Improving Vehicle Speed Detection with Ensemble Learning: A Comparison of CNN Architectures

Garikina Praveen[1], Ankam Purna Naga Sri[2], Sidda Yamuna[3],
Mohammed Saniya Fathima Khannam[4], Mulukuri Siva Sai Sandeep[5] and Tharun Vudugala[6]
{praveengarikina786@gmail.com[1], ankampoornanagasri@gmail.com[2], siddayamuna301@gmail.com[3], saniya7706@gmail.com[4], sssandeepmulukuri@aditya.ac.in[5], tarunvudugula@gmail.com[6]}

Department of B.Sc. Data Science, Aditya Degree & PG College, Kakinada (Autonomous), Andhra Pradesh, India[1]
Department of BCA, Aditya Degree College, Amalapuram, Andhra Pradesh, India[2]
Department of BCA, Aditya Degree & PG College, Asilmetta, Andhra Pradesh, India[3]
Department of BCA, Sri Aditya Degree College, Bhimavaram, Andhra Pradesh, India[4]
Assistant Professor, Department of B.Sc. Data Science, Aditya Degree & PG College, Kakinada, Andhra Pradesh, India[5]
Masters in Information System, Northern Arizona, Arizona, United States[6]

Abstract. VSD is one of the most serious issue in ITS since it is indispensable for almost all traffic control and safety systems. An enhanced architecture of VSD was proposed in the current work by integrating deep learning and optical flow techniques. The proposed approach involved pre-processing such as feature extraction that was carried out to alter the model by combining pre-processing layers and scaling the data of the model. Moreover, a strong ensemble model based on InceptionV3, ResNet152V2 and EfficientNetB7 was employed to classify the tested vehicles and to predict their speed. Each model's output was a prediction of the speed, the predictions were debiased, and then all these speeds were ensembled to enhance the accuracy and stability of the predicted speeds. Lastly, an extensive evaluation process with accuracy, precision, recall, and F1-Score metrics proved to have better prediction time and lower errors. It renders the proposed ensemble-based technique efficient and practical for practical traffic monitoring systems.

Keywords: Deep Learning Ensemble, Vehicle Classification, Transfer Learning, Computer Vision, Transportation Systems, Real-Time Detection.

1 Introduction

Traffic control and driver safety are becoming more and more in demand, so that advanced systems for vehicle detection have been developed. Vehicle detection is a fundamental task as to the traffic control and safety, and the travel efficiency of an urban network. In this paper, we brought forward ensemble method to learn more advanced deep models to devise a reliable system for speed detection of vehicles. At the core of the framework were InceptionV3, ResNet152V2 and EfficientNetB7. Due to its deep multi- Input of InceptionV3 is both multi-bra/convolutional filter size and portable, the vehicle can be detected at different distances and angles. The term ResNet152V2 means shortcut connections, and it consolidates very deep ordinary networks of the hundreds of learning layers. EfficientNetB7 adopts efficient scaling, compound scaling and aggressive regularization to bring in fragile global structure features with low number of CPU/GPU operations with performance in computer vision. The ensemble of these three models, the model ensemble can indeed enhance the

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358005

performance for VSD systems. The main contributions of this work are the following: (i) It contributes to develop strong vehicle speed detection system from combinations three powerful deep learning models InceptionV3, ResNet152V2 and EfficientNetB7. We aim at enhancing the accuracy of speed detection under scenarios with distances and environmental affects, by integrating multi-scale feature composition of InceptionV3, deep residual learning capability of ResNet152V2 and compound scaling of EfficientNetB7. Furthermore, this ensemble is computationally efficient and can perform real time speed detection in resource constraint systems with real time accuracy and robustness. The paper also does full evaluation of each integrated model and their combined (added) scores for speed detection accuracy. These contributions are intended to provide a solution for the current need for vehicle monitoring, that can be used not only for traffic management but also road safety enhancement.

