YORES: An Ensemble YOLO and Resnet Network for Vehicle Detection and Classification

Akansha Singh[1], Krishna Kant Singh[2]

[1] *SCSET, Bennett University, Greater Noida, India*
[2] *Delhi Technical Campus, Greater Noida, India*

Abstract

Vehicle identification is a significant process in Intelligent Transportation System (ITS). The growing number of vehicles on road has led to the need of automated methods for traffic monitoring and control. Autonomous vehicles and driver assistance systems require efficient vehicle detection methods. The real time performance of these methods must be high and efficient. The existing methods for vehicle identification have significant drawbacks like complex computations, poor performance and inability to detect vehicles in traffic videos. Thus, in this research, we offer an ensemble strategy for vehicle detection in traffic videos that combines the advantages of YOLO and Resnet. In contrast to Resnet, which is used for fine-grained detection, YOLO is utilized for coarse object detection. The final detection result is generated by averaging the results of the two algorithms. We test our method using a publicly available collection of traffic films and demonstrate that, when used alone, it beats both YOLO and Resnet. A multipart loss function is used by the YOLO network. The ResNet network uses cross entropy loss function. The global ensemble loss function is used that takes weighted average of these two loss function. The multipart loss function is used to combine the classification as well as vehicle localization losses. Thus, the method identifies the vehicle using classification and gives a bounding box using localization. A detailed comparative analysis of the methods is also done, and it is observed that the proposed method is better than other methods.

Keywords

Deep Learning; ResNet; YOLO; Vehicle Detection; Intelligent Transportation System

1. First level sectioning

Increasing number of vehicles and the corresponding increase in traffic on roads has increased the demand of monitoring and controlling of traffic to reduce the number of fatalities. Intelligent transportation systems (ITS) have become an important area of research in the last decade. To introduce such system which can track all the suspicious conditions on the roads and can report the same to reduce the number of accidents and miss-happenings on the roads (Xiao et al., 2020). The most important and the key component of designing an ITS is vehicle detection. Once the vehicle is detected, the information can be precisely used to classify them, analyse the congestion, tracking the vehicles, removing the occlusions, foreign object detection, and detection of suspicious activities and so on (Xiao et al., 2021; Xu et al., 2022).

The study of how to automatically detect and categorise vehicles is a hot topic in the fields of computer vision and machine learning. There are several things – including other cars, buildings, trees, and more – that can obscure a driver's view of their vehicle on the road. Researching how to create algorithms that can accurately recognise and classify partially visible or obstructed cars is difficult. Detecting automobiles in real time is crucial for several uses, including autonomous driving, traffic monitoring, and surveillance. A difficult area of study is the creation of real-time vehicle detection and classification systems. Vehicle identification and categorization algorithms can be hampered by inclement weather. Researching how to make algorithms that can withstand

ProfIT AI 2023: 3rd International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2023), November 20–22, 2023, Waterloo, Canada

akanshasing@gmail.com (A. Singh); krishnaiitr2011@gmail.com (K. K. Singh)
0000-0002-5520-8066 (A. Singh); 0000-0002-6510-6768 (K. K. Singh)

bad weather is difficult. Classifying vehicles is not a simple yes/no question, but rather a multi-class problem. Creating algorithms that can correctly categorise vehicles including cars, trucks, buses, and motorcycles is a difficult area of study. An unbalanced dataset can lower the quality of results obtained by vehicle identification and classification algorithms. It is a difficult research topic to design algorithms that can correct for bias in datasets. The difficulties listed above are only a small sample of the many that have been studied in the field of autonomous vehicle recognition and categorization. By resolving these issues, the performance of these algorithms can be enhanced, and they can be more widely used in practical settings.

The literature review reveals that vehicle detection in traffic videos is difficult because of the dynamic nature of the situations and the large number of possible vehicle types. It can be difficult for vehicle detection algorithms relying on Haar features or HOG descriptors to function in certain settings.

Thus, in this paper YORES an ensemble YOLO Resnet model is proposed. Recent years have seen significant progress in this area thanks to deep learning-based object detection systems like YOLO and Resnet.

The YOLO algorithm is a well-known example of a single-neural-network object detection system. YOLO can detect objects of varying sizes and aspect ratios quickly and precisely. While Resnet is commonly used for image classification and object detection, it is a deep convolutional neural network. Resnet is well-known for its versatility and adaptability, as it can easily handle complicated visual elements and learn from new da-ta.

By combining YOLO and Resnet, we may overcome the shortcomings of both algorithms and improve our ability to detect vehicles. In this research, we propose an ensemble method for vehicle detection in traffic videos by combining the advantages of YOLO and Resnet. The detection of small items or objects with low contrast against their back-ground may be difficult for YOLO. YOLO predicts the bounding box and class of each object using a single grid cell, which may not be precise enough for localizing small objects. Combining YOLO with other object detection model, ResNet, can help alleviate this shortcoming by leveraging the advantages of each.

