Neural Computing and Applications (2023) 35:4755–4774
https://doi.org/10.1007/s00521-022-07940-9

ORIGINAL ARTICLE

EnsembleNet: a hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models

Usha Mittal[1] · Priyanka Chawla[2] · Rajeev Tiwari[3]

Received: 7 October 2021 / Accepted: 11 October 2022 / Published online: 30 October 2022

Abstract

Due to static traffic management regulations on roadways, traffic flow may become congested as it has been growing on roads. Estimating traffic density impacts intelligent transportation systems as it helps build efficient traffic management. Vehicle recognition and counting are two main steps to estimate traffic density. Vehicle identification systems can use motion, handcrafted features, and convolutional neural network (CNN)-based methods. The utilization of deep learning technologies is increasing daily with the popularity of CNN. Different classification and detection models have been developed using transfer learning. In this study, data are collected from several open-source libraries, including MB7500, KITTI, and FLIR. Image annotation has been done to classify vehicles into different categories. Various data augmentation methods are implemented to increase the dataset size and to reduce class imbalance problem. Image quality has been enhanced by performing the sharpening process. Then, a hybrid model of Faster R-CNN and YOLO using majority voting classifier has been trained on processed data. The proposed model's findings have been compared with its base estimators on the collected datasets. The proposed model has demonstrated detection accuracy of up to 98%, whereas YOLO and Faster R-CNN provide 95.8 and 97.5%, respectively. Additionally, compared to YOLO and Faster R-CNN, experimental results show that the proposed model performs better at estimating traffic density. Hence, the proposed approach can effectively enhance road traffic management.

Keywords Detection · Traffic density · Ensemble · SSD · Faster R-CNN · CNN (Convolutional neural network) · Deep learning

1 Introduction

Traffic management is becoming a critical problem for society as vehicle traffic volume rises. Because vehicles have many different characteristics, including corners, colors, edges, shadows, textures, and more, vehicle identification and categorization is a challenging research field. Classification and localization are the two basic procedures that makeup detection [1, 2]. While localization [4] detects the position of vehicles that are available and depicts detected cars by creating a bounding box around them, classification [3] predicts the class of vehicles that are present in the image or video. High-quality cameras and fast hardware are now possible to use in such systems because of technological advancements. Green signals can be optimized at intersections by figuring out the traffic density. It decreases traffic congestion and shortens the time that vehicles must wait in line overall.

Inductive-loop technology has been used to detect vehicles since the early 1960s. The loop detector consists of an electronic detection unit and a wire loop. The electrically conducting, insulated loops are buried in the pavement. Radars are also a standard traffic monitoring

✉ Priyanka Chawla
priyankac@nitw.ac.in

Usha Mittal
usha.20339@lpu.co.in

Rajeev Tiwari
errajeev.tiwari@gmail.com

[1] School of Computer Science & Engineering, Lovely Professional University, Phagwara, India

[2] Department of Computer Science & Engineering, National Institute of Technology Warangal, Telangana, India

[3] School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

system component that uses radio waves to calculate a vehicle's range, direction, or speed. However, various sensing modalities are now accessible for on-road vehicle detection. In recent years, imaging technology has advanced significantly. Cameras are now more affordable, portable, and high-quality than ever. Additionally, computational power has grown significantly.

The three kinds of vision-based vehicle detection algorithms include motion-based, handcrafted feature-based and CNN-based techniques. Techniques based on motion include optical flow, background subtraction, and frame subtraction, and these techniques can only detect and categorize moving vehicles by taking still-camera photos and videos. The Histogram of Oriented Gradients (HOG) [5], SIFT [6], and Harr-like [7] are examples of handcrafted feature-based methods. Low feature representation can be found in these strategies. Although these techniques are effective with modest amounts of data and do not require specialized hardware, a professional must retrieve features in machine learning. The algorithm's effectiveness largely depends on how accurately characteristics are identified and extracted. Since deep learning (DL) models acquire high-level characteristics from training data, they need a significant amount of it. Creating feature extraction algorithms becomes more straightforward as a result. The fact that these algorithms do many matrix multiplication operations, however, means that they heavily rely on powerful computers. These operations can be highly effectively optimized by graphic processing units (GPUs).

In order to get performance that is superior to that offered by a single model, ensemble learning is used to create algorithms. Any computational problem can be solved using ensemble learning, which is the act of merging various models, such as classifiers or experts. The goal is to offer a DL model that combines a variety of learner models into one robust learner model, increasing accuracy. The stacking method is used in this study to design and apply an ensemble based on deep learning models, such as Faster R-CNN and YOLO.

Algorithms designed using ensemble learning consist of multiple models to get better performance than could be provided by the single constituted model. Ensemble learning is the process of combining several models like classifiers or experts to solve any computational problem. The aim is to provide a DL model which consists of many learner models and form a robust learner model, thus, improving the accuracy. An ensemble-based on deep learning models, such as Faster R-CNN and YOLO, is created and implemented utilizing a stacking method. The following are the main contributions of the proposed work:

(a) The dataset of thermal images and visible images are collected from different sources.
(b) Image pre-processing is performed on collected images to improve their quality.
(c) Data augmentation methods are applied to increase the dataset size as well as to reduce class imbalance problem.
(d) Design and implementation of majority voting classifier for vehicle detection based on ensemble of Faster R-CNN and YOLOv5 with better accuracy.
(e) Counting the number of vehicles and calculating the road's traffic density.
(f) Comparison of the proposed model and its underlying estimators on various benchmarking datasets, including the FLIR dataset (Thermal and RGB), MB7500, and KITTI.

The remaining sections of the paper are structured as follows: A comprehensive literature review is provided in Sect. 2. The suggested approach is illustrated in Sect. 3. The results of the experiment are presented in Sect. 4. In Sect. 5, several comparisons and explanations of the findings are described. Finally, the conclusion is given in Sect. 6.

2 Related work

2.1 Literature review on deep learning detectors

Girshick put forth a novel two-stage object detector in 2014 [2] called the Region-based Convolution Neural Network (R-CNN). It had enhanced detection performance and provided 53.7 percent mAP on Pascal VOC2010 [8]. The ImageNet architecture served as the foundation for applying the transfer learning principle. Prompted by the calculation of spatial pyramid matching (SPM) [9], He et al. suggested using SPP [1] net to both accelerate R-CNN and learn more relevant attributes. Fixed length feature vector extraction is done via the spatial pyramid pooling (SPP) layer. Fast R-CNN [10], in which features were retrieved using the RoI pooling layer, was introduced by Girshick et al. to solve the shortcomings of the SPP net. Even though the performance of Fast R- CNN had improved, in these cases, a selective search was still utilized to create proposals [11]. Faster R-CNN [12], which introduced the region proposal network (RPN) and used supervised learning to learn the features, addressed this issue. The region-based fully convolutional network (R-FCN), given by Dai et al. [13], which shares the computational expense in the classification step, addresses the limitations of the Faster R-CNN. Shallow layers of a deep convolutional neural network provide spatially robust features, while deep levels reflect semantically powerful features. This characteristic is utilized by Lin et al. [14], and Feature

Pyramid Network was created whereby semantically and spatially robust features were merged to enable the detection of objects at various scales.

All regions on the image are examined as potential objects by one-stage detection techniques. Redmon et al. proposed YOLO [5] 1real-time one-stage detector was the first of its kind, and object detection was viewed as a regression problem. In order to get around YOLO's restrictions, Liu et al. [16] proposed the single shot detector (SSD). SSD involved using many feature maps to forecast an object, where each feature map was in charge of identifying objects of a specific scale based on their receptive fields. Due to the need to recognize large items, the network got sophisticated. Despite the accuracy parity between SSD and Faster R-CNN, SSD is still preferable for real-time detection due to its quick inference rate. Redmon et al. suggest YOLOv2 [17], an improved version of YOLO in which anchors were produced by k-means clustering. It was more accurate than YOLO since it could operate effectively on many scales and used a batch normalizing layer. By learning the semantic content of a picture, CNNs are incredibly effective at comprehending visual content. Backbone architectures trained on sizable datasets include ResNet [3, ResNeXt [18], Hourglass [19], VGG16 [11, 13], ResNet [3, 13], and ResNet [3]. Table 1 provides details of the various object detection models based on DL.

