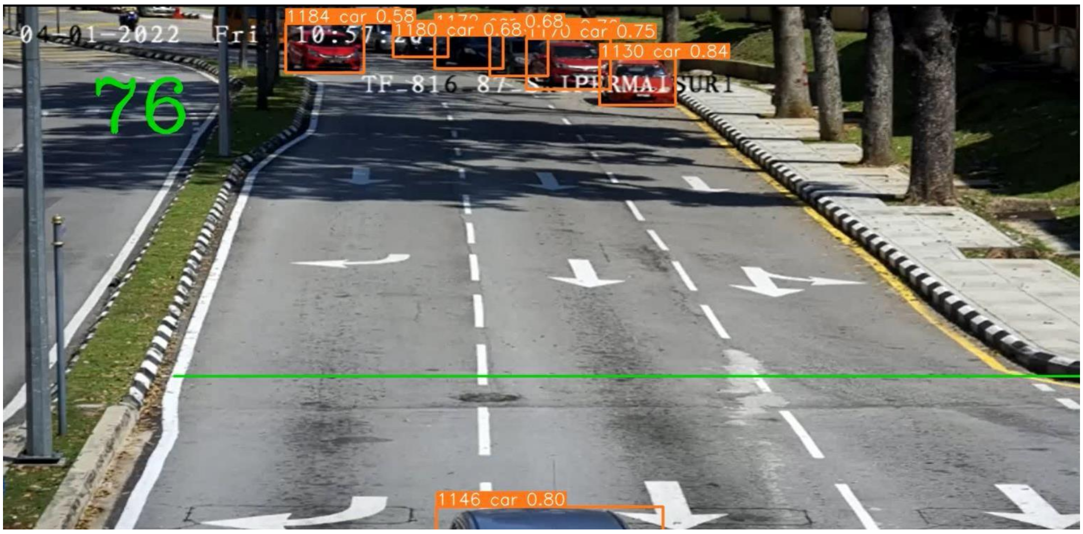

Fig. 10 *Number of vehicles detected using pretrained dataset*

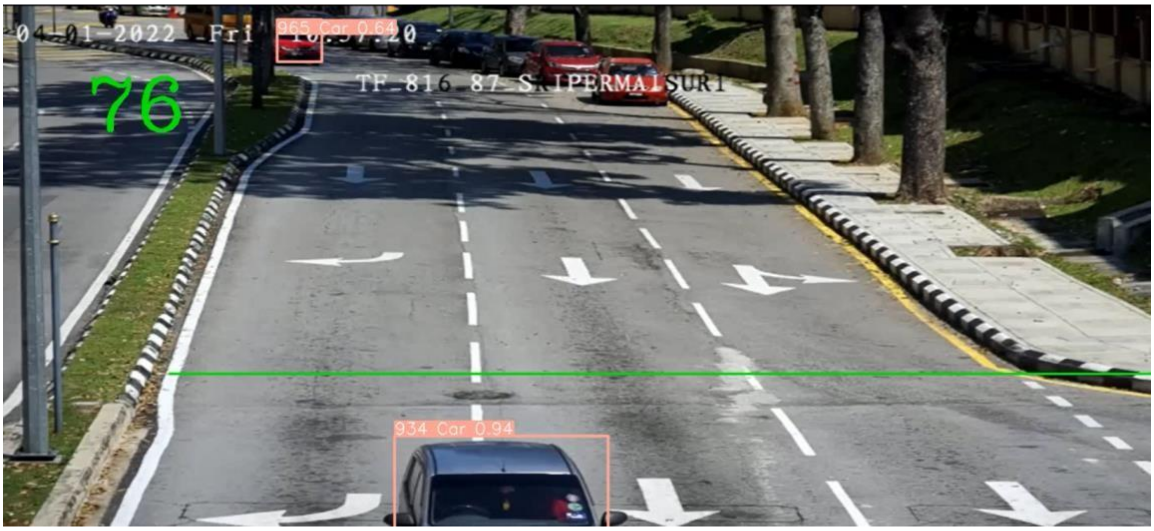

Fig. 11 *Number of vehicles detected using custom dataset*

Figure 12 displays the outcome of the pretrained dataset's estimation of the number of vehicles on each lane, while Figure 13 illustrates the result of the custom dataset. The result analysis of each lane using the pretrained dataset and custom dataset is as shown in Table 1. Table 1 display that both pretrained dataset and custom dataset have a minimum 66.67% accuracy in estimating the number of vehicles in each lane. Despite the accuracy, it is able to roughly estimate number of vehicles in each which can help to determine the congestion level in future work. Moreover, the results show that the feasibility of both models in estimating the vehicles on each lane are as good as each other. The noticeable difference between them is that the pretrained dataset whichis trained with a lot of images is able to detect vehicles which are small and far away in the video while the custom dataset which is trained with 2,800 images is able to detect the near and clearer vehicles only.

