INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Real-time Vehicle Detection implementing Deep Convolutional Neural Network features Data Augmentation Technique

V Sowmya[1]*, **R Radha**[2]

1 Research Scholar, Research Department of Computer Science, SDNBV College for Women, University of Madras, Chrompet, Chennai-600044, India.
2 Associate Professor, Research Dept. of Computer Science, SDNBV College for Women, Chrompet, Chennai-600044, India.

OPEN ACCESS

Received: 25.11.2021

Accepted: 30.12.2021

Published: 21.01.2022

Citation: Sowmya V, Radha R (2021) Real-time Vehicle Detection implementing Deep Convolutional Neural Network features Data Augmentation Technique. Indian Journal of Science and Technology 15(1): 44-53. https://doi.org/10.17485/IJST/v15i1.1908

* **Corresponding author**.

v.sowmy81@yahoo.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background/Objectives: In this progressive Hi-Tech ecosystem, the cutting-edge technologies in the Deep Learning techniques for Vehicle Detection and Classification engendered swift paradigm shifts in diverse operations through the deployment of convolutional neural models in the Traffic Surveillance System. The fundamental element of the Traffic management system constitutes a real-time dynamic image, which forms the base input for vehicle recognition systems. The deep model functionalities on these base static images are highly pragmatic, and a radical approach leads to its successful applicability. **Methods:** This study proposes Faster Region-based Convolutional Neural Network (R-CNN) technique for image-based vehicle detection with significant performance benefits. Essentially, the base network of a pre-trained deep model, fine-tuned VGG-16 is transformed into Faster R-CNN. At this stage, the framework is constructed for a customized finite-capacity vehicle dataset. Subsequently, it is applied to train and test the system. From the performance lens, for further system enhancement, the speedup Bottleneck, and Data Augmentation implementation improve training speed and accuracy. **Findings:** The Experiments demonstrate that the sensitivity factor is 93.5% which provides acceptable results of 87.6% with 0.42s in vehicle detection in aspects of accuracy and execution time. **Novelty :** For our customized dataset, the performance-enhanced detection framework shows an increase of 4% in sensitivity and 3.23s with respect to time as compared to the other existing models. The proposed research is designed for a novel Faster RCNN algorithm that is fine-tuned detection algorithm of vehicles integrating sophisticated approaches for dynamic transformation of the live traffic video stream recording by transposing these real-time traffic videos to image inputs to this optimized detection framework achieving a high sensitivity factor with an efficient computation stack benefiting cost and time.

Keywords: Data Augmentation; Deep Learning; Faster RegionConvolutional Neural Network; Traffic Surveillance System; VGG16 pretrained model; Vehicle Detection

1 Introduction

In the prevailing scenario, object detection is the utmost complex task in the computer-vision domain. Nowadays, the Neural network algorithm has a more prominent thriving impact on object detection. It eliminates the shallow contrarieties in traditional object detection models. The Deep Neural Network achieves significant accuracy in Regions with CNN features. The deeper architecture of CNN can solve the complex feature involved in object detection. The primary Region-Convolutional Neural Network has suggested slightly improved frameworks such as Mask R-CNN, Faster R-CNN, and Fast R-CNN, which achieve better performance, provide accurate results and make real-time object detection tasks and vehicle detection. Nevertheless, it is a challenging method since there is room for occlusion and truncation of vehicles contributing to scale variations in traffic images. This inefficiency in the system has roots in the convolutional neural networks-based object detectors - SSD and Faster R-CNN. These are utilized without the modification of the framework, which is a requirement for the performance enhancement of the existing system. Our proposed approach focuses on modifying the base network to suit the dynamic positioning of different scales by implementing the multi-scale feature maps of CNN [(1)], alternatively using the input images consisting of multiple resolutions.