2.2 Literature review on vehicle detection

In order to work on vehicle recognition and categorization, Zhang et al. [20] utilized a DNN. Finding high-level features from low-level ones was the primary objective. In this study, DNN outperformed a typical neural network for classifying cars with an error rate of 3.34 percent instead of 6.67 percent. The author [21] suggested a better Gaussian mixture model (GMM) with background reduction for vehicle detection. Then, dimensionality was reduced using PCA and LDA, features were extracted using AlexNet and SIFT (scale-invariant feature transform), and classification was completed using SVM (support vector machine). The test results demonstrated that upgraded GMM with AlexNet DNN was more precise at classifying and identifying autos at FC6 and FC7. Author [22] created a deep neural network using the AlexNet DNN for classification and the YOLO approach for detection. Vehicle classification on dark images was done using scene alteration, late fusion, and color transformation methods to improve the performance of DNN.

The author [23] provided a system for identifying a moving car using frame difference. The binary frontal view was created using a symmetrical filter on the car's front image, and the model of the car was identified using a three-layer limited Boltzmann machine. Considering factors like poor lighting and inclement weather, the author [24] developed a vision-based method for identifying cars driving ahead on a roadway. The author of a paper [25, 26] created a technique for tracking solitary objects during a brief timespan using thermal images. Using thermal images taken by unmanned aerial systems, the author [27] also worked on detecting and classifying objects on the sea surface (UAS). It assisted in finding and recovering maritime artifacts. The experimental findings showed 92.5 percent accuracy over a test dataset. The author [28] presented a surveillance system for classifying and recognizing autos during the day and at night. The feature extraction process used several criteria, including textures, entropy, homogeneity, energy, and contrast.

In order to find cars in UAV (unmanned aerial vehicle) photographs, the author [29] employed a catalog-based technique. The author had previously addressed the current method, which was based on screening operations in which asphalted areas were recognized as having the potential to speed up and improve the accuracy of car detection. After recovering the HOG characteristics and finding the actual locations of the automobile using filtering techniques, the 36 directions with the highest similarity value were discovered. The high resolution of the UAV photo allows for several views of a single car. Points from the exact vehicle were pooled to reduce redundancy. SVM was then used to classify the data. The author developed a cloud-based intelligent urban video surveillance system for automatic vehicle detection and tracking [30].

The author created a HOG-based strategy [31] that considered areas with heavy traffic. In order to identify and categorize items, the author [32] employed the hybrid DNN. The non-negative matrix factorization (NMF) was considered for feature extraction and compress data. The author [33] used a background GMM and a shadow removal technique to recognize, track, and divide cars into four groups in the presence of sudden illumination changes and camera shake. For tracking, the Kalman filter was used. The author [34] created a deep Convolutional activation feature (DeCAF) for vehicle recognition and classification. The researcher extracted visual information and compared the accuracy of several methods, including deep CNN and large-scale sparse learning.

Based on the structure of the vehicles, the author [35] proposed a classification scheme. Manual segmentation was done to extract the boundaries, and features were taken out. Fast R-CNN was used by V.V. et al. [36] in their method for detecting cars. This study took images with a camera and balanced the dataset among three classes. A model gave acceptable accuracy assuming noise-free, clear images. The author of the paper [37] used sensor-based data for traffic analysis by placing sensors next to the road. After that, data were chosen from many sources using

Table 1 Major milestones in object detection research based on the deep convolutional neural network since

Researcher	Model	Year	Type	Observations
Krizhevsky [38]	AlexNet	2012	Backbone Architecture	It was a large and complex architecture for computer vision tasks consisting of 650,000 neurons with 60 million parameters. In this, the ReLU activation function was used rather than Tanh, increasing the speed six times with the same accuracy. To avoid over-fitting, the dropout was used
Permanent [39]	OverFeat	2013	One Stage Detector	The original classifier was extended into the detector by considering the last fully connected layer as 1 X 1 convolutional layers to allow arbitrary input. It had shown significant speed strength as compared to two-stage detectors
Simonyan and Zisserman [40]	VGGNet	2014	Backbone Architecture	In this, many 3X3 convolutional layers were used in increasing depth. The features were scaled down by using the max-pooling layer. Finally, the softmax classifier was applied to two fully connected layers consisting of 4096 nodes. The major limitation was its slow training speed and large number of weights
Girshick [2]	R-CNN	2014	Two-Stage Detector	Training and testing time was very high. Hard to get a globally optimal solution
He [1]	SPP-net	2014	Two-Stage Detector	Feature maps were calculated from the whole image, and fixed-length feature vectors were extracted. Detection performance was good, even when objects were at different scales and aspect ratios
Girshick [11]	Fast R-CNN	2015	Two-Stage Detector	Features were extracted using the RoI pooling layer. The optimal solution, high accuracy, and better training and testing speed were significant advantages
Szegedy [39]	GoogleNet	2015	Backbone Architecture	In this, the inception module was used
Ren [14]	Faster R-CNN	2016	Two-Stage Detector	Proposals were generated using Region Proposal Network (RPN). Hard to observe small targets
He [41]	ResNet	2016	Backbone Architecture	It reduced training difficulties. So, it got a more optimal choice
Li [13]	FPN	2016	Backbone Architecture	It was a feature detector that integrated with object detectors
Redmon [15]	YOLO	2016	One Stage Detector	Object detection was considered a regression problem. Difficult to detect small and crowded objects
Newell [42]	Hourglass	2016	Backbone Architecture	It captured both local and global information. It first down-sampled the input image and then up-sampled the feature map
Liu [16]	SSD	2016	One Stage Detector	To avoid negative proposals, hard negative mining was used. Data augmentation also helped in improving detection accuracy. Capable of performing real-time inference
Dai [13]	R-FCN	2016	Two-Stage Detector	Relative position information was provided by a position-sensitive score map of different classes, and features were extracted using ROI pooling
Lin [18]	ResNet	2017	Backbone Architecture	It reduced computation and memory costs. Backbone accuracy is also improved
Huang [43]	DenseNet	2017	Backbone Architecture	Spatially robust features were retained, and the flow of information was improved by mixing the input with the residual output
Chen [44]	DPN	2017	Backbone Architecture	It had the advantages of the ResNet and DenseNet models
Lin [14]	RetinaNet	2017	One Stage Detector	The focal loss was used to subdue the negative samples gradient rather than discarding them. A feature pyramid network was used to detect different size objects
Howard [45]	MobileNet	2017	Backbone Architecture	In this, coordinates were equal to the number of channels of each feature map. Computational cost and number of parameters were reduced significantly. It was specially designed for mobile platforms
Cai [46]	Cascade RCNN	2018	Two-Stage Detector	It worked similar to RefineDet, and proposals were refined in a cascaded manner
Law and Deng [19]	CornerNet	2018	One Stage Detector	Objects were detected as a pair of corners
Duan [47]	internet	2019		In this, bounding boxes were predicted using the CornerNet model, and then center probabilities were predicted to avoid easy negatives
Google Brain Team [48]	EfficientDet	2020	One Stage Detector	In this, ImageNet pre-trained EfficientNet was used as the backbone model. Its computations speed is high than YOLO and AmoeabeaNet

cascade filtering. A CNN-based model was employed to categorize and detect automobiles. This technique offers an accuracy of 98 percent and is more noise-tolerant.

Author [53] introduced a lightweight CNN in which features were optimized, and a joint learning scheme was utilized to classify vehicles based on type. In this study, depth-wise separable convolution was used to minimize the parameters of the network. Softmax loss and contrastive center loss were combined to improve the classification ability of the model. Experimental results were performed on the Car-159 dataset, and the author claimed that the model had less complexity while maintaining its accuracy. Kumar et al. [54] combined the feature values with the bat optimization method to find the optimum feature set. SVM was integrated with local binary patterns to design bounding boxes with confidence scores. Enhanced Convolutional Neural Network (ECNN) was utilized to remove interference area vehicles and moving objects. Experimental results had shown 96.63% accuracy.

Author [55] presented a method to classify small vehicles in the wild by using GANs. Discriminator consisted of two classification modules that could classify whether there was a car, van, or non-vehicle. Furthermore, the author proposed a novel mixed objective function to improve the comprehensive and perceptible information. The proposed model achieved the highest precision of 92.97%. Shvai et al. [56] proposed an ensemble model in which class probabilities obtained from CNN were fused with continuous class probability values obtained from Gradient boosting-based method. The given model was evaluated on a custom real-world dataset and showed an accuracy of 99.03%.

