

Modified-vehicle detection and localization model for autonomous vehicle traffic system

Amit Juyal¹, Sachin Sharma², Shuchi Bhadula²

¹Department of Computer Science and Engineering, Graphic Era Deemed to be University, Graphic Era Hill University, Dehradun, India

²Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India

Article Info

Article history:

Received Mar 25, 2024

Revised Sep 20, 2024

Accepted Sep 30, 2024

Keywords:

Autonomous vehicle

CBAM

CSP

YOLO

YOLOv5-CBAM

ABSTRACT

The modification of vehicles for financial gain is an evolving tendency observed in India. Recognizing and detecting of these modified illicit cars is an important but critical task in autonomous vehicles (AV). It is always possible for a cyclist or pedestrian to traverse obstacles or other fixed objects that appear in front of any moving vehicle. Vehicles that are autonomous or self-driving require a different system to quickly identify both stationary and moving objects. A deep learning model named you only look once version 5 (YOLOv5)-convolutional block attention module (CBAM) is proposed here for the Indian traffic system which is based on YOLOv5m. The proposed algorithm, YOLOv5-CBAM, has three major components. The first module, the backbone module is employed for feature extraction. The second module is to detect static as well as dynamic objects at the same time and the third CBAM module is adopted in the backbone and neck part, which mainly focuses on the more prominent features. Two cross stage partial (CSP) modules were used after every convolutional layer resulting in an additional head to the proposed model. Four head modules equipped with anchor boxes performed the final detection. For the present dataset, the proposed model showed 98.2% mean average precision (mAP), 98.4% precision, and 94.8% recall as compared to the original YOLOv5m.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Amit Juyal

Department of Computer Science and Engineering, Graphic Era Deemed to be University

Graphic Era Hill University

Dehradun, India

Email: amitjuyal26@gmail.com

1. INTRODUCTION

Self-driving is an intelligent autonomous driving feature that improves safety and convenience beside the controls. Some international automakers have already launched and are operating self-driving automobiles. Tesla, Waymo, Zoox, and Cruise are a few examples. Compared to developed nations, traffic conditions are more difficult in emerging nations. India is the most populous and developing nation. Autonomous vehicles (AV) are the most challenging tasks in the present Indian traffic scenario. If automakers are successful in manufacturing AV while concentrating on the traffic conditions in India, then other nations can simply embrace vehicles as well. The current study aims to assess the success of AV in the Indian traffic scenario. To achieve safe and trustable autonomous driving many intelligent software and different hardware have been developed to date and sometimes for better driving accuracy, the integration of intelligent software and hardware is needed. The overall intelligent driving system includes Cameras, radio detection, and ranging (RADAR), and light detection and ranging (LiDAR) as the primary sensors that are used to collect data about nearby traffic as depicted in Figure 1. The visual information is provided by the

camera, the distance and velocities of objects are determined by RADAR, and the three-dimensional (3D) point clouds of nearby traffic are produced by LiDAR. Software based on deep learning, machine learning, pose-estimation-based models, etc further processes this information to estimate the object and intention of the observed object. Based on these outputs, AV controls itself.

Although car manufacturers in India have already deployed advanced driver assistance system (ADAS), fully AV still require significant improvements in the complicated system of AV. On Indian roads, a wide variety of vehicles are in use. People modify their vehicles to meet their specific requirements. Figure 2 demonstrates the modified vehicles that can not be easily recognised as an illustration of such vehicles is seen in Figures 2(a) and 2(b). Therefore, an intelligent vehicle system that can recognize and detect all categories of vehicles, is necessary for safe driving in AV.

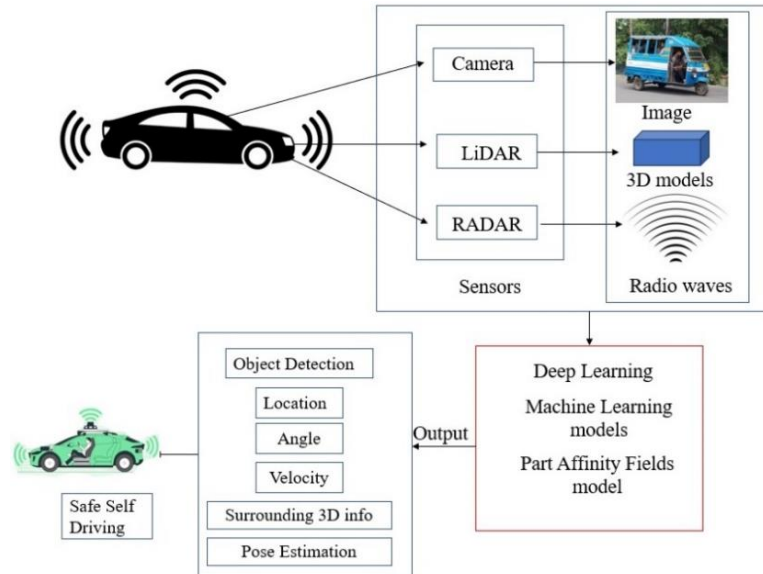


Figure 1. Intelligent system in AV for safe driving



Figure 2. Displays the modified automobiles that are difficult to recognize, as seen in the images; (a) and (b) illustrate examples of modified vehicles by attaching the carrier to the motorcycle, transformed into a cargo or transit vehicle

The safe operation of an autonomous vehicle depends on the accurate detection of fixed and moving objects as illustrated in Figure 3. The illustration shows that AV is following its predetermined path. The AV's planned route may include crossroads. Fixed objects like barricades and traffic cones may be used to restrict one or both sides of the intersection. Barricades and traffic cones can restrict the flow of cars, trucks, buses, and other vehicles, but cyclists and pedestrians can bypass these obstacles. Cyclists and pedestrians can cross in front of an AV by going over permanent barriers like barricades and traffic cones. Accidents may occur due to this circumstance, consequently, in AV fixed and moving object detection is essential for safe driving.

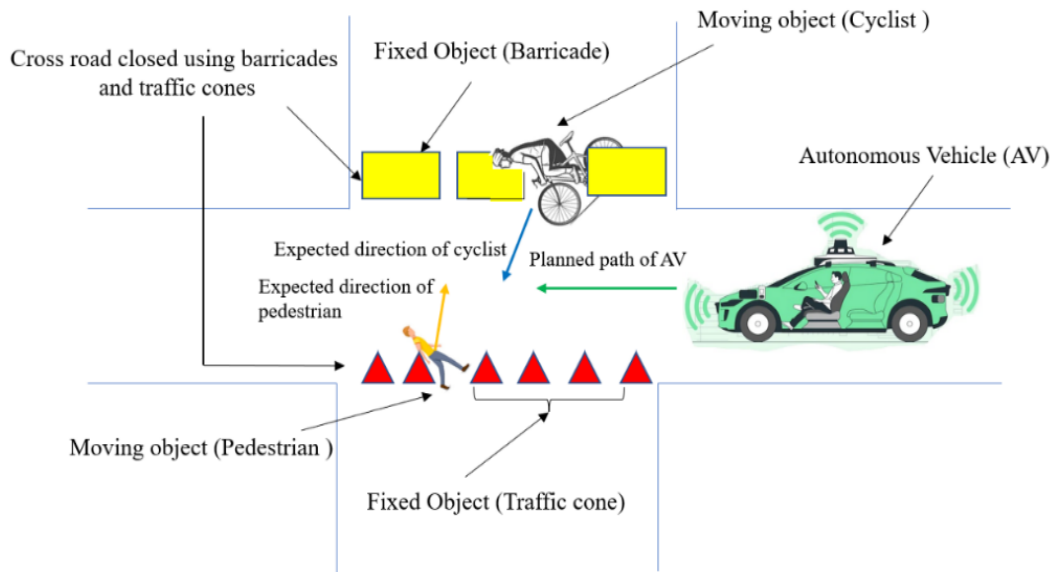


Figure 3. Crossroads with two sides blocked by fixed objects suddenly appeared in the route of the AV

The object detection challenge involves identifying and locating many objects in a digital image. To resolve the issue of object detection till now various machine learning and deep learning techniques have been applied. An important computer-vision issue is real-time object detection, especially when it comes to AVs or autonomous driving. Compared to conventional techniques, convolutional neural network (CNN)-based techniques with good mean average precision (mAP) are being focused on real-time traffic detection in autonomous driving. The object detection method is of two types: two-stage detection and one-stage detection. The methods like R-CNN based on the CNN perform detection in two steps, with first region proposals are generated using segmentation and selective search and second, it tries to find the objects in each region. Some two-stage detector models that have been used in autonomous driving could be described as:

- Region-based CNN (RCNN) [1]: it is a two-step detector that uses the selective search for region proposal, CNN extracts the feature maps and finally linear SVM calculates the weights for each layer in object classification.
- Fast R-CNN [2]: to create a feature map, the complete image with the region of interest (ROI) was fed into numerous convolutional layers. Then, using an ROI pooling layer, a fixed-length vector of features was produced from the feature map for each region proposal.
- Faster RCNN [3]: its working is like fast RCNN except that it generates region proposals more quickly using a separate region proposal network.