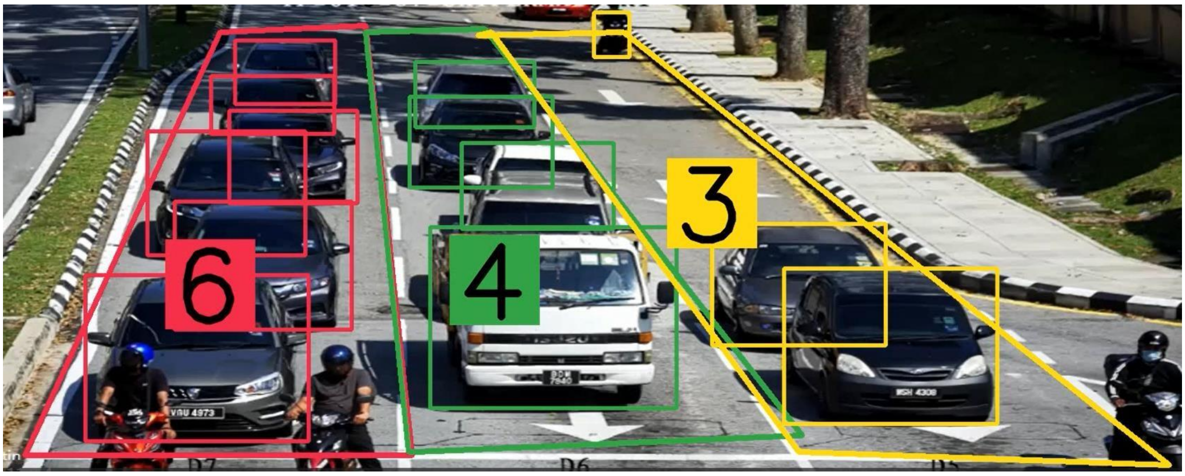

Fig. 12 *Number of vehicles detected in each lane using pretrained dataset*

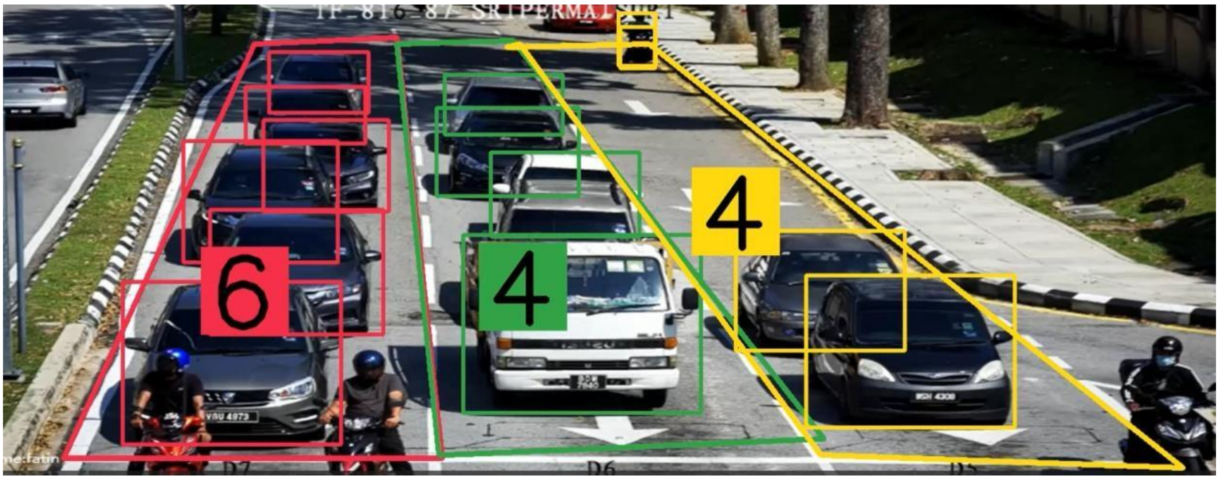

Fig. 13 *Number of vehicles detected in each lane using custom dataset*

Table 1 *Result analysis of each lane*

Lane	Number of vehicles	Pretrained dataset estimation (Accuracy)	Custom dataset estimation (Accuracy)
1 (Red)	8	6 (66.67%)	6 (66.67%)
2 (Green)	5	4 (75%)	4 (75%)
3 (Yellow)	3	3 (100%)	4 (66.67%)

4. Conclusion

In conclusion, the custom dataset trained using YOLOv5 in this work is on par with the pretrained dataset provided by YOLOv5. Although it is not too accurate, we are able to estimate the number of vehicles in each lane with at least 66.67%, which is a step further in designing a vehicle detection system to estimate the congestion on the road using deep learning. In this research, the size of the dataset trained is limited, which could potentially cause biases in the model. In order to improve accuracy, more dataset should be obtained and trained.

Acknowledgement

This research is supported by the Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2019/ICT04/UTHM/02/2), Universiti Tun Hussein Onn Malaysia and Sena Traffic System Sdn. Bhd. The author would like to thank CONVSYS (Pvt) Ltd., Islamabad, Pakistan for the inputs to improve the works.