Awang et al. [57] proposed an improved feature extraction approach based on the Sparse-Filtered CNN with Layer-Skipping method (SF-CNNLS). Three channels of SF-CNNLS were used to extract main and unique features. The proposed model was tested on the BIT benchmark and the custom SPINT datasets. The model showed the highest accuracy of 93%. Author [58] merged Simple Online and Real-time Tracking (SORT) method with Faster R-CNN to detect and classify vehicles by type. The proposed model also estimated the speed of vehicles with an accuracy of 78%.

Zhu et al. [59] proposed MME-YOLO, consisting of two sub-models: the improved inference head and the LiDAR image composite model. The first sub-model could identify duplicate visual clues by feature selection blocks and anchor-based or anchor-free ensemble models. In the other sub-model, the actual point data were analyzed deeply and combined with the visual backbone architecture at different levels, which enabled detecting vehicles under unusual lighting conditions. Experimental results indicated that the proposed model achieved accurate and reliable vehicle detection results. Author [60] worked on two public datasets: MIOvision traffic and the BIT vehicle datasets. Initially, images from datasets were pre-processed to improve their quality using adaptive histogram equalization and GMM. After that, Steerable Pyramid Transform and Weber Local Descriptors were implemented for feature extraction from the detected vehicles. At last, extracted feature vectors were passed as input to the proposed ensemble for vehicle classification.

Hu et al. [61] proposed an improved YOLOv4 model to detect vehicles from video streams. This study proposed an algorithm to improve the detection speed, and experimental tests were conducted. Simulation results showed that the proposed model had good accuracy and could be used for safe vehicle driving decision-making. Author [62] introduced a feature fused SSD model and Tracking-guided Detections Optimizing (TDO) method for accurate and fast vehicle detection from videos. In the feature fused SSD, TDL replaced NMS through which inter-frame vehicles were linked by a fast-tracking method. Hence, propagated inferences could compensate for missed detections, and final results confidence scores were optimized. Experimental results on highway datasets had shown the mAP of 8.2% greater as compared to the base estimator.

Jamiya et al. [63] provided a lightweight model, LittleYOLO-SPP, based on a YOLOv3-tiny network. In the proposed model, spatial pyramid pooling layers were introduced, which comprised pooling layers at different scales for feature concatenation to improve the learning abilities of the network. Further, network performance was improved by considering MSE and generalized IoU (GIoU). Experimental results on the PASCAL VOC dataset achieved an mAP of 77.44%, while 52.95% on the MS COCO dataset. Wang et al. [64] presented a model for vehicle detection from the UAV video. Hue – Saturation and Value (HSV) spatial brightness operations were performed on video frames to enhance the model's adaptability under various lighting conditions. After that, vehicle detection was done using the SSD model. The traditional SSD model was optimized by considering focal loss function. Table 2 lists the results of the literature review.

Lighting conditions and video feed quality are the main issues with vehicle detection [63]. The front, back, edges, and corners of a vehicle can be easily identified during the day, but it is pretty challenging to extract features at night. Therefore, in this study, a dataset of thermal images was used to examine the detection potential and was compared to a dataset of visible photos. According to the literature, some automobiles were missed by algorithms, which also incorrectly labeled them. A hybrid model was created to address this problem to recognize cars at various aspect ratios and sizes. The final label for the vehicle was assigned via a majority voting system.

Table 2 Literature Survey on Vehicle Detection and Classification

Year	Authors	Image Type	Approach	Outcome
2002	Robert F. K. Martin et al. [7]	Monocular images	Monocular images were used, and the camera calibration tool was developed	The tool needed only the bare minimum of scene information to detect, track, and classify vehicles
2011	Zezhi Chen et al. [35]	Images from CCTV cameras	Images were taken from CCTV cameras. The classification was done using SVM & Random forest classifier	It achieved the classification accuracy of 96.26% for SVM, which was much better than the random forest algorithm
2011	Y.M. Chan et al. [24]	CCD camera images	In this, the clustering technique was used, and a comparison between the AdaBoost classifier and the proposed system was given	The model was evaluated considering seven videos under poor illumination conditions and got an accuracy of 92.84%
2013	Sebastian Tuermer et al. [31]	Disparity images	In this, for vehicle detection HOG method was used	The proposed model was quicker and provided more accurate vehicle detection
2014	Thomas Moranduzzo et al. [29]	UAV images	A structured based approach and SVM were used	Greater accuracy was required for a more significant number of possible movement directions
2015	Yongbin Gao et al. [23]	Binary images	Frame difference and symmetrical filters were used	Achieved 100% accuracy on their dataset
2017	Zhaojin Zhang et al. [20]	RGB images	The deep NN was used	Compared to a traditional neural network, which has a 6.67 percent error rate, the deep neural network had a 3.34 percent error rate
2016	Ramanpreet Kaur et al. [49]	RGB images	A probability-based neural network was used	The overall existing system was improved [49]
2017	Ajeet Kumar Bhartee et al. [50]	Thermal images	Utilized heat signatures generated by the objects	It was helpful for security and identifying and locating objects in a natural disaster. It aided vision both day and night. [50]
2018	Yunyoung Nam et al. [28]	Both visible light and thermal images	A gaussian mixture model was used	The visible spectrum images had a 92.7% accuracy, while thermal images had a 65.8% accuracy
2019	V. Vijayaraghavan et al. [36]	RGB Images	A fast R-CNN model was used	The model provided 88% of accuracy on custom data. A significant limitation of this model was that it does not provide acceptable results with noisy images
2020	Google brain team [48]	RGB images	It is an updated and improved version of EfficientNet. It uses BiFPN as an optimization technique	It has better accuracy than Mask-R CNN and YOLO v3
2020	M. Rudin et al.[37]	Data collected from sensors	Sensors were used to collect vehicle data from roads then cascade filtering was used to select data in an ordered manner. Vehicles are classified using CNN	The CNN-based model provided a precision of 98%
2020	Sun et al. [53]	Car-159 dataset	A lightweight CNN was utilized to optimize features and a joint learning method to classified vehicles based on type	Model got precision of 85.34%
2020	Kumar et al. [54]	RGB images based custom data collected	Bat optimization method was used for feature optimization. SVM and ECNN were considered for final detection and classification	The proposed model got an accuracy of 96.63%
2020	Wang et al. [55]	KITTI dataset	GANs were utilized	The proposed model achieved a precision of 92.97%
2020	Shvai et al. [56]	Custom data of visible images were collected	In this, ensemble of CNN was proposed	The proposed model achieved an accuracy of 99.03%
2020	Awang et al. [57]	BIT vehicle dataset	In this, sparse-filtered CNN with layer skipping method was utilized	The proposed model provided an accuracy of 93%
2020	Grents et al. [58]	Custom data of visible images were collected	SORT method along with Faster R-CNN was implemented	The proposed model achieved an accuracy of 78%

Table 2 (continued)

Year	Authors	Image Type	Approach	Outcome
2020	Zhu et al. [59]	Custom data of visible images were collected	MME-YOLO model was implemented	The proposed model provided a precision of 91.18%
2021	Jagannathan et al. [60]	MIO-TCD and BIT vehicle dataset	Image processing methods for image enhancement and ensemble based model was used for detection and classification	The proposed model got an accuracy of 99.13%
2021	Hu et al. [61]	Custom dataset was collected	YOLOv4 model was utilized	The model had an improved speed of 16 FPS
2021	Yang et al. [62]	ImageNet VID dataset	Feature fused SSD + TDO scheme was used	The proposed system had a mAP of 83.5%
2021	Jamiya et al. [63]	PASCAL VOC and MS COCO dataset was used	Little YOLO-SPP was implemented	The proposed model achieved accuracy on PASCAL VOC dataset of 77.44% and on MS COCO dataset was 52.95%
2022	Wang et al. [64]	UAV videos dataset was used	HSV and SSD model was utilized	The proposed model got accuracy of 96.49%

3 Proposed methodology

There are numerous deep learning based object detection models are available. But the efficiency of a particular model depends upon its architecture and dataset on which it is trained. In this paper, Faster R-CNN and YOLOv5 models are investigated and an ensemble of these two architectures named as EnsembleNet is implemented for vehicle detection. To train the models, four different datasets are collected. To improve the performance of the model, data pre-processing is also applied. A flow chart of the proposed methodology is given in Fig. 1.

