

Retraction

Retracted: Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm

Journal of Advanced Transportation

Received 15 November 2022; Accepted 15 November 2022; Published 29 November 2022

Copyright © 2022 Journal of Advanced Transportation. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Advanced Transportation has retracted the article titled “Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm” [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process, and the article is being retracted with the agreement of the Chief Editor.


The authors do not agree to the retraction.

References

- [1] O. O. Khalifa, M. H. Wajdi, R. A. Saeed, A. H. A. Hashim, M. Z. Ahmed, and E. S. Ali, “Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm,” *Journal of Advanced Transportation*, vol. 2022, Article ID 9189600, 11 pages, 2022.
- [2] L. Ferguson, “Advancing Research Integrity Collaboratively and with Vigour,” 2022, <https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/>.

Research Article

Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm

Othman O. Khalifa ^{1,2}, Muhammad H. Wajdi ², Rashid A. Saeed ³,
Aisha H. A. Hashim ², Muhammed Z. Ahmed ⁴, and Elmustafa Sayed Ali ^{5,6}

¹Libyan Centre for Engineering Research and Information Technology, Bani Waleed, Libya

²Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia

³Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

⁴Department of Computer Engineering, University of Maiduguri, Maiduguri, Nigeria

⁵Department of Electrical and Electronic Engineering, Faculty of Engineering, Red Sea University, Port Sudan, Sudan

⁶Department of Electronic Engineering, Faculty of Engineering, Sudan University of Science and Technology, Khartoum, Sudan

Correspondence should be addressed to Elmustafa Sayed Ali; elmustafasayed@gmail.com

Received 27 January 2022; Revised 15 February 2022; Accepted 22 February 2022; Published 15 March 2022

Academic Editor: Muhammad Arif

Copyright © 2022 Othman O. Khalifa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vehicle detection in Intelligent Transportation Systems (ITS) is a key factor ensuring road safety, as it is necessary for the monitoring of vehicle flow, illegal vehicle type detection, incident detection, and vehicle speed estimation. Despite the growing popularity in research, it remains a challenging problem that must be solved. Hardware-based solutions such as radars and LIDAR are been proposed but are too expensive to be maintained and produce little valuable information to human operators at traffic monitoring systems. Software based solutions using traditional algorithms such as Histogram of Gradients (HOG) and Gaussian Mixed Model (GMM) are computationally slow and not suitable for real-time traffic detection. Therefore, the paper will review and evaluate different vehicle detection methods. In addition, a method of utilizing Convolutional Neural Network (CNN) is used for the detection of vehicles from roadway camera outputs to apply video processing techniques and extract the desired information. Specifically, the paper utilized the YOLOv5s architecture coupled with k-means algorithm to perform anchor box optimization under different illumination levels. Results from the simulated and evaluated algorithm showed that the proposed model was able to achieve a mAP of 97.8 in the daytime dataset and 95.1 in the nighttime dataset.

1. Introduction

Visual surveillance of dynamic objectives such as vehicles has been an active research topic as the current existing method: monitoring traffic conditions using control towers controlled by traffic officers is inefficient [1]. This is due to the growing number of traffic surveillance cameras with respect to the increasing number of highways, which means that it would take an increasing number of manual resources, effort, and time to monitor incoming traffic in the

highways. Hence, new methods are required in order to monitor traffic conditions [2].

The rise of technological advancements and high-speed Internet have propelled the need for a more advanced detection of vehicles in traffic settings, which is in line with the sustainable development goals set by the United Nations to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation [3]. Other existing solutions that have currently been used for vehicle detection such as radar and supersonic suffer a severe limitation in their ability to be able to measure important

parameters in traffic, which is required to accurately assess the traffic conditions. This occurs, as hardware-based sensors are unable to provide complete traffic scene information such as vehicle classification, tracking of vehicles, and detection of accidents, monitoring traffic violations, number plate recognition, and more [4].

According to what was mentioned, the aim of this study is to enhance the performance of YOLOv5s architecture by coupling them with k-means algorithm for vehicles detection under different illumination levels. The rest of the paper is organized as follows. In Section 2, different related studies are discussed and summarized. The study methodologies are reviewed in Section 3, which illustrates the methods of data collection, the Convolutional Neural Network (CNN), and the model train. In Section 4, results and discussion are provided, and the paper is concluded in Section 5.

2. Related Works

Within the past few years, the focus on vehicle detection has evolved into trying to improve the rate of vehicle detection through factoring vehicle similarities, illumination changes, and complex environments, pose variations, vehicle occlusion, vehicle variability, camera placement, and different resolutions. The studies detailed in this section are summarized in Table 1, and they illustrate the efforts made by various researchers to produce a model that is used to detect vehicles in real time in traffic cameras.

The research presented by Sang et al. utilized a YOLOv2 model for vehicle detection. To cluster the vehicle bounding boxes in training datasets, the k-means algorithm was proposed and coupled with six differently sized anchor boxes [3]. Their method also applied normalization to improve the detection of the bounding boxes with different aspect ratios. A multilayer feature fusion strategy was opted to improve the feature extraction ability of the network [4]. This was coupled with the removal of repeated convolutional layers in high layers. This proposed model was able to deal with 26 pictures in 1 second and was able to detect vehicles irrespective of the time of day (day or night) and strong weather adaptability and has a high detection rate of vehicles with different aspect ratios [5]. This method however was not able to perform well in datasets that the model is not trained in suggesting that it requires more data to train the model. The authors also did not test the model under heavy occlusion settings [6–8].