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Yew Zun Kam, Chessda Uttraphan, Mohd Norzali Haji Mohd;* ***data collection:*** *Yew Zun Kam, Lim Chiok Chuan;* ***analysis and interpretation of results:*** *Yew Zun Kam, Chessda Uttraphan, Mohd Norzali Haji Mohd, Fawad Salam Khan;* ***draft manuscript preparation:*** *Yew Zun Kam, Chessda Uttraphan, Mohd Norzali Haji Mohd. All authors reviewed the results and approved the final version of the manuscript.*

References

Ceicdata. (2021). Malaysia Number of Registered Vehicles. Retrieved April 5, 2023, from https://www.ceicdata.com/en/indicator/malaysia/number-of-registered-vehicles

Madehow. Traffic Light. Retrieved April 5, 2023 from http://www.madehow.com/Volume-2/Traffic-Signal.html

IBM Cloud Education. Machine Learning, Retrieved April 5, 2023 from https://www.ibm.com/my-en/cloud/learn/machine-learning

Mahbub Hussain, Jordan J. Bird, Diego R. Faria. (2019). “A Study on CNN Transfer Learning for Image Classification.” Advances in Intelligent Systems and Computing, vol 840, pp 191–202. https://doi.org/10.1007/978-3-319-97982-3_16

Parul Sharma, Yash Paul Singh Berwal and Wiqas Ghai. (2020). “Performance analysis of deep learning CNN models for disease detection in plants using image segmentation.” Information Processing in Agriculture, Volume 7, Issue 4, 2020, pp 566-574. https://doi.org/10.1016/j.inpa.2019.11.001

Naranjo-Torres, José, Marco Mora, Ruber Hernández-García, Ricardo J. Barrientos, Claudio Fredes, and Andres Valenzuela. (2020). "A Review of Convolutional Neural Network Applied to Fruit Image Processing" Applied Sciences, vol 10, 3443. https://doi.org/10.3390/app10103443

Goswami, Muskaan, Nikita Goel, Purnima Yadav and Amit Saini. (2020). “SMART TRAFFIC LIGHT SYSTEM USING CCTV.”

