Computers, Materials & Continua
DOI: 10.32604/cmc.2023.029999
Article

Robust Vehicle Detection Based on Improved You Look Only Once

Sunil Kumar[1], Manisha Jailia[1], Sudeep Varshney[2], Nitish Pathak[3], Shabana Urooj[4,*] and Nouf Abd Elmunim[4]

[1]Department of Computer Science, Banasthali Vidyapith, Rajasthan, 304022, India
[2]Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Greater Noida, 201310, India
[3]Department of Information Technology, Bhagwan Parshuram Institute of Technology (BPIT), GGSIPU, New Delhi, 110078, India
[4]Department of Electrical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University, P.O. BOX 84428, Riyadh, 11671, Saudi Arabia
*Corresponding Author: Shabana Urooj. Email: SMUrooj@pnu.edu.sa
Received: 16 March 2022; Accepted: 27 April 2022

Abstract: Vehicle detection is still challenging for intelligent transportation systems (ITS) to achieve satisfactory performance. The existing methods based on one stage and two-stage have intrinsic weakness in obtaining high vehicle detection performance. Due to advancements in detection technology, deep learning-based methods for vehicle detection have become more popular because of their higher detection accuracy and speed than the existing algorithms. This paper presents a robust vehicle detection technique based on Improved You Look Only Once (RVD-YOLOv5) to enhance vehicle detection accuracy. The proposed method works in three phases; in the first phase, the K-means algorithm performs data clustering on datasets to generate the classes of the objects. Subsequently, in the second phase, the YOLOv5 is applied to create the bounding box, and the Non-Maximum Suppression (NMS) technique is used to eliminate the overlapping of the bounding boxes of the vehicle. Then, the loss function CIoU is employed to obtain the accurate regression bounding box of the vehicle in the third phase. The simulation results show that the proposed method achieves better results when compared with other state-of-art techniques, namely Lightweight Dilated Convolutional Neural Network (LD-CNN), Single Shot Detector (SSD), YOLOv3 and YOLOv4 on the performance metric like precision, recall, mAP and F1-Score. The simulation and analysis are carried out on PASCAL VOC 2007, 2012 and MS COCO 2017 datasets to obtain better performance for vehicle detection. Finally, the RVD-YOLOv5 obtains the results with an mAP of 98.6% and Precision, Recall, and F1-Score are 98%, 96.2% and 97.09%, respectively.

Keywords: Image-processing; K-means clustering; CNN; YOLOv5; loss function; deep-learning

1 Introduction

Object detection is a technique based on computer vision that detects the semantic objects instances class of objects [1]. Kim et al. [2] defined object detection as "object detection combining the multi-labelled classification and bounding box regression", where assigning class level and drawing the bounding box for each object refers to image classification and object localization. The rapid growth in vehicles on the road has significantly attracted researchers' attention to traffic safety issues. Therefore, vehicle detection has become an inevitable component of traffic surveillance. Machine learning and deep learning technologies have been used to propose a variety of vehicle detection algorithms. [3]. The methods of vehicle detection are broadly categorized into three types: feature-based, conventional machine learning-based and deep learning-based detection methods. The vehicle detection by feature-based method involves the salient features of the front vehicle, but this method is affected by the different angle views [4]. The feature description operator extracts the features of vehicles by adapting the machine learning-based methods to perform the training of the samples in the conventional machine learning-based method. This method relies upon the prior knowledge of vehicle objects. However, the method based on machine learning is not appropriate for detecting vehicles in different environments [5–7].

Deep learning and computer vision technology have progressed in recent years. The deep learning-based method extracts the features of vehicle objects to perform vehicle detection tasks after classification. There are two types of deep learning-based vehicle detection methods: one-stage detector-based methods and two-stage detector-based methods. The one-stage detector performs localization and classification simultaneously for determining object location and identifying objects. In the case of a two-stage detector, localization and classification are performed sequentially. Therefore, a one-stage detector can detect the object faster than a two-stage detector [8–10]. This paper proposed an improved vehicle detection algorithm, focusing on the one-stage detector based on the deep learning method. The majority of vehicle detection algorithms have the issue of performance in terms of vehicle detection accuracy. Although, the existing algorithms have improved the performance with emerging deep learning technology. The addition of the one-stage detector-based YOLO (You Look Only Once) series version has improved the performance in detection rate [11]. However, these existing algorithms still have room for improvement in the detection rate and performance.

We proposed an efficient and effective vehicle detection based on deep learning to address the shortcomings of the existing techniques described above. The proposed method uses the concepts of data clustering on the datasets by the K-means method for the initial frame of the target and optimization of a loss function. The proposed method uses the YOLOv5 model to generate the feature maps and the bounding box of the vehicle. In the next stage, the overlapping of the bounding box is eliminated by applying the NMS technique. At last, the CIoU (Complete Intersection over Union) loss function is employed for further optimization of an accurate regression bounding box.

1.1 Motivations

The following observations for vehicle detection in this study provide the motivations:

- Recent study has focused on vehicle detection i.e. to identify the vehicle for traffic flow management, road planning, or estimation of air and noise pollution. Hence, our focus is to extract the vehicles for highway surveillance control, management and urban traffic planning.
- When executing on a dataset while selecting a suitable parameter, several vehicle detection techniques are still highly parameter sensitive. As a result, even minor changes in the parameter will have a large impact on vehicle detection.
- Vehicle detection technologies that detect vehicles with high detection rates are limited.

1.2 Contributions

A simple and effective vehicle detection technique (in three stages) has been proposed called RVD-YOLOv5 to calculate precision, recall mAP and F1-Score to measure the performance of the proposed algorithm.

- Stage 1: K-means clustering technique is used to perform the data clustering on MS COCO and VOC PASCAL datasets for better outcome in vehicle detection.
- Stage 2: A YOLOv5 one-stage detector based on deep learning is used to extract the features of vehicles by generating the bounding box corresponding to each vehicle. The NMS method eliminates the overlapped bounding box of the vehicle to further improve the accuracy of proposed RVD-YOLOv5 method.
- Stage 3: The CIoU loss function is employed to further obtain the accurate regression bounding box of the vehicle.

The performance of our proposed method is demonstrated on different datasets and comparisons with state-of-the-art vehicle detection algorithms such as LD-CNN [12], SSD [5], YOLOv3 [13], YOLOv4 [14] and RVD-YOLOv5.