Another study presented by Li et al. provides a YOLO-vocRV model for vehicular detection application, which enables detecting multiple targets of different traffic densities [9]. Through the study evaluation, the authors recognize that the proposed model gives suitable detection rate; however, it gives high false detection rate especially in low training dataset [10]. Sheng et al. provide the concept of using R-CNN model to increase the dataset of traffic detection datasets [11]. Author evaluates the model in detecting vehicles based on different angles and multiple scenes [8]. The results show that the vehicle detection rate increased with a big training dataset. The model has shortcoming of inability to identify the vehicles in fog and snow environments. In

study proposed by Chen et al., authors use the k-means algorithm with Image Net dataset and VGG-16 to design a fully convolutional detection architecture [5]. The model enables detecting vehicles according to different scales and different appearances in a heavy traffic state [4]. It gives high detection rate; however the performance of detection may be degraded in fuzzy environment.

In the study presented by Xu et al., researchers have introduced an improved YOLOv3 model that can detect compounds with higher accuracy [12]. The method of increasing the depth of the network is used to improve the suitability of the network and improve the mechanism of calling maps with higher-level features to increase obtaining detailed data that helps in discovery. Sun et al. proposed an optical flow with detection algorithm based on color space to detect the objects in shadow. The model enables detecting in daytime with high shadow and gives high accuracy [7, 13]. However, the model required long time to make frame removal computations. In addition, it gives low accuracy when tested in nighttime settings [14].

The researchers Bin Zuraimi and Kamaru Zaman demonstrated the possibility of improving the YOLOv4 algorithm to increase the accuracy of vehicle detection systems [15], especially when the number of vehicles increases, which needs high accuracy and fast detection systems and helps in detecting traffic congestion [16]. The researchers used deep learning technology to help detect objects in real time to discover compounds. The detection model is built using the Deep SORT algorithm, which works to calculate the number of vehicles in front of the monitoring camera with high efficiency. Through the analysis, the proposed model gave an increase in detection accuracy of up to 82%.

In study proposed by Alawi et al., authors present the problems facing vehicle detection systems from aerial images using neural networks such as Faster R-CNN. It is sometimes difficult for the comparison between vehicles and objects to distinguish between them [17, 18]. The researchers studied the capabilities of a neural network algorithm in addition to YOLOv3, YOLOv4, and their performance in detection application. These algorithms are analyzed with a number of different factors such as the accuracy of the camera, the size of the object, and the height of the imaging from the ground with number of 52 training experiments. The studies gave results that both YOLOv4 and YOLOv3 give the best performance compared to Faster R-CNN.

3. The Vehicles Detection Methodology

This section discusses the methodologies employed to build the proposed model framework for image-processing tool developed for capturing images in real-time vehicles detection. The image capturing mechanism is set up to capture live traffic data and provide the information to the data collection stage. The data preparation and assumptions are used to acquire the best quality of data from live camera. The YOLOv5s program is used to identify the moving vehicles on the road. The following

TABLE 1: Summary of related research.

Author	Method	Strengths	Limitations
A. Ammar et al. (2021)	Compare Faster R-CNN and YOLOv3, YOLOv4 with UAV imaging datasets	(i) YOLOv3, YOLOv4 gave higher accuracy than Faster R-CNN	(i) Possible low detecting accuracy with high AUV speed
Bin Zuraimi et al. (2021)	Improved YOLOv4 algorithm	(i) Increase the accuracy of vehicle detection systems	(i) Weak performance in case of dense traffic (i) Longer computational time than frame removal method
W. Sun et al. (2020)	Optical flow with shadow detection algorithm based on the HSV color space	(i) High accuracy in high shadow settings in daytime	(ii) Unable to deal with nighttime settings (iii) Not tested with vehicles in different scales
B. Xu al. (2019)	Improved YOLOv3	(i) Improve the mechanism of calling maps to vehicle detection (ii) High accuracy	(i) Weak vehicle detection in complex traffic environment
L. Chen et al. (2018)	VGG-16 with Image Net dataset, k-means algorithm, feature fusion techniques, fully convolutional architecture	(i) Increased the detection rate of the vehicles with respect to (a) different scales (b) vehicles with different appearances and with heavy vehicle occlusion	(i) Not tested in change within illumination environments
M. Sheng et al. (2018)	Tested the efficiency of R-CNN & Faster R-CNN model through the increase of training dataset	(i) Able to detect and recognize vehicles from different angles and multiple scenes (ii) Increase in training datasets results in an increase in vehicle detection rate	(i) Slow detection rate (ii) Low accuracy rate in vehicle detection (iii) Unable to identify vehicles under extreme conditions such as fog and snow
X. Li et al. (2018)	YOLO-vocRV	(i) Suitable for the multiple target detection of different traffic densities for vehicles	(i) False detection rate is high (ii) Requires large training dataset (iii) Not tested in low-light settings
J. Sang et al. (2018)	YOLOv2 model for vehicle detection	(i) Improve the detection of the bounding boxes	(i) Not suitable for datasets that the model is not trained in

subsections will provide a brief concept about data collection, the implementation of CNN algorithm, model training, and YOLOv5s program.

3.1. Data Collection. For this work, the location that is chosen is the Duke Highway at Taman Selasih, Gombak, and Selangor. This area is chosen due to a few reasons [6].

- (i) Ease of access for data collection and placing camera not interrupted by any occlusion from any objects such as billboards or trees
- (ii) The traffic was bidirectional, which allowed for greater coverage
- (iii) There is adequate lighting for nighttime settings to provide a good reference to traffic illumination
- (iv) The location of the camera is not blinded by the sunlight

The camera was placed directly across the bidirectional highway using a tripod and located directly for centering purposes (see Figure 1). The recordings will take approximately 20-minute duration for each scenario. The conditions are divided in two scenarios: afternoon (high level of illumination) and evening (low level of illumination) at a fixed weather state (see Figure 2). This variation is intended to find the effect of illumination on the system.