Ayesha Atta, Sagheer Abbas, M. Adnan Khan, Gulzar Ahmed and Umer Farooq. (2020). “An adaptive approach: Smart traffic congestion control system.” Journal of King Saud University - Computer and Information Sciences, Volume 32, November 2020, pp 1012-1019. https://doi.org/10.1016/j.jksuci.2018.10.011

Adil Hilmani, Abderrahim Maizate, Larbi Hassouni. (2020). "Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8841893, 28 pages, 2020. https://doi.org/10.1155/2020/8841893

Manpreet Singh Bhatia, Alok Aggarwal, Narendra Kumar. (2020). “Smart Traffic Light System to Control Traffic Congestion”. PalArch’s Journal of Archaeology of Egypt / Egyptology 17 (9):7093 - 7109. https://archives.palarch.nl/index.php/jae/article/view/5389

Fawad Salam Khan, Mohd Norzali Haji Mohd, Raja Masood Larik, Muhammad Danial Khan, Muhammad Inam Abbasi and Susama Bagchi. (2021). “A Smart Flight Controller based on Reinforcement Learning for Unmanned Aerial Vehicle (UAV).” 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2021, pp. 203-208. https://doi.org/10.1109/ICSIPA52582.2021.9576806

Aymen Fadhil Abbas, Usman Ullah Sheikh, Mohd Norzali Haji Mohd. (2020). “Recognition of vehicle make and model in low light conditions.” Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 2, April 2020, pp. 550~557, ISSN: 2302-9285. https://doi.org/10.11591/eei.v9i2.1865

Aymen Fadhil Abbas, Usman Ullah Sheikh, Fahad Taha AL-Dhief and Mohd Norzali Haji Mohd. (2021). “A comprehensive review of vehicle detection using computer vision.” TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 19, No. 3, June 2021, pp. 838~850, ISSN: 1693-6930. https://doi.org/10.12928/TELKOMNIKA.v19i3.12880

Mahmuda Akhtar and Sara Moridpour. (2021). “A Review of Traffic Congestion Prediction Using Artificial Intelligence.” Journal of Advanced Transportation, vol. 2021, Article ID 8878011, 18 pages. https://doi.org/10.1155/2021/8878011

Upulie Handalage and Lakshini Kuganandamurthy. (2021). “Real-Time Object Detection using YOLO: A review.” https://doi.org/10.13140/RG.2.2.24367.66723

Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. (2015). “You Only Look Once: Unified, Real-Time Object Detection.” https://doi.org/10.48550/arXiv.1506.02640

Glenn Jocher et al. (2022). “ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.” https://doi.org/10.5281/zenodo.3908559

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai and Bo Ma. (2022). “A Review of Yolo Algorithm Developments.” The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021), Procedia Computer Science 199, pp. 1066-1073. https://doi.org/10.1016/j.procs.2022.01.135

Ricardo Pereira, Guilherme Carvalho, Luís Garrote and Urbano J. Nunes. (2022). "Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics" Applied Sciences 12, no. 3: 1319. https://doi.org/10.3390/app12031319

Mansour Mohamed, Abuelgasim Saadeldin Mansour Mohamed, and Muhammad Mahbubur Rashid. (2022). “Video-Based Vehicle Counting and Analysis Using YOLOv5 and DeepSORT with Deployment on Jetson Nano”. Asian Journal of Electrical and Electronic Engineering 2 (2):11-20. https://journals.alambiblio.com/ojs/index.php/ajoeee/article/view/34

Suruchi Kumari and Deepti Agrawal. (2022). “Video Based Vehicle Detection and Tracking using Image Processing”, International Journal of Research Publication and Reviews, Vol 3, no 8, pp 735-742, August 2022, ISSN 2582-7421.

Tuan Linh Dang, Gia Tuyen Nguyen and Thang Cao. (2020). “Object Tracking Using Improved Deep_Sort_YOLOv3 Architecture” ICIC Express Letters, Volume 14 Number 10, pp. 961-969. https://doi.org/10.24507/icicel.14.10.961

Bo Gao. (2022) “Research on Two-Way Detection of YOLO V5s+Deep Sort Road Vehicles Based on Attention Mechanism”, Journal of Physics: Conference Series, Volume 2303. https://doi.org/10.1088/1742-6596/2303/1/012057

Yunus Egi, Mortaza Hajyzadeh and Engin Eyceyurt. (2022). “Drone-Computer Communication Based Tomato Generative Organ Counting Model Using YOLO V5 and Deep-Sort”, Agriculture 2022. https://doi.org/10.3390/agriculture12091290.