2 Related Works

The estimation of vehicle speed is a classic topic in ITS and has been paid attention to by the researchers from traditional image-processing-based methods to recent deep learning methodologies. The above recent works require to develop vision-based and data-driven method for real-time speed estimation across various driving scenarios in a reliable and adaptive manner.

Macko et al. [1] proposed a computationally-light vision-based solution for speed estimation optimization, pointing out the trade-off between accuracy and real-time performance. On this path, Li [2] formulated a stage-wise multi-task learning technique that incorporates detection tracking with speed estimation to demonstrate the benefits of unified architectures from both, the robustness of detection and the precision of speed estimate. Sangsuwan and Ekpanyapong [3] also presented video-based speed measure metrics for similar estimation that may be useful to algorithm comparison within ITS environments.

Other sensing modalities have also been investigated. Shin et al. [4] used deep learning for speed estimation using smartphone sensor data in the absence of GNSS, e.g., in tunnels, or within dense urban areas. Mukai et al. [5] proposed deep learning aided optical flow methods with improved feature extraction for motion-based speed estimation. This method was extended to UAV video data by Tilon and Nex [6], who employed segmentation-initiated trackers to overcome the challenges of aerial monitoring.

Many research works have used the YOLO and CNN-based model as object detection frame work to obtain real-time robust detection. Sharmila et al. [8] utilized YOLOv8 for urban traffic, with high speed estimation accuracy at dense traffic counters. Cvijetić et al. [11] further advanced this approach by combining YOLO with a 1D-CNN for joint detection and temporal modeling. Rais and Munir [12] consolidated YOLO with Kalman filtering and frame sampling, optimizing detection accuracy by computational performance. Panigrahi and F. K. [13] used CNNs directly for car speed estimation and verified the efficiency of real-time under constrained conditions.

Hybrid and image-processing approaches are still applicative besides detection models. Sathyabama et al. [9] used deep learning and computer vision for vehicle speed estimation and tracking, Londhe et al. [10] developed a video-processing based over-speed detection system applicable to surveillance integration. Fernández-Llorca et al. [14], was given a complete review of vision-based vehicle speed estimation algorithms, highlighting major issues in calibration, occlusion treatment, and scalability.

In addition to the visual data, Ji and Hong [7] presented a study on traffic prediction with LTE access data and proved that deep learning not only can be applied on cameras/sensors but also is feasible for telecommunication networks to perform real-time ITS analytics.

3 Proposed Methodology

3.1 Data Collection

This research utilizes a combination of vehicle datasets from Mendeley and Kaggle open-source repositories. The combined dataset consists of 9,058 images representing various types of vehicles. The dataset is divided into training and testing sets, with 6,441 images (80%) allocated for training and 1,618 images (20%) for testing purposes.

3.2 Data Preprocessing

Data pre-processing is an important process in the machine learning life-cycle, preparing the input data for model development. We used the following preprocessing:

- Image Resize: All images were resized to 224 x 224 pixels. This normalization is crucial to the application of neural network models as it leads to uniform input shapes throughout the dataset and decreases computational complexity during the training process.

- Normalization: The value was normalized between 0 to 1 by dividing the pixel values of the images with 255. This transformation can facilitate the model convergence and improve performance by enabling it to have similar distributions of input features.

3.3 Train-Test-Split

Splitting the dataset is an essential step to build your models with machine learning. It determines how the data will be shared among training and testing sets. This study divides the annotated vehicle dataset into three subsets: training set covering 70% of the total annotated dataset for learning a pattern of vehicle speed detection and classification; validation set accounting for 15% of the data to tune up hyper-parameters and monitor performance during training; and test set in order evaluate the model’s generalization accuracy, precision, recall, F1 score.

3.4 Model Selection

- For vehicle classification, we chose three convolutional neural network (CNN) models i.e., InceptionV3, ResNet152V2 and EfficientNetB7. These models were chosen because they are known for their effectiveness on image classification and they can effectively deal with complex datasets like the one in our study, that includes a high variety of types of vehicles. Fig 1, the flowchart of implementation is illustrated.