1.3 Roadmap

The remainder of the paper is organized as follows: Section 2 describes a survey of the literature on several existing deep learning-based vehicle detection systems. Section 3 shows framework of a proposed method for vehicle detection using clustering of datasets and optimization in the loss function. Section 4 shows the results of simulation analysis and performance on MS COCO and VOC PASCAL datasets. Finally, Section 5 discusses the conclusion and future work of our research paper.

2 Related Work

In this section, several vehicle detection algorithms based on deep learning have been discussed. The R-CNN and YOLO series are two of the most used object detection algorithms nowadays. The detection rate of the R-CNN series is better compared to the YOLO series in target detection when more precision is required, although its detection speed is slower. As a result, these approaches are not suited for real-time vehicle detection. The YOLO series is preferred over the R-CNN algorithm to solve the speed problem, which employs regression to improve the performance by learning generalized characteristics of the object. The YOLO algorithms [15–18] detect object position and classification using a one-stage neural network. There are various state-of-the-art methods such as R-CNN, Fast R-CNN, Faster R-CNN, LD-CNN, SSD and YOLO.

In object detection, vehicle detection is a commonly discussed problem. Various vehicle detection algorithms have been proposed to detect vehicles as vehicle detection is crucial for traffic management. Furthermore, vehicle detection has been used in a wide variety of applications. Based on deep learning algorithms, vehicle detection techniques can be categorized into two-stage and one-stage. With emerging deep learning and computer vision technology, many detection algorithms have been proposed to detect vehicles based on two-stage and one-stage categories. The approaches based on two-stage category detect vehicles by extracting the region of interest (ROI) and performing bounding box regression and classification of Region of Interest (ROI) images. The method based on one-stage performs localization and classification in the same stage by employing regression ideas for object detection. The result of detection accuracy is high in two-stage based algorithms compared with one-stage based algorithms. However, these algorithms have obtained low real-time performance [19–23].

As a result, one-stage approaches are chosen over two-stage methods, despite having a somewhat lower accuracy but a faster detection speed. The most popular deep learning-based method for the detection of vehicles from the two-stage detector and one-stage detector are LD-CNN [12], SSD [5], YOLOv3 [13] and YOLOv4 [14]. In Section 4, All of these methods are compared with our proposed method, called RVD-YOLOv5. Using the MobileNet architecture to generate the base convolutional layer in Faster R-CNN, Kim et al. [24] have presented an improved Faster R-CNN technique for fast vehicle detection. In this method, the soft NMS algorithm replaced the NMS algorithm for solving the issue of duplicate proposals. Shen et al. [12] developed the LD-CNNs model, which is based on deep convolutional neural network detection. This model enhances detection accuracy while reducing computing costs. One-stage techniques are the best models in object detection networks. These models are incredible at predicting objects fast and accurate. The one-stage detector includes the YOLO series and SSD [5]. Cao et al. [5] has developed a model, called SSD, the improved Single-shot multi-box detector (SSD) was introduced for vehicle detection by including significant improvements in the basic architecture of the SSD model during the weighted mask's network training and advancement to the loss function. This model adopted multitask loss function with positioning and confidence errors. The following formula can express the loss function by Eq. (1).

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{1}$$

where l stands for the detection box, g for the real box, c for the multi-class object's confidence, and N for the number of detected boxes after matching with the real box. Confidence loss is denoted by L_{conf}, whereas position loss is denoted by L_{loc}. The weight coefficient of position loss and confidence loss is denoted by α.

The position loss is derived using the smooth L1 loss between the detection and real boxes. The following formula can be used to determine the position loss by using Eq. (2).

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k}.smooth_{L1}(l_i^m - \hat{g}_j^m) \tag{2}$$

Where Pos denotes the total number of positive samples, x_{ij}^{k} denotes whether the i^{th} detection box's predicted object category k corresponds to the j^{th} actual box's classification label., l_i^m indicates the coordinates of the i^{th} detection box, while g_j^m denotes the coordinates of the j^{th} real box.

The confidence loss function is expressed as follows by Eqs. (3) and (4).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log\left(\hat{c}_i^p\right) - \sum_{i \in Neg} \log(\hat{c}_i^0) \tag{3}$$

$$\hat{c}_i^p = \exp\left(c_i^p\right) / \sum_{p} \exp(c_i^p) \tag{4}$$

The i^{th} detection box predicted the object category p. Where object category is represented by p, x_{ij}^{p} represents the matching of j^{th} real real box with The i^{th} detection box predicted the object category p, $\hat{c}_i^p$ denote the probability that the object category predicted by the i^{th} detection box is p.

Zhao et al. [25] introduced a model that uses YOLOv3 [13] and a modified deep sort with a Kalman filter to predict vehicle position and calculate Mahalanobis, cosine, and Euclidean distances. Bag of Freebies [26] and Bag of Specials are two of YOLOv4's [14] most important features. The backbone of the network is CSPDarknet-53 [15], with the Spatial Attention Module (SAM) [15], Path Aggregation Network (PAN) [27], and Cross-iteration Batch Normalisation (CBN) [28] being used. SAM, PAN, and CBN were employed with minor adjustments, and mosaic augmentation was used

for augmentation. CutMix [29], DropBlock regularisation [30], class level smoothing, completing IoU (CIoU) loss [31], Self-Adversarial Training (SAT), and multiple anchors for single ground are some of the improvements made without compromising inference time during training. It is comprised of a CSB block with a focus layer, convolution, batch normalisation, SiLU [32], and a network of C3 blocks in the case of YOLOv5. The focus layer operates as a space-to-depth transformation. This server lowers the cost of 2D convolution, lowers the spatial resolution, and increases the number of channels. Zhang et al. [15] have used confidence's balanced weight by selecting a loss function. The mean square error calculates loss function in the training stage. In addition, the mean square error helps in calculating the candidate box. Therefore, the square root is to be calculated to weaken the weight of the boxes by setting the size, scale and target type for every box. The mean square error formula can be expressed as the following by using Eq. (5).

$$MSE = \frac{\sum_{i=1}^{n} \left(y_i - y_i^p\right)^2}{n} \tag{5}$$

The Eq. (6) represents the loss function as follows:

$$Loss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{i,j}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(\left(y_i - \hat{y}_i\right)^2\right) \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{i,j}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_{i_i}}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_{i_i}}\right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{i,j}^{obj} \left(c_i - \hat{c}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{i,j}^{obj} \left(c_i - \hat{c}_i\right)^2 + C \tag{6}$$

$$Loss = [\lambda_{coord} * Coordinate\ prediction\ error + (Box\ confidence\ prediction\ error\ with\ target$$
$$+ \lambda_{noobj} * Box\ confidence\ prediction\ error\ without\ target) + Classification\ error]$$