Because of the two stages in object detection, RCNN family methods are considered slow in real-time object detection. To speed up the detection, one-stage detection methods such as YOLO [4] and SSD [5] are preferred. YOLO and SSD are considered faster than two-stage region-based CNN methods as they transform the detection problem into a unified regression problem. Some one-stage detector models that are being used in Autonomous driving could be described as:

- SSD: in SSD, the visual geometry group (VGG)-16 [6] network processes all the images to create feature maps. Convolutional layers utilize these feature maps afterward, performing the actual detection and producing multiple bounding boxes for each object.
- YOLO: an object detection method called “you only look once” was first put out by Redmon *et al.* [4]. The entire image is split into a grid by YOLO and then generates the anchor boxes for each grid cell. It is the fastest method in real-time object detection [7]. YOLO is a well-known object detection method because of the way it quickly and precisely detects objects. Using the Pytorch framework, Jocher *et al.* [8] unveiled YOLOv5 . The existing YOLOv5 architecture is shown in Figure 4. YOLOv5 automatically extracts features for object detection from input images. These features are then passed to a prediction system to form boxes around objects to recognize their classes. The YOLO is the first end-to-end object detection network that can predict bounding boxes in addition to class labels.

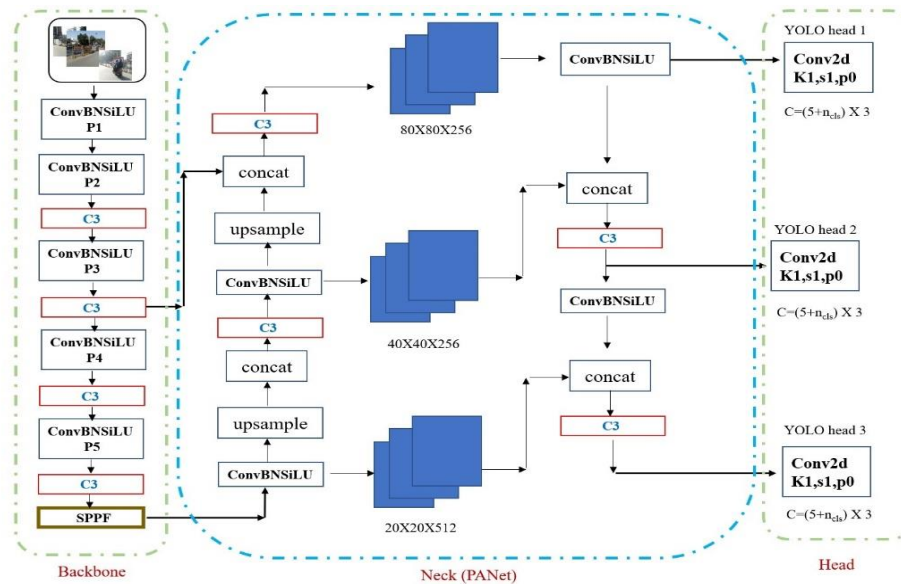


Figure 4. Original YOLOv5 architecture

The YOLOv5 network is comprised of three critical components. CNNs and optimized spatial pyramid pooling (SPPF) generate visual features at various granularities in the backbone. Neck, several layers that integrate and mix visual features before passing them for prediction. The head part of the network processes the features from the neck to perform bounding box prediction with class labels. However, in real-time detection systems like in AV, more accurate and agile detection is required for safe driving. The present study focuses on developing a deep-learning model that can detect both stationary and moving objects for safer driving. Secondly, the study aims to detect the intention of cyclists or pedestrians to cross the road. Based on the literature review of previous work, the present study suggested the use of an improvised YOLOv5 model. It has been reported that the original YOLOv5 has 3 heads for detection, while in the improvised YOLOv5-convolutional block attention module (CBAM) model 1 another head has also been added in the present work for better detection of identical and small objects. Other improvisations contributed:

- i) Based on the architecture of YOLOv5, an improved model has been presented to detect 18 different types of objects.
- ii) A complex self-driving Indian vehicle dataset with 18 types of dynamic and fixed object images is presented.
- iii) Cross-stage local network (CSP) [9] and attention module CBAM [10] are added to enhance the predictions of anchor boxes.

The present study elaborated on the development, functioning, and outcomes of improvised/enhanced YOLOv5-CBAM. The observations described that the use of improvised models that have been implemented on the custom dataset produces better results for fixed and dynamic Indian traffic object detection. However, it is more challenging to recognize the objects, especially to evaluate the intentions of multiclass objects like vehicles, barriers, traffic cones, cyclists, and pedestrians, as they all belong to different classes. So the study continued with the use of the PAF model along with an improvised deep-learning model for better detection.

2. REVIEW OF LITERATURE

Recently, the field of artificial intelligence in autonomous driving has seen a lot of study interest in the problem of multi-object identification. The one-stage YOLO technique was initially developed to tackle object detection as a regression problem. One of the most recent developments, YOLO, can detect objects with a good confidence score, but the model's spatial constraint reduces the number of objects that may be predicted. To predict the motion of nearby traffic objects, [11] presents a GAMM model for AV vehicles. The model is capable of forecasting traffic agents' movements under various restrictions like kinematics, geometric, and behavioral. Han *et al.* [12] combine LiDAR and color images for object detection to improve the YOLO model. YOLO is improved to detect low-intensity objects like pedestrians and fixed objects.

Original YOLO was trained to get important parameters. Color images and depth images are used to improve accuracy. Dreossi *et al.* [13] presented CNN-based framework to classify vehicles. The framework uses an image generator for synthetic images which is used to validate the CNN model. Faster YOLO by [14] combines the proposed random kernel convolutional network with auto-encoder hidden layers network for feature extraction. Zhang *et al.* [15] introduce an obstacle-detection algorithm. YOLO combines light field camera images that contain depth images and high-resolution images. YOLO takes red, green, and blue (RGB) images as input and generates object detection information using a depth map. An object detection system for AV using modified YOLO with seven convolutional layers was proposed [16]. Another researcher proposed a YOLO v3-live model for real-time object detection [17]. For feature extraction, multi-scale receptive fields were used. The overall model was constructed by modifying the structure and network parameters. Wong *et al.* [18] introduce YOLO Nano for object detection. The network architecture of YOLO Nano is based on single-shot object detection network architecture. With a highly compact network design, YOLO nano achieved 69.1% mAP. In another study, multispectral images for object detection were used [19]. Multispectral images contain multilateral information that can help detect objects that cannot be visualized in normal RGB images. The proposed multispectral ensemble detection pipeline is divided into two single spectral detection models and an ensemble part. SqueezeDet which is a fully CNN for object detection was also proposed in a study [20]. Feature map, class probability, and bounding box were generated using convolutional layers.

3. METHOD

Real-time detection of fixed objects as well as moving objects is an important task for safe driving. The primary risk concern for self-driving in real traffic scenarios is the detection of multiclass objects like dynamic objects (cyclists, pedestrians, and various types of vehicles) as well as static or fixed objects (barricades, traffic cones).

Multiclass object localization (MOL) is an essential problem in computer vision containing numerous applications like self-driving cars, security systems, X-rays, and many others. The challenge derived from the above discussions is known as MOL and it plays an important role for the autonomous vehicle because due to this technique various objects present in the surroundings of the vehicle can be detected, classified, and located in terms of their exact position.

Figure 5, displays the entire diagram of the suggested framework for the detection of multiclass dynamic and fixed objects in real traffic scenarios. In the first step moving and fixed objects are detected using an improved YOLOv5-CBAM model. To enable AV to operate safely, the next stage is to detect the objects that have been modified. Additionally, recognizing stationary objects like traffic cones, barricades, and traffic signs is crucial for self-driving automobiles.