A further shortcoming of independent object identification models is their potential inability to distinguish between background clutter and actual objects. By combining the benefits of YOLO and REsNet with alternate architectural designs, training data, or input representations, model ensembles can help overcome this restriction. The resulting object detection system may be better equipped to deal with the wide variety of conditions found in the real world.

There are two phases to our ensemble method. In the first phase, coarse object detection is carried out with the help of YOLO. YOLO can swiftly detect the presence of auto-mobiles in an image or video because it has been trained on a vast collection of traffic footage. When cars are identified, YOLO generates bounding boxes that contain those places.

Resnet is utilized for fine-grained detection in the second step. Resnet, which was trained on a more limited set of traffic recordings than YOLO, is able to improve upon the latter's detections by pinpointing the exact position and orientation of the vehicles. Resnet's output is also a set of bounding boxes that are associated with the observed vehicle locations.

In order to arrive at a conclusive detection result, the results from both algorithms are integrated via weighted average. Each algorithm's performance on a validation dataset is used to determine the weights, which can be tweaked to give more weight to speed or accuracy.

2. Proposed Method

The videos are converted to frames for further processing and identification of vehicles. Noise may be present in the frames due to different illumination, weather, and camera calibrations. Filtering techniques are applied during the pre-processing stage and all the frames are converted into a normalized size of 224 × 224 × 3. The details of the complete method are described in the sections below (figure 1).

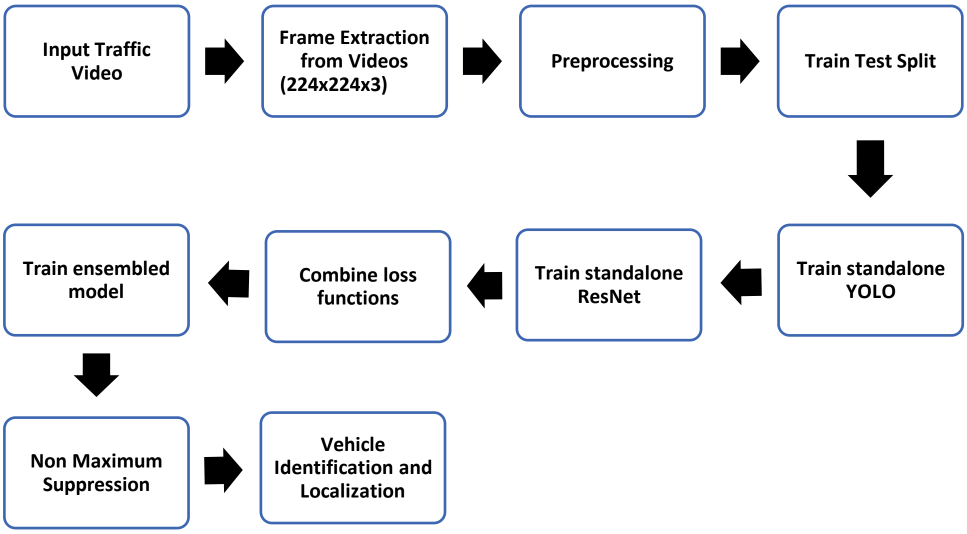


Figure 1: Proposed vehicle detection method

2.1. Conversion of Video Data to frames

The traffic scenes to be processed are generally captured by the CCTV cameras installed on roads. These cameras capture the vehicles as a video. The processing of these videos cannot be done directly. Thus, the conversion of the videos to image frames captured at different time frames is required. A video taken over a time interval T may be represented as shown in equation (1).

$$v_T \epsilon \{f_1, f_2, f_3, \dots \dots \dots \dots f_n\} \tag{1}$$

where v_T = Traffic Video recorded at time interval T .
f_n = image frame.
n = number of frames per second.

2.2. Pre-processing

The pre-processing of the retrieved video frames is important. As these frames suffer from poor quality due to different capturing conditions. They may also have noise due to the problems in the image sensors. All these will lead to poor results and therefore some pre-processing is required. After pre-processing the data will be ready for input to the model. The main issue is presence of noise. Thus, the input frames are filtered using Butterworth low pass noise removal filter (Basu, 2002) for removing the noise and smoothening the images. The mathematical equation for the same is given in eq. (2).

$$B(x, y) = \frac{1}{1+\left[\frac{D(x,y)}{D_0}\right]^{2m}} \tag{2}$$

where D_0 is the cut $-$ off frequency and $D(x, y) = \sqrt{x^2 + y^2}$
where $x \; and \; y$ are individual pixels of HSI layers obtained in previous step.

2.3. Proposed Network Architecture

In recent years deep learning has shown very good results for object detection and classifications in image/videos. In this paper, we have used a Resnet-50 network for detecting various vehicles on the road. The network comprises of the convolution layer network which extracts various important features from the image applying convolutions. The second part of the network is feature localization network which comprises of region proposal networks and pooling combined with non-maximum suppressions to detect the bounding boxes around the vehicles. The backbone network used in the proposed work for initial feature extractions is ZF network (Zeiler & Fergus, 2014). The network has very fast training and testing speed and is very useful in designing real time object detections. The network uses small size kernels which maintain even lower-level details in the frames with max pooling. This reduces the time and complexity in network processing.