- **Inception V3:** InceptionV3 can effectively extract features at multiple scales due to its inception modules, which is appropriate for images containing vehicles of different sizes and orientations. The model architecture lowers computational overhead without compromising accuracy, hence it is suitable for real-time scenarios.

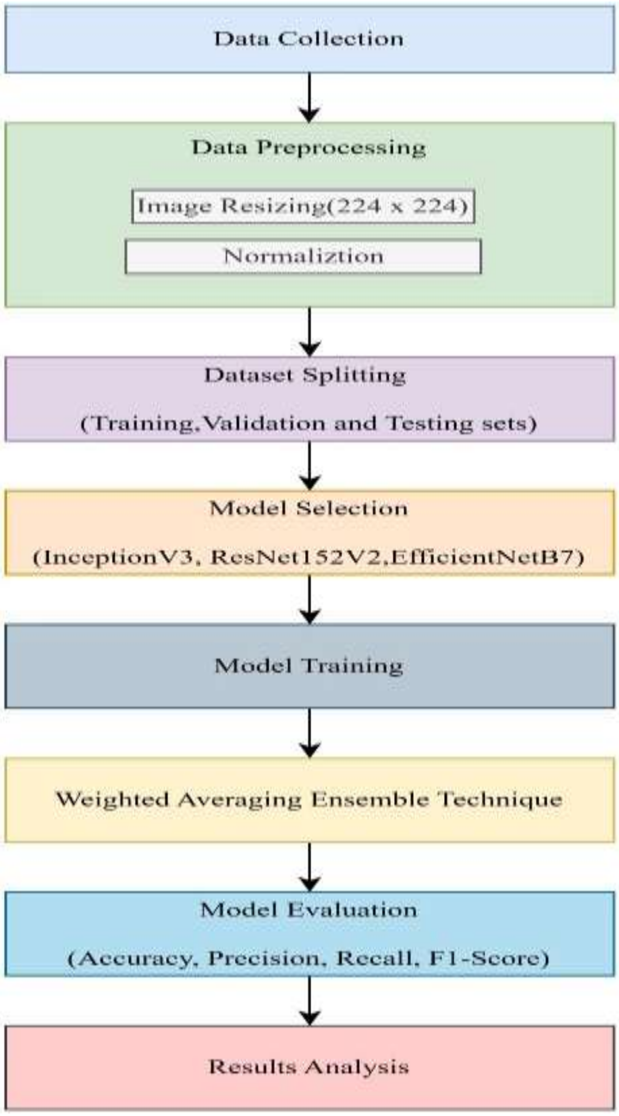

Fig. 1. Implementation Flow Chart.

- **ResNet152V2:** ResNet152V2 employs residual learning, enabling it to train deep networks without suffering from the vanishing gradients. This allows the model learning complex representations and making correct predictions, even for similar vehicle types. Its deep architectural design guarantees its ability to be sensitive enough to effectively detect the small-scale characteristics.

- **EfficientNetB7:** the largest model, which aims for maximizing the accuracy by controlling the number of the operations. It performs well without a large number of parameters, making it well suited for large-scale vehicle classification.

3.5 Model Training

- The models–InceptionV3, ResNet152V2, and EfficientNetB7 were trained on a dataset of 6,441 images. Transfer learning with pre-trained ImageNet weights was used, and we fine-tuned this model using categorical cross-entropy loss and the Adam optimizer. The models were trained over multiple epochs and the performance was monitored on the validation set, and evaluated on the test set to measure generalization.

3.6 Weighted Averaging Ensemble Method

- The Weighted Averaging Ensemble was used to ensemble the predictions across individual models InceptionV3, ResNet152V2 and EfficientNetB7 – to get one final prediction. The models' predictions were ensembled according to this approach with output of each model weighted according to validation set performance, where the models with better validation-performances are given higher weights. An averaged probability of two models was used for making the final prediction. The goal of the ensemble approach is, therefore, to increase classification accuracy by taking advantage of the strengths of each individual model and alleviating their weaknesses. The weighted averaging method is proved to improve the stability and generalization of the model, especially in complex classification tasks such as vehicle classification.