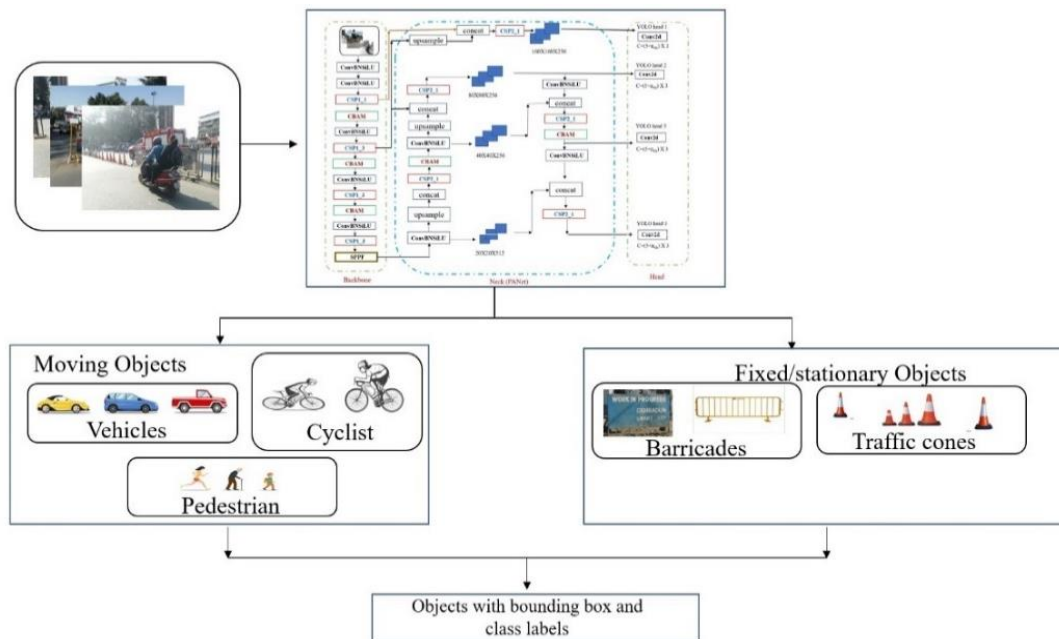


Figure 5. The overall functionality of the suggested technique

Although barricades and traffic cones are used as fixed items to control traffic, it has been observed that pedestrians, cyclists, and two-wheeler riders often disregard these types of barriers and can emerge abruptly in front of any car. As a result, in the Indian traffic scene, pedestrians, cyclists, and two-wheelers are seen as potential accident causes. Therefore, it is crucial for safe driving to detect pedestrians, cyclists, and two-wheelers.

3.1. Improved model

The original YOLOv5 object detection network normally comprises the neck, the backbone, and the detecting head. The features retrieved in the middle network are noticeably low, which causes the detection results of YOLOv5 for the Indian traffic scenes to be less than optimal. Even though the original YOLOv5 can detect and identify automobiles in vehicle images with good accuracy, it only requires a small number of parameters to suit the information because there are similar vehicle images that can be detected with fewer parameters.

The technique's key goal is to enhance recognition of both static and moving Indian vehicular traffic items. YOLOv5 has been modified to increase accuracy while considering the need for less computation. Certain changes have been made in the original YOLOv5 to propose our improvised model YOLOv5-CBAM for the present study. The proposed YOLOv5-CBAM model addresses the inadequacies of the YOLOv5 model in the vehicle detection job, and provides three major contributions as follows:

3.1.1. Inclusion of CSP network

As shown in Figure 6, a cross-stage local network called CSP is employed to integrate and generate image features on many lightweight CNNs for better adjusting the depth of the network. The base layer's feature map is split into two portions using the CSP approach, which then integrates them utilizing a cross-stage hierarchy. Unlike the YOLOv4 network, the YOLOv5-CBAM network structure has two CSP structures, CSP1 and CSP2, with the first type mostly utilized throughout the backbone and the other one being primarily used throughout the Neck. The YOLOv5-CBAM network produces 4 output feature maps having various sizes in comparison to the YOLOv5 network.

CSP:- the CSP network is a technique commonly used in CNN specifically in efficientnet to improve the performance of the network through segmentation and fusion of feature maps. The CSP network addresses the problem of vanishing gradient, a problem witnessed in a neural network where the gradients of the loss function concerning the parameters reduce to insignificant amounts are the gradients are transported backwards through the network during the optimization stage of gradient descent. Computing efficiency is also increased throughput in processing only the partitioned part. The CSP can be mathematically expressed as:

An image represented by feature vector I is an input feature map.

$$I \in R^{C \times W \times H} \{C = \text{channels}, W = \text{width}, H = \text{height}\} \quad (1)$$

Together the channel dimension divides the incoming feature map into two parts:

$$\text{split } I \text{ into two parts } I_a, I_b = \frac{I}{2}$$

where $I_a \in R^{\frac{C}{2} \times W \times H}$ and $I_b \in R^{\frac{C}{2} \times W \times H}$.

Execute a sequence of convolutional operations (f_1) on I_a .

$$M_1 = f_1(I_a : \theta_1) \quad (2)$$

Identity mapping: retain the value of I_b without any changes.

$$M_2 = I_b \quad (3)$$

Concat (1) and (2).

$$M = M_1 (+) M_2 \quad (4)$$

$$M \in R^{C \times W \times H}$$

Transition: next, proceed with the application of an additional convolutional layer to the combined output.

$$N = f_2(M : \theta_2) \quad (5)$$

Where the transition convolutional layer is represented by f_2 , which has parameters denoted by θ . CSP block as an entirety is denoted as.

$$Z = f_2((f_1(\text{split}(I)a; \theta_1) + \text{split}(I)b); \theta_2) \tag{6}$$

3.1.2. Inclusion of CSP network

The efficiency with which the intermediary layers extract the crucial data determines the success of object detection. The YOLOv5 backbone module extracts the image’s features and uses them to detect objects in the digital image. From the extracted features, some features have a greater influence on the outcomes than others. As a result, we require a module that can solely concentrate on key features. Therefore, we have added an attention component in YOLOv5m to focus on more important features. For the attention module, the CNN distributes higher weights to more important features and lower weights to less essential information. The weighted data is then combined and evaluated. Three categories can be used to classify the common attention modules, the hybrid attention module [21], the spatial attention module, and the channel attention module. The hybrid attention module is a lightweight attention module that generates a channel attention map and uses this information, to generate a final fine-tuned feature in the spatial attention module. In the channel attention module, each of the channels of a feature map is viewed like a feature detector, that concentrates on “what” is important given an input image, and the spatial attention module is responsible for allocating weights for the spatial data. The CBAM, which is made up of the channel and spatial attention modules CAM and SAM respectively, is the attention module that is added to the YOLOv5m network architecture. In the Improved YOLOv5 CBAM, the CBAM layer is added in the backbone after every CSP1 layer, and in the Neck part, it is added after every CSP2 layer so that the network can focus on more important features. The primary motivation for utilizing CBAM in the improved YOLOv5 CBAM is that it is an impressive attention module for CNN that can efficiently extract information about channel and spatial attention. The CBAM module is an easy-to-integrate, less computational cost component of CNNs. CBAM training has minimal overheads and can be trained end-to-end together with the network backbone. Every CSP module is followed by the addition of CBAM, which assigns higher weights to features that are more crucial to concentrate on features that are crucial for improving the accuracy of comparable traffic object detection.

CBAM: a relatively small lightweight module called the CBAM. CNN, in particular, can easily incorporate CBAM as a type of neural network. It is a strong enhancement strategy that can help achieve more important and relevant features on the channel and in the spatial domain. CBAM and DSAM mainly have two structures namely CAM and SAM.

In (CAM) the global average pooling (GAP) and global max pooling (GMP) proposed are used on the input feature map and on the channel on spatial information which contains more effective features. Extracted features from GAP and GMP are then dispatched to the shared multi-layer perceptron (MLP). These features are passed through the various hidden layers of MLP and produce two feature maps as its output. To obtain the final channel attention map, the obtained maps are summed over channels and then passed through the sigmoid activation. Finally, the most significant channels (feature maps) are obtained from the multiplication of the channel attention map and the original input feature map. Mathematically CAM is represented in (7).

$$Cmap(F) = \sigma(Smlp(GAP(F)) + Smlp(GMP(F))) \tag{7}$$

Where channel attention map, is denoted by $Cmap(F)$, $Smlp$ is a shared multi-layer perceptron and sigmoid function is denoted by σ .