[(1)] Proposes a pre-trained mobile net framework that is modeled for base networking in the Faster R-CNN. Further, a soft NMS and context-aware ROI-pooling are the methods applied for detecting a fast vehicle. Experiments are conducted and tested on KITTI and LSVH datasets, and the results are 89.20%, 87.8%, 74.72% of average precision on different categories of easy, moderate, and difficult vehicle detection. The accuracy of object detection applying the automatic threshold requires the extensions. [(2)] Proposes a model called Region Proposal Network (RPN) that shares the full image of CNN features. RPN and Fast R-CNN are consolidated into a single network to improve the object bounds and score at each position. Experiments are tested on COCO and PASCAL VOC 2007, 2012 dataset and achieve the result of 78.8% mean Average Precision(mAP). But there is a limitation in handling the objects in the highly congested situation. [(3)] Proposes a vehicle detection method that implements the Faster R-CNN. The combination of Kalman filter and Camshaft are the methods used. Further, a multithread technique is deployed for the detection and tracking of a vehicle. The mean average compute time of detection equals 0.935s per image, and the tracking equals 0.0244s per image. Due to the exposure issue, the system limits the performance of the images in extreme illumination conditions. [(4)] Propose a Fast R-CNN for the detection of a vehicle. Regional proposal and Object recognition are used as separate entities because the object recognition entity is redundant, and it is removed during the training process. The SHRP2NDS database is used to test the method. It achieves 89.2% for the precision and 98.5% for the recall public dataset. Then the results were compared to show the efficacy of the system. [(5)] Proposes a real-time detection method using Faster R-CNN. The Resnet-50 is used as a base network and trained through Faster R-CNN. The experiment shows that a 0.985 sensitivity factor is achieved. Detecting objects in real-time scenes might be improved. [(6)] Propose a method called ERPN (Enhanced Regional Proposal Network). Initially, the Deconvolutional FPN (Feature Pyramid Network) is used to enhance the quality of the regional proposals. Further, PSO (Particle Swarm Optimization) -SVM is applied to distinguish the anchors. Finally, the multi-loss function in RPN is removed. The experiment shows that 31.7% mAP on the COCO dataset and 74.4% mAP on VOC 2012 is obtained. The overall classification accuracy and mAP might be improved.[7] Propose a system called detection of the vehicle using Fast R-CNN. Convolutional Neural Network is implemented in the form of a base network, the raw image fed as input, and the bounding box creation estimates probabilities of features as output. There is Noise while capturing images. Public Dataset is not used to show the efficacy of the system.[8] Proposes SVM with HoG is used to categorize the vehicles from distinct viewpoints. The experiment shows 77.1% of precision, 73.3% of recall, and 0.78 of F-Score. The overall performance measures can be improved.[9] Proposes Accurate Vehicle Proposal Network (AVPN) incorporates feature maps in the process of detecting small-sized objects. AVPN comparison with large and small datasets but shows less in performance.[10] Proposes Hyper Regional Proposal Network (HRPN) that enhances the recall by applying a technique similar to[9]. Next, the RPN replaces the cascade classifier for minimizing the occurrences of false alarms. The model achieves 78.3% of sensitivity with 3.93s.[11] Proposes a paper on vehicle detection; and the recognition of aerial images. This system uses a CNN for regressing the spatial density map in a vehicle over aerial images. The two public datasets, such as Munich and overhead imagery research are used for evaluation. The experiments show 89.5% precision and a 90.51% recall rate. This detection method has an accuracy that still has room for improvement.

The above methods still need improvement and have the limitation of vehicle detection in real-time. To overcome the above challenges, this study proposes a performance-driven approach in real-time vehicle detection utilizing the Faster R-CNN algorithm. The pre-trained VGG-16 is implemented as a base network and trained in a custom vehicle dataset. An image data augmentation technique and bottleneck features are applied for improving accuracy, speed of training, and avoiding overfitting. The performance is evaluated on different metrics - Precision, F-score, Sensitivity/Recall, Specificity, Accuracy, and Confusion matrix. The experiment shows that this proposed system achieves better Simulation time, Sensitivity factor and significant accuracy comparison with other deep learning-based object detection. The primary objective of the proposed method is the detection of multiple vehicles from the surveillance video feed. This detection mechanism is significant in the field of surveillance video analytics The focus of the proposed paper as follows:

- Fine-tuning the Convolutional layers, Max-pooling and Region of Interest layers are used for improving detection accuracy in terms of Sensitivity factor.
- Bottleneck features and Image data augmentation are applied to eliminate overfitting of the proposed model particularly heavy vehicles.
- Better real-time vehicle detection performance even in a crowded scene.
- Better Simulation time is achieved when compared to other models.

Further sections of the paper are formatted as follows: Proposed methodology is explained in detail in section 2. Experiment results and discussion are discussed in section 3, and the conclusion in section 4.

2 Proposed Methodology

The framework is constructed with the performance aspect of the whole system. The architecture of the proposed system constitutes the following; the foundational design incorporates the VGG-16 that is implemented as a base network in the Faster R-CNN model. In the subsequent layer, in Region Proposal Network, the speed Bottleneck feature enables improvement in the speed for training. Further, for enhancement, the Image Data Augmentation technique is employed for over fitting reduction in the model, and the NMS is applied to limit the overlapping of the proposed region. Later, the proposal region is transferred to the ROI pooling layer to produce a fixed-size of R-CNN feature maps. At the final stage, the classifier is built based on VGG16 architecture, which has two outputs – Classification and Regression. Classification facilitates identifying the particular vehicle which is trained on the custom vehicle dataset, and Regression supports the creation of the refined bounding box per detected vehicle. Finally, OpenCV technique is employed to test the images on the trained method. The Faster R-CNN Vehicle Detection Framework on the custom vehicle dataset is shown in Figure 1.