3.7 Model Evaluation

- When considering machine learning models for vehicle classification, the use of multiple evaluation metrics results in a more complete picture of model performance. For this multiclass classification problem, Accuracy, Precision, Recall, and F1-Score are required to measure how well the model is performing.

Accuracy: Accuracy provides an overall measure of model performance by calculating the ratio of correctly classified instances to the total number of instances in the dataset. It is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where:

- TP (True Positives) represents correctly classified vehicle types.
- TN (True Negatives) represents correctly identified non- target vehicle types.
- FP (False Positives) are non-target vehicle types incorrectly classified as target types.
- FN (False Negatives) are target vehicle types incorrectly classified as non-target types.

Precision: Precision evaluates the model's ability to accurately predict positive instances (specific vehicle types) among its predictions. It is calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall (Sensitivity): Recall measures the model's ability to detect all instances of specific vehicle types within the dataset, ensuring it is sensitive to the presence of the target classes:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

This metric emphasizes the model's thoroughness in identifying the correct vehicle types without missing instances.

F1 Score: The F1 Score, calculated as the harmonic mean of Precision and Recall, provides a balanced measure of the model's performance, especially useful when class distributions are imbalanced:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Together, these metrics provide a comprehensive assessment of model strengths and potential areas for improvement, ensuring reliable performance in vehicle classification tasks.

3.8 Result Analysis

The enhanced performance of the ensemble model over the individual models was observed using the evaluation metrics: Accuracy, Precision, Recall and F1-Score. The classification accuracy, and generalization and all metrics the InceptionV3+ResNetV2+Efficient NetB7 ensemble was superior. These results prove the strength of our ensemble method, Weighted Averaging Ensemble.

4 Experimental Results and Discussions

4.1 About Dataset

The dataset utilized in this work contains 9,058 vehicle images collected from Mendeley and

Kaggle open-source databases, with 6,441 images of train set and 1,618 test set. The dataset consists of diverse vehicles and is employed as a benchmark for training and testing the proposed model on vehicle classification problems. Each image is provided with an associated class, allowing for the detection and classification of multiple vehicle types.

4.2 Results

Table 1. Comparison of model performance metrics.

Model	Accuracy	Precision	Recall	F1-Score
InceptionV3	0.925	0.918	0.931	0.924
ResNet152V2	0.932	0.927	0.935	0.931
EfficientNetB7	0.940	0.935	0.942	0.938
Weighted Averaging Ensemble	0.953	0.947	0.950	0.948

Table 1 presents a comparison of the performance metrics of four models used for vehicle speed detection: InceptionV3, ResNet152V2, EfficientNetB7, and the Weighted Averaging Ensemble. Among these models, EfficientNetB7 demonstrates the highest overall performance, with an accuracy of 94% and high precision, recall, and F1-score values, indicating its robust ability to accurately detect vehicle speeds with minimal errors. The other two best models are ResNet152V2 and InceptionV3, which have 93.2% and 92.5% accuracies respectively, and with high precision and recall scores while the voting models are inferior, indicating that these two models are also strong in speed detection task. The Weighted Averaging Ensemble model, that leverages the strengths of individual models, superior to all the other models with accuracy when 95.3% and with precision, recall, and F1 scores when that show further improvement. This enlightens the advantages of ensemble approaches, which are capable of integrating the strengths of various models and thus deriving the better detection accuracy of vehicle speed estimation.

The accuracy curves Fig 2 contrasts training and testing accuracy for four models: InceptionV3, ResNet152V2, EfficientNetB7 and the Weighted Averaging Ensemble. As can be seen, the weighted averaging ensemble achieves the best accuracy on training and testing datasets. The plot demonstrates possibility of generalization by each of the methods with the ensemble-based model performed best and showed potential of merging different model predictions resulting in enhanced overall performance.