The motivation behind the spatial attention module is simply to draw a focus on important areas of the feature map. Then, average pooling and max pooling operations are applied across channel dimensions so, we have two 2D maps. These maps are concatenated and passed through a convolutional layer followed by a sigmoid activation to obtain a spatial attention map. The spatial attention map is then element-wise multiplied by the feature map which has already passed through channel attention in order to further enhance the relevant areas. Mathematically SAM is represented in (8).

$$Smap(F) = \sigma(f_{7 \times 7}([Poolavg + Poolmax])) \tag{8}$$

Where spatial attention module is $Smap(F)$, σ is sigmoid function, $f_{7 \times 7}$ 7×7 filter convolution operation, + is concatenation.

In the proposed improved YOLOv5-CBAM, the backbone and neck part of the model architecture included CSP and CBAM modules to enhance the model’s accuracy. The addition of CSP and CBAM in backbone is mathematically represented in the (9).

$$X_{backbone}^i = CBAM(CSP(X^{i-1})) \quad (9)$$

In (3), for the i th CSP block, $X_{backbone}^i$, is the representation of the i th CSP block with CBAM. X^{i-1} is the original feed-forward input to this block is the input feature map.

The addition of CSP and CBAM in Neck is mathematically represented in the (10).

$$X_{Neck}^j = CBAM(CSP(X_{backbone}^j)) \quad (10)$$

In (4), for the j th neck layer, X_{Neck}^j , is the representation of the j th Neck layer with CBAM, $X_{backbone}^j$, is the feature map from the backbone, which is the corresponding feature map here for the input image.

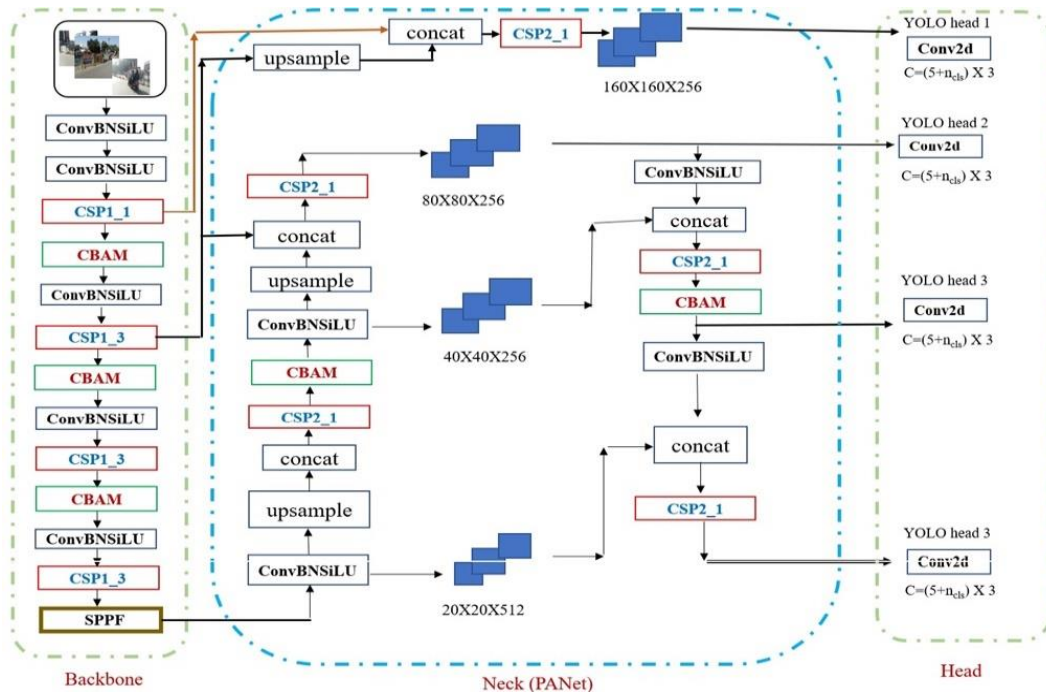


Figure 6. Proposed YOLOv6-CBAM model for real-time Indian traffic object detection

3.1.3. Inclusion of head

In the present study we have added an extra head in the improved YOLOv5 CBAM to generate more features. YOLOv5-CBAM with four heads, in YOLOv5, which was originally developed, three detection heads make it possible to detect objects of different sizes. In the improved YOLOv5-CBAM there are four head detection layers, regarding object detection, the number of detection heads is equal to the number of feature maps/scales on which the predictions are made. The sizes of objects in images can be regulated in a large range. Thus, introducing one more head with the other three detection heads will improve the detection capability of the model, and the model can effectively predict smaller objects as well as large objects. The detection head is used to predict the bounding boxes, which means, the regions of interest of the object and the class probabilities at several scales.

$$Y_{bounding\ box,class\ score}^i = H^i(X_{Neck}^i) \quad (11)$$

In (11), $Y_{bounding\ box,class\ score}^i$ is the output of predictions (bounding boxes, class scores) from the i th detection head, X_{Neck}^i is the feature map from the previous module Neck.

In the modified YOLOv5 CBAM, the feature graph extracted after the first CSP1_1 is concat along with an upsample feature from CSP2_1 in the Neck part which corresponds to an additional $160 \times 160 \times 256$ feature detection head 1. The redesigned network includes four prediction heads with dimensions of $20 \times 20 \times 255$, $40 \times 40 \times 255$, $80 \times 80 \times 255$, and $160 \times 160 \times 255$ for detection. Additionally, we use concatenation to

combine a previous network feature map with our upsampled features. This allows us to obtain more significant semantic information based on the upsampled features as well as finer-grained information from the previous feature map. Which helps the network to differentiate between similar vehicles (modified loader vehicles using motorcycles). Then, using this merged feature map, we process it with a few additional convolutional layers and finally predict a comparable tensor.

3.2. Estimation of the pose intention of a cyclist and a pedestrian

In this section, based on the detection of both cyclists and pedestrians, the technique of using pose estimate as the primary information to ascertain whether the cyclist and pedestrians will cross the blocked road using fixed objects is investigated. In the proposed model, the intention recognition algorithm is integrated with the object detection algorithm to anticipate the intention of detected cyclists and pedestrians. Figure 7 shows the flow of object detection along with pose estimation using the PAF method.

To construct a 2D pose for both the pedestrian and the cyclist, the first step is to use YOLOv5-CBAM to detect a pedestrian and then input that location into the part affinity fields (PAFs) method [22]. The orientation and position of pedestrian limbs are represented by 2D vector fields of PAF. The PAF architecture is used in the pose estimation process to acquire the 2D features of the individual present in Figure 8.

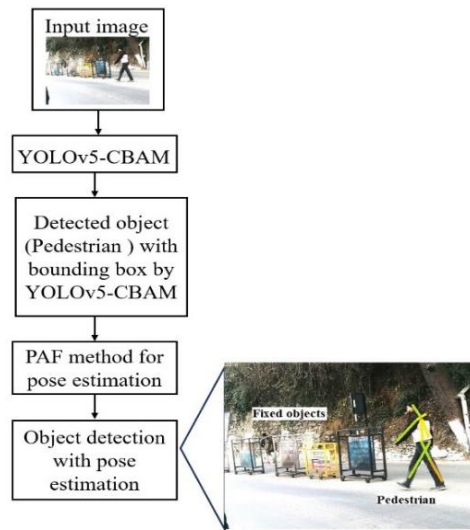


Figure 7. Pose estimation and detection method using proposed YOLOv5-CBAM with PAF method