The second network used is YOLO which is efficient and fast object detection network (Diwan et al., 2023). The network architecture for YOLO is shown below:

1. Input Layer: This layer is responsible for receiving the input video frames (RGB) from the traffic videos.
2. Backbone Network: The EfficientNet design serves as the foundation for the backbone network, which is made up of numerous convolutional layers and includes the following:
    a. Convolutional layers: The backbone network contains a total of 9 convolutional layers, each with a different number of filters and kernel sizes.
    b. Bottleneck layers: The backbone network is comprised of 2 bottleneck layers, each of which utilizes a combination of 1x1 and 3x3 convolutional layers to minimize the total number of input channels.
    c. Depthwise separable convolutions: The backbone network also includes two depthwise separable convolutional layers. These layers make use of a combination of depthwise and pointwise convolutions in order to reduce the number of computations that are necessary for feature extraction.
3. The Neck Network: The neck network is what connects the head network to the backbone network. It is made up of a few convolutional layers and includes the following components:
    a. SPP layer: The neck network incorporates a spatial pyramid pooling (SPP) layer, which implements max pooling at many scales to capture features at various granularities of detail.
    b. Convolutional layers: The neck network also incorporates a number of convolutional layers, which further refine the features that were extracted by the backbone network.
4. Head Network: The head network is the part of the system that is in charge of producing bounding boxes and detecting objects. The head network is made up of a number of convolutional layers, including the following:
    a. Levels of prediction that are based on anchors: The head network has three levels of prediction that are based on anchors. Each of these layers predicts the class and location of objects by making use of anchor boxes that range in scale.
    b. Convolutional layers: The head network also includes a number of convolutional layers, which further refine the predictions that were provided by the anchor-based prediction layers.
5. Output Layer: The output layer is responsible for generating the final detection results, which include the category and position of each object that was found.

2.4. Ensembling Technique

Let I be an input video frame and let $YOLO(I)$ be the output of $YOLO$ on I, which consists of a set of bounding boxes $B = \{b_1, b_2, b_3, \ldots b_n\}$, where each $b_i = (x_i, y_i, w_i, h_i)$ represents the location and size of a detected vehicle.

Let $ResNet(I)$ be the output of $ResNet$ on I, which also consists of a set of bounding boxes $B' = \{b_1', b_2', b_3', \dots b_n'\}$, where each $b_i' = (x_i', y_i', w_i', h_i')$ represents the location and size of a detected vehicle.

We can combine the outputs of YOLO and Resnet using a weighted average:

$$B_{final} = w_1 B + w_2 B' \tag{3}$$

where $B_{final} = \left\{b_1{}^{final}, b_2{}^{final}, b_3{}^{final}, \dots b_n{}^{final}\right\}$ is the final set of bounding boxes, and w_1 and w_2 are the weights assigned to YOLO and Resnet, respectively. We can choose these weights based on the performance of each algorithm on a validation dataset, and we can adjust them to prioritize speed or accuracy depending on the application.

Both the YOLO and ResNet models produce a significant number of ideas for each vehicle. The abundance of proposals poses a challenge in the process of filtering and identifying a singular bounding box for each vehicle. Therefore, the application of the non-maximum suppression technique is employed to filter the bounding boxes and reduce them to a single box per vehicle. The NMS algorithm takes as input a set of proposal boxes B, their corresponding confidence scores (S), and a user-selected threshold value (N). The filtered proposals (D) are acquired as the resulting output of the method.

2.5. YOLO LOSS Function

The employed loss function in this study is a multifaceted loss function. The losses employed in this context are mean squared error losses, which incorporate the IoU score to quantify the discrepancy between expected and actual values. The loss function comprises three components, namely coordinate loss, confidence loss, and classification loss.

$$L_{YOLO} = f_{coordinate-loss} + f_{confidence-loss} + f_{classification-loss} \tag{4}$$

where

$$f_{coordinate-loss} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \tag{5}$$

$$f_{confidence-loss} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(\sqrt{w_i} - \sqrt{\widehat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h}_i}\right)^2 \tag{6}$$

$$f_{classification-loss} = \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \tag{7}$$

where

λ_{coord} : weight of coordinate loss
x_i, y_i : centre coordinates
w_i: width of the bounding box
h_i: height of the bounding box
C_i: Confidence Score
$P_i(c)$: i_{th} grid cell class probability

2.6. ResNet LOSS Function

The cross-entropy loss function is defined as the difference between the true probability distribution y and the anticipated probability distribution.

$$L_{ResNet} = \sum y_i \log(\widehat{y_i}) \tag{8}$$

Where $\boldsymbol{y_i}$ is the ith element of the true probability distribution y and $\boldsymbol{\widehat{y_\iota}}$ is the corresponding element of the predicted probability distribution ŷ. The summation is taken over all elements i of the distributions.

