

Research Article

ObjectDetect: A Real-Time Object Detection Framework for Advanced Driver Assistant Systems Using YOLOv5

Jamuna S. Murthy,¹ G. M. Siddesh,² Wen-Cheng Lai ,^{3,4} B. D. Parameshachari ,⁵ Sujata N. Patil,⁶ and K. L. Hemalatha⁷

¹Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, India

²Department of Information Science Engineering, M S Ramaiah Institute of Technology, India

³Bachelor Program in Industrial Projects, National Yunlin University of Science and Technology, Taiwan

⁴Department of Electronic Engineering, National Yunlin University of Science and Technology, Taiwan

⁵Department of TCE, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka 570016, India

⁶Department of Electronics and Communication Engineering, KLE Dr. M.S. Sheshgiri College of Engineering and Technology, Belagavi, 590008 Karnataka, India

⁷Department of ISE, Sri Krishna Institute of Technology, Bengaluru, India

Correspondence should be addressed to Wen-Cheng Lai; wenlai@mail.ntust.edu.tw

Received 22 February 2022; Revised 25 March 2022; Accepted 3 May 2022; Published 14 June 2022

Academic Editor: Alessandro Bazzi

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

In the recent times, there has been a lot of speculation related to advanced driver-assistance system (ADAS) which provides best driving experience for the drivers. ADAS technology helps to detect the unhealthy driving conditions which lead to road accidents today. Road accidents are basically caused due to distracted driving, over speeding, drink and drive, foggy weather, no proper headlights, or due to some object which suddenly trespasses the vehicle. Today the major advancements in ADAS include parking assistance, road traffic detection, object detection on highways, and lane detection. But the major risk limitation in ADAS system is the speed and time at which the object is detected and tracked. Several algorithms such as R-CNN, Fast R-CNN, and YOLO were used for effective object detection and tracking earlier, but sometimes, the system do fail to detect due the speed factor. Hence, the proposed work presents a novel approach called “A Real-Time Object Detection Framework for Advanced Driver Assistant Systems” by implementing the state-of-the-art object detection algorithm YOLOv5 which improves the speed in detection of object over real-time. This paper provides a comparison between other state-of-the-art object detectors such as YOLOv3 and YOLOv4. Comparison is done based on mean average precision (mAP) and frames per second (FPS) on three benchmark datasets collected as a part of research findings. YOLOv5 proves to be faster and 95% accurate than the other object detection algorithms in the comparison. This framework is used to build a mobile application called “ObjectDetect” which helps users make better decisions on the road. “ObjectDetect” consists of a simple user interface that displays alerts and warnings.

1. Introduction

According to the WHO, approximately 1.3 million people die each year due to road traffic crashes [1]. With a rise in accidents and with the increase in the number of vehicles, ADAS has become a vital part of the driving experience. Prior warnings seconds before an incident can help the driver handle the situation in a better manner. ADAS has

emerged as an extremely vital tool with respect to safety in the automobile industry. Notable automotive giants such as MG Astor, BMW, and Mahindra XUV700 have stepped in to integrate ADAS into their models [2]. Existing ADAS technologies operate on visual cameras [3], RADARs [4, 5], and LiDARs [6] for the object detection. ADAS mainly depends on features such as high speed, high accuracy, low cost, and low power consumption. Apart from these factors

ADAS should effectively work at three situations, i.e., traveling in rural roads, urban roads, and in highways. By using sensor technology, the goal of object detection is achieved; higher rate sensors are too costly and consume more power. Also, sensors become weak based on continuous operation. Hence, by realizing the importance of speed and cost factor, our state-of-the-art methodology included deep learning approach to address this issue by implementing system called “A Real-Time Obstacle Detection Framework for Advanced Driver Assistant Systems” by implementing the state-of-the-art object detection algorithm YOLOv5.

The ADAS is one of these technologies. It allows the autonomous car to get real-time support in particular traffic circumstances and detect threats from nearby objects using on-board sensors. The development of ADAS technology has accelerated the transition to autonomous driving. Based on visual data collected from sophisticated sensors like cameras, a TSR system may detect one or more traffic signs. Similarly, a greater grasp of road sceneries leads to a better awareness of the surroundings, which relates to the driving space of cars on the side of the road terrain. For example, employing on-board sensors, ADAS allows an autonomous car to get real-time support in certain traffic situations and detect risks related to adjacent objects. ADAS technology has facilitated the rapid evolution of autonomous driving. An object detection system may identify one or more traffic lights based on visual input from sophisticated sensors such as cameras. Similarly, a better understanding of road scenes leads to a better understanding of the surrounding environment, which is relevant to vehicle driving space on the side of the road terrain.

Major advancements in the creation of revolutionary advanced technologies, as well as the widespread deployment of fixed and mobile sensors, such as image sensors, have aided their usage in the road traffic management and monitoring process. Because of advancements in computer vision research, intelligent transportation systems (ITS) have undergone a significant transformation in order to lessen the effect of human lives lost as a consequence of road accidents and rising traffic congestion [7]. Furthermore, significant progress in the computer vision domain has been made due to the rapid evolution of machine learning algorithms, particularly with the enormous growth in traffic data volumes (big data), the emergence of deep neural networks (DNNs), and the development of powerful computers with graphics processors, such as graphics processing units (GPUs). However, some vision-based applications, such as real-time embedded systems, need a significant quantity of memory and fast processing rates. Indeed, segmentation-based road recognition is one of the most difficult problems in computer vision [8], which entails investigating and detecting the vehicle’s surroundings. Unlike traditional approaches that rely on hand-crafted features such as edges and corners, deep learning models are trained incrementally using enormous amounts of data, automating the process of obtaining and training hierarchical feature representations.