2.1 Dataset

The custom vehicle dataset consists of an aggregate of 3192 images comprising of truck, bus, and motorbike. In this dataset, the input-image data is sourced from diverse reliable resources. The dataset comprises a specific amount of the images downloaded from icrawler, and the major portion of the dataset is from the real-time traffic videos captured in different traffic-congested localities in Chennai (conversion into images using OpenCV technique), and a limited number of images are obtained from the internet (without copyright issues).

These images are then tested on varying conditions. The dataset is distributed as training and testing sets. The vehicle training dataset is divided into two independent entities with an 8:2 ratio, containing 2681 and 511 images in the training and test datasets, respectively.

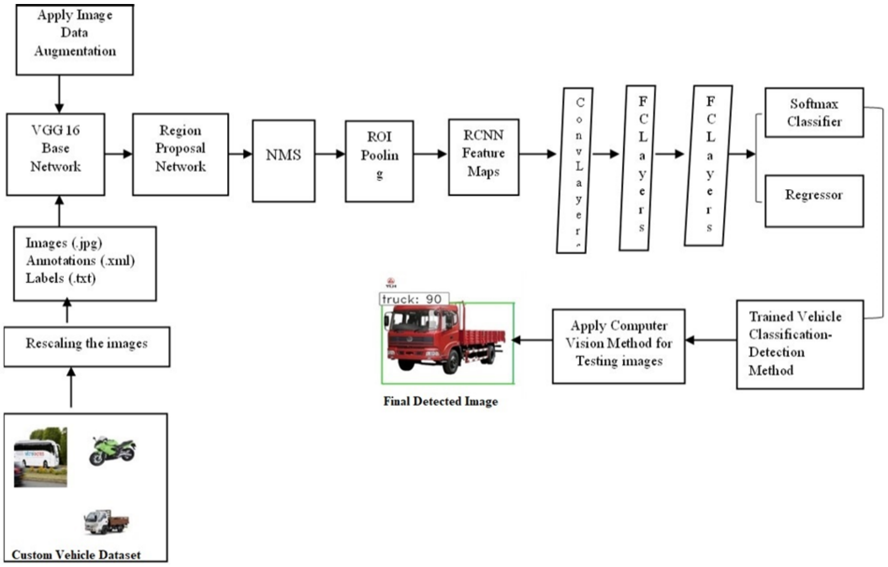


Fig 1. Faster R-CNN Vehicle Detection Framework

2.2 Base Network

R-CNN was first built by Ross Girshick et al., in 2014, an improved version is Fast R-CNN, and then finally with Faster R-CNN in 2016. The Faster R-CNN system is developed integrating the pre trained fine-tuned VGG-16 architecture. The VGG-16 comprises thirteen convolutional layers in conjunction with the two fully-connected layers, max-pooling, and a softmax classifier. In this study, the VGG-16 is used as the base network in Faster R-CNN for vehicle detection. The input image size of VGG-16 is 224x224x3 for classification. The final layer of the network, which adopts fully-connected layers, requires a fixed-length, which is performed by flattening the output of the final convolutional layer. In this proposed model, the Bottleneck features are applied in pre-trained VGG-16 architecture to benefit and build an efficient framework. Bottleneck features aids training of a smaller dataset (multi-class) in the pre-trained model, and besides, eliminate the over-fitting and the significance of the power of feature classification in terms of classes helps faster responses of recognition of images. All these factors contribute to the efficiency attributes of the model, ultimately improving accuracy. A selective search of 2000 regions for every image is performed, which populates a CNN feature vector for each image. The final feature map is 512x7x7.

2.3 Regional Proposal Network (RPN)

Faster R-CNN uses an RPN for generating regions to detect objects. RPN ranks anchor boxes that help to detect objects of different sizes. There are nine anchors at a position of an image. The specific ratios such as 1:1, 1:2, 2:1 and sizes 128x128, 256x256, 512x512 are defined. In this system, the final feature map is 7x7x512 dimensions from a pre-trained model VGG-16. Thus, the output feature map 7x7 produces 441 anchor boxes. RPN's prediction of the foreground and background feature improves the precision of the anchor boxes drawn for the object detection. RPN is networked to the convolutional layers, which include 3x3 filters, 1 padding, and 512 output channels. Eventually, the classifier and regressor will determine the occurrence of objects.

2.4 Data Augmentation

Image Data Augmentation is a prevalent technique to optimize the results by the elimination of over-fitting. The Keras neural network library consists of the features to accommodate models utilizing data augmentation by enabling the image data generator class. This generator can be linked directly to a training dataset for training the model During the training of the dataset, the image data generator class creates an augmented tensor of bottleneck features. Further, the utilization of transformation characteristics enables the data augmentation parameters to improve accuracy. Figure 2 depicts the horizontal and vertical flipping, height shifting, 90^0 rotations of images parameters used in the proposed model. Data augmentation can also increase prediction accuracy. Faster R-CNN adopts the Bottleneck feature and Data Augmentation to accelerate the execution of the training process to achieve high precision by training a defined set of images.