When training a deep learning network, "N" annotated photos $\{x_1, x_2, \ldots, x_N\}$ are provided, and for each image, "x_i," there are M_i objects from C categories:

$$y_i = \left\{ \left(c_1^i, b_1^i\right), \left(c_2^i, b_2^i\right), \ldots, \left(c_{M_i}^i, b_{M_i}^i\right) \right\} \quad (1)$$

where $c_j^i (c_j^i \in C)$ and b_j^i indicate categorical and spatial labels of the j^{th} object in x_i, respectively. For x_i, the prediction y_{pred}^i shares the same format as y_i:

$$y_{pred}^i = \left\{ \left(c_{pred1}^i, b_{pred1}^i\right), \left(c_{pred2}^i, b_{pred2}^i\right), \ldots \right\} \quad (2)$$

Over C + 1 categories, a multi-class classification model is trained, where C refers to actual classes plus one background class.

3.1 Data collection

Four distinct datasets—FLIR thermal, FLIR RGB, MB7500, and KITTI—have been gathered to be used in studies. The FLIR dataset, which includes roughly 14 K thermal, RGB, and 10 K videos, was published in 2018 [39]. Images from the afternoon and the nighttime make up 60 and 40% of the dataset, respectively, during the clear to cloudy months of November to May. The MB7500 dataset includes roughly 7500 images taken in windy conditions with a Phantom 4 drone and a high-definition camera. KITTI datasets has been taken from Karlsruhe's mid-size city, rural areas, and on highways.

3.2 Data pre-processing

In this study, thermal images, as well as visible camera images, are considered. Before feeding images to the model, images are pre-processed to improve the quality. Vehicle detection relies primarily on the edges of the

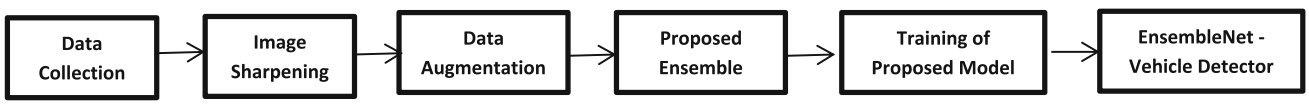


Fig. 1 Flow chart of the Proposed Methodology

Fig. 2 Original image (**A**) and processed image (**B**) from FLIR thermal images Dataset

Fig. 3 Original Image (**A**) and Processed Image (**B**) from FLIR RGB images Dataset

Fig. 4 Original Image (**A**) and Processed Image (**B**) from MB7500 Dataset

object. There are situations when edges in the photos are not visible due to hardware resolution problems and environmental factors. In image processing, sharpening is a technique that enhances the edges of the object. The collected images are therefore subjected to image sharpening as the first pre-processing step. Figures 2, 3, 4 and 5 show the original image and processed (sharpened) image from the different datasets.

Fig. 5 Original Image (**A**) and Processed Image (**B**) from KITTI Dataset

Fig. 6 Original Image

Fig. 7 Horizontal Flipped Image

3.3 Data annotation

The process of classifying and labeling data is known as data annotation. Each image in this work has been labeled using the labelImg tool into six categories: bicycle, two-wheeler, light vehicle, heavy vehicle, bus, or truck. Models are subsequently trained using the annotated datasets.

3.4 Data augmentation

Data augmentation is also used to boost the diversity of the data used to train the models. Three transformations—horizontal flip, rotation, and Gaussian noise—have been used to balance the dataset. An original image of a bus is shown in Fig. 6, whereas reinforced images of the same bus—horizontally flipped, rotated, and noisy—are shown in Figs. 7 through 9.

3.5 Detection of vehicles using the Ensemble method

The Faster R-CNN and YOLOv5 DL models have been examined in this study. The details of both the models are given below:

Faster R-CNN: Faster R-CNN is a widespread two-stage object detection deep learning model. It mainly comprises of two sub-networks. One sub-network, the region proposal network (RPN) is responsible for producing region suggestions. At the same time, another sub-model uses created proposals for object detection. The key benefit of RPN

Fig. 8 Rotated Image

Fig. 9 Noisy Image

networks is cost-effectiveness because selective search takes much time.

YOLO: YOLO stands for You look only once. In 2020, Ultralytics published YOLOv5, their most recent single-stage object detection model. The main components are three, that is.

a. Model Backbone
b. Model Neck
c. Model Head

The model backbone uses cross-stage partial networks (CSP) to find useful features from the input images.

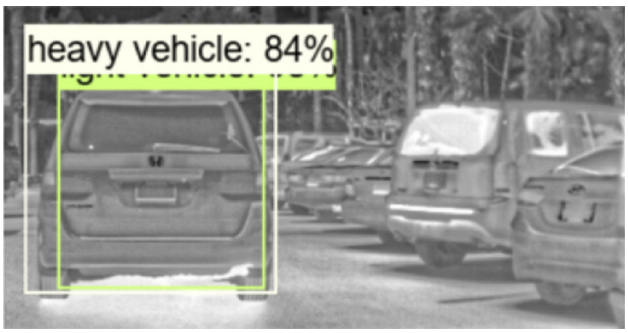


Fig. 10 Prediction with Duplicate Detections

CSPNet's processing speed is rapid. As a result, the feature extraction procedure is quick. The next step is to create feature pyramids using the model neck. A model can recognize related objects of various scales and sizes using feature pyramids. Several feature pyramid approaches exist, including FPN, BiFPN, and PANet. PANet is used in YOLOv5 to obtain feature pyramids. The model head makes the final forecasts. When features are applied using anchor boxes, final output with class probabilities, bounding boxes, and objectness scores are generated. The leaky ReLU and sigmoid activation functions are employed in YOLOv5. Stochastic gradient descent (SGD) is considered an optimization function. Binary cross-entropy with Logits Loss was utilized to calculate the loss.

Considering these two deep learning models (Faster R-CNN and YOLOv5), an ensemble model has been proposed to increase the overall detection accuracy. The Faster R-CNN model exhibits multiple detections in some scenarios, which causes the same object to be identified simultaneously in two categories and produces false results [51]. Figure 10 shows duplicate detections made by Faster R-CNN. Similarly, from experiments, it has been observed that it is hard to detect crowded objects and small objects using YOLO. Hence, to enhance the predictions of base models, and an ensemble using a majority voting approach has been implemented.

Ensemble learning is a process that involves creating many machine learning or deep learning models, then combining their results. It is frequently used to increase prediction performance, function approximation, and classification model accuracy. Predictions made by base estimators Faster R-CNN and YOLOv5 are stored in two variables, P_{faster} and P_{yolo}, respectively. Based on bounding box coordinates, predictions from the two models are contrasted to see whether any predictions are unique or redundant. If the difference between the coordinate values of two predictions is less than or equal to a threshold, then both models have predicted a particular vehicle. The confidence scores of both model predictions are compared to retain the single bounding box.

Suppose Faster R-CNN's confidence score (P_{faster}) is higher than YOLOv5's predicted confidence score (P_{yolo}). In that case, detection produced by Faster R-CNN is considered a final prediction, and YOLOv5's detection is rejected or vice versa. Pseudocode of the proposed EnsembleNet is given in algorithm 1. Figure 11 shows the flow chart of EnsembleNet for vehicle detection.

total of each model's predictions is calculated by multiplying each vehicle by the associated units.

$$density = \sum predicted\_category_{\mathrm{i}} * n\_units \quad (3)$$

where $predicted\_category_i$ denotes the vehicle class detected by the model and n_units defines the units assigned to that class.

Algorithm 1: Proposed EnsembleNet Algorithm

INPUT: an image (Visible / Thermal)

OUTPUT: Bounding boxes for detected classes along with their objectness score that is [$P_{proposed}$: bbox , class, score]

Set up the threshold value. And give input as an image to the trained model.

Begin

Obtain the output from the base estimators. Let P_{faster} and P_{yolo} are the detections made by faster R-CNN and YOLOv5 models, which output coordinates of the bounding boxes, vehicle category, and objectness score.