Proposed framework consists of three major modules, object extraction, object detection-tracking, and object visualization. Visualization module is applied to build an

interactive mobile application called “ObjectDetect” which assists the user by notifying them with unique alerts and warnings. ObjectDetect is aimed towards providing alerts and warnings a few seconds prior based on the real-time data. In order to build proposed “ObjectDetect” framework, a survey was conducted on multiple object detection algorithms such as R-CNN, Fast-RCNN, Faster RCNN, YOLOv3, and YOLOv4. Since the research work is aimed at improving the speed and accuracy factors which were the limitations on previous works, finally, YOLOv5 was chosen. The proposed model is not a per-trained model and is aimed at including a system that will be able to assist drivers in compromising situations by giving a heads up with significant speed and accuracy.

The article is organized in the following manner. Section 2 discusses about the recent studies on autonomous vehicles and object detection methodologies. Section 3 presents the proposed ObjectDetect mechanism on obstacle detection and driver assistance using YOLOv5 model. Section 4 details the experiment configuration and results evaluated for the ObjectDetect method. And Section 5 concludes the contribution of research and advantages of method; then, Section 6 discusses the future extension of ObjectDetect model.

2. Literature Review

Numerous researches are done on different aspects of ADAS and Autonomous vehicles. The IoT-based occlusion technique called multiple targets tracking in occlusion area with interacting object models in urban environments was used for autonomous vehicles to solve the problem of object detection by Chen et al. by using a laser scanner [9]. The different observed shapes on each laser scan made it difficult to identify the object. Hence, proposed system is developed using machine learning approach using YOLOv5 which reduces the occlusion issue. ADAS also includes driver monitoring systems. Driver monitoring system (DMS) helps in keeping track of various facial features of the driver like eyelid and mouth movement. One such system was proposed by Kato et al. [10].

There is a lot of research done in object detection since it plays a crucial role in many of the technologies, to get a better understanding of state-of-the-art object detection techniques and models, cloud-based. Liu et al. conducted a survey of most of the research that provides a clear picture of these techniques. The main goal of this survey was to recognize the impact of deep learning techniques in the field of object detection that has led to many ground breaking achievements. This survey covers many features of object detection ranging from detection frameworks to evaluation metrics [11, 12].

For many region-based detectors, like Fast R-CNN [13], a costly per-region subnetwork is applied several times. In order to address this, Girshick introduced R-FCN by proposing location-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection [14]. One of the major challenges of object detection was to detect and localize multiple objects across a large spectrum of scales and

locations, due to which the pyramidal feature representations were introduced. In this, an image is represented with multiscale feature layers. Feature pyramid network (FPN), one such model to generate pyramidal feature representations for object detection, presents no difficulty and as well as effective but may not be the optimal architecture design. For image classification in a vast search space, the neural architecture search (NAS) algorithm demonstrates favorable results on the productive discovery of outstanding architectures. Hence, inspired by the modularized architecture proposed by Zoph et al., Dai et al. proposed the search space of scalable architecture that generates pyramidal representations. They proposed architecture, called NAS-FPN, which provides a lot of flexibility in building object detection architecture and is adaptable to a variety of backbone models, on a wide range of accuracy and speed tradeoffs [15].

Various detection systems repurpose classifiers by taking a classifier for an object and evaluating it at multiple locations and scales in a test image. For example, R-CNN uses region proposal methods to first produce bounding boxes that are likely to appear in an image and then, on these suggested boxes, run a classifier. These intricate pipelines were slow and hard to optimize. Hence, Ghiasi et al. proposed you only look once (YOLO), an algorithm that is a single convolutional network that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. Unlike R-CNN and other similar algorithms, YOLO is found to be extremely fast and sees the entire image during training and testing, hence making fewer background errors. When trained on natural images and tested on the artwork, YOLO outperforms other algorithms by a wide margin. But YOLO was yet found to lag behind state-of-the-art detection systems in accuracy and struggled to localize some objects precisely [16]. Redmon et al. by focusing mainly on improving recall and localization while maintaining classification accuracy, proposed YOLOv2. It was then found that detection methods are constrained to a small set of objects; hence, they as well proposed a joint training algorithm that allows one to train object detectors on both detection and classification data, using which they trained the YOLO9000 algorithm which was built by modifying YOLOv2 [17].

The majority of the accurate CNN-based object detectors required high GPU power and training in order to achieve their optimal accuracy. High GPU power is essential for achieving accuracy and speed in real-time since it is vital in a car collision or obstacle warning model. Redmon and Farhadi proposed a modified version of the state-of-the-art object detection models, YOLOv5, with significant improvement in the speed and accuracy of the models. An impressive aspect of this model is that it can operate in real-time on a conventional GPU and training as well requires only a single GPU. Hence, using conventional GPUs such as 1080Ti or 2080 Ti, we can train an accurate and extremely fast object detector [18]. Since YOLOv5 outperforms other frameworks, our proposed framework is based on it.