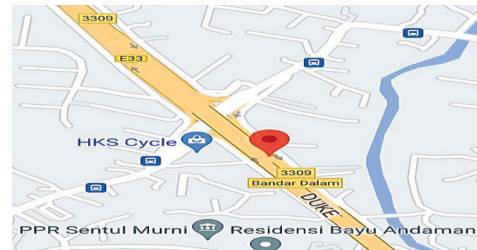


FIGURE 1: Data collection location (Duke Highway at Taman Selasih, Gombak, Selangor).

A target of 750 images for each dataset is set. Both datasets collected randomly are split into 70% for training, 20% for validation, and 10% for testing. Further, since manual annotations of the dataset are time consuming and expensive, the presented study utilizes data augmentation while training the network. Figure 3 shows the examples of data augmentation used for training.

Data augmentation also helps to avoid overfitting while in this process. Specifically, the flipping of the image vertically and horizontally is adopted in order to increase the dataset collected. Tables 2 and 3 show how the data was split based on data augmentation and objects, respectively. The detailed implementation for the data collection and preparation section is described in Figure 4.

Hence, the flow in Figure 4 is to ensure that the image data will be able to help assist with the model's

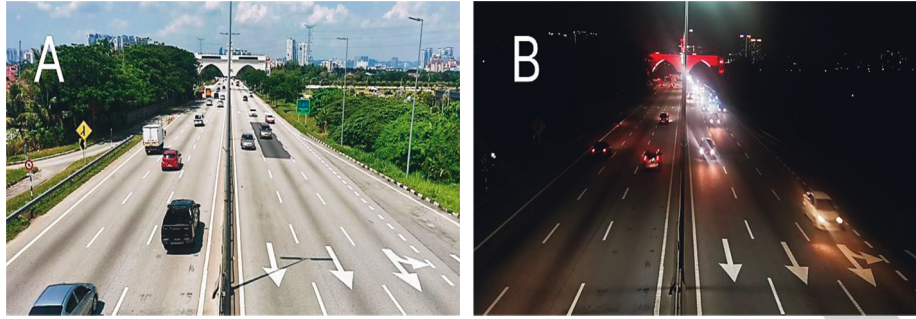


FIGURE 2: Examples of images from the camera taken at daytime (a) and nighttime (b) without weather interruptions.

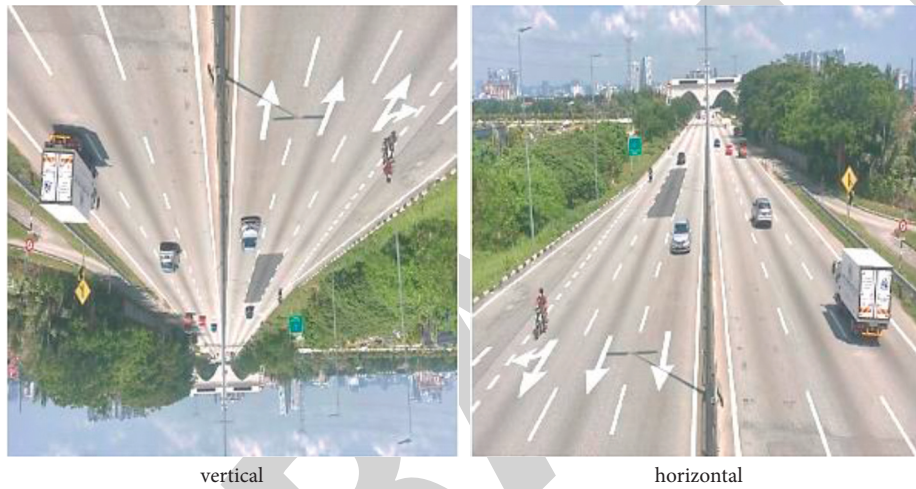


FIGURE 3: Examples of data augmentation used for training.

TABLE 2: Split data into training, validation, and testing for daytime (A) and nighttime (B).

Dataset	Augmentation	Number of images
Training A	Yes	1200
Training B	No	1200
Validation A	No	165
Validation B	Yes	150
Test A	No	73
Test B	No	75

TABLE 3: Split of data into training, validation, and test based on objects.

Labelled objects	Dataset	Number of images
Car	Night time (B)	4406
Car	Day time (A)	3767
Motorcycle	Night time (B)	932
Motorcycle	Day time (A)	1103

performance in detection rate of vehicles. To establish standardized image data sets for the vehicle detection, each image dataset would be required to follow the syntax format set by YOLO. Each image is associated with a text file with the same name, contains object classes, and coordinates as follows: <object-class> <x_center> <y_center> <width> <height>.

Three files were created: classes.name, train.txt, and test.txt. Similar to the image dataset, the names of the objects are also required to follow the convention set by YOLO: objectn_name. The train.txt and test.txt file will contain the path file to the training and testing images that will be used.

3.2. Implementation of Convolutional Neural Network (CNN). The framework for this project such as training method, YOLOv5 implementation, and testing the performance of the model is discussed. In this study, k-means clustering is applied to the training dataset to perform clustering analysis on the size and scale of the vehicle bounding boxes [7]. Traditionally, the algorithm used for object detection used a sliding window in order to generate a candidate proposal [9]. This generation method is time intensive.

The candidate proposals produced by Faster R-CNN and SSD is less than the sliding window as it uses aspect ratios [0.5, 1, 2] which means that the aspect ratio is not optimized to be used for specific object detection application such as vehicle detection [8]. By utilizing k-means cluttering, suitable number and size for anchor boxes can be obtained and selected to reduce the time consumption and improve the positioning accuracy.