[P_{faster}: bbox, class, score]
[P_{yolo}: bbox, class, score]
If P_{faster}.bbox – P_{yolo}.bbox <= threshold
  If P_{faster}score > P_{yolo}.score
    $P_{Ensemble}$.bbox = P_{faster}.bbox
    $P_{Ensemble}$.class = P_{faster}.class
    $P_{Ensemble}$.score = P_{faster}score
  Else
    $P_{Ensemble}$.bbox = P_{yolo}.bbox
    $P_{Ensemble}$.class = P_{yolo}.class
    $P_{Ensemble}$.score = P_{yolo}.score
  End
End
End

3.6 Traffic density estimation

Instead of using the number of cars, density is computed using the number of units currently on the road. Each vehicle of a particular brand is unique in size and shape. They take up varying spaces and cross the intersection at various speeds. In this study, vehicles are divided into six classes: bicycles, two-wheelers, light (less than 2 m in height), heavy (more than 2 m in height), buses, and trucks. The number of units allotted to each type of vehicle is given in Table 3. Unit values are considered in such a manner so that from total density values, the number of vehicles in each category can be obtained.

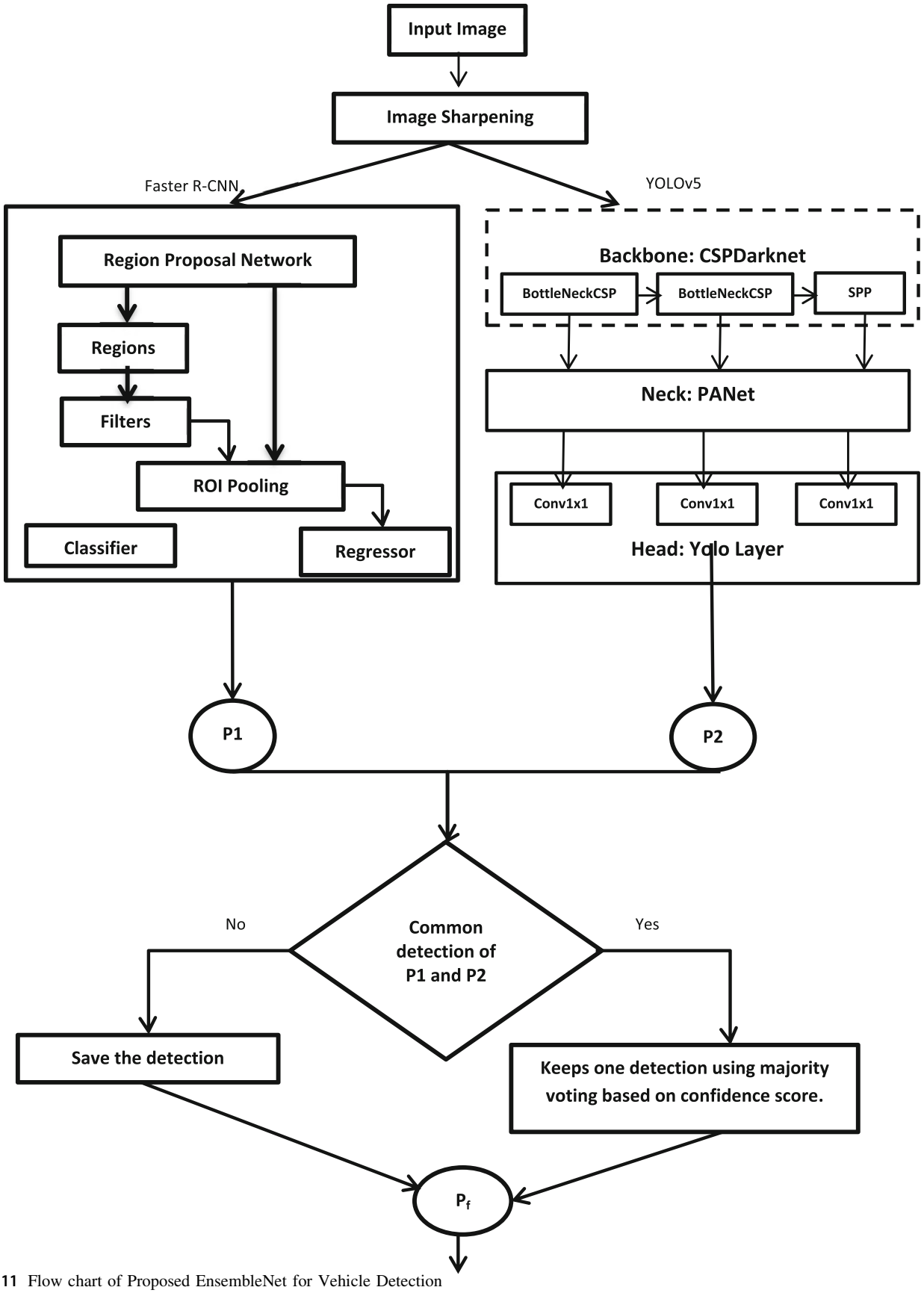
The lane's width is an essential parameter for calculating traffic density. In this study, density is calculated by considering the roads of the National Highway Authority of India (NHAI) India, which is 45 m wide for four lanes. The

4 Results and discussions

This study prefers a camera with better accuracy in low visibility conditions that can produce high-quality images in low-light environments. The camera must have a maximum covered range of 100 m and should be able to operate in the temperature range of -400 to + 550C. The code is implemented on Google Colab and is written in Python using TensorFlow and Keras. Figures 12, 13, 14 and 15 show differences between thermal images and visible images of the same scene taken from the FLIR dataset.

4.1 Results of vehicle detection

Precision vs. recall curves (PR curve) and mean average precision (mAP) of models have been computed to



◀**Fig. 11** Flow chart of Proposed EnsembleNet for Vehicle Detection

Table 3 Units allocated to various vehicle

Vehicle Type	Number of Units
Cycle	1.0
Two-Wheeler	1.3
Light Vehicle	1.7
Heavy Vehicle	2.1
Bus/Truck	2.3

compare the YOLOv5, Faster R-CNN, and proposed models. Figure 16, 17, 18 and Fig. 19 show the precision vs. recall graphs of the precision-recall curve of the FLIR thermal dataset, FLIR RGB, MB7500, and KITTI dataset, respectively. Table 5 compares three models on different datasets using average precision.

The precision vs. recall curve obtained using the FLIR thermal dataset and the proposed methodology is shown in Fig. 16. A better model has higher precision and recall values. It is clear from the figure that the presented

(A) **(B)**

Fig. 12 Daytime image from the visible camera (**A**) and Thermal Camera (**B**) from FLIR Dataset

(A) **(B)**

Fig. 13 Sunlight affected images from the visible camera (**A**) and Thermal Camera (**B**) from FLIR Dataset

(A) **(B)**

Fig. 14 Morning time image from the visible camera (**A**) and Thermal Camera (**B**) from FLIR Dataset

(A) (B)

Fig. 15 Nighttime image from the visible camera (**A**) and Thermal Camera (**B**) from FLIR Data