2.7. Global Ensemble LOSS Function

When combining YOLO with ResNet, we can use a loss function that is a weighted sum of the losses from both models. For instance, we can compute the loss as a weighted sum of the YOLO and ResNet loss functions by giving each loss term in the YOLO loss function a certain value. Training model parameters can also be updated using a weighted combination of individual model's optimization techniques. The relative success of each model on the validation data will inform the decision of how much weight to give each feature.

YORES ensembles YOLO and ResNet – and that their loss functions, L(YOLO) and L(ResNet), have been assigned weights of alpha and beta, respectively. The ensemble loss function is then calculated as:

$$L_{ensemble} = \alpha L_{YOLO} + \beta L_{ResNet} \tag{9}$$

Here, alpha and beta are scalar weights that specify how much emphasis should be placed on either of the two loss functions. These weights can be determined by looking at how well each model does on validation data and giving more weight to the model that does better.

Minimizing the global loss function $L_{ensemble}$ as a function of the model parameters is the target of the optimization. Backpropagation is used to compute the gradients of the global loss function with respect to the model parameters during training, and an optimization technique like stochastic gradient descent (SGD), Adam, or RMSProp is used to update the model parameters.

The loss functions of YOLO and ResNet are combined in the ensembled model by weighting the individual loss functions and then summing them to get the final loss function.

3. Experiments and Results

In this section the experiments are discussed. The proposed model is implemented using Python programming language. The Python modules used include keras and tensorflow. Other supporting modules are also used. The model training is done using GPU support as the dataset is very large and training will ot be possible with simple CPU. The dataset contains two subsets localization and classification. Using these datasets an annotated csv file is created. The csv file comprises the bounding box position of each object. This is used as the ground truth for training. The model is trained with 10000 iterations on the selected dataset. After the trained network is fully trained it can identify the vehicles. The non max suppression threshold value is selected as 0.45. The model is then applied to test videos and images. Each detected object shows vehicle bounded by a box. The name of the vehicle also appears on the box. The results of various steps of the proposed method are shown below. Classification results for all categories of vehicles are shown in figure 3.

3.1. Data Set Used

The experiments are conducted using the publicly available datasets. Numerous publicly available datasets are available for vehicle classes. But in this work one of largest vehicle dataset MIO-TCD (Luo et al., 2018) is used. This dataset is divided into two parts the classification and localization dataset. The localization is used for the object position and classification for vehicle class. The distribution of the MIO-TCD dataset is shown in table 1. The dataset contains vehicles from different field of views, illumination condition and weather. Some of the sample images from the dataset are shown in figure 2.

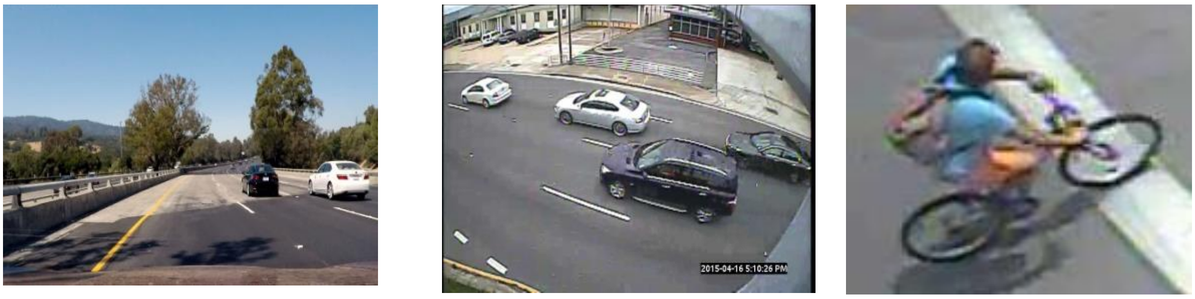


Figure 2: Example frames from video dataset.