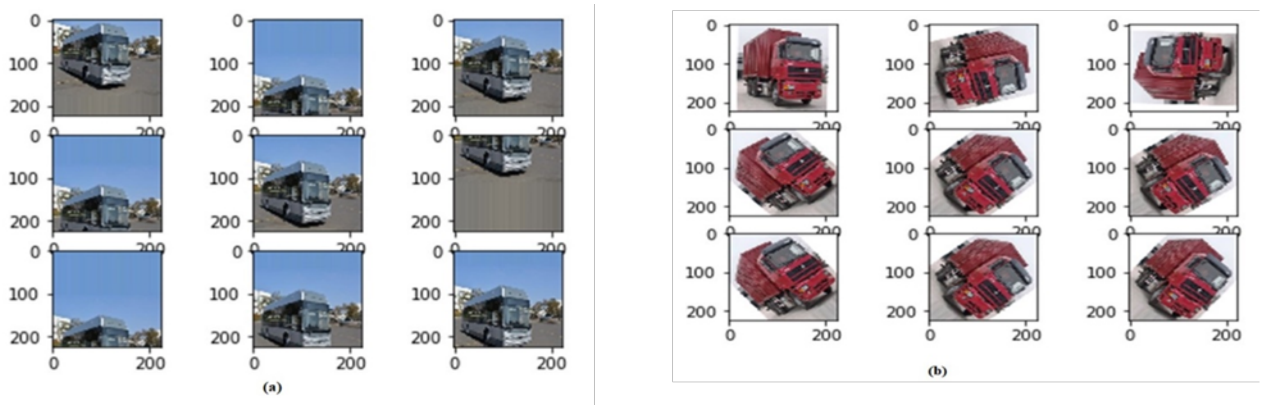


Fig 2. Image Data Augmentation (a) Height-Shifting (b) Rotation

2.5 Non-Maximum Suppression (NMS)

The bounding box regressor predicts a significant number of bounding boxes. Consequently, there is a high possibility of producing several bounding boxes in a single image. Thereby causing inefficiencies during the detection process leading to high computation cost and inaccurate float-prediction. The non-maximum suppression algorithm is applied in this system to enable solid prediction and reduce the unnecessary resource-intensive computational cost during the image-detect transaction of a single bounding box per picture. The improvement and selection of boxes are based on the model's score-weighting system with an overlap_threshold of 0.7. At the final stage of the network, it merges multiple high-overlapping bounding boxes into a single accurate bounding box.

2.6 ROI (Region of Interest) Pooling

After RPN, the obtained proposed regions are considered as substandard structure as it attributes to intensive encoding and decoding different dimensions of CNN feature maps leading to computing cost. Accordingly, ROI pooling is used in this system to enhance the image-processing in this transit stage, the recalibration of the size of individual input feature maps into an identical size is performed. ROI pooling performs partitioning of the input feature map to a finite number "N" of virtually equivalent regions, and then applies max-pooling on each of the targeted regions. The resizing of regions will serve as standardized input for the proximate layer in the framework.

2.7 Soft-max Classifier

The soft-max classifier is the last stage in this model. After post extraction of features for each proposal in RoI pooling, the features are entitled to classification. This study uses deep convolution VGG16 architecture that builds the classifier. The classifier includes the classification layer, the regression layer, and two fully connected (FC) layers. The middle part of the classifier which is the first of Fully-Connected layer transitioned to the soft-max layer to calculate the confidence possibilities of background and vehicle. The next layer, which is the last part of the classifier and the second fully-connected layer involves linear activation functions that return the bounding box. All convolutional layers are aligned in a certain arrangement that has the ReLU layer and the batch normalization layer. Faster R-CNN uses regular L1 loss at the position (x1, y1) of the top-left edge of the box and in the logarithm of the height and width. The loss function is calculated using Equation (1) as follows,

$L_{loc}\left(t^{u}, v\right) = \sum_{i\in\left(x_{1}, y_{1}, w, h\right)} Smooth_{L1}\left(t_{i}^{u} - v_{i}\right)$

where,

$$\mathrm{Smooth}_{L1}(x) = \begin{cases} 0.5x^{2} \text{ if } |x| < 1 \\ |x| - 0.5 \text{ otherwise} \end{cases} \tag{1}$$

The proposed model is guided by the following algorithm:

Step 1: Extraction of data from multiple sources.

Step 2: Data Pre-processing:

- Resizing and Cropping the images by OpenCV method into 600x600 (width and height).
- For each image, annotation file is created.

Step 3: Train the dataset for vehicle detection applying fine-tuned Faster RCNN algorithm

Step 4: Fine-tuned VGG-16 is used as a base network in the Faster-RCNN algorithm.