Traditionally, traffic sign identification has been based on colour and form patterns, with two associated stages: detection and classification [19, 20]. After many preprocessing processes, such as data transformation and normalisa-

tion, which consists of identifying areas of interest (ROI) based on colour segmentation and “sliding window” manner, traffic signs are detected in the image. Following the pattern recognition step, the classification stage involves classifying each sign feature into categories such as “speed restrictions” and “pedestrian crossing.” The template-matching technique was used to improve the feature classification process in [19]. The probable traffic indicators are then classified using a shallow neural network (i.e., a multi-layer perceptron (MLP)). Hmida et al. [20] suggested a hardware design that uses a template-matching approach to classify traffic indicators. Similarly, for successful feature extraction and classification, some studies have used shallow classifiers, such as support vector machines (SVMs) or random forests, in combination with local descriptors like the histogram of oriented gradient (HOG), such as [21]. Hmida et al. [22], for example, presented a traffic sign identification system based on linear SVMs and the MNIST dataset. Gecer et al. [23] used a high-performance technique for traffic sign identification based on blob detectors and SVM classifiers, which increased the model’s colour discriminating capacity by obtaining an accuracy rate of 98.94 percent. However, because of the broad variety of road signs in unexpected locations, obscured and tiny road signs, and fluctuating weather conditions (e.g., shadows and lightning), it is difficult to distinguish them using conventional approaches, which is why deep learning techniques are used.

Many studies have applied facial features and convolution network-based object detection models for the autonomous assistance of drivers based on obstacle detection. These models do not possess optimal architecture design and region identification mechanism. The existing methods provide better accuracy of detection of objects with speed of detection as a trade-off metric. Pipelining of existing detectors was unable to detect the larger object spaces. Hence, the proposed solution comes with utilization of conventional GPU power-based pipelined and accurate YOLOv5 framework for obstacle detection on a higher speed.

3. Proposed Work

ADAS is developed with the help of YOLOv5 model with efficient obstacle detection mechanism and faster speed. The object detection is done with the help of mobile application and alerts to the user. Car processing unit detects the real time video of the driver’s view and fed to the model for the accurate and fast detection of objects in urban roads. The input video is processed as frames; each of which acts as input to the object recognition and detection algorithm (YOLOv5). Each frame is processed along three stages in the algorithm, namely, backbone, neck, and head as shown in Figure 1.

- (i) Backbone [24]: CSPDarknet53
- (ii) Neck: concatenated path aggregation networks (PANet) with spatial pyramid pooling (SPP) additional module
- (iii) Head: YOLO layer

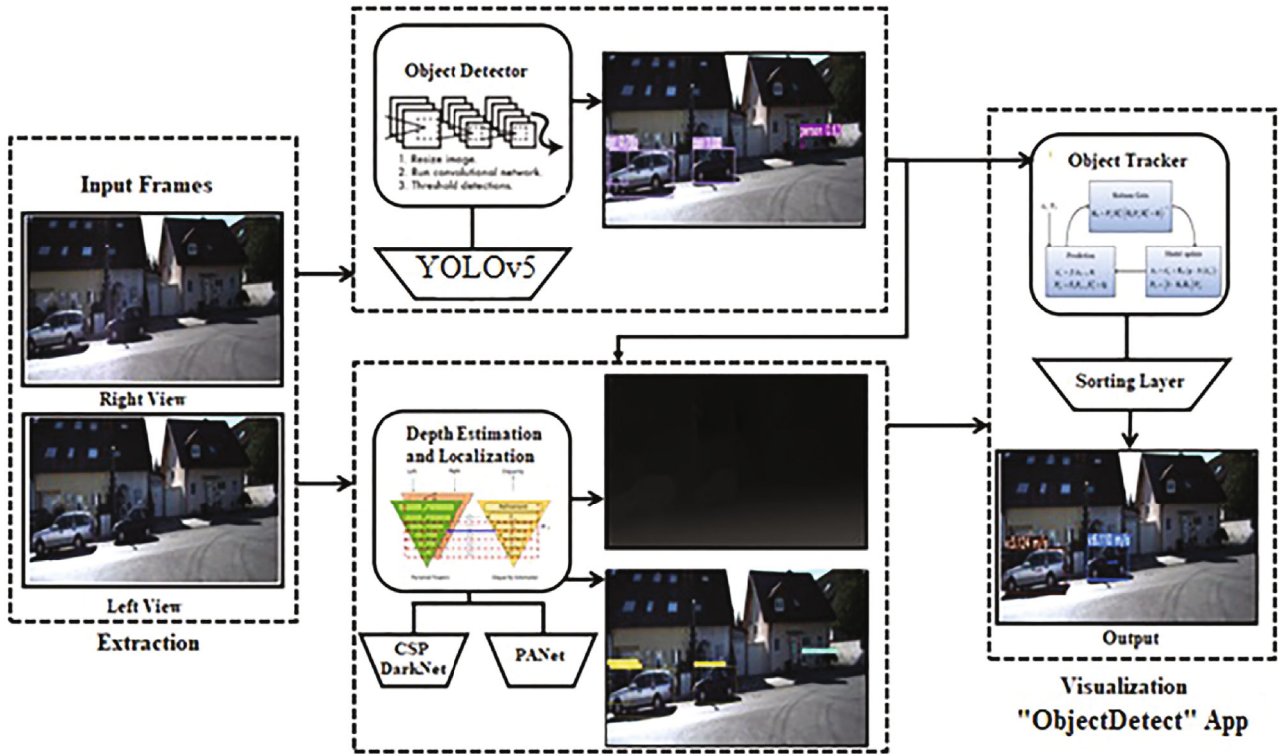


FIGURE 1: Proposed “ObjectDetect” framework for object detection and tracking.