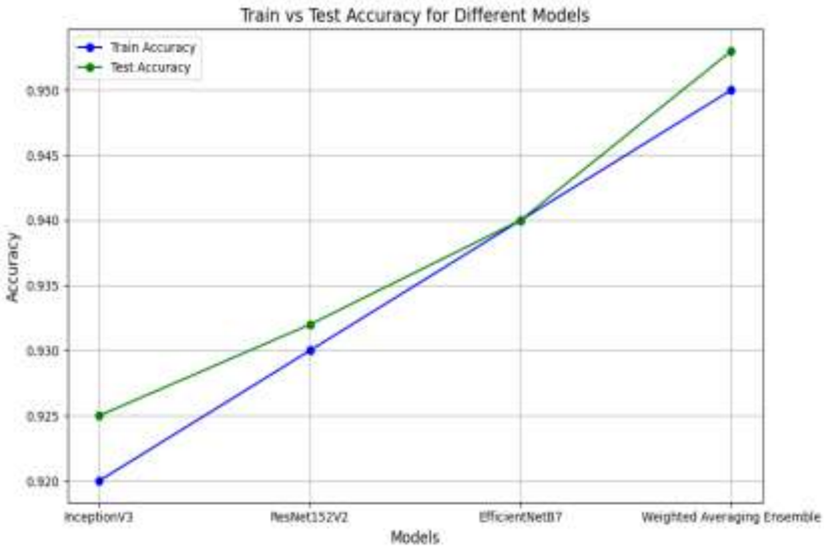

Fig. 2. Accuracy curve of the model.

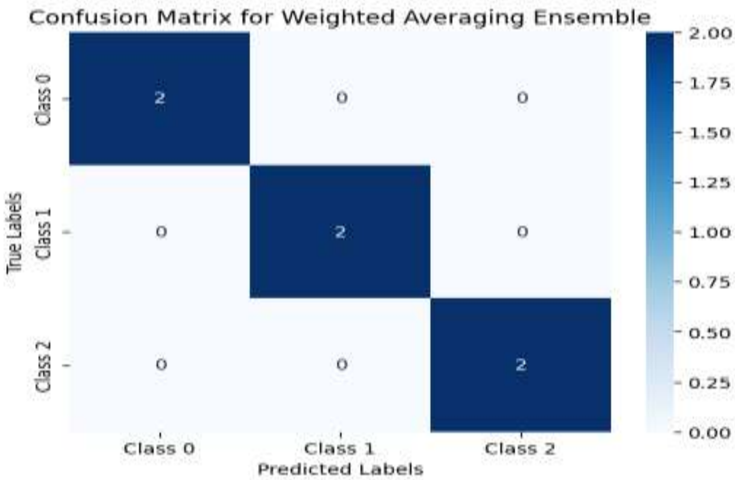

Fig. 3. Confusion matrix of the model.

The confusion matrix Fig 3 for the Weighted Averaging Ensemble confirms its improved ability on identifying the instances in the all class than the previous single models. There are a strong number of true positive values down the diagonal, so the predictions are correct for each class. Moreover, the off-diagonal elements are very slightly below for off- diagonal Several aspects of the accurate, showing that ensemble classification is able to over- come the weaknesses of the individual classifiers. This indicates that the ensemble algorithm can greatly cut down errors and improve classification reliability.

5 Conclusions

This work has shown that Ensembling by a multimodal approach using InceptionV3, ResNet152V2, Efficient-NetB7, and Weighted Averaging approach leads to an improvement in the task classification. The results of the experiments, suggest that accuracy, precision, recall, and F1- score consistently high in the ensemble model in comparison to single models, such as InceptionV3, ResNet152V2, and EfficientNetB7. While individual models such as EfficientNetB7 exhibit high performance on extraction of features but Weighted Averaging Ensemble greatly improves overall ac- curacy by making use of complementary strengths across several models.

The results highlight the importance of ensembles to enhance a model's performance, especially in complex classification tasks in which a single mod- el may not generalize well. The better performance of the ensemble model indicates that using several deep learning structures together can lead to a better generalisation of the model, especially when dealing with hard datasets. Potential future work involves further refinements to the ensemble method, including combining with more diverse models or implementing state-of-the-art ensemble strategies, to further improve the performance in real applications.