Step 5: In RPN (Regional Proposal Network), apply speed-up bottleneck and data augmentation features during the training of the dataset. This improves the training speed and reduces the over-fitting of the proposed model.

Step 6: NMS is adopted to check the overlapping of the proposed region. The overlap_threshold=0.7. The final feature map is 7x7x512.

Step 7: The classifier is built at the final stage. The custom dataset is trained for detecting the vehicles.

Step 8: The efficacy of the model is measured and compared with other models.

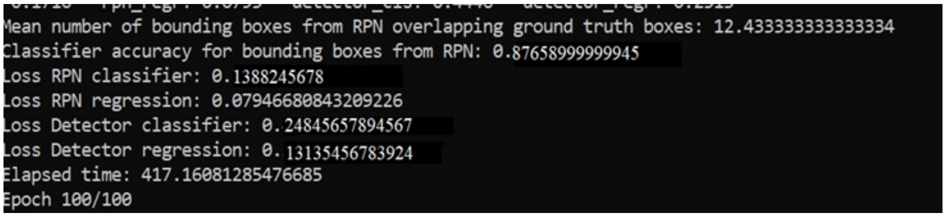
Step 9: Finally, Test the proposed model with real-time static images and visualize them via computer vision.

3 Results and Discussion

The experiment is conducted in Anaconda3 with the underlaying open-source deep-learning framework Keras integrated with Tensor Flow background as the development environment. OpenCV is used to test the Classifier in real-time static images.

3.1 Training a Dataset

In this proposed system, the fine-tuned VGG-16 pre-trained framework is utilized as a base network. To accelerate the training, the weights (model_frcnn.h5py) are downloaded in the pre-trained model. The custom vehicle dataset for Faster R-CNN simulation includes images, annotations, labels, the input file criteria for images are. JPEG, annotations are .xml files and labels are .txt format. This dataset is populated into the model (pre-trained). Initially, the parameters of the base network and RPN are trained. Then, rpn_max_overlap=0.8 and rpn_min_overlap=0.5 are the range of proposals generated by the RPN to differentiate 'positive', 'negative' for each anchor. Bottleneck feature and Image Data Augmentation are the methods used in this system to improve speed and avoid overfitting. The final region output feature map is 7x7x512. Then train the classifier with two fully-connected layers and with the necessary dropout. The final step is to apply a softmax classifier for classification to detect the vehicle and linear activation for regression to produce a refined bounding box along with weights. The loss functions are optimized using the Adam algorithm. The initial learning rate is 0.0001 per batch, with a decline rate of 0.0005. For 100 epochs, the network is trained. Figure 3 depicts the training loss of the finite-capacity dataset.



```
Mean number of bounding boxes from RPN overlapping ground truth boxes: 12.433333333333334
Classifier accuracy for bounding boxes from RPN: 0.87658999999945
Loss RPN classifier: 0.1388245678
Loss RPN regression: 0.07946680843209226
Loss Detector classifier: 0.24845657894567
Loss Detector regression: 0.13135456783924
Elapsed time: 417.16081285476685
Epoch 100/100
```

Fig 3. Training Loss of the Custom Vehicle Dataset

3.2 Evaluation Metrics

This study applies the Intersection over Union (IoU), Average Precision (AP), Sensitivity, and Confusion matrix to assess and weigh the performance of this model in a custom vehicle dataset. The Average Precision score of the proposed model is 0.882. The IoU is used for overlapping an object during training. Therefore, the anchors represented as foreground, which has higher than 0.7 IoU overlap with ground truth boxes and 0.3 IoU overlap with ground truth boxes as background. There are 441 anchors with 256 foreground and background classes in the proposed system. NMS is applied to limit the overlapping of the proposed region. The overall Sensitivity Factor is 0.935, which calculates in terms of Macro and Micro Average Scores as shown in Figure 4(a). The Confusion Matrix uses normalized values for the custom vehicle dataset as shown in Figure 4(b). The results for AP are as follows 0.94, 0.88, 0.81 for Bike, Bus, and Truck. The overall sensitivity factor is 0.935. The classification report includes the sensitivity and specificity factors, precision score, F1 score, and accuracy as described in [Table 1]. The following

Equation (2), (3), (4), (5), and (6) are used to calculate hyper elements of deep neural networks that might affect the performance of the classifications.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall/Sensitivity} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F-Score} = 2 * \frac{P*R}{P+R} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN+TP} \quad (5)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