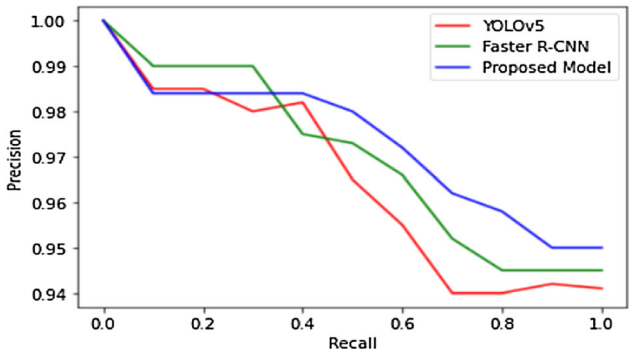


Fig. 16 PR Curve on FLIR Thermal dataset

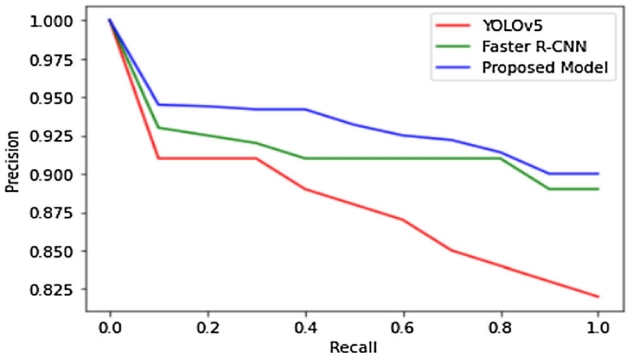


Fig. 17 PR Curve on FLIR RGB dataset

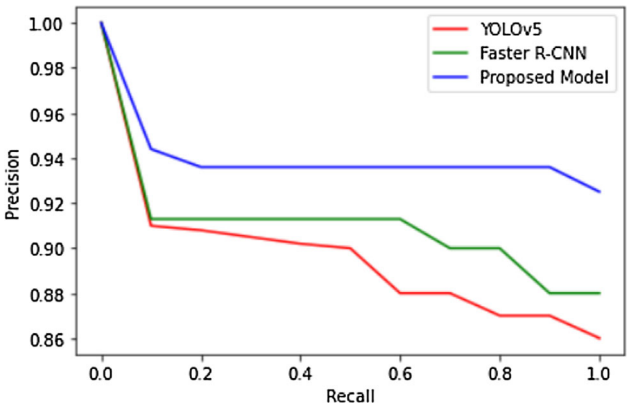


Fig. 18 PR Curve on MB7500 dataset

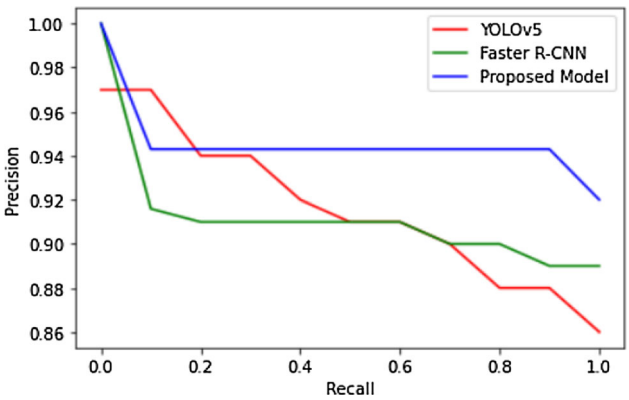


Fig. 19 PR Curve on the KITTI dataset

approach outperforms the Faster R-CNN and YOLOv5 model. Similar precision vs. recall curves for the FLIR RGB dataset, MB7500 dataset, and KITTI dataset are shown in Figs. 17, 18, and 19, respectively. It is apparent from all of the graphs that the proposed technique outperforms base models.

In contrast to YOLOv5 and Faster R-CNN, which offer average precision of 95.8 and 97.5%, respectively, Table 4 demonstrates that the proposed model offers an average precision of up to 98%. Table 5 and Fig. 19 both provide comparisons based on mAP performance. Maximum mAP is achieved by the proposed ensemble on FLIR Thermal dataset which is 94%. Proposed ensemble provided a mAP of 93, 91 and 89% on FLIR RGB, KITTI and MB7500 dataset. Figure 20 demonstrates that the proposed model detects the vehicle more accurately than its base estimators.

4.2 Results of density estimation

Some experimental findings from Fig. 21A through Fig. 26C on various datasets are generated using YOLOv5,

Table 4 Comparative analysis of YOLOv5, Faster R-CNN, and Proposed model on different datasets based on average precision

	Model		
Dataset	YOLOv5	Faster-R CNN	Proposed model
FLIR thermal	95.8%	97.5%	98.%
FLIR RGB	87.6	91.9%	94.%
MB7500	84.5%	91.7%	94.5%
KITTI	86.6%	92.%	95.0%

Faster R-CNN, and the presented model. Each identified vehicle in the image is depicted by a bounding box that includes the vehicle's label and confidence value. Along with the figures, actual and predicted densities are also presented. A suffix in the figure number, such as (A), (B), and (C), denotes the detection performed on the same image using YOLOv5, Faster R-CNN, and the proposed approach.

Figures 21A, B and C consists of one heavy vehicle and three light vehicles. Therefore, the actual vehicle density in the image is 7.4 units. Figure 21A depicts the results of the YOLOv5 model. It predicts more number of vehicles as compared to the present in the image. It detects two heavy vehicles and four light vehicles. Thus, the estimated vehicle density is 13.6. While Figs. 21B and C show the results of Faster R-CNN and the proposed method, respectively, in which two heavy vehicles and two light vehicles are detected. Hence, the computed density in the image is 7.6 units. Proposed models keep the detections of the Faster R-CNN model as the confidence score of detection made by Faster R-NN is higher than provided by YOLOv5. Duplicate detection made by YOLOv5 is discarded based on the threshold value.

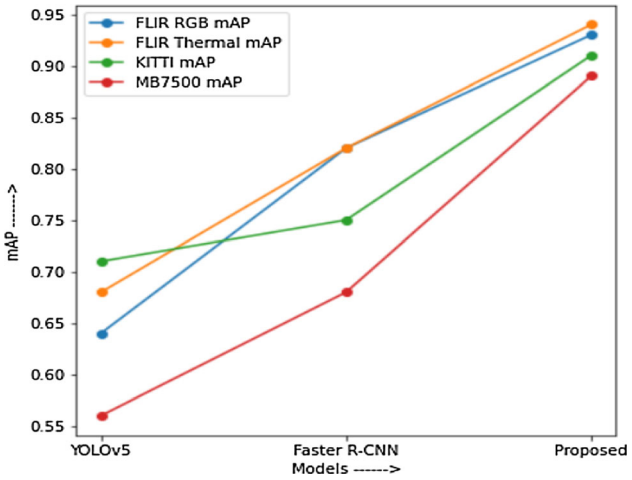


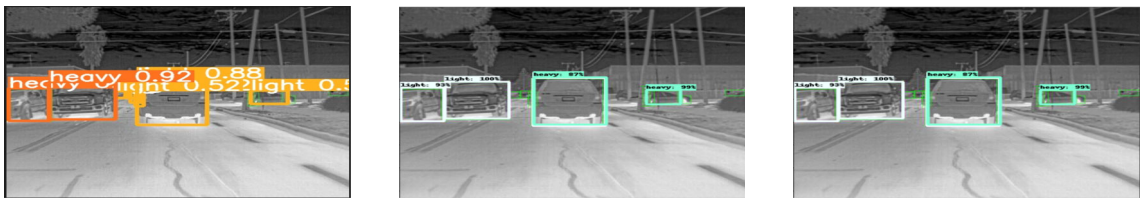
Fig. 20 mAP of YOLOv5, Faster R-CNN, and proposed ensemble on FLIR RGB, Thermal, KITTI and MB7500 dataset

Figure 22A, B and C contains one light vehicle and one heavy vehicle; that is, vehicle density is 3.8 units. Figure 22A depicts the results of the YOLOv5 model in which predictions are correct, and the estimated density is the same as the actual density. Figure 22B shows the results of Faster R-CNN in which one light vehicle is detected correctly, but a heavy vehicle is detected as a light vehicle. Hence, the computed density is 3.4 units. Figure 22C contains the predictions made by proposing method, i.e., one light vehicle and one heavy vehicle, and the computed vehicle density is 3.8 units. Out of Faster R-CNN and YOLOv5 detections, confidence scores provided by YOLOv5 are higher. Thus, the proposed model keeps the detections with a higher confidence score.

Figure 23A, B, and C shows that there are three cycles and one light vehicle present in the image. Thus, the total vehicle density is 4.7 units. Figure 23 (A) contains detections made by the YOLOv5 model, which detects two

Table 5 Comparison of YOLOv5, Faster R-CNN, and Proposed Ensemble on FLIR RGB, Thermal, KITTI and MB7500 dataset based on mAP

Method	Dataset	mAP	CY	TW	LV	HV	TR	BU
YOLOv5	FLIR RGB	0.64	0.68	0.57	0.7	0.75	0.6	0.54
	FLIR Thermal	0.68	0.7	0.68	0.66	0.7	0.85	0.5
	KITTI	0.71	0.6	0.62	0.65	0.68	0.7	1.0
	MB7500	0.56	0.56	0.52	0.64	0.5	0.5	0.62
Faster R-CNN	FLIR RGB	0.82	0.84	0.82	0.8	0.92	0.89	0.67
	FLIR Thermal	0.82	0.86	0.91	0.76	0.87	0.86	0.67
	KITTI	0.75	0.75	0.75	0.84	0.89	0.77	0.5
	MB7500	0.68	0.68	0.6	0.59	0.94	0.6	0.67
Proposed method	FLIR RGB	**0.93**	1.0	0.93	0.91	0.91	1.0	0.83
	FLIR Thermal	**0.94**	0.95	0.91	0.89	0.97	1.0	0.91
	KITTI	**0.91**	0.9	0.94	0.95	0.89	0.9	0.89
	MB7500	**0.89**	0.89	1.0	0.96	0.94	0.78	0.78



(A)
Actual density=7.2
Predicted density=13.6

(B)
Actual density=7.2
Predicted density=7.6

(C)
Actual density=7.2
Predicted density=7.6

Fig. 21 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. FLIR thermal dataset