K-means cluttering is a method of vector quantization that is also a popular cluster analysis. It is used to classify objects into its attributes or features into K number of clusters. Here, K

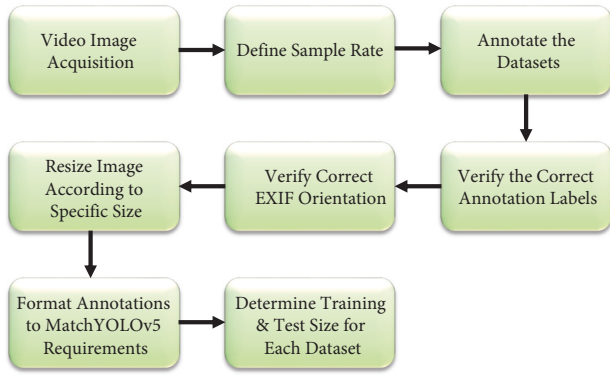


FIGURE 4: Overview of the process and implementation of data collection.

represents a positive integer value [10, 14]. K-means begins with the selection of a single centroid at random [19]. The cluttering method can be formulated with the equation below.

$$d_i = \sum_{i=1}^k \sum_{x \in C_i} x - \mu_{i2}^2, \quad (1)$$

where x represents the sample, k represents the centroid, and μ_i represents the average vector of C_i .

The probability of choosing a centroid will be directly proportional to the nearest distance. The k-mean is used on various k on vehicle sizes and aspect ratios to find the best k to increase the mAP [13]. The flowchart to represent k-means cluttering is shown in Figure 5.

In order to implement this method in Python, the following codes shown in Algorithm 1 are used. The function shown below is for extracting features from the images. In addition, the tensor flow methods are used to handle the processes at backend.

Then process fetches the image features and names using the function in Algorithm 2. This then is followed by making the k-means clustering model and training it using the features, which were extracted from the images. $K=7$ in which there is an improvement of about 5% in accuracy for the mAP of the vehicle dataset.

3.2.1. Model Training. One of the most important parts of this methodology is the training of the Convolutional Neural Network (CNN). In order to ensure the effectiveness of the training, the following process has been adopted.

The model architecture as specified earlier is the YOLOv5 (see Figure 6). Once the labelling of the image dataset along with the classes has been completed, the model configuration needs to be set and determined when performing the training. The model configurations are discussed in the following sections.

3.2.2. YOLOv5 Program. YOLOv5 was released a month after YOLOv4 and implemented in PyTorch which makes it easier for implementation for IoT devices like speed cameras. Being built with a CSP network as the backbone and PANet as its neck makes it an attractive choice to be used for vehicle

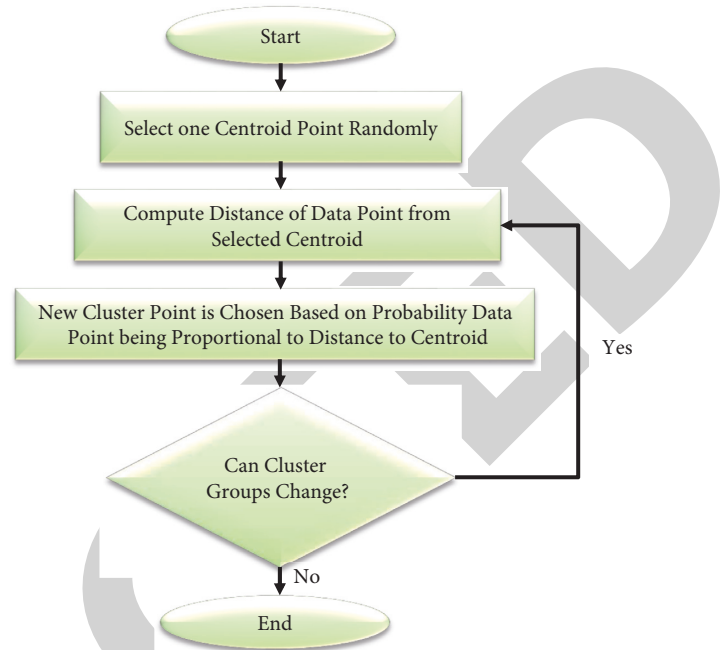


FIGURE 5: Flow of k-means clustering in object detection.

detection in real time. YOLOv5s, which is the smallest variation of the YOLOv5 family, is chosen, as it is the most lightweight. In order to implement this, the following process in Figure 7 is used.

One of the requirements for YOLO is that the images must be in ratios of 32. Hence, here the image for datasets will be set as 416×416 pixels. Once the environment for the YOLOv5 is configured, the pretrained weights and custom datasets, which are defined in the earlier sections, will be imported. The structure of the YOLOv5 such as max_batches, batch, divisions, width, and height will be set and changed according to the performance to find the optimal configurations for this setup. To implement YOLOv5s, a notebook developed by Roboflow Ai was utilized [16]. The details of the architecture for the YOLOv5 used are shown in Figures 8 and 9.

4. Results and Discussion

This section focuses solely on displaying and interpreting the results obtained using the proposed methodology. The results are classified into sections: A) reviews the performance of the system with respect to the metrics discussed earlier and, B) represents the comparison with the benchmark paper. Further, the limitations of this system are discussed.

4.1. Performance of Proposed System. In this part of the discussion, variables such as mAP, IoU, recall, and precision are tested and measured [20]. The performance in the proposed model for the training and testing period is seen in the graph shown in Figures 10 and 11. This figure shows the different performance metrics across training and validation sets for the two different datasets [21].

```

(1) Def image feature (direc):
(2) Model = InceptionV3 (weights = "imagenet," include_top = false)
(3) features = [];
(4) img_name = [];
(5) For i in tqdm (direc);
(6) fname = 1;
(7) img = image.load_img (fname, target_size = (224, 224))
(8) x = img_to_array (img)
(9) x = np.expand_dims (x, axis = 0)
(10) x = preprocess_input (x)
(11) feat = model.predict (x)
(12) feat = feat.flatten ()
(13) features.append (feat)
(14) Img_name.append (i)
(15) Return features, img_name

```

ALGORITHM 1: Running tensor flow.

```

(1) img_features, img_name = Img_features, img_name (all)
(2) len (img_features)
(3) img_features
(4) k = 7;
(5) Clusters = KMeans (k, random_state = 40)
(6) Clusters.fit (Img_features)

```

ALGORITHM 2: Running the K-means clustering model.