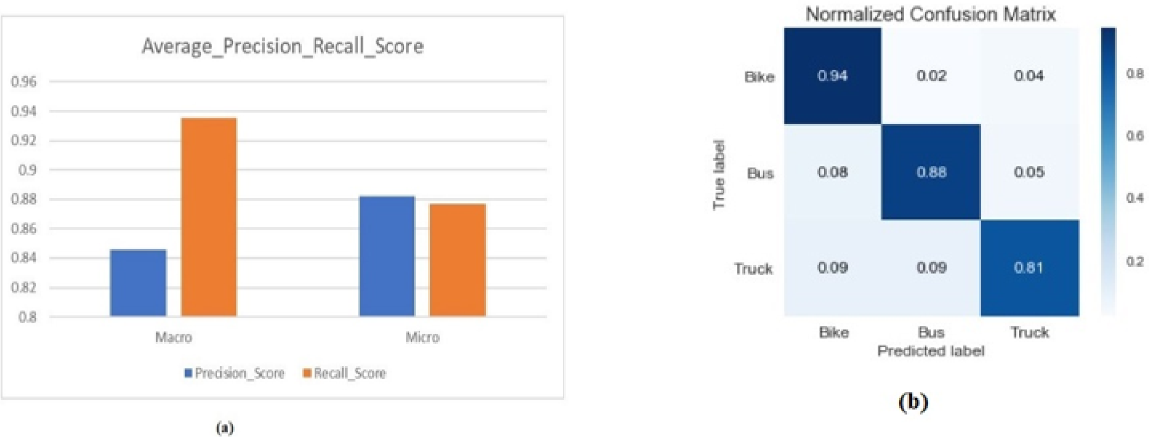


Fig 4. (a) Precision and Recall/Sensitivity based on Macro and Micro Average Score (b) Normalized Confusion Matrix

The performance metrics of the methods such as Faster-RCNN are evaluated based on different aspects such as Precision, Recall, F-score, Sensitivity, Specificity, Accuracy, and. Inference Time is shown in [Table 1].

Table 1. Vehicle Detection Classification Results

Method	Precision	Recall/ Sensitivity	F-Score	Specificity	Accuracy	Inference Time
Proposed Model	**88.6%**	**93.5%**	**0.88**	**0.88**	**87.6%**	**0.42s**

3.3 Testing a model with CIFAR Test Dataset

CIFAR dataset is applied for evaluation of this algorithm. The dataset consists of 600 images per category, including the annotation files. [Table 2] demonstrates the object detection and recognition results of the CIFAR test dataset. The other methods use CNN algorithm for proposal generation and Alexnet [12], Inception v3 [13] for feature extraction. Our method uses a selective search algorithm to obtain the object proposal and VGG16 for feature extraction. [Table 2] points out the average precision of the class of the proposed method is better than others.

Table 2. Comparison ofaverage precision of each class using CIFAR test dataset

Method	Bike	Bus	Truck
Alexnet-CNN [12]	93.2	84.0	95.9
Inceptionv3-CNN [13]	73.2	71.7	65.2
Our Proposed Model	**94.5**	**88.2**	**81.6**

3.4 Comparative Testing of the Proposed model with Existing Methods

This study initially assesses the systematic efficiency of the module and later compares the performance with other recent sophisticated approaches as depicted in [Table 3]. Figure 5 shows the Sensitivity factor/Recall value of the proposed system in comparison with the state of art methods such as (6,8–11).

In ERPN (6), the Deconvolutional FPN is used to enhance the quality of the regional proposals. Further, PSO (Particle Swarm Optimization) - SVM is applied to distinguish the anchors and achieves a low recall value. (8) SVM with HoG is used to categorize the vehicles from distinct viewpoints and achieves 73.3% of recall value. Accurate Vehicle Proposal Network (AVPN) (9) incorporates feature maps for detecting small sized objects and achieves recall values of 74.7% (small) and 77.0% (large). Hyper Regional Proposal Network (HRPN) (10) is utilized to enhance the recall by applying a technique similar to (9). The convolutional regression neural network (11) uses spatial density map in a vehicle over aerial images and achieves a recall rate of 90.5%. Compare to other test models, our proposed method with finite-capacity dataset enabled with speedup bottleneck and Image data Augmentation feature achieves better Sensitivity of 93.5%, F-Score of 0.88, simulation time of 0.42s (adequate iterations in less time -100 epochs) and significant accuracy of 87.6% in detection.

Table 3. Performance Comparison between the Proposed method and other methods

Method	Precision	Sensitivity /Recall	F-Score	Time
ERPN (6)	45.6%	46.4%	0.44	5.8s
SVM_HOG+CNN (8)	77.1%	73.3%	0.78	5.0s
AVPN_basic+Fast-RCNN (9)	88.9%	74.7%	0.82	4.05s
AVPN_large (9)	87.8%	77.0%	0.82	3.65s
H-Fast (10)	86.2%	74.0%	0.80	3.65s
HPRN+Cascade Classifier (10)	89.2%	78.3%	0.83	3.93s
Convolutional Regression Neural Network (11)	89.5%	90.5%	0.88	9.7s
Our Proposed System	**88.6%**	**93.5%**	**0.88**	**0.42s**