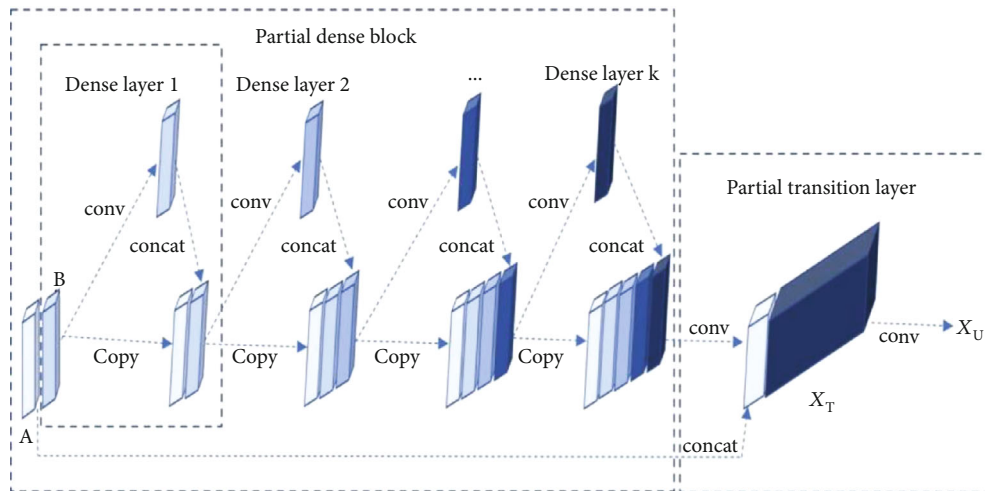


FIGURE 2: The overview of CSPDenseNet.

Proposed system comprises three major modules. These 3 modules are

- (i) Object extraction (backbone and neck)
- (ii) Object detection and tracking (head)
- (iii) Object visualization

3.1. *Object Extraction.* The backbone and neck take images (each of the frames) as input to extract the feature maps using CSPDarknet53 and SPP, PANet path-aggregation.

Darknet53 comprises 53 convolutional layers. For detection tasks, 53 layers stacked on to the original architecture of 53 layers give us 106 layers of architecture.

Step 1. Input: the video input is processed frame by frame

Step 2. CSPDarknet53: cross-stage-partial-connections are concerning used to eliminate duplicate gradient information that occurs while using conventional DenseNet [25]

- (i) In CSPDenseNet, the base layer is divided into 2 parts; here, part A and part B

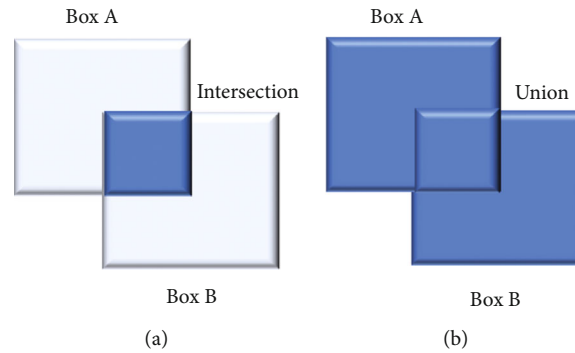


FIGURE 3: (a) The intersection of bounding boxes. (b) The union of bounding boxes.

- (ii) One part will go into the original dense block and is processed accordingly; here, part B is processed in the dense block
- (iii) The other part will directly skip to the transition stage

As a result of this, there is no duplicate gradient information; it also reduces a lot of computations, as shown in Figure 2.

Step 3. Additional layers are added between the backbone and the head using the neck. To aggregate the information, the YOLOv5 algorithm applies a modified path aggregation network [26] with a modified spatial attention module and a modified SPP (spatial pyramid pooling) [27]. Concatenated path aggregation networks [28] with spatial pyramid pooling (SPP) additional modules [26] are used to increase the accuracy of the detector

3.2. Object Detection. Each frame processed in the backbone and neck is then transferred to the head which involves the YOLOv5 algorithm which works using the following techniques:

Step 1. Residual blocks: initially, the input frame is divided into grids. Each grid cell is responsible for detecting the objects present in its cell

Step 2. Bounding box regression: the YOLO algorithm runs such that bounding boxes and confidence scores are predicted around every object present in that particular grid

Every bounding box consists of these attributes: width (bw), height (bh), bounding box center (x, y), and confidence score (c). The confidence score represents how confident and accurate the algorithm is of a particular object in that bounding box. Together with these attributes, YOLO uses a single bounding box regression to predict the probability of an object appearing in the bounding box. Figure 3 shows the YOLOv5 algorithm being run in real-time on a webcam. The algorithm detected objects in the frames by indicating the classes they belong to and the confidence scores representing how sure it is of the objects.