The loss function aims to balance the coordinates (x, y, w, h), confidence, and error of classification. During the training step, we simply want to establish a single correlation between the bounding box and the target. hence, the IOU of the bounding box and the ground truth are determined [16]. YOLOv5 is the fifth generation of the YOLO target detection network. It is built on YOLOv3 and YOLOv4 and is the result of ongoing integration and innovation. Second, YOLOv5 has obtained significant detection results on PASCAL VOC and COCO datasets; thus, this paper employs the YOLOv5 detection network to generate bounding box for objects [33]. According to the official, the YOLOv5 comes in four different versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The three versions of this model, YOLOv5m, YOLOv5l, and YOLOv5x, are the outcome of the YOLOv5s model being developed further. The input, backbone, neck, and prediction elements of the YOLOv5 network are subdivided into four sections. The input component of the YOLOv5 performs data improvement, the backbone employs the focus structure and CSP structure, the neck uses the Feature Pyramid Networks (FPN) and Path Aggregation Network (PAN) structure, and the prediction section uses the CIOU Loss and GIOU Loss functions in the target object detection.

3 Methodology

Our proposed vehicle detection framework is described in this section in Fig. 1. First, in Section 3.1, we introduced the K-means technique for data preparation. Then, in Section 3.2, we show how to perform bounding box clustering. Then, in Section 3.3, YOLOv5 is used to perform a feature concatenate to integrate high-level and low-level feature maps and how to create candidate

anchor boxes on various feature maps. Then, Section 3.4 applied Non-Maximum Suppression (NMS) technique to eliminate the overlapping of the bounding box in the detected image. Finally, Section 3.5 uses the CIoU loss function to further obtain the accurate regression bounding box of the vehicle. The proposed framework using YOLOv5 for vehicle detection takes captured video or a set of input images and detects the vehicle by generating a high-efficiency bounding box. The vehicle detection algorithm steps are shown in Fig. 1.

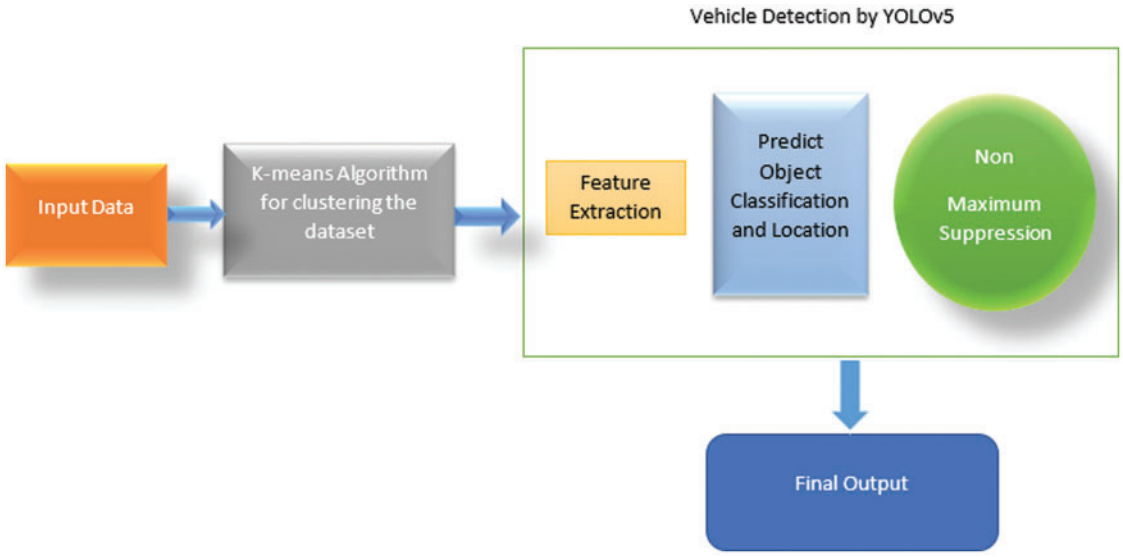


Figure 1: Block diagram of proposed vehicle detection algorithm

The K-means clustering algorithm is used to perform the data clustering during the training stage. After that, calculations for bounding box clustering and anchor boxes are obtained. Next, the backbone network generates the extracted feature maps by using the focus structure, which is used to perform convolution and slicing operations. The NMS technique is then used to eliminate bounding box overlapping in the next stage.

3.1 K-Means Clustering

K-means Clustering groups the unlabelled dataset into different clusters using an Unsupervised Learning Algorithm. The number of pre-defined clusters is denoted by K. The algorithm divides the unlabelled dataset into K groups, with each dataset belonging to a single group with similar attributes. The K-means algorithm is implemented by performing the three stages listed below until convergence is achieved: Determine the coordinates of the centroid, compute the distance between each data feature and the centroids, to get the closest centroid by grouping the data based on the nearest distance.

The Fig. 2 shows the data clustering before K-means and after K-means. The data coordinates are distributed in the second and third steps using the centroid and nearest neighbours. The cluster $CL_K(m)$ uses the data coordinates C_j is shown by Eq. (7) and (8).

$$|C_j - M_K(m)| < |C_j - M_C(m)|, \text{Where } \forall C = \{1, 2, \ldots K \; and \; C \neq K\} \forall C = \{1, 2, \ldots KandC \neq K\} \quad (7)$$

Assume $C = \{C_1, C_2, \ldots, C_N\}$ represents the datasets and number of clusters are represented by K.

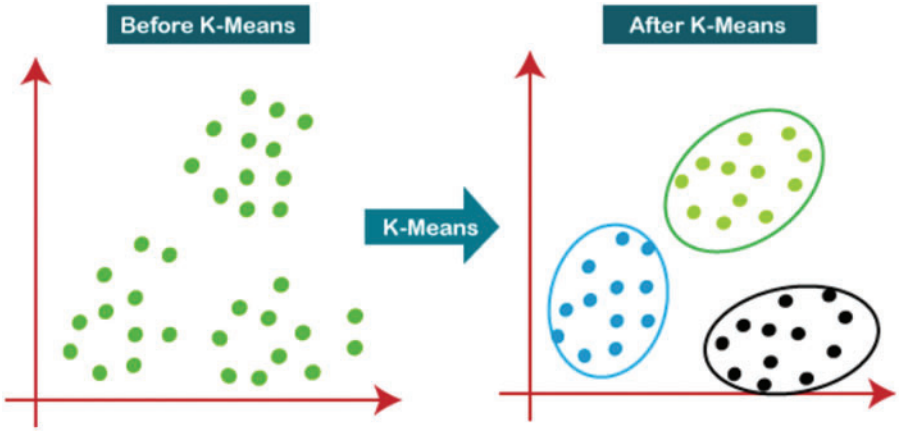


Figure 2: Clustering before K-means and after K-means

where M_l indicates the centroid of cluster CL_l and the centroid of cluster is calculated by the following formula:

$$M_l(m) = \frac{1}{n_l} \sum_{C \epsilon CL_l} C, l = 1, 2, 3, \ldots\ldots\ldots\ldots, K \tag{8}$$

Where n_l represents the cluster features and l indicate the number of clusters, m is used for number of iterations and N is used to represent the number of features.