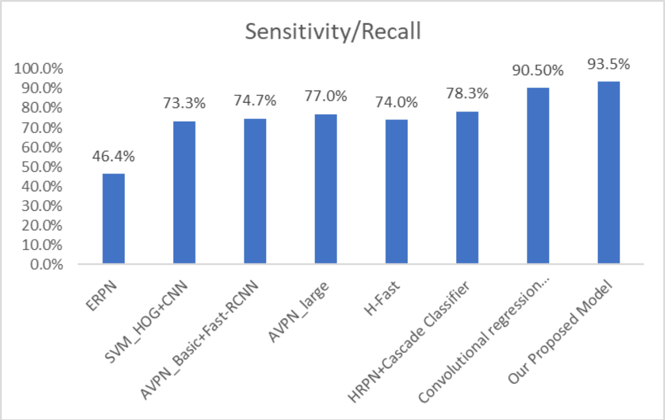


Fig 5. Performance Comparison with different methods in terms of Sensitivity factor

Figure 6 shows the various vehicle detection with real-time images extract from the video using the fine-tuned Faster RCNN features data augmentation method. The real-time static still images are used to test a model. The images are extracted from the real-time video in the chennai zone and other sources include newspaper articles from different parts of the country.

3.5 Limitations

The study observed inefficiencies in the data distribution and usage in the system that causes suboptimal classification accuracy, but it is sufficient to maintain the precision score. Thus, compromising the reliability of the framework. Besides, our study is experimented with adopting a customized dataset with a data capacity of 3200 images. The results may vary in other baselined datasets - PASCAL-VOC, KITTI, and COCO datasets. The limitation in hardware capacity the system achieves low accuracy in one of the aspects. In the future, the accuracy score has scope for optimization using different algorithms and attribute selection.