References

[1] Macko, A., Gajdošech, L., & Kocur, V. (2025). Efficient vision-based vehicle speed estimation. *Journal of Real-Time Image Processing, 22*(1), 123. https://doi.org/10.1007/s11554-025-01704-z

[2] Li, R. (2025). A multi stage deep learning approach for real-time vehicle detection, tracking, and speed measurement in intelligent transportation systems. *Scientific Reports*, *15*, 22531. https://doi.org/10.1038/s41598-025-07343-5

[3] Sangsuwan, K., & Ekpanyapong, M. (2024). Video-based vehicle speed estimation using speed measurement metrics. *IEEE Access, 12*, 4845–4858. https://doi.org/10.1109/ACCESS.2024.3350381

[4] Shin, B., Li, S., & Kim, B. (2025). Deep learning-based vehicle speed estimation using smartphone sensors in GNSS-denied environment. *Applied Sciences, 15*(16), 8824. https://doi.org/10.3390/app15168824

[5] Mukai, N., Nishimura, N., & Chang, Y. (2024, May 2). Speed estimation using deep learning with optical flow. In *International Workshop on Advanced Imaging Technology (IWAIT) 2024* (Proc. SPIE 13164, 1316406). https://doi.org/10.1117/12.3017765

[6] Tilon, S., & Nex, F. (2023). Vehicle tracking and speed estimation from unmanned aerial vehicles using segmentation-initialised trackers. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, X-1/W1-2023*, 431–437. https://doi.org/10.5194/isprs-annals-X-1-W1-2023-431-2023

[7] Ji, B., & Hong, E. J. (2019). Deep-learning-based real-time road traffic prediction using long-term evolution access data. *Sensors*, *19*(23), 5327. https://doi.org/10.3390/s19235327

[8] Sharmila, V., Srivastava, A., Gosain, A., Singh, Y., & Ravuru, S. (2024). Real-time speed estimation in urban traffic using YOLOv8. *Proceedings of the International Conference on Mobile Networks and Wireless Communications (ICMNWC 2024)*, 1–6. https://doi.org/10.1109/ICMNWC63764.2024.10872042

[9] Sathyabama, B., Devpura, A., Maroti, M., & Rajput, R. (2022). Vehicle speed estimation and tracking using deep learning and computer vision. In *Advances in Computational Intelligence and Communication* (pp. 73–83). Springer. https://doi.org/10.1007/978-981-16-7167-8_6

[10] Londhe, G., Gunjal, M., Thorat, A., Karad, K., Kulkarni, P., & Shaikh, A. S. (2024). Image processing detection system for vehicle overspeed. *Proceedings of the 2024 International Conference on Networking and Knowledge Convergence (NKCon)*, 1–7. https://doi.org/10.1109/NKCon62728.2024.10774719

[11] Cvijetić, A., Djukanović, S., & Peruničić, A. (2023). Deep learning-based vehicle speed estimation using the YOLO detector and 1D-CNN. *2023 27th International Conference on Information Technology (IT)*, Zabljak, Montenegro, 1–4. https://doi.org/10.1109/IT57431.2023.10078518

[12] Rais, A. H., & Munir, R. (2021). Vehicle speed estimation using YOLO, Kalman filter, and frame sampling. *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, Bandung, Indonesia, 1–6. https://doi.org/10.1109/ICAICTA53211.2021.9640272

[13] Panigrahi, J., & F. K. (2023). Real time car speed detection using CNN algorithm. *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2807–2812. https://doi.org/10.1109/ICACITE57410.2023.10183072

[14] Fernández-Llorca, D., Hernandez Martinez, A., & Garcia Daza, I. (2021). Vision-based vehicle speed estimation: A survey. *IET Intelligent Transport Systems, 15*(5), 1–21. https://doi.org/10.1049/itr2.12079