Figure 8. Pose estimation using the PAF method

The PAF network is divided into two branches, with the top branch predicting the confidence maps and the bottom branch predicting the affinity fields. The first stage of each branch receives a set of feature maps that were produced as a result of the convolutional network’s analysis of the picture (which was initiated by the first 10 layers of the VGG-19 [23] and then finetuned). Every PAF relates to a limb and

consists of two signals(channels), one being the x-coordinate and the other being the y-coordinate of the involved vector. For the input image I, first obtain the confidence map for each body part p. $M_p(x,y)$ illustrates the confidence map of key point p in the body or joint. For each limb point type c, The PAF is $V_c(x,y)=(V_c(x,y), V_c(x,y))$. $V_c(x,y)$, $V_c(x,y)$ are 2D vectors at pixel (x,y). The CNN has two branches: one for the confidence maps and one for the PAFs. These networks will have a similar structure, and they will consist of three layers. After confident maps and PAFs are created, the match graphs between the detected body parts and human skeletons are obtained by using a bipartite matching scheme. Non-maximum suppression, on the confidence maps (M_p), is applied to detect key points for each of the body parts p. Candidate connections are created between the key points for each body part (m,n) based on PAFs M_p , which is connected by limb type c. To calculate the score of candidate connection is scored by integration of PAF values corresponding to the line segment connecting p_m and p_n .

$$score(p_m, p_n) = \int_0^1 M_p(p_m + t(p_n - p_m)) \cdot \frac{p_n - p_m}{\|p_n - p_m\|} dt$$

Finally bipartite matching algorithm is used to generate the graph which associates the detected keypoints to the limbs. It is clearly shown in

3.3. Customized dataset for testing and training of improved model

YOLOv5 already has good performance in vehicle detection. The uncertainty in India's traffic scenario makes it more challenging for YOLOv5 in real-time traffic objects. Figure 9, demonstrates the modified vehicles that are detected wrongly as Figures 9(a) to 9(c), clearly illustrates that YOLOv5 produces false results due to a lack of Indian vehicle information. In Figure 9(a), YOLOv5 predicted the object as a "Motorcycle" but it is a modified vehicle which is built using an old motorcycle. Similarly in Figures 9(b) and 9(c), YOLOv5 wrongly predicted objects as "Truck" although in real both are Indian vehicles "Vikram" and "E-RickSha" respectively. For the present study, the custom dataset is required so that our proposed YOLOv5 based model can detect normal Indian vehicles but also modified automobiles. To assess the effectiveness of our suggested methodology, we created an entirely novel set of Indian traffic objects. Images of vehicles and stationary objects from Indian traffic situations have been collected in Dehradun, India. 2,740 color images have been gathered.

The data augmentation process is done to increase the number of images in the dataset. After data augmentation image count increases from 2,740 to 5,098 images. By utilizing data augmentation methods, the model is improved and more accurate because the data is plentiful and enough. Data augmentation methods rotation, grayscale, saturation, blur, noise, etc were used. In rotation 90° clockwise and 90° counter-clockwise images were rotated. Images were converted from RGB to grayscale in 25% of images, between -25% and +25% saturation was done, and images were blurred up to 2.5px, from the total pixels 5% pixels noise was added in the image. The dataset is divided into training, validation, and testing. There are a total of 18 classes in the dataset. The vehicles and other static objects are manually annotated as shown in Figure 10. Class names: ['ambulance', 'auto', 'barrier', 'bus', 'car', 'cyclist', 'e-rikshaw', 'illegal vehicle', 'loader', 'motorbike', 'pedestrian', 'scooter', 'tractor', 'traffic cone', 'traffic police stand', 'truck', 'vikram', 'pedestrian'].

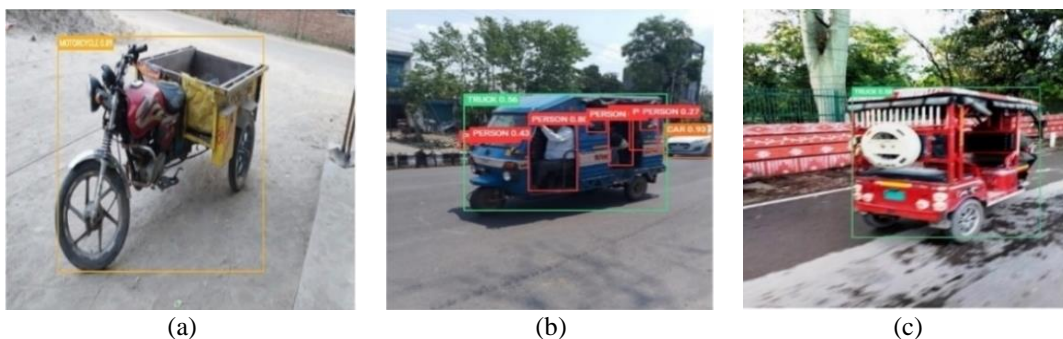


Figure 9. The original YOLOv5, when applied to detect the modified vehicles then it is able to detect the vehicle but inaccurately classifies them, for illustration, (a) modified vehicle is recognized as motorcycle, Indian public transport vehicle "Vikram" is misidentified as (b) truck and E-Rikshaw is recognized as (c) truck

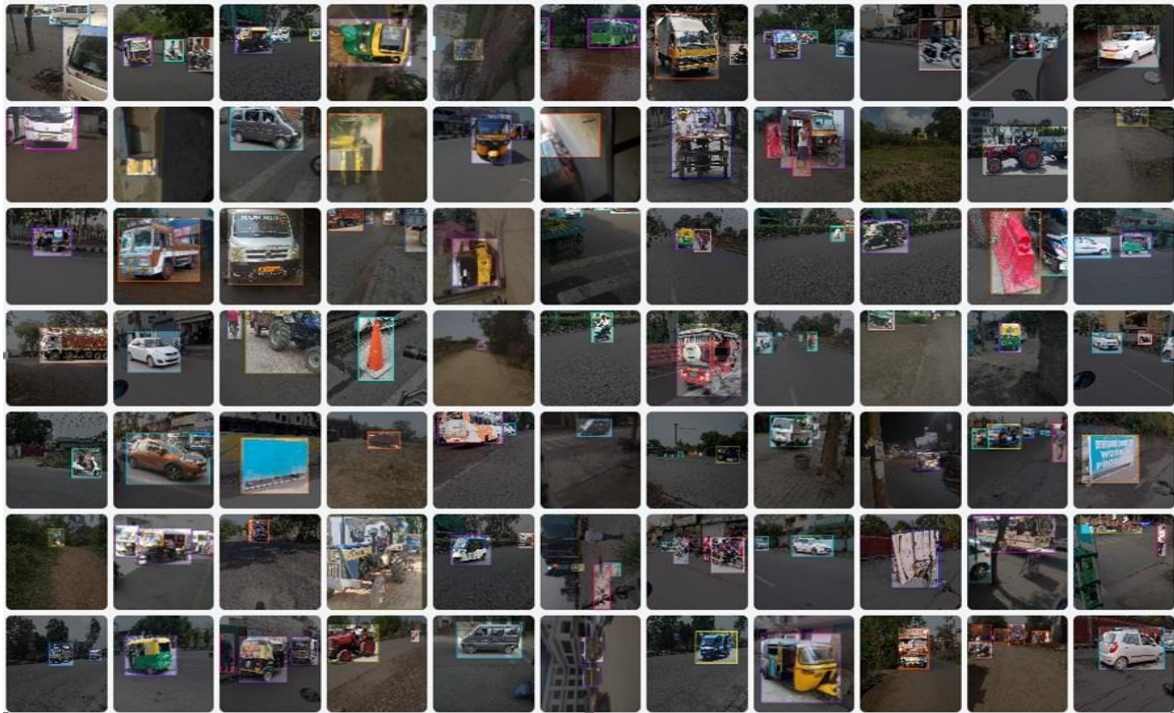


Figure 10. Sample labeled images of the custom dataset collected and annotated manually

3.4. Experimental hardware configuration

The design and implementation of our model and baseline procedures were implemented on the torch framework. Which is an open-source machine-learning library for building and training neural networks. The actual model training was done on the NVIDIA A100-SXM4-40GB GPU provided by Google Colab. Memory-usage 2493 MiB/40960 MiB. For the model training the dataset is split into 80% training, 10% validation, and 10% testing.

3.5. Model evaluation matrices

To evaluate and compare the prediction accuracy of various models, mAP is used. For mAP, recall and precision matrices are utilized.