(A)
Actual density=3.8
Predicted density=3.8

(B)
Actual density=3.8
Predicted density=3.4

(C)
Actual density=3.8
Predicted density=3.8

Fig. 22 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. FLIR thermal dataset

(A)
Actual density=4.7
Predicted density=3.7

(B)
Actual density=4.7
Predicted density=4.7

(C)
Actual density=4.7
Predicted density=4.7

Fig. 23 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. FLIR RGB dataset

(A)
Actual density=6.1
Predicted density=5.5

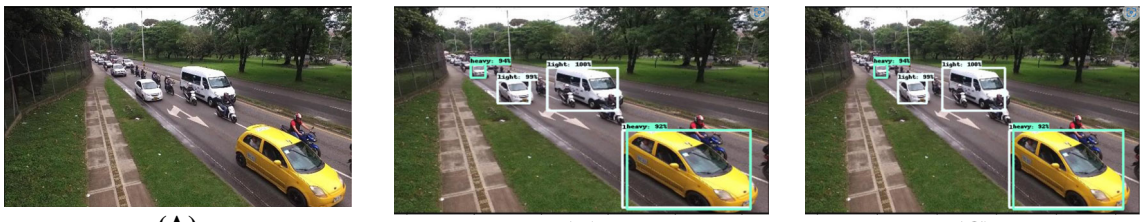
(B)
Actual density=6.1
Predicted density=3.4

(C)
Actual density=6.1
Predicted density=5.5

Fig. 24 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. FLIR RGB dataset

cycles correctly with a confidence score of 47 and 22%, respectively. The third cycle is predicted as the light vehicle with a confidence score of 10%, and the light vehicle is missed. Thus, the computed vehicle density is 3.7 units. At the same time, Faster R-CNN predicts all the cycles and light vehicles with a confidence score of more than 80%. So, the computed density is 4.7 units. Predictions made by the proposed method are correct, and the estimated density is 4.7 units.

Figure 24 A, B, and C consists of one light vehicle, one heavy vehicle, and one truck. Therefore, the actual vehicle density is 6.1 units. Figure 24A depicts the results of the

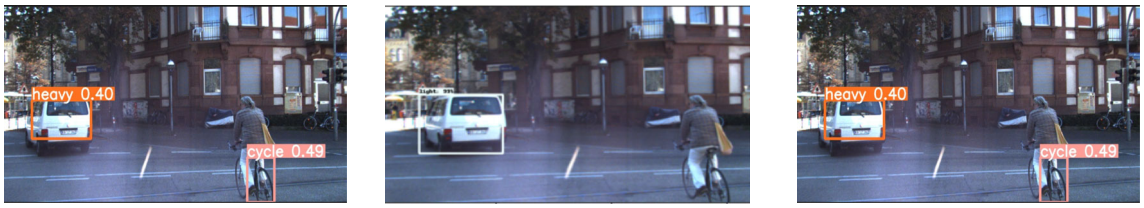


(A)
Actual density=13.7
Predicted density=0

(B)
Actual density=13.7
Predicted density=7.6

(C)
Actual density=13.7
Predicted density=7.6

Fig. 25 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. MB7500 dataset



(A)
Actual density=3.1
Predicted density=3.1

(B)
Actual density=3.1
Predicted density=2.1

(C)
Actual density=3.1
Predicted density=3.1

Fig. 26 Actual and predicted density of (**A**) YOLOv5 model, (**B**) Faster R-CNN model, and (**C**) Proposed ensemble w.r.t. KITTI dataset

YOLOv5 model showing correct detection for light vehicles; one heavy vehicle is predicted as a light vehicle, and one truck is predicted as one heavy vehicle. Hence, the estimated vehicle density is 5.5 units. While Fig. 24B shows the results of Faster R-CNN. Faster R-CNN detects the light vehicle correctly. However, a heavy vehicle is predicted as a light vehicle, and they miss a truck. Thus, the estimated density is 3.4 units only. The proposed method makes the predictions as made by the YOLOv5 model; hence, the computed density by it is close to the actual density, which is 5.5 units.

Figure 25 A, B, and C is a dense image that contains three light vehicles, one heavy vehicle, and five two-wheelers. Thus, the actual density is 13.7 units. Results of YOLOv5 are shown in Fig. 25A. When the image is dense, YOLOv5 cannot make correct predictions. Thus, it is not detecting any vehicle, and the computed density is 0 units. Predictions made by Faster R-CNN are shown in Fig. 25B, which predicts two light vehicles and two heavy vehicles. No two-wheeler is predicted by Faster R-CNN also. Thus, the estimated density is 7.6 units. Predictions made by the proposed method are shown in Fig. 25C, which provides the estimated density of 7.6 units.

Figure 26 A, B, and C consists of one cycle and one heavy vehicle that is 3.1 units of density. YOLOv5 predicts the accurate predictions and returns the density the same as the actual density. While Fig. 26B shows the results of Faster R-CNN in which cycle is missed, and the heavy vehicle is predicted correctly. Thus, the computed density is 2.1 units. Predictions made by the proposed ensemble are shown in Fig. 26C and provide the correct predictions with 3.1 units of density.

In addition to detection accuracy, inference speed can be used to measure the effectiveness of the detection model. The inference speed of the various models, including the YOLOv5, Faster R-CNN, and proposed technique, is shown in Table 6. YOLOv5 takes the least amount of time, whereas the proposed ensemble consumes the most.

Although the proposed model takes a little longer to process an image than other models, when the detection results and inference speed are combined, it is concluded that the proposed model is producing better results in terms of detection and density calculation.

Table 6 Comparison of inference speed of Faster R-CNN, YOLOv5, and proposed method

Model	Inference speed
Faster R-CNN	15.67 Sec
YOLOv5	7.5 Sec
Proposed ensemble	18.5 Sec

5 Conclusion

This research presents and evaluates an ensemble-based deep learning architecture. To detect vehicles, the current approach combines two models, Faster R-CNN and YOLOv5 and proposed a vehicle detector known as EnsembleNet. Overall predictions are improved by the application of the majority voting principle. From the experimental results, it has been found that YOLOv5 provides better results when the traffic density is less but fails to give appropriate results in dense images, while Faster R-CNN returns efficient results in dense images. Hence, by designing a hybrid system, the overall performance of the detection has been improved. Although using two deep learning models makes the model more time-consuming, it also improves overall accuracy. Traffic density estimation is done in addition to the detection and classification tasks and provides the road occupancy rate at a specific period. Our results' comparison to baseline models demonstrates that the proposed model is capable of producing superior outcomes. In order to recognize automobiles and optimize signal controllers depending on density, the proposed model can be employed in intelligent transportation applications. Green signals can be optimized at intersections by figuring out the traffic density. It decreases traffic congestion and shortens the time that vehicles must wait in line overall. The running time complexity and computational complexity optimization are two of the main drawbacks of the proposed study. Work will be done in the future to speed up the computation of the detection model.

Funding No funding support for this work.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflicts of interest We declare no any conflict of interest. This manuscript is submitted only in this journal and is not in parallel submitted or in review at any other venue.