Step 3. Intersection over union (IoU): if no object exists in a grid cell, then the confidence score is zero; else, the confidence score must be equal to the intersection over union (IoU) between the predicted box and ground truth. Here, the ground truth boxes are manually predefined by

the user; hence, greater IoU means greater confidence score, which means higher accuracy of prediction by the algorithm. Filtration of those boxes with no objects is done based on the probability of objects in that box. Nonmax suppression processes eliminate the unwanted bounding boxes, and the box with the highest probability or confidence score will remain [29]

$$\text{IoU} = \frac{\text{Area of } (\text{BoxA} \cap \text{BoxB})}{\text{Area of } (\text{BoxA} \cup \text{BoxB})}. \quad (1)$$

The above Equation (1) IoU calculation is used to measure the overlap between two proposals.

Nonmax suppression [30]: this is used to find the appropriate bounding box among the predicted bounding boxes by the algorithms based on the confidence scores. This is represented in Algorithm 1 below.

Step 4. Final detection: the algorithm detects the object and class probabilities with confidence scores. This is depicted in Figures 4(a)–4(c)

4. Visualization

The final module of the proposed system involves an android-based application. The application inputs a real-time video stream from the device; camera runs an object detection algorithm on it and notifies the user under any case of any condition that requires to be brought to the user's attention and needs to be acknowledged [31]. These conditions could be any obstacle or collision ahead or the user being in close contact with respect to the user's position. With the YOLOv5 algorithm, the system is powerful enough to run object detection in various weather conditions. The alerts are of 3 categories:

- (i) The green alert is shown when there are no threats detected
- (ii) The yellow alert is shown when the threat detected is of low priority, such as stationary objects in front of the vehicle, like animals or pedestrians crossing
- (iii) The red alert is shown when the threat detected is of the highest priority such as objects approaching the car at high speeds

```

Input: Set of proposal boxes  $A$ , corresponding confidence scores  $C$  and overlap threshold  $T$ .
Output: A list of filtered proposals  $F$ .
procedure Sup( $A, m$ )
   $A_{sup} \leftarrow \emptyset$ 
  for  $a_i \in A$  do
    remove  $\leftarrow false$ 
    for  $a_j \in A$  do
      if  $match(a_i, a_j) > \alpha_{sup}$  then
        if  $value(m, a_j) > value(m, a_i)$  then
          remove  $\leftarrow true$ 
    else remove then
       $A_{sup} \leftarrow A_{sup} \cup a_i$ 
  return  $A_{sup}$ 

```

ALGORITHM 1: Non_Max_Supression (Sup).

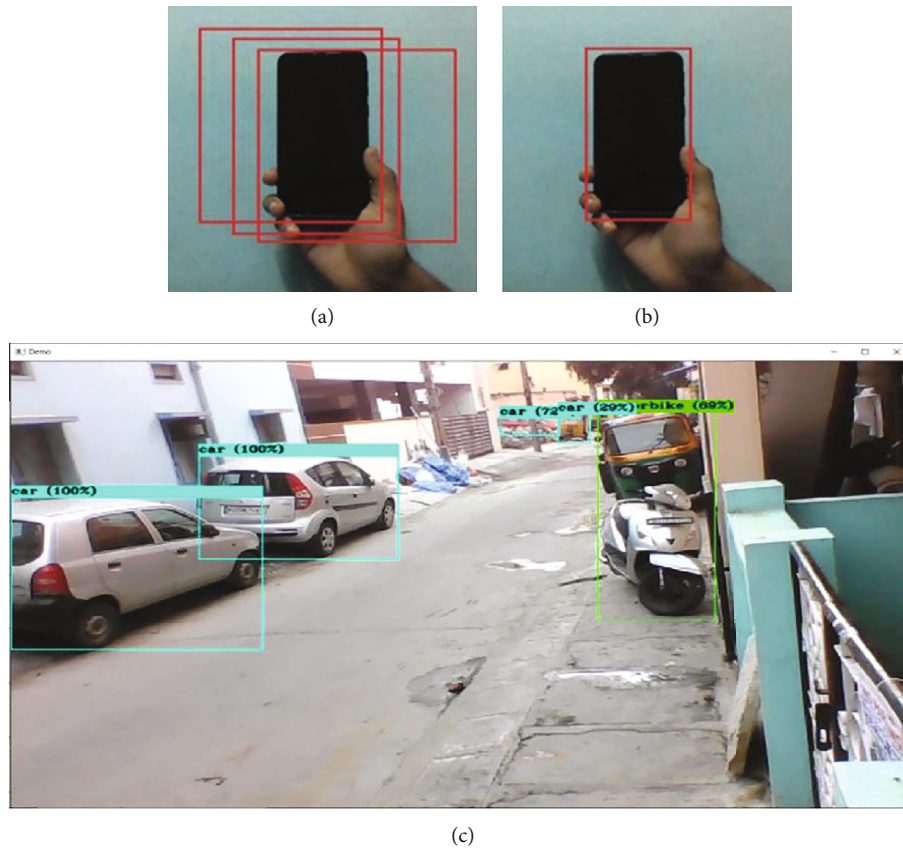


FIGURE 4: (a, b) Bounding boxes and class predictions. (c) Confidence scores around the objects when the algorithm is run on a webcam in real-time.

5. Evaluation Results

With the aim of creating a CNN for real-time operation on a conventional GPU, YOLOv5 was introduced. In the process of doing so, various training improvement methods on the accuracy of the classifier on the ImageNet dataset were tested, and their influence was noted along with the accuracy

of the detector on the MS COCO dataset with the following configuration.