The direct usage of an original priori box is not sufficient to increase vehicle detection accuracy. As a result, the K-means clustering technique was employed to cluster the target frame of the labelled dataset. The goal is to enhance the intersection ratio between the anchor and detection frames, so that the optimal a priori frame could be chosen. The formula can be expressed using Eq. (9).:

$$d = 1 - IOU \tag{9}$$

Where, the intersection ratio between the predicted frame and the true frame is represented by IOU.

3.2 Bounding Box Clustering

The methods based on the traditional vehicle detection algorithm generates the candidate proposal using sliding windows. However, these methods are slower than deep learning-based methods as they are time-consuming. Therefore, the candidate proposals are calculated using aspect ratio [0.5,1,2] in less time than a sliding window in Faster R-CNN and SSD. The first problem is that the aspect ratio is selected manually. The results are obtained to predict detections in the network based on the priors for a dataset. The second difficulty is that the aspect ratios are created with datasets like PASCAL VOC [15] and MS COCO [15–18]. Therefore, this approach is not very effective in detection of vehicles. K-means algorithm is used to overcome these issues without selecting the aspect ratio manually. Hand-picked anchor boxes are different from cluster centroids. The k-means algorithm can be denoted by Eq. (10) as follows.

$$E = \sum_{i=1}^{K} \sum_{i \epsilon C_i} ||x - \mu_i||^2 \tag{10}$$

where x denotes the sample, μ_i denotes the average vector of C_i, and K denotes the clustering centre. K-means is executed on the different values of vehicle size and aspect ratio. Let vehicle weight and height K = 5 and aspect ratio to be K = 3. It improves the mAP on vehicle datasets. The K-means technique is a standard clustering algorithm that minimises the distance between the sample and the cluster centre to find the optimal K cluster centres [8]. this article employs the K-means technique to obtain new anchor boxes by performing the re-clustering the dataset used in this paper. The anchor boxes are obtained based on the size and shape to ground-truth bounding boxes during training. The best-fit anchor boxes for the MS COCO and PASCAL dataset are used for vehicle detection using the K-mean clustering approach with various K values to reduce the proposed method's training time, which improves the accuracy of the proposed method.

Algorithm 1: K-means Clustering Algorithm

Input: Dataset X = $\{X_1, \ldots\ldots X_n\}$, K number of clusters
Output: Clustered dataset P = $\{Y_1, \ldots\ldots Y_n\}$
1. **Initialization:**
2. r←0
3. $\rho \leftarrow \varphi$
4. $\forall\ \mu_i$, $1 \leq i \leq K$, compute random μ_i
5. **BEGIN:**
6. for each point x ∈ X do
7. $Y_i^r = \left\{X_j : d\left(X_j, \mu_i\right) \leq d\left(X_j, \mu_h\right) \textit{ for all } h = 1, \ldots\ldots K\right\}$ // assign each sample X_j to cluster set
8. $\mu_i^{(r+1)} = \frac{1}{|Y_i^{(r)}|} \sum_{X_j \in Y_i} X_j$ // update the cluster set
9. $\rho^r = \left\{Y_1^r, \ldots\ldots Y_K^r\right\}$
10. if r ≥ *Max or* $\rho^r = \rho^{r-1}$ then
11. return ρ^r
12. end if
13. end for
14. End

3.3 Architecture of YOLOv5

The YOLOv5 network is the most recent addition to the YOLO architectural family. This network model achieves high accuracy and a fast inference speed, with the fastest detection speed reaching 140 frames per second. The weight file size of YOLOv5 is 90% less than that of YOLOv4. As a result, after deploying with embedded devices, the YOLOv5 model can be employed for real-time detection. As a result, the yolov5 network's advantages include its high detection accuracy, lightweight properties, and fast detection speed.

The backbone network, neck network, and detect network are the three key components of the YOLOv5 architecture. Instead of using Darknet, YOLOv5 uses PyTorch and the CSPDarknet53 as a backbone. The backbone of the YOLOv5 model solves the repetitive gradient information. It is also used to incorporate gradient changes into feature maps. As a result, model accuracy improves while inference speed decreases. Finally, by reducing the parameters, the model's size is reduced. It boosts information flow by using a path aggregation network (PANet) as a neck. PANet employs the feature pyramid network (FPN) to improve the propagation of low-level features. The image is first given to CSPDarknet53 for feature extraction. The backbone network generates feature maps of different sizes from the input image [34–39]. PANet's neck network integrates feature maps from multiple levels with

feature maps of various sizes to obtain more contextual information and reduce data loss. Finally, the results are generated by the YOLO layer. Fig. 3 shows the architecture of the YOLOv5 model.