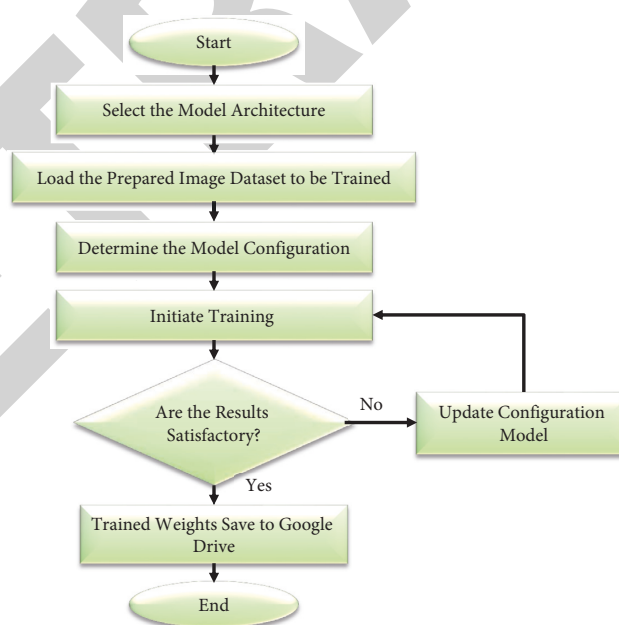


FIGURE 6: Process for training the YOLOv5.

From the above, there are three different types of losses that are shown as objectness loss, box loss, and classification loss. Objectness loss refers to the probability that an object is within the proposed region of interest [15]. The higher the objectivity, the higher the probability that the image window contains the object. Box loss refers

to how effective the algorithm is in locating the center of an object and how effective the prediction of the bounding box covering an object is. Finally, the classification loss refers to the measure of the algorithm's effectiveness in correctly predicting the class of a given object [12, 18].

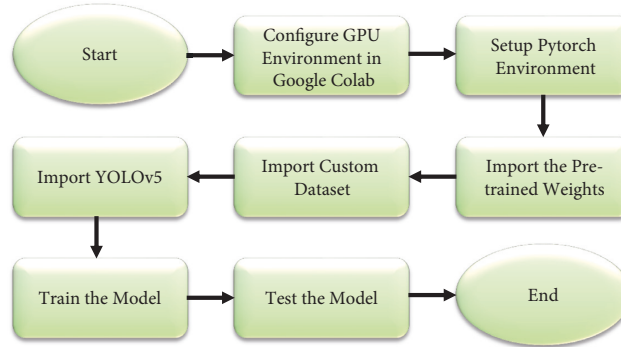


FIGURE 7: Process of implementing YOLOv5.

```

# parameters
nc: {num_classes} # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, BottleneckCSP, [128]], # 2-P2/4
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, BottleneckCSP, [256]], # 4-P3/8
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, BottleneckCSP, [512]], # 6-P4/16
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]], # 8-P5/32
  [-1, 3, BottleneckCSP, [1024, False]], # 9-P5/32
  ]
  
```

FIGURE 8: Parameters, anchors, and backbone used for YOLOv5 model.

```

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, BottleneckCSP, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, BottleneckCSP, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, BottleneckCSP, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, BottleneckCSP, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]
  
```

FIGURE 9: Displaying the information of the head of the YOLOv5 architecture used.

For the daytime dataset (A), the model improved significantly and swiftly concerning precision, recall, and mAP before remaining stagnant after about 140 epochs. The

objectness, box, and classification losses also show a rapid decline at the same epoch level. The nighttime dataset (B) also showed similar results to (A) which indicates that the performance of the system across different illumination settings is stable and similar [22].

Once the training was completed, the model was tested with complete new sets of images it has not seen before in the test set [23]. Figures 12 and 13 show that the model is able to detect cars and motorcycles to a high degree of certainty without mistaking it for other vehicles that have not been labelled in the datasets such as Lorries and trucks [24]. However, as the two classes move further and appear smaller, that certainty of prediction decreases significantly indicating that the model struggles to correctly differentiate vehicles with large scale of variance (see Figure 14) [24].

For the nighttime dataset (B), despite the presence of low light, it was still able to considerably detect the cars and motorcycles classes well. Much like the daytime dataset (A), the nighttime dataset was also able to avoid miss-classing vehicles such as trucks and accurately detect the predefined classes as shown in Figure 15. Further, one of the biggest differences between the daytime (A) and nighttime (B) dataset is that, in the latter, there is a presence of glaring illumination from the headlights of the vehicle as shown in Figure 16. The proposed algorithm was still able to detect it with high degree of accuracy despite that difference. However, as seen in Figures 17 and 18, when the vehicles are present at a darker area with lower level of illumination, which is present at the left-hand side of the image, the algorithm is not able to detect the moving vehicle.

4.2. The Performance Comparison. To evaluate the effectiveness of the proposed network, the benchmark paper [25], which utilized YOLOv4 coupled with optimization of the bounding box prediction using k-means algorithm as well as the baseline YOLOv5 without k-means clustering (baseline), was used for comparison. The results contrasted to the proposed solution and benchmark paper are as shown in Tables 4 and 5.