PC specification:

Central processing unit: 11th generation Intel® Core™ i7

Graphic processing unit: NVIDIA®, 16 GB graphic card

Hard disk capacity: 1 TB

OS requirement: iOS/Windows 10/Ubuntu 18



FIGURE 5: YOLOv5 on rural road dataset.



FIGURE 7: YOLOv5 on highway dataset.

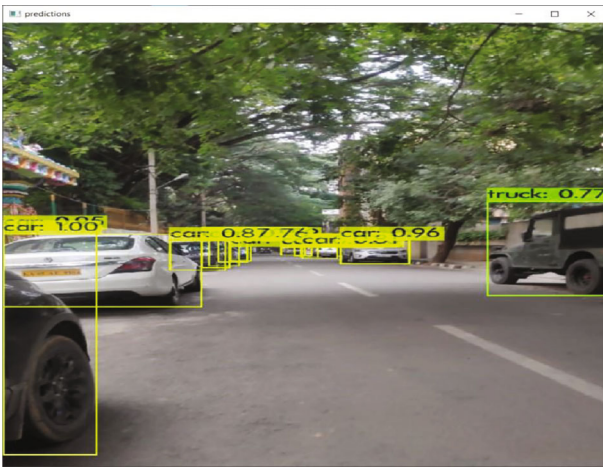


FIGURE 6: YOLOv5 on urban road dataset.

Random access memory: 16-32 GB
 Car computing unit specification:
 NVIDIA Jetson Xavier NX 16 GB
 Central processing unit: ARM®v8.2
 Graphic processing unit: NVIDIA®, CUDA®
 Hard disk capacity: 1 TB
 OS requirement: iOS/Windows 10/Ubuntu 18
 Random access memory: 16-32 GB

While comparing YOLOv5 with other state-of-the-art object detectors, it was found that YOLOv5 improved YOLOv4's AP by 10% and FPS by 12%, and within comparable performance, YOLOv5 ran twice faster than Efficient-Det [32]. It was found that the classifiers' accuracy was enhanced by proposing features such as CutMix and Mosaic data augmentation, class label smoothing, and Mish activation [33, 34].

In order to evaluate the proposed framework, 3 different types of datasets based on three categories such as rural roads, urban roads, and highways were used. The 3 datasets were created by labelled annotations of images which were captured as a novel part of research work. 8% of the collected data consisted of blurry images and images with low visibility. An example of each dataset is shown in Figures 5–7. The 3 different datasets were categorized as explained in Table 1.

TABLE 1: Datasets.

Sl. no.	Datasets	Number of images
1	Rural roads	1,20,000
2	Urban roads	1,24,000
3	Highways	1,50,000

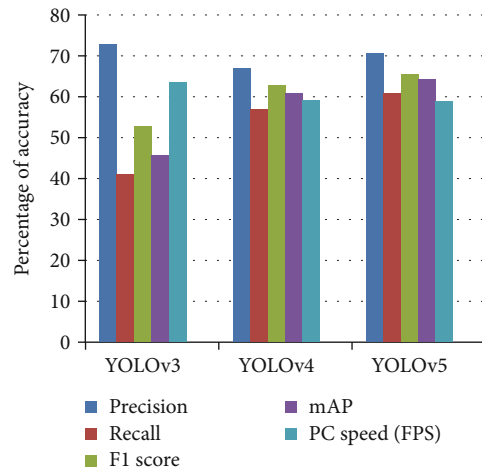


FIGURE 8: Comparative analysis of state-of-the-art models of rural road dataset.

The main goal of the research work is to reduce and increase the accuracy and speed at which the objects are detected; hence, here, the mAP (mean of average precision) and FPS (frames per second) play a very important role. The measures such as precision, recall, *F*-measure, PC speed (FPS), and Jetson speed (FPS) are used to compare the proposed model against two classic algorithms such as YOLOv3 [35, 36] and YOLOv4. The measures are listed below.

$$mAp = \sum_{q=1}^Q \frac{AveP(q)}{Q}, \quad (2)$$

$$F1score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}, \quad (3)$$

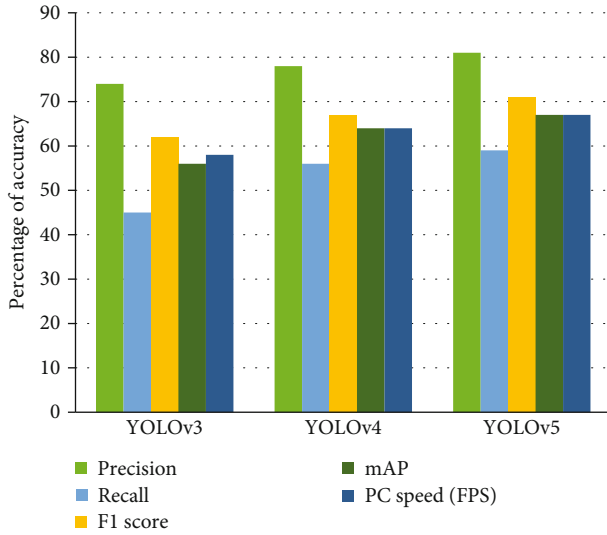


FIGURE 9: Comparative analysis of state-of-the-art models of urban road dataset.