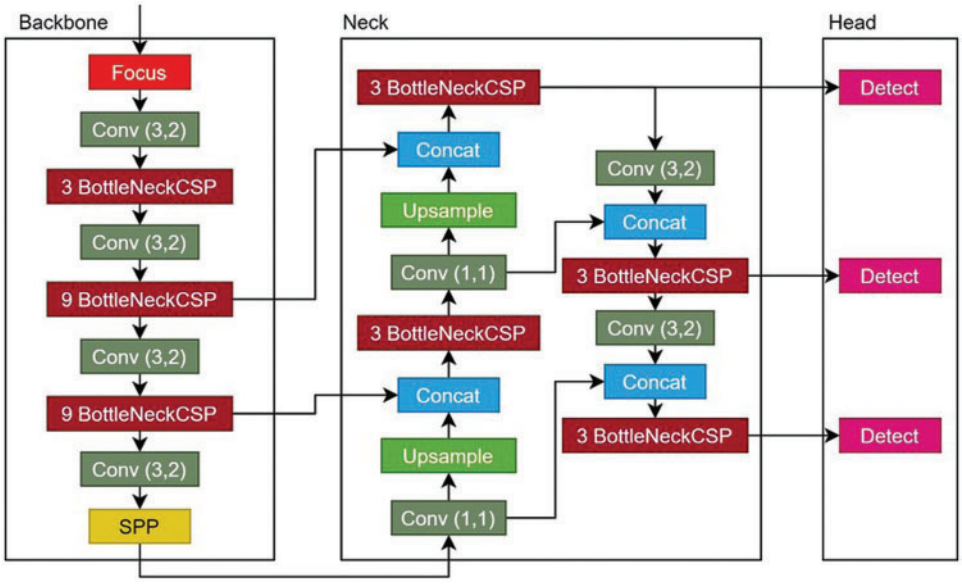


Figure 3: Architecture of YOLOv5

3.4 Non-Maximum Suppression (NMS) Technique

Multiple target objects may be present in an image, and the objects may be of various shapes and sizes. As a result, the target objects may be perfectly captured with a single bounding box. In a single input image, the YOLOv5 generates multiple overlapping bounding boxes (B_Box). As a result, YOLOv5 needs to generate a single bounding box for object of input image. Therefore, it is necessary to eliminate the overlapping bounding boxes. The Non-Maximum Suppression technique (NMS) eliminates the overlapping bounding boxes, selecting a single B_Box out of several overlapping B_Boxes to identify the objects in an image.

The NMS technique eliminates duplicate identifications and selects the best match for the final identification. Algorithm 2 demonstrates the NMS method. Furthermore, to solve the erroneous computation of non-overlapping B_Boxes, YOLOv5 choose GIoU loss as the B_Boxes regression loss function, which is defined by Eq. (11).

$$I_{GIOU} = 1 - IoU + \frac{\left|C_B - PB \cup B^{GT}\right|}{C_B} \tag{11}$$

Where B^{GT} denotes the ground-truth box, PB shows the prediction of the box, C_B denotes the smallest box covering PB and B^{GT} and $IoU = \frac{PB \cap B^{GT}}{PB \cup B^{GT}}$.

Algorithm 2: Non-Maximum Suppression (NMS) Technique

Input: B_{Box}=$\{b_1, b_2 \ldots\ldots b_n\}$, where B_{Box} represents the preliminary bounding boxes
$C_{score} = \{s_1, s_2 \ldots\ldots s_n\}$, where C_{score} represents the array of confidence score
Output: D_{Box}= set of final detected bounding box after applying NMS algorithm

1. **Initialization:**
2. $D_{Box} \leftarrow \{\ \}$
3. While $D_{Box} \neq \varnothing$
4. $K \leftarrow ArgMax\ CScore$
5. $D\_Box \leftarrow D\_Box \cup b\_K$
6. $B_{Box} \leftarrow B_{Box} - b_K$
7. $C_{score} \leftarrow C_{score}$-$s_K$
8. ***BEGIN:***
8. For $b_i \in B_{Box}$ do
9. if $IoU(b_K, b_i) \geq Th_{old}$ Then
10. $B_{Box} \leftarrow B_{Box} - b_i$
11. $C_{score} \leftarrow C_{score}$-$s_i$
12. ***End***

3.5 Optimized Loss Function

The YOLOv5 loss function can be expressed by Eqs. (11)–(13) to generate bounding box and these are also used to calculate the GIOU_Loss. the optimization of the overlapping cannot be achieved if there is phenomenon between the detection box and the real box. Positive and negative samples are employed by two category and cross-entropy loss functions to determine confidence and category loss.

$$Loss = GIOU\_Loss + Loss_{Conf} + Loss_{Class} \tag{12}$$

$$GIOU_{Loss} = 1 - GIOU = 1 - (IOU - \frac{|Q|}{C}) \tag{13}$$

Where C denotes the smallest bounding box rectangle between the detected and prior frames, and Q denotes the difference between the union of two boxes and the smallest bounding box rectangle. Which is defined by Eqs. (14) and (15).

$$\begin{aligned} \mathrm{Loss_{Conf}} = & \sum_{\mathrm{i=0}}^{\mathrm{S^2}} \sum_{\mathrm{j=0}}^{\mathrm{B}} I_{ij}^{obj} \left[\hat{\mathrm{C}}_{\mathrm{i}}^{\mathrm{j}} \log \mathrm{C}_{\mathrm{i}}^{\mathrm{j}} + (1 - \hat{\mathrm{C}}_{\mathrm{i}}^{\mathrm{j}}) \log(1 - \hat{\mathrm{C}}_{\mathrm{i}}^{\mathrm{j}}) \right] \\ & - \lambda_{\mathrm{noobj}} \sum_{\mathrm{i=0}}^{\mathrm{S^2}} \sum_{\mathrm{j=0}}^{\mathrm{B}} I_{ij}^{noobj} \left[\hat{\mathrm{C}}_{i}^{j} \log C_{i}^{j} + \left(1 - \hat{\mathrm{C}}_{i}^{j}\right) \log \left(1 - \hat{\mathrm{C}}_{i}^{j}\right) \right] \end{aligned} \tag{14}$$

$$Loss_{class} = \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} \left[\hat{P}_i^j \log P_i^j + (1 - \hat{P}_i^j) \log(1 - \hat{P}_i^j) \right] \tag{15}$$

Where I_{ij}^{obj} and I_{ij}^{noobj} denote the target of the the j^{th} detection frame in the i^{th} grid, λ_{noobj} indicates the positioning error loss weight, $\mathrm{C_i^j}$ and P_i^j considered values for training, and $\hat{\mathrm{C}}_{\mathrm{i}}^{\mathrm{j}}$ and $\hat{P}_i^j$ represent the values for prediction. The modified loss function of Eqs. (16)–(18) was chosen to solve the above

discussed issues. The *CIOU_LOSS* loss function is used in the modified algorithm's bounding box to strengthen the aspect ratio limitation mechanism.

$$CIOU\_LOSS = 1 - (IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v) \tag{16}$$

where $\rho()$ denotes the Euclidean distance, c indicates enclosed rectangles length and α denotes the weight coefficient.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^{p}}{h^{p}}\right)^2 \tag{17}$$

Where, $\alpha = \frac{v}{(1-IOU)+v}$ where v is a parameter that measures the aspect ratio's constancy.

$$Focal\_Loss = \begin{cases} -\alpha\,(1-p)^{\prime\gamma}\log P\prime & y = 0 \\ (1-\alpha)\,p^{\prime\gamma}\log(1-p^{\prime}) & y = 1 \end{cases} \tag{18}$$

3.6 Performance Metrics

The accuracy of the proposed method is estimated by different parameters of the performance metrics for vehicle detection. The true positive (TP) represents the number of detected vehicles, while the false positive (FP) indicates the number of detected non-vehicles. The average precision (AP) is estimated as the sum of all precision by calculating the TP and FP. The accuracy of each category represents the detection performance of the algorithm by Eq. (19).

$$Precision_{vehicle} = \frac{TP}{FP + TP} \tag{19}$$

Where the average precision is represented by $Precision_{vehicle}$. the true positives (TP) represents the category of vehicle, and the false positives (FP) is the category of non-vehicle.