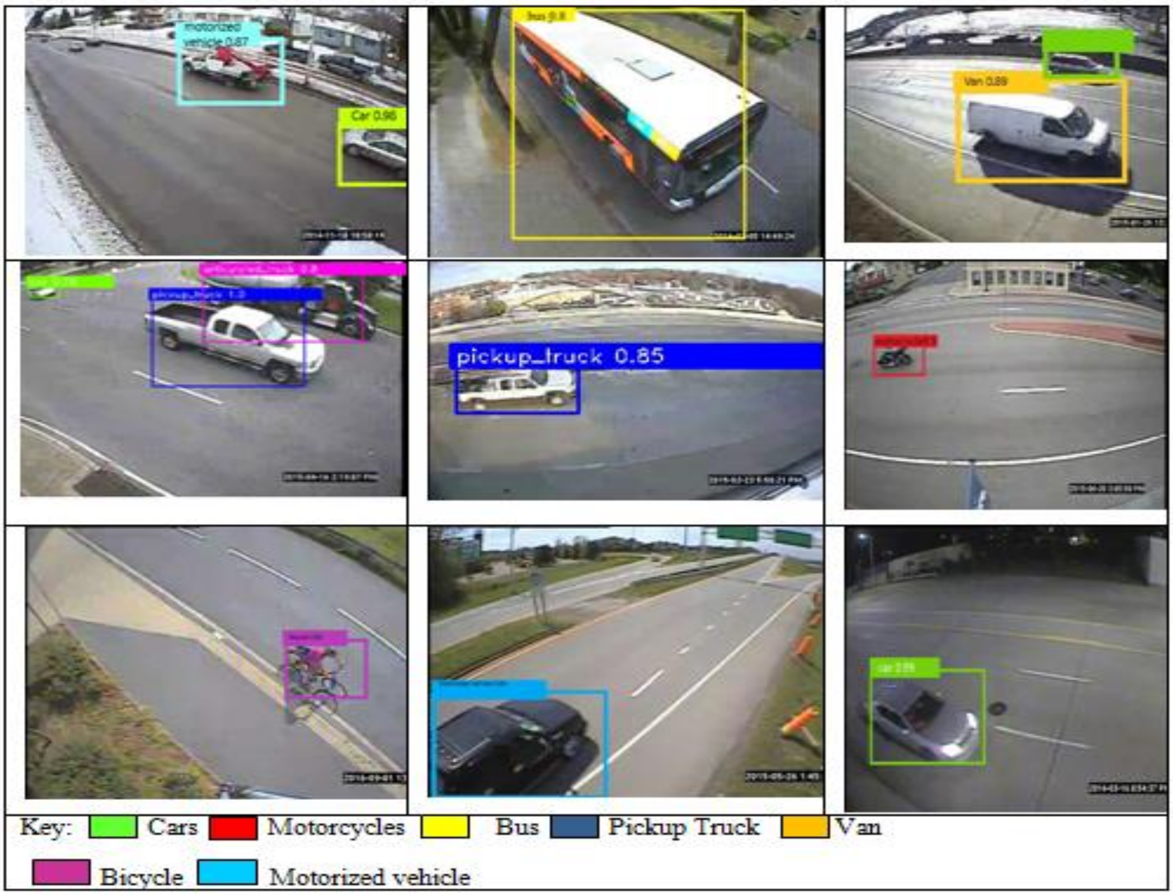


Figure 3: Detection results for different vehicle categories

Table 1.
Distribution of dataset

Category	Training	Testing
Articulated Truck	10346	2587
Bicycle	2284	571
Bus	10316	2579
Car	260518	65131
Motorcycle	1982	495
Non-Motorized Vehicle	1751	438
Pick up Truck	50906	12727
Single Unit Truck	5120	1280
Work Van	9679	2422
Background	160000	40000

3.2. Evaluation Metric

To quantitatively analyse the performance of the classification and detection method following metrics are being used.

Total Accuracy: The total accuracy demonstrates the percentage of the total number of vehicles currently identified as vehicles.

$$AC = \frac{TCI}{tota\ lnumber\ of\ vehicles} \tag{10}$$

where TCI = total number of correctly identified vehicles

Mean Recall and Mean precision

The dataset which we are using is having different number of images or frames for different datasets we have used another two metrics for rectifying this imbalance namely mean recall and mean precision. These are obtained by taking the average of precision and recall overall category of the vehicles.

$$MRE = \frac{\sum_{i=1}^{11} RE_i}{11} \tag{11}$$

$$MPR = \frac{\sum_{i=1}^{\mathbf{11}} \boldsymbol{PR_i}}{\mathbf{11}} \tag{12}$$

where $RE_i = \frac{TP_i}{TP_i + FN_i}$ and $PR_i = \frac{TP_i}{TP_i + FP_i}$

TP_i, FN_i and FP_i are true positives, false negatives and false positives for each category.

The overall results for all category of vehicle are shown in Table 2.

Table 2
Accuracy of method for all classes of vehicles and background

Category	Accuracy (%)	MRE	MPR
Articulated Truck	98.7	0.74	0.78
Bicycle	85.2	0.78	0.83
Bus	98.2	0.96	0.79
Car	99.8	0.82	0.81
Motorcycle	100	0.89	0.92
Motorized Vehicle	67.8	0.63	0.55
Non-Motorized Vehicle	71.2	0.71	0.67
Pick-up Truck	98.2	0.92	0.91
Single Unit Truck	78.3	0.71	0.75
Work Van	97.8	0.87	0.91
Background	98	0.87	0.92