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

Average precision (AP)= the AP is the average of all precisions. It is the weighted sum of the precisions at each threshold. Calculate the difference between the most recent recall and the next recall which is then multiplied by the most recent precision

$$AP = \sum_{n=0}^{n=m-1} [Recall(n - 1) + Recall(n + 1)] * Precision(n)$$

Recall(n)=0, precision(n)=1, and n=number of thresholds equations should.

mAP: by comparing the predicted box to the ground-truth box, the mAP determines a score. The model’s detections become increasingly accurate as the score increases.

$$mAP = \frac{1}{K} + \sum_{c=1}^{c=K} APc$$

APc=average precision of class c.

K=number of classes.

3.6. Implementation algorithm

The subsequent algorithm 1, provides a systematic elucidation of the practical execution of the proposed models. The dataset is divided into train, validation, and test set. Each folder contains two subfolders, image and label respectively. The image folder contains the images and in the label folder, there is a corresponding text file that contains information about object class and ground truth coordinates.

Algorithm 1. The overall process for practical implementation of YOLOv5-CBAM

```

Input: Image of static and dynamic Indian vehicles and traffic objects
Begin
  Collection of images for a custom dataset
  Do
    Image annotated and ground truth to calculate Intersection over Union IoU
  End
  Begin
  Data preprocessing and augmentation
  90° Rotate: Clockwise, Counter-Clockwise,
  Grayscale: Apply to 25% of images,
  Saturation: Between -25% and +25%,
  Blur: Up to 2.5px, Noise: Up to 5% of pixels
  Dataset
    |-----Training set
    |-----Images folder: contains images
    |-----Label folder: Contains class label and bounding box coordinates.

    |-----Validation set
    |----- Images folder: contains images
    |-----Label folder: Contains class label and bounding box coordinates.
    |-----Test set
    |----- Images folder: contains images
    |-----Label folder: Contains class label and bounding box coordinates.
  End
  Begin
  YOLOv5-CBAM
  Do
    Implementation using Pytorch
    Edit yml file for a custom dataset
    Edit network architecture
    Images resize 640X640
    Set the number of epochs
    Save weights
    Use mAP to evaluate model
  End
  Run the model for the test set
  End
  Output: Test image with bounding box and confidence score

```

4. RESULTS

This section depicts the results of real-time testing based on the mAP score of the developed model.

4.1. Comparative analysis of YOLOv5 and improvised YOLOv5-CBAM

The experimental findings indicate that the YOLOv5-CBAM network outperforms the YOLOv5 network. Figure 11 illustrates the initial YOLOv5m model achieving a mAP of 41.3% after being trained for 300 epochs. There has been a significant improvement in the average recall rate, average recognition precision, and other performance measures, particularly for a classification task with similar items. As a result, it is appropriate and reasonable to use CSP and CBAM to extract the key features, which in turn serve to improve network detection layers and increase the identification efficiency of similar Indian traffic objects, modified cars, and static objects. With its detection results of 98.2% mAP, 98.4% precision shown in Figure 12, and 94.8% recall rate, which satisfy the requirements of static and dynamic object detection. It may be said that the suggested YOLOv5-CBAM network performs the best overall in the task of object detection task.

In the following section, the proposed YOLOv5-CBAM is compared with other deep-learning models. Table 1, represents the comparative values of mAP, precision, and recall acquired by our model. With original YOLOv5m model. It is clearly shown that our model has achieved better results in comparison to YOLOv5-CBAM. Table 2, illustrates the performance of our model in detecting objects in the image of size 640 by 640 Due to the CSP and CBAM layers our model has a large number of parameters but good accuracy.

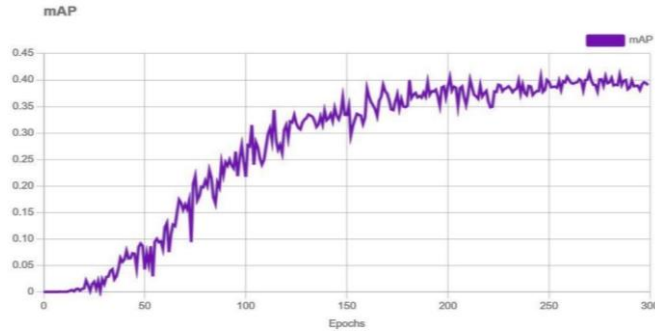


Figure 11. Graph plot of mean average precisions original YOLOv5 mAP

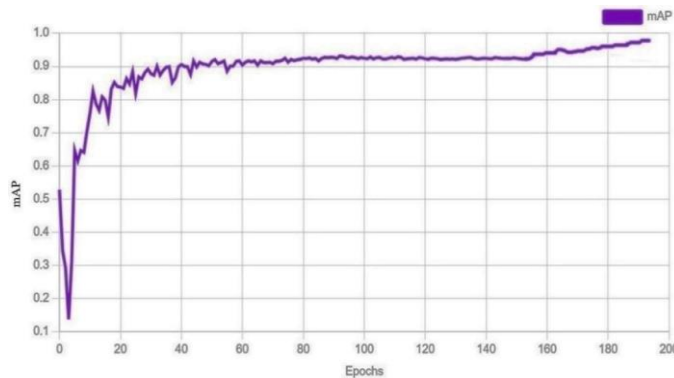


Figure 12. Graph plot of mean average precisions 98.2 mAP in 194 epochs, achieved by proposed YOLOv5-CBAM

Table 1. The model performance on different scales

| Model | mAP | Precision | Recall |
|-------------|-------|-----------|--------|
| YOLOv5m | 41.3% | 61.9% | 38.6% |
| YOLOv5-CBAM | 98.2% | 98.4% | 94.8% |

Table 2. Comparative analysis of proposed YOLOv5-CBAM with different sizes of YOLOv5

| Model | Size | mAP | Param | Flops |
|-----------|------|------|-------|-------|
| YOLOv5s | 640 | 37.0 | 1.9 | 4.5 |
| YOLOv5m | 640 | 41.3 | 21.2 | 49.0 |
| YOLOv5x | 640 | 50.7 | 86.7 | 109.1 |
| YOLO-CBAM | 640 | 98.2 | 90.5 | 115.2 |

4.2. Effectiveness of YOLOv5-CBAM

YOLO algorithm divides the image into grids and predicts bounding boxes with a confidence score for each grid cell. The box with the highest confidence score will result in the best detection. Therefore, the loss, which is the sum squared error between the predicted box and ground truth, should be minimal. The box that has the highest intersection over union (IoU) is considered to calculate the loss for the true positive. For this purpose, the sum squared error between the ground truth and the predicted box is calculated. Losses that could happen in YOLOv5 include class loss (BCE loss) - loss of objectness (BCE loss) - loss of location (CIoU loss). Class loss (Lcls), object loss (Lobj), and location loss (Lloc) are illustrated in (12) and λ is the balance coefficient.

$$Loss = \lambda_1 Lcls + \lambda_2 Lobj + \lambda Lloc \tag{12}$$

The class loss (Lcls).

$$Lcls = \sum_{a=0}^m \sum_{b=0}^B I_{a,b}^{obj} [p^{gt}(c) \log(p^{pred}(c)) + (1 - p^{gt}(c)) \log(1 - (p^{pred}(c)))] \tag{13}$$

Where in (13) Ia,b illustrate the fixed and dynamic object present in the bth anchor box of the ath cell. Small c is the actual class label and p^{gt}(c) is the ground truth and p^{pred}(c) is the predicted box. The box loss (Lbox),

$$Lbox = \lambda_{coord} \sum_{a=0}^m \sum_{b=0}^B I_{a,b}^{obj} (1-CIoU) \quad (14)$$

$$CIoU = IoU - D^2 \frac{D^2}{L^2} - \alpha v \quad (15)$$

$$IoU = \frac{|Box^{pred} \cap Box^{gt}|}{|Box^{pred} \cup Box^{gt}|} \quad (16)$$

$$\alpha = \frac{v}{(1+IoU)+v'} \quad (17)$$

$$v = \frac{4}{\pi} \left(\tan^{-1} \frac{w'}{h'} - \left(\tan^{-1} \frac{w}{h} \right) \right)^2 \quad (18)$$

the object loss,

$$Lobj = \lambda_{obj} \sum_{a=0}^m \sum_{b=0}^B I_{a,b}^{obj} (X^{gt} - X^{pred})^2 = \lambda_{absobj} \sum_{a=0}^m \sum_{b=0}^B I_{a,b}^{obj} (X^{gt} - X^{pred})^2 \quad (19)$$

In (14) λ_{coord} is the location coordinates coefficient and illustrates the bth anchor box of the ath cell I which object is present. In (15)-(18) Box^{pred} is the predicted box around the object and Box^{gt} is the ground truth. So IoU is the division of the common area by the total area covered by both boxes. C is the most accurately fitted box that covers the Box^{gt}. L is the diagonal length of the C. D is the distance between the predicted and ground truth box. A α is a positive trade-off and v is the aspect ratio. The width and height of the box are w and h. In (19), X^{pred} is the confidence score of the predicted box Box^{pred}. And X^{gt} is the confidence score of the ground truth box Box^{gt} are the box weight that contains the object. Box loss measures the effectiveness of the model as how effectively it locates the center of the predicted object and how effectively the predicted bounding box surrounds an object. Classification loss indicates whether the algorithm is effective in the prediction of the appropriate class of an object. As shown in Figure 13, the losses for box, class, and distribution focal loss loss decreased while training.