Precision as a performance metric may be inadequate because most datasets are severely unbalanced. Even though many performance evaluation criteria have been presented, they all revolve around average precision. The F Score, also known as the F1-Score or the F measure, is a standard metric for determining the percentage of objects truly identified as a result. The F1-Score is defined by Eq. (20) as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

Precision and recall are both considered in this score. With a value of 1, the F1-Score achieves the maximum value, i.e. complete precision and recall.

4 Results and Discussion

The goal of this simulation is to determine the efficiency of the RVD-YOLOv5 algorithm proposed in this work. A number of iterations have been performed on the MS COCO and VOC PASCAL datasets to ensure that RVD-YOLOv5 is suitable for real-time performance. The simulations are carried out in Python programming, and the performance environment consists of a machine with an Intel(R) Core (TM) i7-4770 processor, 6 GB of RAM, and Windows 10 installed. Taking performance metrics into account, we compare the performance of our proposed RVD-YOLOv5. The first step of the process is to perform clustering on our datasets. The precision, average precision (AP), recall, mean average precision (mAP) and F1-Score of RVD-YOLOv5 are compared with various existing algorithms in the second part.

4.1 Data Set Description

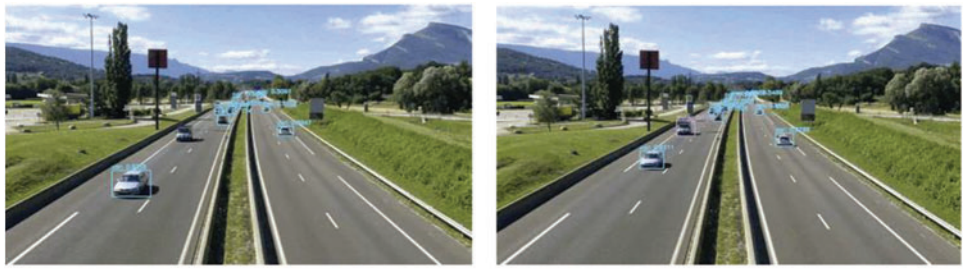
The MS COCO 2017 dataset and PASCAL 2007, 2012 used to train our proposed method, called RVD-YOLOv5 is, to carry out the performance analysis. The proposed method, RVD-YOLOv5, is firstly trained on PASCAL VOC 2007, 2012. The data is being clustered on the basis of only required classes, and data of not required classes are removed. The image size 416 × 416 is used as an input in the network. The size of the batch is taken 64, and the 8 is the size of a mini-batch. This dataset contains the total number of trained images is 16,551, and the total number of test images is 4952.

4.2 Results

The performance metric is used to analyse the performance of the proposed RVD-YOLOv5 method based on different sets of parameters. The different sets of parameters considered for measuring the performance of the proposed method are precision, recall, mAP and F1-Score. The work is carried out in three stages: the first stage performs data clustering on the datasets for improving the detection accuracy rate. In the second stage, the YOLOv5 detects vehicles by generating the bounding box corresponding to each vehicle. Then, the NMS method is applied to eliminate the overlapping of the bounding box. Finally, the loss function is enhanced to further improve the accuracy of the bounding box of the vehicle. The obtained results are compared with the existing methods of vehicle detection discussed in the literature section to analyse the performance of the proposed method, called RVD-YOLOv5. The outcome of this method is shown in Figs. 4 and 5 for the detection of vehicles. The PASCAL VOC 2007, 2012 and MS COCO 2017 datasets are used to analyse the RVD-YOLOv5. The comparative results are shown in Tabs. 1 and 2. The performance metric of object detection includes various performance measure parameters such as precision, recall, mean average precision (mAP) and F1-Score [28]. The existing models such as LD-CNN, SSD, YOLOv3 and YOLOv4 had mAP values of 86.91 per cent, 91.76 percent, 87.88 percent and 96.54 percent, respectively. The proposed method, RVD-YOLOv5, obtained an mAP value of 98.6% based on the calculation of mean average precision. There has been an increase of around 2.06% in mean average precision for this proposed system. Tab. 1 shows a comparison of the performance metrics of RVD-YOLOv5 for the MS COCO dataset, while Tab. 2 shows a comparison of performance on the VOC PASCAL 2012 dataset. The average precision (AP) is calculated for the RVD-YOLOv5 shown in Tab. 3. The results of Tab. 4 show that our proposed RVD-YOLOv5 is significantly faster than the LD-CNN, SSD, YOLOv3 and YOLOv4 algorithms. Figs. 5a and 5b represent the comparison of the performance metric on MS COCO and VOC PASCAL 2012 datasets. Figs. 6a and 6b show the comparative performance of RVD-YOLOv5 with the existing methods discussed in Section 2.

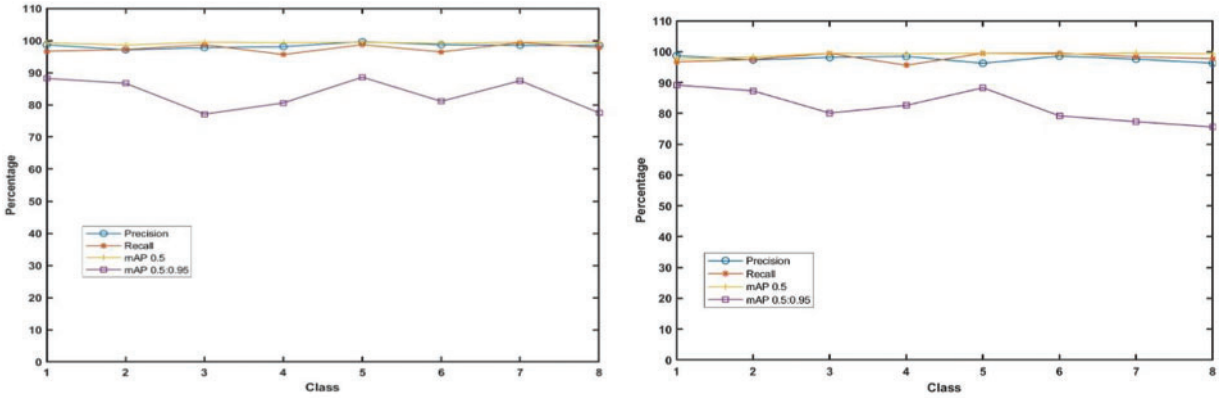
(a) (b)