Based on the tables above, despite the difference in the datasets between the benchmark paper and the proposed solution, the proposed solution can consistently detect vehicles of varying sizes a lot better than the benchmark paper. The reason for this is possibly since the model trained on a

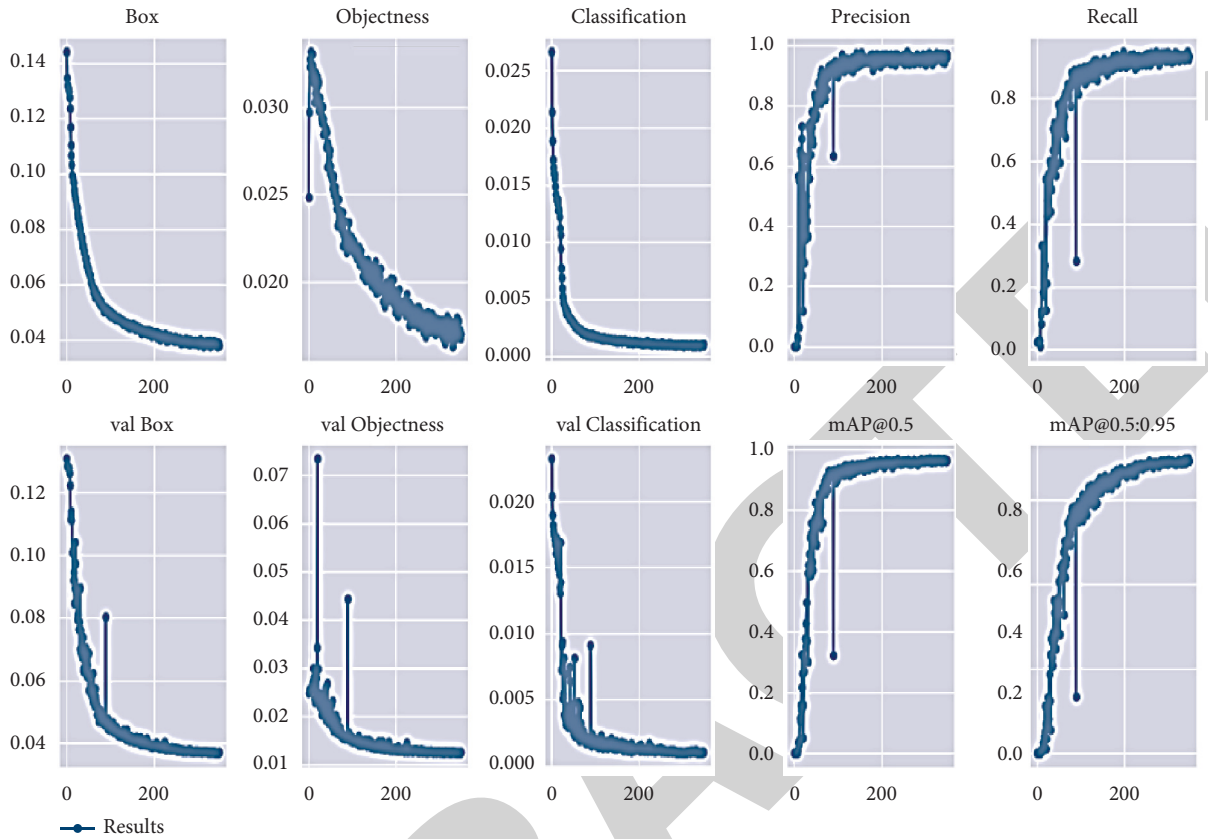


FIGURE 10: Plots of box loss, objectness loss, classification loss, precision, recall, and mean average precision (mAP) for daytime dataset (A) for training and validation.