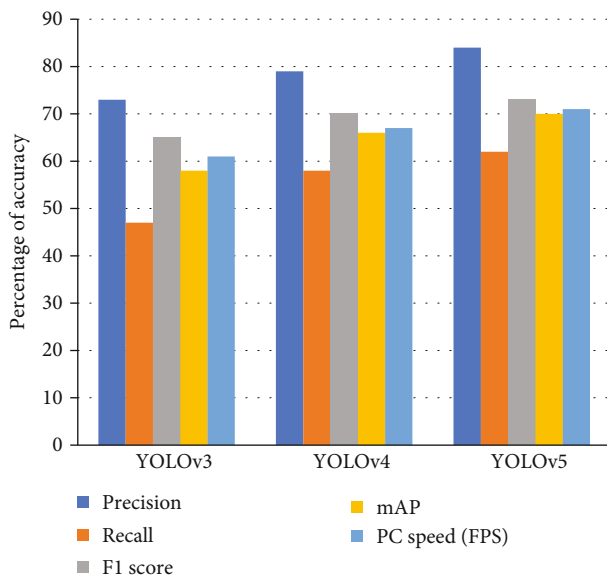


FIGURE 10: Comparative analysis of state-of-the-art models of highway dataset.

$$\text{Precision} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Positive})}, \quad (4)$$

$$\text{Recall} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Negative})}. \quad (5)$$

Figures 8–10 present the comparative analysis of YOLOv5 against other state-of-the-art models in rural roads, urban roads, and highway datasets. Here, YOLOv3 has good precision but has very bad recall and F -measure. Also, the mAP and FPS are very low. But YOLOv4 and YOLOv5 comparatively has balanced scores, but YOLOv5 outperforms other two algorithms because since the proposed works are

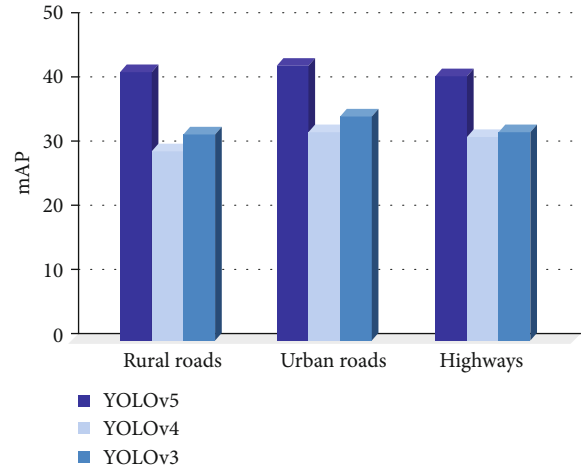


FIGURE 11: Comparison of state-of-the-art object detectors on the custom dataset.

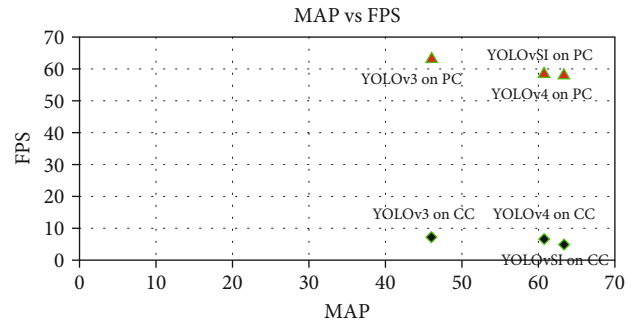


FIGURE 12: Graphical analysis of mAP vs. FPS with PC and CC specifications.

aimed at increasing speed and accuracy, YOLOv5 is the best fit state-of-the-art model for problem definition.

The mAP of the state-of-the-art object detectors such as YOLOv3, YOLOv4, and YOLOv5 was compared using the three datasets, i.e., rural roads, urban roads, and highway datasets given below. The results of this comparison are represented in Figure 11. With respect to mAP, it is clearly seen that YOLOv5 outperforms the other two by a significant margin.

Figure 12 represents a comparative analysis of YOLOv5 with other state-of-the-art object detection algorithms regarding mean average precision (y -axis) and frames per second (x -axis) for PC and CC specifications as listed below in table. From Figure 12, it can be inferred that the YOLOv5 algorithm performs better than others in real-time detection. It achieves an average precision between 67 and 70 and frames per second between 65 and 124.

6. Conclusions

The proposed framework is intended to provide real-time object detection with optimal speed and accuracy to assist the driver. This framework is achieved by implementing the state-of-the-art YOLOv5 algorithm. The whole framework is implemented in the form of three major modules, namely, extraction, detection, and visualization. The first