Figure 4: (Continued)



(c) (d)

Figure 4: Vehicle detection using RVD-YOLOv5



(a) (b)

Figure 5: Performance of RVD-YOLOv5 on MS COCO and PASCAL datasets

Table 1: Performance metrics of RVD-YOLOv5 for MS COCO dataset

Class	Targets	Precision (%)	Recall(%)	mAP 0.5 (%)	mAP 0.5:0.95 (%)
All	116	98.10	98.75	98.34	81.30
1	116	98.70	96.72	97.68	89.20
2	116	97.35	97.50	98.25	87.30
3	116	98.15	99.45	99.50	80.10
4	116	98.50	95.60	99.30	82.60
5	116	96.25	96.50	99.50	88.30
6	116	98.60	99.50	99.15	79.20
7	116	97.60	98.35	99.60	77.70
8	116	96.30	97.75	98.30	75.60

Table 2: Performance metrics of RVD-YOLOv5 for VOC PASCAL dataset

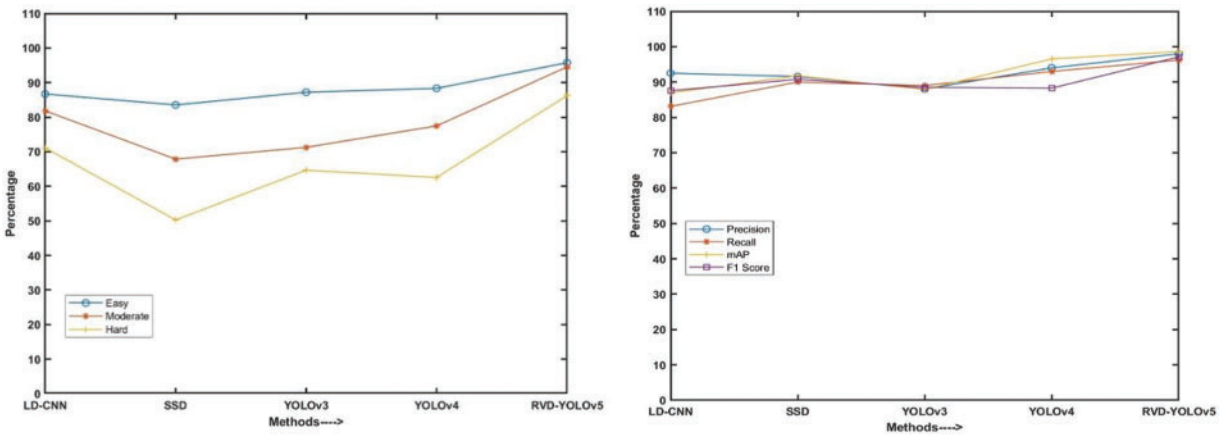
Class	Targets	Precision (%)	Recall(%)	mAP 0.5 (%)	mAP 0.5:0.95 (%)
All	116	80.60	98.10	99.30	80.60
1	116	98.70	96.65	99.30	88.20
2	116	97.10	97.22	98.60	86.70
3	116	97.80	98.70	99.50	77.10
4	116	98.10	95.65	99.30	80.60
5	116	99.65	98.72	99.50	88.60
6	116	98.70	96.45	99.10	81.10
7	116	98.50	99.35	99.50	87.50
8	116	98.50	97.85	99.50	77.50

Table 3: Results of vehicle detection methods on the MS COCO datasets

Methods	Average precision		
	Easy (%)	Moderate (%)	Hard (%)
LD-CNN	86.71	81.84	71.12
SSD	83.55	67.87	50.27
YOLOv3	87.22	71.28	64.67
YOLOv4	88.35	77.49	62.57
RVD-YOLOv5	**95.76**	**94.55**	**86.23**

Table 4: Comparative analysis of existing vehicle detection with proposed method on MC COCO dataset

Methods	Precision	Recall	mAP	F1-score
LD-CNN	92.5%	83.10%	86.91%	87.54%
SSD	91.56%	90%	91.76%	90.77%
YOLOv3	88%	89%	87.88%	88.49%
YOLOv4	84%	93%	96.54%	88.27%
RVD-YOLOv5	**98%**	**96.2%**	**98.6%**	**97.09%**



(a) (b)

Figure 6: Comparative analysis of the proposed RVD-YOLOv5 with existing methods

5 Conclusion and Future Work

In the present exposition, a high precision method using K-means clustering and the YOLOv5 has been proposed for vehicle detection. The proposed method, RVD-YOLOv5, uses K-means for data clustering on datasets. The YOLOv5 has been used for extracting the features maps and anchor bounding box of vehicles. The bounding box overlapping is eliminated using the NMS technique. The CIoU loss function is used for the estimation of the accurate regression bounding box of the vehicles. The performance of the YOLOv5 after data clustering and improved loss function is compared on the different datasets. The images from the datasets and captured videos are used to detect the vehicle using the proposed algorithm in the simulations. The proposed algorithm using K-means for data clustering on datasets trained on the MS COCO and VOC PASCAL datasets is more precise and obtains high efficiency while detecting the target. The proposed method, RVD-YOLOv5, shows a significant improvement as compared to LD-CNN, SSD, YOLOv3 and YOLOv4. Thus, the proposed method achieves the results with an mAP of 98.6% and Precision, Recall, and F1-Score are 98%, 96.2% and 97.09%, respectively.

This paper mainly discusses real-time vehicle detection with high precision. As a future work, the real-time vehicle detection can be integrated for vehicle tracking and counting.

Acknowledgement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R79), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This research is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R79), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V. K. Papastathis and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1210–1224, 2005.

[2] M. Kim, J. Jeong and S. Kim, "ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image," *Remote Sensing*, vol. 13, no. 23, pp. 4851, 2021.

[3] X. Z. Chen, C. M. Chang, C. W. Yu and Y. L. Chen, "A real-time vehicle detection system under various bad weather conditions based on a deep learning model without retraining," *Sensors*, vol. 20, no. 20, pp. 5731, 2020.

[4] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.

[5] J. Cao, C. Song, S. Song, S. Peng, D. Wang *et al.,* "Front vehicle detection algorithm for smart car based on improved SSD model," *Sensors*, vol. 20, no. 16, pp. 4646, 2020.