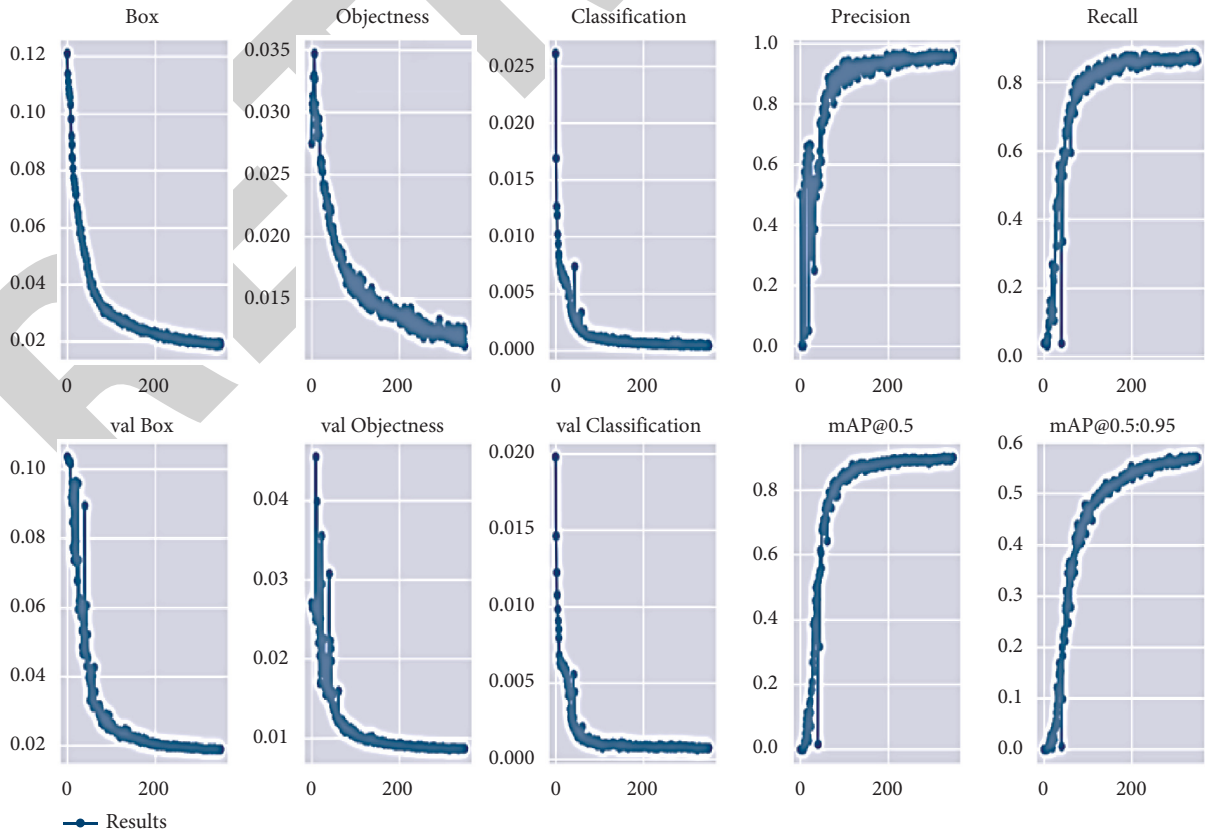


FIGURE 11: Atlas of graphs of geared kinematic chains.

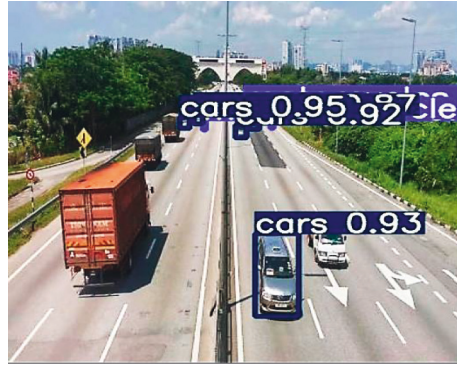


FIGURE 12: Model accurately detecting the predefined classes in daytime dataset (A).



FIGURE 13: Model detecting motorcycle class in daytime dataset (A).



FIGURE 16: Presence of illumination from headlight.

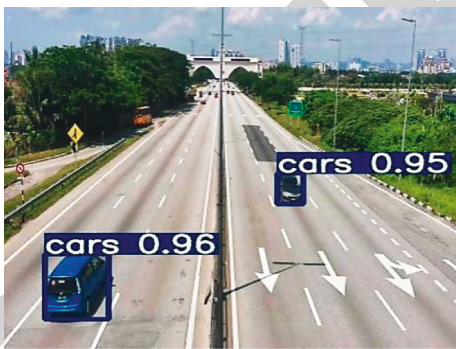


FIGURE 14: Model not being able to detect vehicles at small scale in daytime dataset (A).



FIGURE 17: Model was not able to detect the car in the nonlighted area.

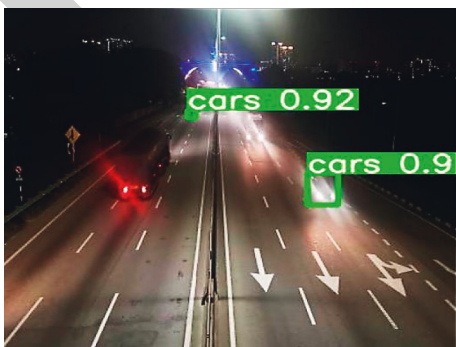


FIGURE 15: Model is able to detect it accurately in nighttime dataset (B).



FIGURE 18: Another example of model not detecting the car in the nonlighted area.

TABLE 4: Results comparison of proposed, baseline, and benchmark paper.

Benchmark		Proposed solution		YOLOv5 without k-means	
Class	Average precision	Class	Average precision	Class	Average precision
Car	77.16	Car (A)	97.9	Car (A)	93.7
Other	31.28	Motorcycle (A)	97.7	Motorcycle (A)	90.9
Van	49.71	Car (B)	95.9	Car (B)	91.6
Bus	83.63	Motorcycle (B)	94.4	Motorcycle (B)	87.2

TABLE 5: mAP results for both solutions.

	Benchmark	Proposed		Baseline	
		A	B	A	B
mAP	61.725	97.8	95.1	92.3	89.4

dataset has smaller sized vehicles, when compared to the dataset size of the benchmark, it is a lot smaller. Further, the presence of the k-means algorithm was significant in the proposed solution as it was able to achieve a mAP increase by 5.62% in the daytime dataset (A) and increase by 5.99% for the nighttime dataset. This proves that, through optimizing the anchor box selection in object detection models such as YOLOv5, the detection rate can be increased.

The proposed solution also performs slightly worse in low level of illuminations. A few possible reasons for this could include that, during nighttime, the colors present are significantly lower so it is harder for the models to extract features, as the surrounding shades are almost identical to one another. This can be seen when the model was unable to detect vehicles that were in the region where no streetlights were protruding. Next, the glaring illumination from the headlights also plays a factor as it increases the brightness of the image, which makes the camera and the model unable to see the actual shape of the car.

5. Conclusion

Different vehicle detection methods have been reviewed, analyzed, and evaluated based on the respective strengths and weaknesses of those methods in chapter two. From this, a new implementation based Convolutional Neural Network (CNN) was proposed to study its effectiveness in traffic parameters, specifically under illumination variance. The vehicle detection was implemented through the YOLOv5s architecture, which was coupled with k-means to optimize the anchor boxes. The performance of the system such as the accuracy, IoU, and recall under different traffic conditions is measured. In order to study the effectiveness of this proposed method, it is compared with works done on the research area as well as the baseline YOLOv5s model.

Data Availability

The datasets/codes generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Disclosure

The paper reflects the authors' views on this research.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' Contributions

All authors contributed equally to this study.