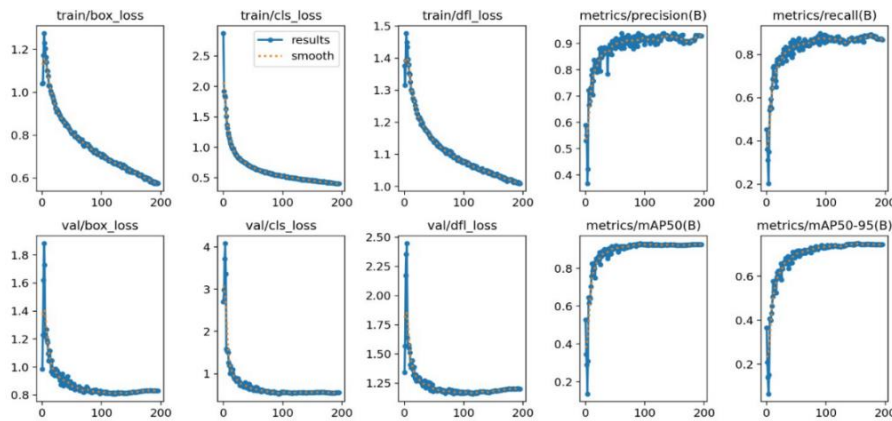


Figure 13. Comparative losses during training and validation and mAP at 0.5 and 0.95 thresholds respectively

The box loss graph represents the model efficiency in predicting the bounding box compared to the ground truth. After training, in the box loss graph epochs are on the x-axis, and box loss is represented on the y-axis. when the training starts, initially the box graph loss (train/box_loss) starts at a high level and steadily decreases with the progress of training. The box loss graph (val/box_loss) during validation is also decreasing after every epoch. YOLOv5-CBAM is a multi-class object detection method that can localize and predict the class of objects in real time. Class loss graphs represent how efficiently YOLOv5-CBAM predicts the class score. Class loss during training and validation should be minimal. The class loss graphs (train/cls_loss and val/cls_loss) illustrate that the proposed model is doing good in predicting the class score because class loss is gradually heading toward the minimum. Another loss function known as distribution focal loss (df_l_loss)

captures the object detection aspect of the capability of the model if the model is extended to perform multiclass object detection. Rather than accepting the bounding box coordinates as scalar values, DFL takes bounding box coordinates to be distributions. The position of an object in an image can be described more correctly if it is set as a probability density rather than a point. The focal mechanism in DFL gives higher emphasis on making samples that are challenging for selection and more difficult to predict during the learning process. During training and validation, the `dfl_loss` is leading towards minimum value which represents that the proposed model is performing better object detection with multi-class prediction. Graph plot of mAPs in Figure 12, showing that 98.2% mAP in 194 epochs and around 90% mAP at 0.5 and .95 threshold values respectively, achieved by proposed YOLOv5-CBAM predictions were generated for the novel and unknown images in our test set after training our model. Figure 10 demonstrates that the proposed approach can more accurately detect both static and moving traffic items. However, it struggles to detect illegal vehicles especially if it is placed far from the camera.

The proposed YOLOv5-CBAM successfully performs object detection on test images. The Figure 14, shows the overall detection with the class score. It is shown that YOLOv5-CBAM successfully detects objects. YOLOv5-CBAM first detects the objects then the PAF method estimates the pose of pedestrians detected by the model. Figure 15, shows that the pedestrian is in a walking pose and crossing the road bypassing a fixed object (barricade) using the PAF method.



Figure 14. Images from the test dataset illustrate the effectiveness of detecting traffic objects with class confidence

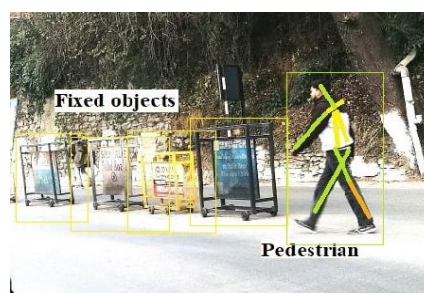


Figure 15. Fixed and dynamic object detection with pedestrian pose estimation. Posture estimation will help in identifying the intention of the pedestrian whether he is crossing the road or not

5. DISCUSSION

In the context of the best results, the mAP is the measure to evaluate the model performance. The proposed YOLOv5-CBAM achieved good results in object detection in comparison with other YOLO-based model networks. Table 3, shows the comparative analysis of predicted accuracy achieved by different object detection models in recent years. It is clearly visible that our model outperforms other models.

Table 3. Comparative analysis of proposed YOLOv5-CBAM with different object detection models

| Year | Model | Dataset | mAP |
|---------------|---|---|--------|
| [24], 2019 | Tinier-YOLO | PASCAL VOC and COCO | 65.7% |
| [25], 2020 | Vehicle detection and classification model based on SSD | Traffic vehicles | 77.31% |
| [26], 2021 | Am improved Yolo v5-Ghost | Vehicle images from virtual environment CARLA | 80.76 |
| [27], 2021 | YOLOv2 based model | KITTI dataset | 94% |
| [28], 2022 | Improved YOLOv5 for vehicle detection | Vehicle images captured by UAV | 89.6% |
| [29], 2022 | YOLO based model | Thermal infrared images unmanned aerial vehicles (UAVs) | 88.69% |
| Present model | YOLOv5-CBAM | Static object images (traffic cone, traffic police stand, barrier) and movable objects (vehicles) | 98.2% |

Finally, the key findings of the present study emphasize the following points.

- i) Features play an important role in object detection therefore maintaining the important features from the previous layer of the network helps in more accurate object detection and recognition.
- ii) The uncertainty in India's traffic scenario makes it more challenging for AV in the Indian scenario, people are used to modifying vehicles for their business, therefore, we have prepared a custom dataset of Indian vehicles to evaluate the performance of the proposed model.
- iii) CSP network is used in the proposed model to improve object detection. It divides the features map to generate more diverse and rich features and enhances feature learning additionally CSP module also helps in reducing the gradient vanishing problem.
- iv) CBAM helps the proposed model in accurate detection by extracting more important and relevant features on the channel and in the spatial domain. CBAM and DSAM mainly have two structures namely CAM and SAM.
- v) The proposed YOLOv5-CBAM can accurately detect multiclass objects in Indian traffic scenarios. In the first step model performs object detection and secondly by adding the PAF model with YOLOv5-CBAM we can detect the pedestrian structure and by seeing the posture model can predict the motion of a pedestrian.

Autonomous driving is an emerging area to research due to its vast application in transportation. It focuses on its applicability in transportation for people who cannot drive. Autonomous driving could be a solution to transportation for physically challenged people as well as elderly people. In developed countries, AV are being prepared and are now in their testing phase. But in developing countries like India research on AV is still in its early stages the reason behind this is the traffic scenario is complicated and uncertain. So to develop fully automated vehicles for Indian traffic, lots of research and development are still required. The present study proposed an improvised deep learning model to develop an autonomous driving vehicle that could able to detect Indian vehicles more accurately and can also predict the uncertain intention of pedestrians to cross the road. The study will help the development of an autonomous vehicle for Indian society. A significant limitation of the present study is the vast diversity of Indian cars. For the success of fully automated vehicles in Indian traffic, a big and more extensive dataset is required. This will enable AV to accurately detect and identify various types of vehicles, allowing them to make appropriate decisions based on this information.