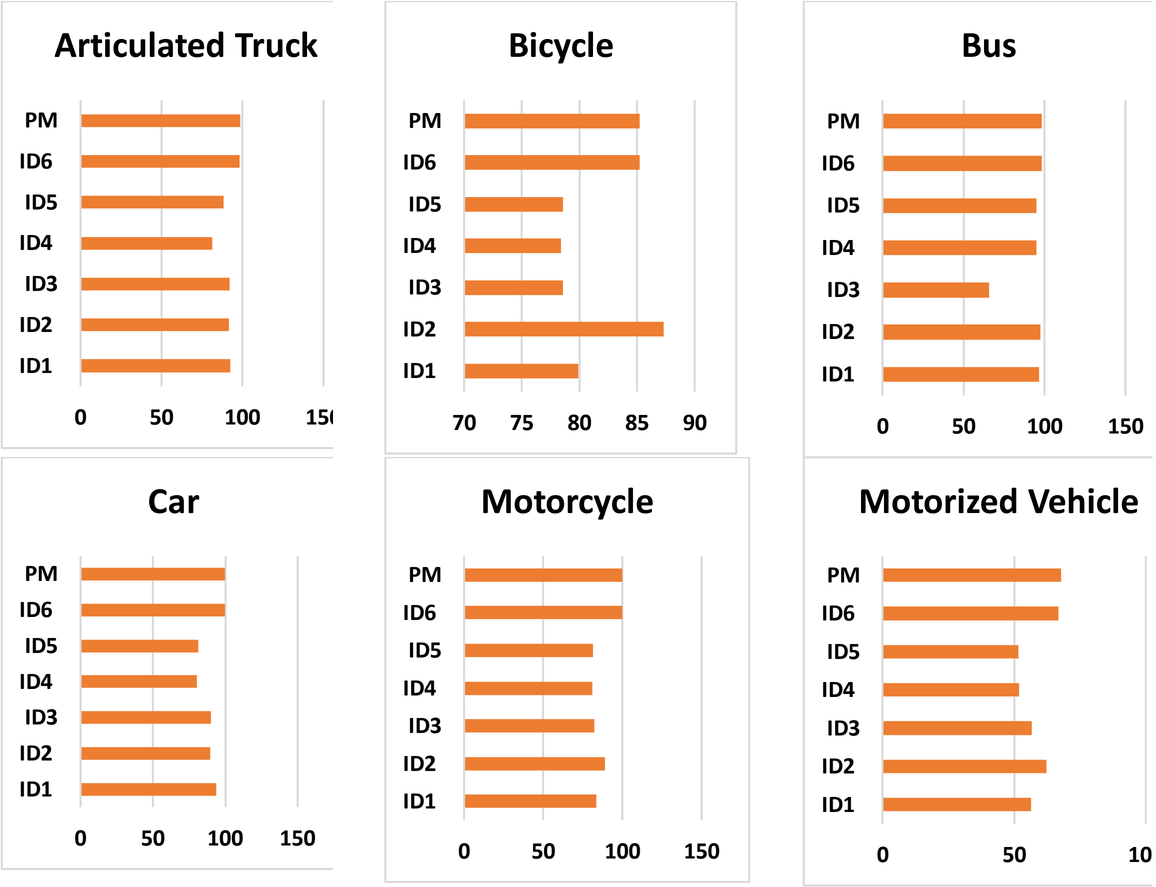
The method was also compared with other state of the art classification methods. All these methods have used feature extraction followed by a classifier network. The comparative results for all the methods have been shown in Table 3.

Table 3
Comparative analysis of accuracy

Method	Articulated Truck	Bicycle	Bus	Car	Motorcycle	Motori-zed Vehicle	Pick up Truck	Non-Motorized Vehicle	Single unit truck	Van	Average Accuracy

ID1 (Jung et al., 2017)	92.5	79.9	96.8	93.8	83.6	56.4	92.8	58.2	73.8	79.6	80.7
ID2 (Theagarajan et al., 2017)	91.6	87.3	97.5	89.7	88.8	62.3	92.3	59.1	74.4	79.9	82.2
ID3 (Wang et al., 2019)	92.1	78.6	66	90	82.3	56.8	90	58.8	74	76	81
ID4 (YOLO v2(P)) (Luo et al., 2018)	81.3	78.4	95.2	80.5	80.9	52	84.6	56.5	70	70	75
ID5 (YOLO v2(M)) ((Luo et al., 2018)	88.3	78.6	95.1	81.4	81.4	51.7	86.5	56.6	69.2	69.2	76
ID6 (Sharmate al., 2021)	98.4	85.2	98.2	99.8	99.8	66.8	71.2	98.2	79.3	97.8	90
Proposed Method (PM)	98.7	85.2	98.4	99.8	100	67.8	71.4	98.4	80.2	97.8	90.5

The graphical representations of the same are shown in figure 4.