module, extraction, is used to obtain the feature map of the given input. The detection module identifies and localizes the object present in the input. The last module is used to provide an interface that comprises alerts and warnings. The proposed framework is applied to build the android application called “ObjectDetect” which assists the user by notifying them of significant events that require the user to analyze and decide based on it. The proposed application, “ObjectDetect,” relies majorly on a camera. With the help of some sophisticated cameras, this system can operate under challenging weather conditions. Hence, in the future, we can integrate this system with other sensors, such as LIDAR, to enhance speed and accuracy. The visualization can be improved by integrating “ObjectDetect” with other driver assistance technologies, such as Google Maps and voice assistant. In the future, with the help of a cloud-based approach, the processes can be recorded and analyzed. The cloud-based approach also helps in increasing the accessibility of the application. Raspberry pi can also be used in order to have a smooth flow in the processes and increased efficiency. In the future, the proposed framework can be integrated with the electronic control unit (ECU) present inside the vehicles.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] “National motor vehicle crash causation survey,” *National Highway Traffic Safety Administration Technical Report DOT HS*, vol. 811, 2008.
- [2] <https://www.newsbytesapp.com/news/auto/most-affordable-ad-as-equipped-cars-available-in-india/story>.
- [3] P. Kiran, B. D. Parameshachari, J. Yashwanth, and K. N. Bharath, “Offline signature recognition using image processing techniques and back propagation neuron network system,” *SN Computer Science*, vol. 2, no. 3, pp. 1–8, 2021.
- [4] L. Wang, Y. Zhang, and J. Wang, “Map-based localization method for autonomous vehicles using 3D-LIDAR,” *IFAC-Papers OnLine*, vol. 50, no. 1, pp. 276–281, 2017.
- [5] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.
- [6] S. N. Shivappriya, M. Priyadarsini, A. Stateczny, C. Puttamadappa, and B. D. Parameshachari, “Cascade object detection and remote sensing object detection method based on trainable activation function,” *Remote Sensing*, vol. 13, no. 2, p. 200, 2021.
- [7] Q. Cabanes and B. Senouci, “Objects detection and recognition in smart vehicle applications: point cloud based approach,” in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 287–289, Milan, Italy, 2017.
- [8] P. Jagannathan, S. Rajkumar, J. Frnda, P. B. Divakarachari, and P. Subramani, “Moving vehicle detection and classification using Gaussian mixture model and ensemble deep learning technique,” *Wireless Communications and Mobile Computing*, vol. 2021, 15 pages, 2021.
- [9] J. F. Chen, C. C. Wang, and C. F. Chou, “Multiple target tracking in occlusion area with interacting object models in urban environments,” *Robotics and Autonomous Systems*, vol. 103, pp. 68–82, 2018.
- [10] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.
- [11] S. Shaily, S. Krishnan, S. Natarajan, and S. Periyasamy, “Smart driver monitoring system,” *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 25633–25648, 2021.
- [12] L. Liu, W. Ouyang, X. Wang et al., “Deep learning for generic object detection: a survey,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.
- [13] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: a survey,” 2019, <http://arxiv.org/abs/1905.05055>.
- [14] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [15] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: object detection via region-based fully convolutional networks,” in *Advances in neural information processing systems*, vol. 29, pp. 379–387, 2016.
- [16] G. Ghiasi, T. Y. Lin, and Q. V. Le, “Nas-fpn: learning scalable feature pyramid architecture for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [18] J. Redmon and A. Farhadi, “YOLO 9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
- [19] A. Hechri, R. Hmida, and A. Mtibaa, “Robust road lanes and traffic signs recognition for driver assistance system,” *International Journal of Computational Science and Engineering*, vol. 10, no. 1/2, pp. 202–209, 2015.
- [20] R. Hmida, A. Ben Abdelali, and A. Mtibaa, “Hardware implementation and validation of a traffic road sign detection and identification system,” *Journal of Real-Time Image Processing*, vol. 15, no. 1, pp. 13–30, 2018.
- [21] J. Greenhalgh and M. Mirmehdi, “Real-time detection and recognition of road traffic signs,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1498–1506, 2012.
- [22] R. Hmida, A. B. Abdelali, and A. Mtibaa, “Speed limit sign detection and recognition system using SVM and MNIST datasets,” *Neural Computing and Applications*, vol. 31, no. 9, pp. 5005–5015, 2019.
- [23] B. Gecer, G. Azzopardi, and N. Petkov, “Color-blob-based COSFIRE filters for object recognition,” *Image and Vision Computing*, vol. 57, pp. 165–174, 2017.
- [24] C. L. Su, W. C. Lai, J. Y. Wu, and P. Y. Wang, “Design and implementation of edge computing for detection on embedded electromobility,” in *4th International Conf. on Artificial Intelligence for Industries (AII)*, Laguna Hills, CA, USA, 2021.
- [25] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv5: optimal speed and accuracy of object detection,” 2020, <http://arxiv.org/abs/2004.10934>.

- [26] C. Y. Wang, H. Y. M. 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," in *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.
- [27] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [29] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.
- [30] A. Amidi and S. Amidi, *VIP cheatsheet: convolutional neural networks*, 2018.
- [31] J. Redmon, *YOLOv3: an incremental improvement*/Joseph Redmon, Ali Farhadi-University of, Washington, 2018.
- [32] Y. H. Lee and Y. Kim, "Comparison of CNN and YOLO for object detection," *Journal of the semiconductor & display technology*, vol. 19, pp. 85–92, 2020.
- [33] 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.
- [34] C. Lee, H. J. Kim, and K. W. Oh, "Comparison of faster R-CNN models for object detection," in *In 2016 16th international conference on control, automation and systems (iccas)*, pp. 107–110, Gyeongju, Korea (South), 2016.
- [35] C.-L. Su, W.-C. Lai, and C. Te Li, "Pedestrian detection system with edge computing integration on embedded vehicle," in *International Conf. on Artificial Intelligence in Information and Communication (ICAIIIC)*, Jeju Island, Korea (South), 2021.
- [36] C. L. Su, W. C. Lai, Y. K. Zhang, T. J. Guo, Y. J. Hung, and H. C. Chen, "Artificial intelligence design on embedded board with edge computing for vehicle applications," in *IEEE 3rd International Conf. on Artificial Intelligence and Knowledge Engineering (AIKE)*, Laguna Hills, CA, USA, 2020.