References

1. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV
2. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR
3. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR
4. Mittal U, Srivastava S, Chawla P (2019) Review of different techniques for object detection using deep learning. In: Proceedings of the third international conference on advanced informatics for computing research - ICAICR '19, pp. 1–8, https://doi.org/10.1145/3339311.3339357
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: CVPR
6. Lowe DG (1999) Object recognition from local scale-invariant features. In: ICCV
7. Lienhart R, Maydt J (2002) An extended set of haar-like features for rapid object detection. In: International conference on image processing
8. Fidler S, Mottaghi R, Yuille A, Urtasun R (2013) Bottom-up segmentation for top-down detection. In: CVPR
9. Kleban J, Xie X, Ma W-Y (2008) Spatial pyramid mining for logo detection in natural scenes. In: Multimedia and Expo, 2008 IEEE international conference on
10. Girshick R (2015) Fast r-cnn, In: ICCV
11. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. In: IJCV
12. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS
13. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: NeurIPS
14. Lin T-Y, Dolĺar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: CVPR
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: CVPR
16. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg A C (2016) SSD: Single shot multibox detector, In: ECCV
17. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: CVPR
18. Lin T-Y, Goyal P, Girshick R, He K, Dolĺar P (2017) Focal loss for dense object detection. In: ICCV
19. Law H, Deng J (2018) Cornernet: Detecting objects as paired keypoints. In: ECCV
20. Zhang Z, Xu C, Feng W (2017) Road vehicle detection and classification based on Deep Neural Network. In: Proceedings of the 7th IEEE international conference on software engineering and service science, Aug. 26–28, IEEE Xplore Press: Beijing, China, pp. 675–678. https://doi.org/10.1109/ICSESS.2016.7883158
21. Harsha SS, Anne KR (2016) Gaussian mixture model and deep neural Network based Vehicle Detection and Classification. (IJACSA) Int J Adv Comput Sci Appli, Vol. 7(9): pp. 17–25
22. Zhou Y, Nejati H, Do T T, Cheung NM, Cheah L (2016) Image-based Vehicle Analysis using Deep Neural Network: A Systematic Study. In: IEEE international conference on digital signal processing (DSP). Beijing, China: IEEE
23. Gao Y, Lee HJ (2015) Moving car detection and model recognition based on deep learning. Adv Sci Technol Lett, pp. 57–61
24 Chan YM, Huang SS, Fu LC, Hsiao PY, Lo MF (2012) Vehicle detection and tracking under various lighting. IET Intell Trans Sys 6:1–8
25. Berg A, Ahlberg J, Felsberg M (2015) A thermal object tracking benchmark. In: 12th IEEE international conference on advanced video and signal based surveillance (AVSS) (pp. 1–7). Karlsruhe, Germany: IEEE
26. Mittal U, Srivastava S, Chawla P (2019) Object detection and classification from thermal images using region based convolutional neural network. J Comput Sci 15(7):961–971. https://doi.org/10.3844/jcssp.2019.961.971
27. Rodin CD, Lima LN, Andrade FA, Haddad DB, Johansen TA, Storvold R (2018) Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. Int Joint Conf Neural Netw (IJCNN) 2018:1–8

28 Nam Y, Nam Y-C (2018) Vehicle classification based on images from visible light and thermal cameras. EURASIP J Image Video Process. https://doi.org/10.1186/s13640-018-0245-2

29. Moranduzzo T, Melgani F (2014) Detecting cars in UAV images with a catalog-based approach. IEEE Trans Geosci Remote Sens 52(10):6356–6367

30. Chen Y-L, Chen T-S, Huang T-W, Yin L-C, Wang S-Y, Chiueh T-C (2013) Intelligent Urban video surveillance system for automatic vehicle detection and tracking in clouds. In: IEEE 27th international conference on advanced information networking and applications (AINA) (pp. 814–821). Barcelona, Spain: IEEE

31. Tuermer S, Kurz F, Reinartz P, Stilla U (2013) Airborne vehicle detection in dense Urban areas using HoG features and disparity maps. IEEE J Selected Topics Appl Earth Observ Remote Sens 6(6):2327–2337. https://doi.org/10.1109/JSTARS.2013.2242846

32. Prabha C, Shah I (2016) Study of vehicular traffic using hybrid deep neural network. Int J Innov Res Comput Commun Eng pp. 4334–4338.

33. Chen Z, Ellis T, Velastin SA (2012). Vehicle detection, tracking and classification in Urban Traffic. In: 15th international IEEE conference on intelligent transportation systems (pp. 951–956). Anchorage, Alaska, USA,: IEEE

34. He D, Lang C, Feng S, Du X, Zhang C (2015). Vehicle detection and classification based on convolutional neural network. In: Proceedings of the 7th international conference on internet multimedia computing and service, (pp. 1–5). Zhangjiajie, Hunan, China

35. Chen Z, Ellis T, Velastin SA (2011) Vehicle type categorization: a comparison of classification schemes. In: Proceedings of the 14th international IEEE conference on intelligent transportation systems washington, DC, USA, pp: 74–79. https://doi.org/10.1109/ITSC.2011.6083075

36. Vijayaraghavan V, Laavanya M (2019) Vehicle classification and detection using deep learning. Int J Eng Adv Technol (IJEAT) 9(15):24–28

37. Ma R, Zhang Z, Dong Y, Pan Y (2020) deep learning based vehicle detection and classification methodology using strain sensors under bridge deck. Sensors 20(18):5051. https://doi.org/10.3390/s20185051

38. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NeurIPS

39. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: CVPR

40. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: arXiv preprint arXiv:1409.1556

41. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: ECCV, Springer

42. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: ECCV

43. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR, 2017

44. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks. In: NeurIPS, 2017, pp. 4467–4475

45. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications, in: arXiv preprint arXiv:1704.04861

46. Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In: CVPR

47. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q (2019) Centernet: Keypoint triplets for object detection, in: arXiv preprint arXiv:1904.08189

48. Maithani M (2020). EfficientDet: Guide to state of the art object detection model. Retrieved from https://analyticsindiamag.com/efficientdet/

49. Kaur R, Talwar M, (2016) Automated vehicle detection and classification with probabilistic neural network. IJARIIT

50. Bhartee AK, Srivastava KM, Sharma T (2017) Object Identification using thermal image processing. Int. J. Eng. Sci. Computing

51. Mittal U, Potnuru R, Chawla P (2020) Vehicle detection and classification using improved faster region based convolution neural network. In: 2020 8th International conference on reliability, infocom technologies and optimization (Trends and Future Directions) (ICRITO), 2020, pp. 511–514, https://doi.org/10.1109/ICRITO48877.2020.9197805

52. Oliveira DC, Wehrmeister MA (2018) Using deep learning and low cost RGB and thermal cameras to detect pedestrians in aerial images captured by multirotor UAV. Sensors (Basel) 18(7):2244. https://doi.org/10.3390/s18072244

53. Sun W, Zhang G, Zhang X, Zhang X, Ge N (2020) Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. Multimed Tools Appl 80(20):30803–30816. https://doi.org/10.1007/s11042-020-09171-3

54. Ranjeeth Kumar C, Anuradha R (2020) RETRACTED ARTICLE: Feature selection and classification methods for vehicle tracking and detection. J Ambient Intell Human Comput 12(3):4269–4279. https://doi.org/10.1007/s12652-020-01824-3

55. Wang X, Chen X, Wang Y (2020) Small vehicle classification in the wild using generative adversarial network. Neural Comput Appl 33:5369–5379. https://doi.org/10.1007/s00521-020-05331-6

56. Shvai N, Hasnat A, Meicler A, Nakib A (2020) Accurate classification for automatic vehicle-type recognition based on ensemble classifiers. IEEE Trans Intell Transp Syst 21(3):1288–1297. https://doi.org/10.1109/tits.2019.2906821

57. Awang S, Azmi NMAN, Rahman MdA (2020) Vehicle type classification using an enhanced sparse-filtered convolutional neural network with layer-skipping strategy. IEEE Access 8:14265–14277. https://doi.org/10.1109/access.2019.2963486

58. Grents A, Varkentin V, Goryaev N (2020) Determining vehicle speed based on video using convolutional neural network. Transportation Research Procedia 50:192–200. https://doi.org/10.1016/j.trpro.2020.10.024

59. Zhu J, Li X, Jin P, Xu Q, Sun Z, Song X (2020) MME-YOLO: Multi-sensor multi-level enhanced YOLO for robust vehicle detection in traffic surveillance. Sensors 21(1):27. https://doi.org/10.3390/s21010027

60 Jagannathan P, Rajkumar S, Frnda J, Divakarachari PS (2021) Moving vehicle detection and classification using gaussian mixture model and ensemble deep learning technique. Wireless Commun Mobile Comput 2021:1–15. https://doi.org/10.1155/2021/5590894

61. Hu X, Wei Z, Zhou W, (2021) A video streaming vehicle detection algorithm based on YOLOv4. In: 5th advanced information technology, electronic and automation control conference (IAEAC), pp. 2081–2086, https://doi.org/10.1109/IAEAC50856.2021.9390613.

62. Yang Y et al (2021) A fast and effective video vehicle detection method leveraging feature fusion and proposal temporal link. J Real-Time Image Proc 18(4):1261–1274. https://doi.org/10.1007/s11554-021-01121-y

63 Sri Jamiya S, Esther Rani P (2021) LittleYOLO-SPP: A delicate real-time vehicle detection algorithm. Optik 225:165818. https://doi.org/10.1016/j.ijleo.2020.165818

64. Wang X (2022) Vehicle image detection method using deep learning in UAV video. Comput Intell Neurosci 2022:1–10. https://doi.org/10.1155/2022/8202535