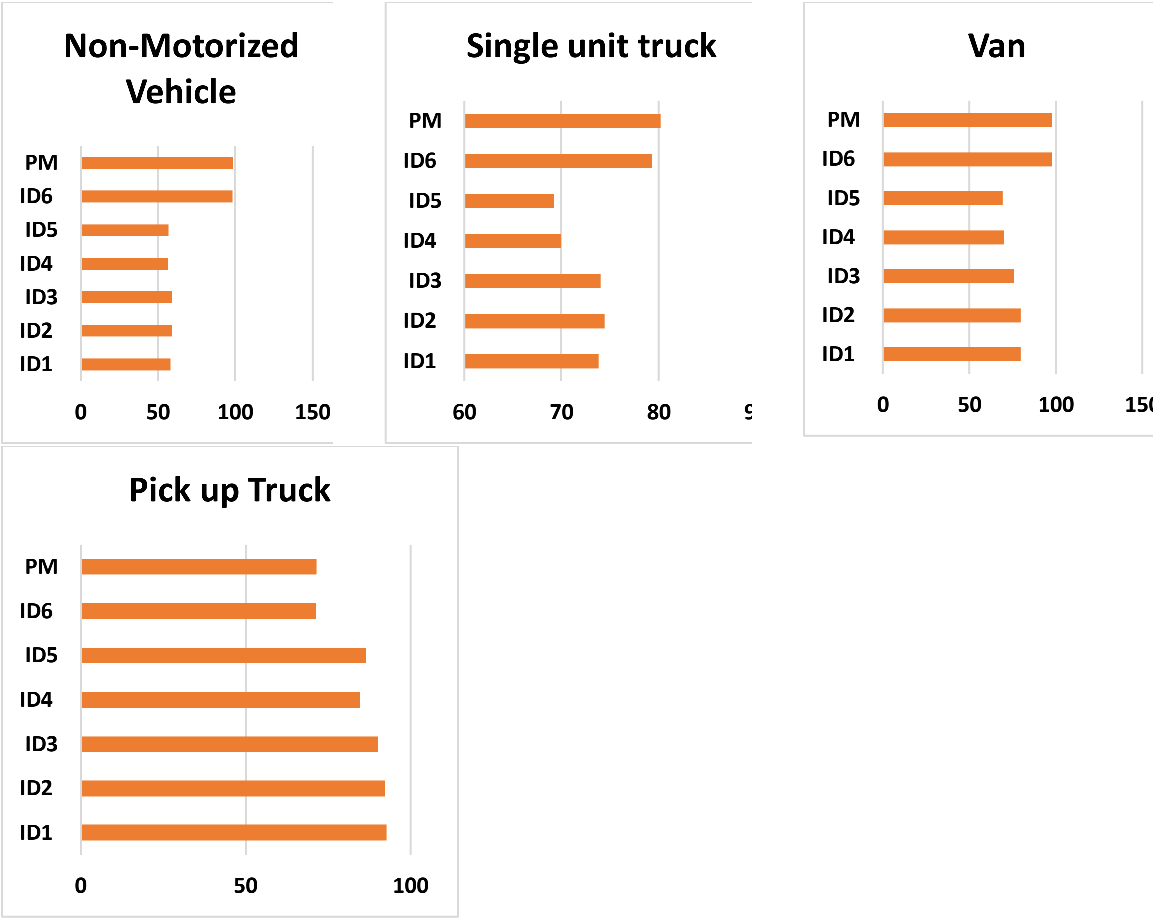


Figure 4: Comparative Results

The average accuracy comparison analysis with different methods is shown in figure 9.

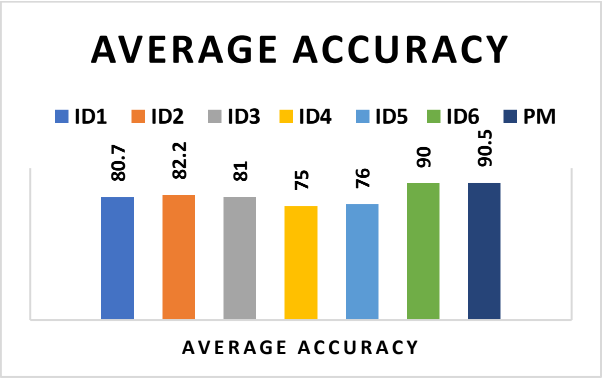


Figure 5: Comparative analysis with other methods

4. Conclusions and Future Work

In this research, we offer an ensemble method to the problem of vehicle detection in traffic videos by combining the advantages of the YOLO and Resnet algorithms. YOLO is used for coarse object detection, and Resnet is used for fine-grained detection in our approach. YOLO is used to identify large groups of objects. A weighted average assembly is used to aggregate the results of both of these processes. The YOLO and ResNet loss functions are combined in the ensembled model by first assigning weights to each loss function and then computing the weighted sum of these individual loss functions as the overall loss function. We can effectively combine the strengths of YOLO and ResNet and increase the performance of car detection from traffic videos by improving the ensembled model using the overall loss function. This allows us to effectively mix YOLO and ResNet.

Some of the limitations of standalone object detection models can be circumvented with the help of ensembling and YOLO. The proposed approach accurately identifies eleven classes of vehicles, achieving state-of-the-art results. The comparison of the proposed approach with six other methods demonstrates its superiority in terms of accuracy, speed, and robustness. Therefore, the proposed approach has significant potential for practical applications in traffic surveillance and management, such as traffic flow optimization and accident detection. Further studies can investigate the scalability and generalizability of the proposed approach to various traffic scenarios and different environments. Overall, this research contributes to the development of intelligent transportation systems and paves the way for future research in this field.