[6] A. Arunmozhi and J. Park, "LBP and Haar-like features for on-road vehicle detection," in *IEEE 2018 7th Int. Conf. on Educational and Information Technology (ICEIT)*, USA, pp. 362–367, 2018.

[7] S. Jabri, M. Saidallah, A. E. B. El Alaoui and A. El Fergougui, "Moving vehicle detection using Haar-like, LBP and a machine learning Adaboost algorithm," in *2018 IEEE Int. Conf. on Image Processing, Applications and Systems (IPAS)*, France, pp. 121–124, 2018.

[8] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE 2014 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, Columbus, OH, USA, pp. 580–587, 2014.

[9] R. Girshick, "Fast r-cnn," in *IEEE 2015 Int. Conf. on Computer Vision (ICCV)*, Santiago, Chile, pp. 1440–1448, 2015.

[10] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[11] J. Zhou, P. Jiang, A. Zou, X. Chen and W. Hu, "Ship target detection algorithm based on improved YOLOv5," *Journal of Marine Science and Engineering*, vol. 9, no. 8, pp. 908, 2021.

[12] J. Shen, N. Liu, H. Sun and H. Zhou, "Vehicle detection in aerial images based on lightweight deep convolutional network and generative adversarial network," *IEEE Access*, vol. 7, pp. 148119–148130, 2019.

[13] H. Song, H. Liang, H. Li, Z. Dai and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review*, vol. 11, no. 1, pp. 1–16, 2019.

[14] V. Sowmya and R. Radha, "Heavy-vehicle detection based on YOLOv4 featuring data augmentation and transfer-learning techniques," *Journal of Physics: Conference Series*, Chennai, India, vol. 1911, no. 1, pp. 12029, 2021.

[15] S. Zhang, L. Wen, X. Bian, Z. Lei and S. Z. Li, "Single-shot refinement neural network for object detection," in *IEEE 2018 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, Salt Lake City, Utah, pp. 4203–4212, 2018.

[16] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.

[17] W. Sun, G. Dai, X. Zhang, X. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. https://doi.org/10.1109/TITS.2021.3130403.

[18] A. T. Tajar, A. Ramazani and M. Mansoorizadeh, "A lightweight tiny-YOLOv3 vehicle detection approach," *Journal of Real-Time Image Processing*, vol. 18, no. 6, pp. 2389–2401, 2021.

[19] L. Zhu, X. Geng, Z. Li and C. Liu, "Improving YOLOv5 with attention mechanism for detecting boulders from planetary images," *Remote Sensing*, vol. 13, no. 18, pp. 3776, 2021.

[20] M. A. Hassan, A. R. Javed, T. Hassan, S. S. Band, R. Sitharthan and M. Rizwan, "Reinforcing Communication on the internet of aerial vehicles," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1288–1297, 2022. https://doi.org/10.1109/TGCN.2022.3157591.

[21] A. R. Javed, S. Ur Rehman, M. U. Khan, M. Alazab and T. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1456–1466, 2021.
[22] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan and M. S. Haghighi, "Anomaly detection in automated vehicles using multistage attention-based convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4291–4300, 2020.
[23] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas and X. Liu, "Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles," *Trans Emerging Tel Tech*, pp. e4088, 2020. https://doi.org/10.1002/ett.4088.
[24] M. Kim, J. Jeong and S. Kim, "ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image," *Remote Sensing*, vol. 13, no. 23, pp. 4851, 2021.
[25] Y. Zhao, X. Zhou, X. Xu, Z. Jiang, F. Cheng *et al.,* "A novel vehicle tracking ID switches algorithm for driving recording sensors," *Sensors*, vol. 20, no. 13, pp. 3638, 2020.
[26] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE 2017 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, pp. 7263–7271, 2017.
[27] Y. Liu, R. Wang, S. Shan and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in *IEEE 2018 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, Salt Lake City, Utah, pp. 6985–6994, 2018.
[28] P. Zhou, B. Ni, C. Geng, J. Hu and Y. Xu, "Scale-transferrable object detection," in *IEEE 2018 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, Salt Lake City, Utah, pp. 528–537, 2018.
[29] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe *et al.,* "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *IEEE 2019 Int. Conf. on Computer Vision (ICCVF)*, Seoul, Korea, pp. 6023–6032, 2019.
[30] G. Ghiasi, T. Y. Lin and Q. V. Le, "Dropblock: A regularization method for convolutional networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1810–12890, 2018.
[31] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye *et al.,* "Distance-IoU loss: Faster and better learning for bounding box regression," in *Int. Conf. on Artificial Intelligence (ICAI)*, Larache, Morocco, vol. 34, pp. 12993–13000, 2020.
[32] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan *et al.,* "Feature pyramid networks for object detection," in *IEEE 2017 Int. Conf. on Computer Vision and Pattern Recognition (ICCVPR)*, Salt Lake City, Utah, pp. 2117–2125, 2017.
[33] Z. Jin, P. Qu, C. Sun, M. Luo, Y. Gui *et al.,* "DWCA-YOLOv5: An improve single shot detector for safety helmet detection," *Journal of Sensors*, vol. 2021, no. 6, pp. 1–12, 2021.
[34] M. D. Zeiler, G. W. Taylor and R. Fergus, "Adaptive deconvolutional networks for mid and high-level feature learning," in *IEEE 2011 Int. Conf. on Computer Vision (ICCV)*, Barcelona, Spain, pp. 2018–2025, 2011.
[35] S. Singh, A. Suri, J. N. Singh, M. Singh and D. K. Yadav, "Object identification and tracking using YOLO model: A CNN-based approach," in *Springer 2021 Int. Conf. Machine Learning and Information Processing (ICMLIP)*, Hyderabad, India, pp. 153–160, 2021.
[36] M. Rai, R. Sharma, S. C. Satapathy, D. K. Yadav, T. Maity *et al.,* "An improved statistical approach for moving object detection in thermal video frames," *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 9289–9311, 2022.
[37] S. Mishra, D. K. Yadav, F. Tabassum and D. Kumar, "Detection of moving vehicle in foggy environment through google's firebase platform," *Turkish Online Journal of Qualitative Inquiry*, vol. 12, no. 6, pp. 9892–9901, 2021.
[38] S. Yadav, D. K. Yadav, A. K. Budati, M. Kumar and A. Suri, "Automating the Indian transportation system through intelligent searching and retrieving with amazon elastic compute cloud," *IET Networks*, vol. 10, no. 3, pp. 123–130, 2021.
[39] A. Suri, S. K. Sharma and D. K. Yadav, "Detection of moving vehicles on highways using fuzzy logic for smart surveillance system," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 1, pp. 419–431, 2021.