References

- [1] G. Wang, D. Xiao, and J. Gu, "Review on vehicle detection based on video for traffic surveillance," in *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008*, pp. 2961–2966, Qingdao, China, September 2008.
- [2] E. S. Ali, M. K. Hasan, R. Hassan et al., "Machine learning technologies for secure vehicular communication in internet of vehicles: recent advances and applications," *Security and Communication Networks*, vol. 2021, Article ID 8868355, 23 pages, 2021.
- [3] J. Sang, Z. Wu, P. Guo et al., "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, pp. 4272–12, 2018.
- [4] Eltahir, A. Ahmed, and R. A. Saeed, "V2V communication protocols in cloud-assisted vehicular networks," in *Vehicular Cloud Computing for Traffic Management and Systems*, J. Grover and P. Vinod, Eds., IGI Global, Hershey, PA, USA, 2018.
- [5] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," *Eurasip Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–7, 2018.
- [6] N. M. Elfatih, M. K. Hasan, Z. Kamal et al., "Internet of vehicle's resource management in 5G networks using AI technologies: current status and trends," *IET Communications*, pp. 1–21, 2021.
- [7] G. Giuffrida, G. Meoni, and L. Fanucci, "A YOLOv2 convolutional neural network-based Human-Machine Interface for the control of assistive robotic manipulators," *Applied Sciences*, vol. 9, no. 11, pp. 2243–11, 2019.
- [8] S. E. Ibrahim, R. A. A. E. Saeed, and A. Mukherjee, "Resource management in vehicular cloud computing," in *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing*, Management Association and Information Resources, Eds., IGI Global, Hershey, PA, pp. 2448–2470, 2021.
- [9] X. Li, Y. Liu, Z. Zhao, Y. Zhang, and L. He, "A deep learning approach of vehicle multitarget detection from traffic video," *Journal of Advanced Transportation*, vol. 2018, Article ID 7075814, 11 pages, 2018.

- [10] K. Adi, A. P. Widodo, C. E. Widodo, A. Pamungkas, and A. B. Putranto, "Automatic vehicle counting using background subtraction method on gray scale images and morphology operation," *Journal of Physics: Conference Series*, vol. 1025, no. 1, pp. 012025-012026, 2018.
- [11] M. Sheng, C. Liu, Q. Zhang, L. Lou, and Y. Zheng, "Vehicle detection and classification using convolutional neural networks," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS 2018*, pp. 581-587, Enshi, China, May 2018.
- [12] B. Xu, B. Wang, and Y. Gu, "Vehicle detection in aerial images using modified YOLO," in *Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 1669-1672, Xi'an, China, October 2019.
- [13] W. Sun, M. Sun, X. Zhang, M. Li, and D. Wanchun, "Moving vehicle detection and tracking based on optical flow method and immune particle filter under complex transportation environments," *Complexity*, vol. 2020, Article ID 3805320, 15 pages, 2020.
- [14] Z. Ahmed, R. A. Saeed, and A. Mukherjee, "Vehicular Cloud Computing models, architectures, applications, challenges and opportunities," in *The Book with Title Vehicular Cloud Computing for Traffic Management and Systems*, J. Grover and P. Vinod, Eds., IGI Global, USA, IGI Global, June 2018, ISBN13: 9781522539810, ISBN10: 1522539816.
- [15] M. A. Bin Zuraimi . and F. H. Kamaru Zaman, "Vehicle detection and tracking using YOLO and DeepSORT," in *Proceedings of the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 23-29, Penang, Malaysia, April 2021.
- [16] M. Abdelgadir, R. A. Saeed, and A. Babiker, "Mobility routing model for vehicular ad-hoc networks (VANETs), smart city scenarios," *Vehicular Communications*, vol. 9, pp. 154-161, July 2017.
- [17] M. Alawi, R. Saeed, A. Hassan, and O. Khalifa, "Internet access challenges and solutions for vehicular ad-hoc network environment," in *Proceedings of the IEEE International Conference on Computer & Communication Engineering (ICCCCE2012)*, pp. 3-5, Malaysia, July 2012.
- [18] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, "Vehicle detection from aerial images using deep learning: a comparative study," *Electronics*, vol. 10, pp. 1-31, 2021.
- [19] A. E. Amal, R. A. saeed RA, A. mukherjee, and M. K. Hasan, "Evaluation and analysis of an enhanced hybrid wireless mesh protocol for vehicular ad-hoc network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1-11, Article ID 169, 2016.
- [20] M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, and P. E. Kummervold, "Short communication: detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5," *Algorithms*, vol. 14, no. 4, p. 114, 2021.
- [21] J. Chen, G. Ding, Y. Yang et al., "Dual-modality vehicle anomaly detection via bilateral trajectory tracing," *IProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4011-4020, Nashville, TN, USA, June 2021.
- [22] K. E. B. Ahmed, R. A. Mokhtar, and R. A. Saeed, "A new method for fast image Histogram calculation," in *Proceedings of the International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICNNEE)*, pp. 187-192, Khartoum, Sudan, September 2015.
- [23] L. E. Alatabani, E. S. Ali, and R. A. Saeed, "Deep learning approaches for IoV applications and services," in *Intelligent Technologies for Internet of Vehicles. Internet of Things (Technology, Communications and Computing)*, N. Magaia, G. Mastorakis, C. Mavromoustakis, E. Pallis, and E. K. Markakis, Eds., Springer, Cham, 2021.
- [24] E. S. Ali, M. B. Hassan, and R. A. Saeed, "Machine learning technologies on internet of vehicles," in *Intelligent Technologies for Internet of Vehicles. Internet of Things (Technology, Communications, and Computing)*, N. Magaia, G. Mastorakis, C. Mavromoustakis, E. Pallis, and E. K. Markakis, Eds., Springer, Cham, 2021.
- [25] Q. Wang, "Traffic lights detection and recognition method based on the improved YOLOv4 algorithm," *Sensors*, vol. 22, p. 200, 2022.