References

[1] Anan, L., Zhaoxuan, Y., & Jintao, L. Video vehicle detection algorithm based on virtual-line group. In *APCCAS 2006-2006 IEEE Asia Pacific Conference on Circuits and Systems* (pp. 1148-1151). IEEE. (2006)

[2] Basu, Mitra. "Gaussian-based edge-detection methods-a survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32.3 (2002): 252-260.

[3] Dong, Zhen, et al. "Vehicle type classification using a semisupervised convolutional neural network." *IEEE transactions on intelligent transportation systems* 16.4 (2015): 2247-2256.

[4] Fei, Mengjuan, Jing Li, and Honghai Liu. "Visual tracking based on improved foreground detection and perceptual hashing." Neurocomputing 152 (2015): 413-428.

[5] Hassaballah, M., Mourad A. Kenk, and Ibrahim M. El-Henawy. "Local binary pattern-based on-road vehicle detection in urban traffic scene." Pattern Analysis and Applications 23.4 (2020): 1505-1521.

[6] J., Xu., (2022). The Improvement of Road Driving Safety Guided by Visual Inattentional Blindness. IEEE Transactions on Intelligent Transportation Systems, 23(6), 4972-4981. doi: 10.1109/TITS.2020.3044927

[7] Jung, Heechul, et al. "ResNet-based vehicle classification and localization in traffic surveillance systems." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017.

[8] Li, Shuguang, et al. "Video-based traffic data collection system for multiple vehicle types." *IET Intelligent Transport Systems* 8.2 (2014): 164-174.

[9] Luo, Zhiming, et al. "MIO-TCD: A new benchmark dataset for vehicle classification and localization." *IEEE Transactions on Image Processing* 27.10 (2018): 5129-5141.

[10] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

[11] Pérez-Hernández, Francisco, et al. "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance." *Knowledge-Based Systems* 194 (2020): 105590.

[12] S.M.M.Rahman. https://mahbubur.buet.ac.bd/resources/DatabaseEBVT.htm ; 2015.

[13] Sengar, Sandeep Singh, and Susanta Mukhopadhyay. "Moving object area detection using normalized self adaptive optical flow." *Optik* 127.16 (2016): 6258-6267.

[14] Sharma, Poonam, et al. "Automatic vehicle detection using spatial time frame and object based classification." Journal of Intelligent & Fuzzy Systems 37.6 (2019): 8147-8157.

[15] Sharma, Poonam, et al. "Vehicle identification using modified region based convolution network for intelligent transportation system." *Multimedia Tools and Applications* (2021): 1-25.

[16] Sivaraman, Sayanan, and Mohan Manubhai Trivedi. "Active learning based robust monocular vehicle detection for on-road safety systems." *2009 IEEE intelligent vehicles symposium*. IEEE, 2009.

[17] Sobral, Andrews, and Antoine Vacavant. "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos." *Computer Vision and Image Understanding* 122 (2014): 4-21.

[18] Theagarajan, Rajkumar, Federico Pala, and Bir Bhanu. "EDeN: Ensemble of deep networks for vehicle classification." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017.

[19] Tomikj, Nikola, and Andrea Kulakov. "Vehicle Detection with HOG and Linear SVM." Journal of Emerging Computer Technologies 1.1 (2021): 6-9.

[20] Tsai, Luo-Wei, Jun-Wei Hsieh, and Kuo-Chin Fan. "Vehicle detection using normalized color and edge map." *IEEE transactions on Image Processing* 16.3 (2007): 850-864.

[21] Wang, Xinchen, et al. "Real-time vehicle type classification with deep convolutional neural networks." *Journal of Real-Time Image Processing* 16.1 (2019): 5-14.

[22] Xiao, Y., Zhang, Y., Kaku, I., Kang, R., & Pan, X. (2021). Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption. Renewable & sustainable energy reviews, 151, 111567. doi: 10.1016/j.rser.2021.111567

[23] Xiao, Y., Zuo, X., Huang, J., Konak, A., & Xu, Y. (2020). The continuous pollution routing problem. Applied mathematics and computation, 387, 125072. doi: 10.1016/j.amc.2020.125072

[24] Xu, J., Zhang, X., Park, S. H., & Guo, K. (2022). The alleviation of perceptual blindness during driving in urban areas guided by saccades recommendation. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 16386-16396.

[25] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

[26] Zhang, Fukai, Ce Li, and Feng Yang. "Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation." Sensors 19.3 (2019): 594.

[27] Zhang, Zehan, et al. "Maff-net: Filter false positive for 3d vehicle detection with multi-modal adaptive feature fusion." arXiv preprint arXiv:2009.10945 (2020).

[28] Zhang, Zhaoxiang, et al. "EDA approach for model based localization and recognition of vehicles." 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007.

[29] Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, *82*