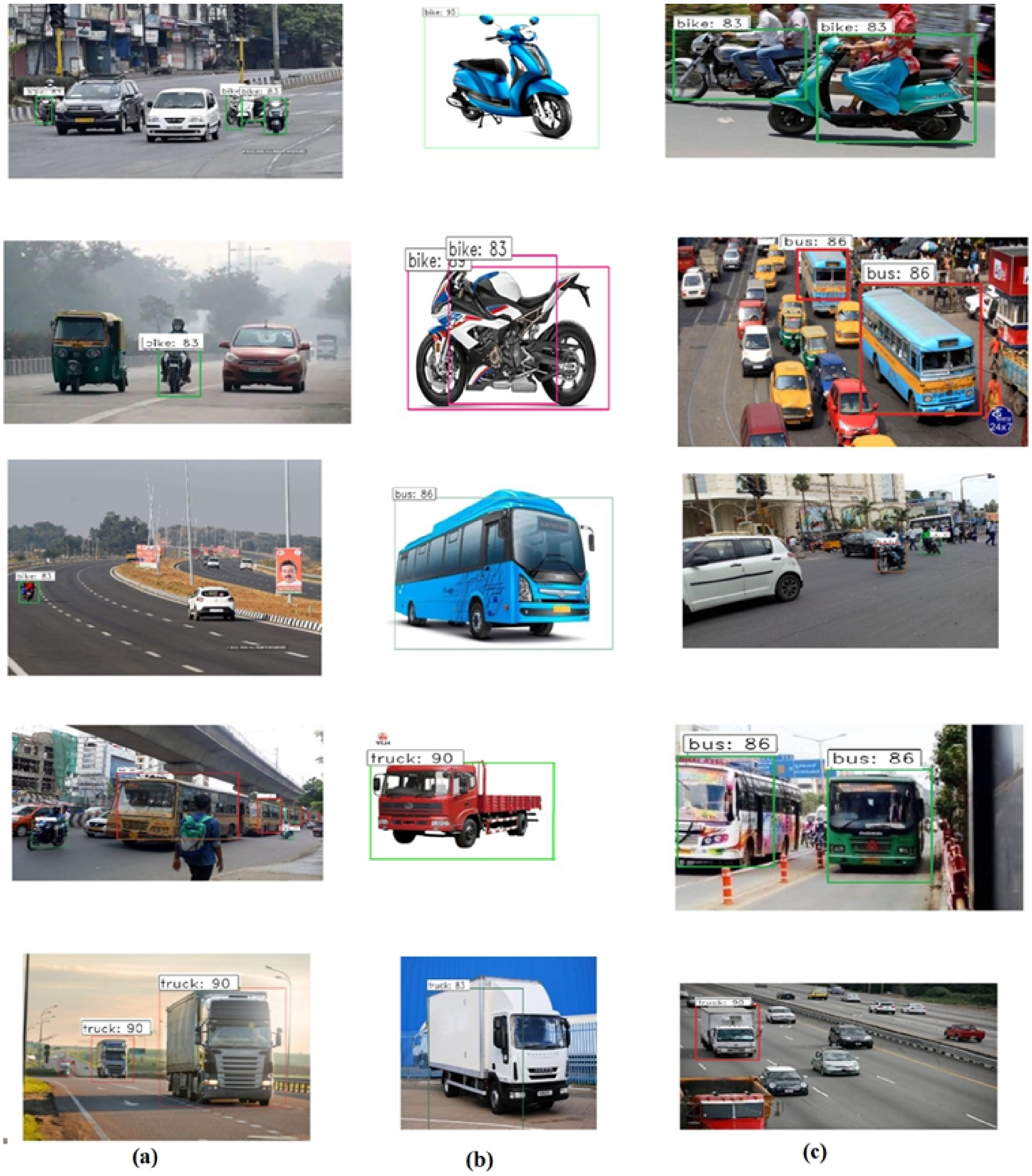


Fig 6. Detection results of the proposed method in various scenarios (a) highways (b) individual vehicle detection (c) crowded places

4 Conclusion and Future work

An optimized Faster R-CNN method for rapid and precision-oriented is implemented for vehicle detection. The efficacy of vehicle detection is accomplished by making the best use of the core components of the framework. The adoption of VGG-16 architecture as a base network of the Faster R-CNN model, and integrating the RPN, besides enabling the Speed Bottleneck features, and Data Augmentation techniques for improvement of velocity, precision, and reduction of over-fitting aimed to fine-tune the inefficiencies to secure optimal performance. The proposed optimized approach exhibits significant results with a Sensitivity factor of 93.5% and detection accuracy of 87.6% with a processing time of 0.42s. The findings corroborate that the proposed approach is efficient. This study shall progress with the latest advanced algorithms for additional improvements in future researchers.

References

1) Nguyen H. Improving Faster R-CNN Framework for Fast Vehicle Detection. *Mathematical Problems in Engineering*. 2019;2019(3):1–11. Available from: https://dx.doi.org/10.1155/2019/3808064. doi:10.1155/2019/3808064.
2) Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Computer Vision and Pattern Recognition*. 2016;p. 1–14. Available from: https://arxiv.org/abs/1506.01497v3.
3) Zhang Y, Wang J, Yang X. Real-time vehicle detection and tracking in video based on faster R-CNN. *Journal of Physics: Conference Series*. 2017;887(887):012068–012068. Available from: https://dx.doi.org/10.1088/1742-6596/887/1/012068. doi:10.1088/1742-6596/887/1/012068.
4) Hsu SC, Huang CL, Chuang CH. Vehicle detection using simplified Fast R-CNN. In: and others, editor. International Workshop on Advanced Image Technology (IWAIT). 2018.
5) Tourani A, Soroori S, Shahbahrami A, Khazaee S, Akoushideh A. A Robust Vehicle Detection Approach based on Faster R-CNN Algorithm. *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*. 2019;p. 119–123.
6) Chen YP, Li Y, Wang G. An Enhanced Region Proposal Network for object detection using deep learning method. *PLOS ONE*. 2018;13(9):e0203897–e0203897. Available from: https://dx.doi.org/10.1371/journal.pone.0203897. doi:10.1371/journal.pone.0203897.
7) Vijayaraghavan V, Laavanya M. Vehicle Classification and Detection using Deep Learning. *International Journal of Engineering and Advanced Technology*. 2019;9(1S5):24–28. doi:10.35940/ijeat. A1006.1291S52019.
8) Fathima KSA, Merriliance K. SVM with Hog Based on Classification Using Vehicle's Different Viewpoints. *International Journal of Engineering Research and Technology*. 2021;14(2):164–173. Available from: http://www.irphouse.com/.
9) Deng Z, Sun H, Zhou S, Zhao J, Zou H. Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2017;10(8):3652–3664. Available from: https://dx.doi.org/10.1109/jstars.2017.2694890. doi:10.1109/jstars.2017.2694890.
10) Tang T, Zhou S, Deng Z, Zou H, Lei L. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. *Sensors*. 2017;17(2):336–336. Available from: https://dx.doi.org/10.3390/s17020336. doi:10.3390/s17020336.
11) Tayara H, Soo KG, Chong KT. Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network. *IEEE Access*. 2018;6:2220–2230. Available from: https://dx.doi.org/10.1109/access.2017.2782260. doi:10.1109/access.2017.2782260.
12) Sharma N, Jain V, Mishra A. An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*. 2018;132:377–384. Available from: https://dx.doi.org/10.1016/j.procs.2018.05.198. doi:10.1016/j.procs.2018.05.198.
13) Hussain M, Jordan J, Bird DR, Faria. A Study on CNN transfer learning for image classification. UK Workshop on Computational Intelligence systems - Nottingham;vol. 840. United Kingdom. Springer. 0202. Available from: https://jordanjamesbird.com/a-study-on-cnn-transfer-learning-for-image-classification/.