6. CONCLUSION

Intelligent transportation is a necessity in modern traffic scenarios. Self-driving or autonomous vehicle, which is an integration of artificial intelligence and advanced hardware, is the key technology in intelligent transportation. Real-time decision-making is a very important factor for driving in an autonomous vehicle. Computer vision shows excellent possibilities in AV. It is an artificial intelligence field that can provide vital visual data from nearby traffic that can be helpful for AV to decide whether to go left or right or to accelerate or apply the brakes accordingly. Real-time detection of various static or moving traffic participants is critical in self-driving cars. Fixed objects in traffic scenes include barriers, barricades, and traffic cones. A real-time detection method for the Indian traffic system is proposed here on a custom dataset

containing images of dynamic and fixed traffic objects. Concerning precision, recall, and mAP, the enhanced YOLOv5-CBAM model and the original YOLOv5 network have been trained and evaluated on different scales. The experimental result showed that the proposed YOLOv5-CBAM achieved better mAP than the original YOLOv5 on multiclass object detection. Pedestrians and cyclists may come across the AV. The possibility of AV collisions with pedestrians or cyclists is handled using the PAF approach which identifies the biker or pedestrian's intent to cross the road.

Future studies have the potential to expand the current work. Future studies can expand the dataset to include more diverse images of different illuminations, blurry images due to motion, and images with noise from unwanted lights in rain or snow. The proposed model can detect images in real time. AV can be used it for safe driving in uncertain traffic, such as in Indian traffic scenes. Apart from this, it is also meant for discovering new ways to enhance detection accuracy and speed in real-time applications like AV.




REFERENCES

- [1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [2] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.
- [3] 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. 2015-January, pp. 91–99, 2015.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [5] W. Liu *et al.*, "SSD: single shot multibox detector," in *Computer Vision—ECCV 2016: 14th European Conference*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.
- [6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [7] A. Juyal, S. Sharma, and P. Matta, "Deep learning methods for object detection in autonomous vehicles," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, Jun. 2021, pp. 751–755, doi: 10.1109/ICOEI51242.2021.9452932.
- [8] G. Jocher, R. Munawar, A. Octopus, and S. Waxmann, "Ultralytics YOLOv5 architecture," *docs.ultralytics.com*, 2024. https://docs.ultralytics.com/yolov5/tutorials/architecture_description/#1-model-structure.
- [9] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: a new backbone that can enhance learning capability of CNN," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.
- [10] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.
- [11] Y. Luo, P. Cai, Y. Lee, and D. Hsu, "GAMMA: a general agent motion model for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3499–3506, Apr. 2022, doi: 10.1109/LRA.2022.3144501.
- [12] J. Han, Y. Liao, J. Zhang, S. Wang, and S. Li, "Target fusion detection of LiDAR and camera based on the improved YOLO algorithm," *Mathematics*, vol. 6, no. 10, p. 213, Oct. 2018, doi: 10.3390/math6100213.
- [13] T. Drossi, S. Ghosh, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Systematic testing of convolutional neural networks for autonomous driving," *arXiv preprint arXiv:1708.03309*, 2017, doi: 10.48550/arXiv.1708.03309.
- [14] Y. Yin, H. Li, and W. Fu, "Faster-YOLO: an accurate and faster object detection method," *Digital Signal Processing*, vol. 102, p. 102756, Jul. 2020, doi: 10.1016/j.dsp.2020.102756.
- [15] R. Zhang, Y. Yang, W. Wang, L. Zeng, J. Chen, and S. McGrath, "An algorithm for obstacle detection based on YOLO and light filed camera," in *2018 12th International Conference on Sensing Technology (ICST)*, Dec. 2018, pp. 223–226, doi: 10.1109/ICSensT.2018.8603600.
- [16] M. H. Putra, Z. M. Yussof, K. C. Lim, and S. I. Salim, "Convolutional neural network for person and car detection using YOLO framework," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1–7, pp. 67–71, 2018.
- [17] S. Chen and W. Lin, "Embedded system real-time vehicle detection based on improved YOLO network," in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Oct. 2019, pp. 1400–1403, doi: 10.1109/IMCEC46724.2019.8984055.
- [18] A. Wong, M. Famuori, M. J. Shafiee, F. Li, B. Chwyl, and J. Chung, "YOLO nano: a highly compact you only look once convolutional neural network for object detection," in *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, Dec. 2019, pp. 22–25, doi: 10.1109/EMC2-NIPS53020.2019.00013.
- [19] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Multispectral object detection for autonomous vehicles," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, Oct. 2017, pp. 35–43, doi: 10.1145/3126686.3126727.
- [20] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017, pp. 129–137, doi: 10.1109/CVPRW.2017.60.
- [21] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12993–13000, Apr. 2020, doi: 10.1609/aaai.v34i07.6999.
- [22] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, pp. 7291–7299, doi: 10.1109/CVPR.2017.143.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014, doi: 10.48550/arXiv.1409.1556.




- [24] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: a real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2019, doi: 10.1109/ACCESS.2019.2961959.
- [25] Y. Rao, G. Zhang, W. Zhou, C. Wang, and Y. Lv, "Deep convolutional neural network based traffic vehicle detection and recognition," in *IoT as a Service: 5th EAI International Conference, IoTaaS 2019*, Xi'an, China: Springer, 2020, pp. 427–438.
- [26] T.-H. Wu, T.-W. Wang, and Y.-Q. Liu, "Real-time vehicle and distance detection based on improved Yolo v5 network," in *2021 3rd World Symposium on Artificial Intelligence (WSAI)*, Jun. 2021, pp. 24–28, doi: 10.1109/WSAI51899.2021.9486316.
- [27] X. Han, J. Chang, and K. Wang, "Real-time object detection based on YOLO-v2 for tiny vehicle object," *Procedia Computer Science*, vol. 183, pp. 61–72, 2021, doi: 10.1016/j.procs.2021.02.031.
- [28] Z. Chen, L. Cao, and Q. Wang, "YOLOv5-based vehicle detection method for high-resolution UAV images," *Mobile Information Systems*, vol. 2022, 2022, doi: 10.1155/2022/1828848.
- [29] C. Jiang *et al.*, "Object detection from UAV thermal infrared images and videos using YOLO models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102912, 2022, doi: 10.1016/j.jag.2022.102912.

BIOGRAPHIES OF AUTHORS






Amit Juyal    Ph.D. Scholar in Department of Computer Sciences and Engineering, Graphic Era Deemed to be University. He received a bachelor degree in Science from HNB Garhwal University. He has done her Masters of Computer Application from Uttarakhand Technical University and subsequently received an M. Tech. degree in Computer Science and Engineering from Graphic Era University. His research interest includes machine learning and deep learning. He can be contacted at email: amitjuyal26@gmail.com.



Prof. (Dr.) Sachin Sharma    associate dean, international affairs and professor, Department of Computer Science and Engineering at Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. He is also Co-Founder and Chief Technology officer (CTO) of IntelliNexus LLC, Arkansas, USA based company. He also worked as a Senior Systems Engineer at Belkin International, Inc., Irvine, California, USA. He received his Philosophy of Doctorate (Ph.D.) degree in Engineering Science and Systems specialization in Systems Engineering from University of Arkansas at Little Rock, USA with 4.0 out 4.0 GPA and M.S. degree in Systems Engineering from University of Arkansas at Little Rock with 4.0 out 4.0 GPA. He has authored/coauthored over 200 publications in the form of books, patents, and papers in refereed journals and conference proceedings. He holds thirty US, Indian and international patents in the area of IoT, wireless communication and AI. He has also served as a guest reviewer for several special issues of IEEE/ACM transactions. His research interests include wireless communication networks, IoT, vehicular ad hoc networking, and network security. He can be contacted at email: sachin.cse@geu.ac.in.



Shuchi Bhadula    associate professor in Department of Computer Sciences and Engineering, Graphic Era Deemed to be University. She is meticulous and self-motivated IT professional having an experience of 14+ years in the field of teaching. She received her bachelor's degree in Science from HNB Garhwal University. She has done her Masters of Computer Application from Uttarakhand Technical University and subsequently received an M. Tech. degree in Computer Science and Engineering from Graphic Era University. She received her Philosophy of Doctorate (Ph.D.) degree in Computer Science and Engineering from Graphic Era Deemed to be University. Her research interest includes healthcare, AI, IoT, vehicular network, and software engineering. She can be contacted at email: bhadula.shuchi@gmail.com.