Kathford Journal of Engineering and Management,
ISSN: 2661-6106.
© 2024 *Kathford Journal of Engineering and Management*
https://kathford.edu.np/Journal/

VOL. 4 ISSUE 1 DECEMBER 2024, PP. 21-30

Multi-model Deep Learning Approaches for Vehicle Speed Estimation

Anish Thapa Magar[1], Subhadra Osthi[1], Neha Adhikari[1], Saban Kumar K.C.[,1]*

[1]Department of Computer and Electronics, Communication & Information Engineering, Kathford International College of Engineering and Management, Lalitpur, Nepal

**Corresponding Author: er.saban@kathford.edu.np*

ABSTRACT–Accurate vehicle speed estimates are critical for traffic monitoring and management. Traditional speed estimation approaches, such as radar guns or pixel displacement methods, are resource-intensive and frequently struggle in real-time or dynamic traffic situations. This paper describes a combination of YOLO for real-time object identification and Long Short-Term Memory (LSTM) networks for vehicle speed prediction that overcomes these constraints. The ability of YOLO to recognize cars with high precision, along with the capability of LSTM in processing sequential data, makes this approach extremely good for calculating speed from video data. Using the VS13 dataset of 400 single-vehicle videos, YOLOV5 detects vehicles and generates bounding box areas for LSTM to estimate speed. The performance across 13 car models averaged 5.79 km/hr RMSE. The accuracy and generalizability of the model can be improved in the future by adding outside variables like weather, road conditions, and driver behavior.

KEYWORDS– ***YOLOV5, RNN-LSTM, RMSE, VS13 Dataset***

1. INTRODUCTION

With rising urbanization and growing concerns about road safety and traffic management, reliable vehicle speed prediction is vital to maintaining efficient transportation networks. Vehicle speed estimation is an important component of traffic monitoring and smart city programs around the world, as it helps to optimize traffic flow, reduce congestion, and improve road safety. These technologies offer real-time decision-making, which is critical for effectively addressing modern transportation concerns.

In Nepal, traffic congestion and accidents have become major worries. According to a Department of Transport Management assessment, vehicle densities in the Kathmandu Valley surpass 1,000 cars per kilometer during peak hours, with traffic creating travel delays of up to 50%. Furthermore, vehicle accidents claimed over 2,800 lives in 2022, underlining the critical need for effective traffic monitoring systems. Despite these challenges, modern, automated traffic management solutions continue to be underutilized in developing countries such as Nepal. Traditional speed estimation technologies, such as radar guns and manual speed traps, are resource-intensive, time-consuming, and incompatible with real-time applications.

This paper presents a multi-model approach that combines YOLO for real-time vehicle recognition with Long Short-Term Memory (LSTM) networks for sequential speed estimation. YOLO's great precision in object detection, along with LSTM's capacity to analyze temporal data, overcomes the limitations of prior approaches. Traditional systems frequently fail to handle dynamic traffic circumstances or offer the real-time data required for effective traffic management. Meanwhile, many contemporary deep learning models are not robust enough to handle a wide range of traffic circumstances.

This approach is well suited to Nepal's traffic circumstances, where high vehicle numbers, unpredictable road conditions, and limited resources necessitate novel solutions. Using these modern technologies, our work presents a scalable and reliable technique for estimating vehicle speed, contributing to smarter and safer traffic control systems in Nepal and other developing countries.

1.1 LITERATURE REVIEW

Traditional computer vision approaches for estimating vehicle speed frequently rely on assessing the distance traveled by a vehicle over time or the time required to go a known distance. (Keattisak et al., 2020) used YOLOv3, DeepSORT, GoodFeatureToTrack, and the Pyramidal Lucas-Kanade optical flow algorithm to recognize and track automobiles in video data. By constructing virtual intrusion lines on the road, the system estimates speed by measuring pixel displacement and trip duration, with a mean absolute error (MAE) of 3.38 km/h and a root mean square error (RMSE) of 4.69 km/h. (Llorca et al., 2021) conducted a thorough study of vision-based vehicle speed estimation systems, highlighting the cost savings and possibility for reliable recognition without expensive range sensors. They classified different approaches and addressed the problems and benefits of vision-based systems.

Deep learning has transformed the field of vehicle speed prediction, delivering more accurate and resilient solutions. The YOLO (You Only Look Once) algorithm, developed by (Redmon et al., 2016), is a popular choice for object recognition due to its real-time processing capabilities and accuracy. YOLOv5, the most recent iteration, continues to improve on these characteristics, making it a popular tool among many academics. (Cvijetić et al., 2023) employed YOLO to detect vehicles and a one-dimensional convolutional neural network (1D-CNN) to estimate speeds. Their method involves estimating the change in the bounding box area around the vehicle as it moves, resulting in an average inaccuracy of 2.76 km/h on the VS13 data set.

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are excellent for sequential data and time-series analysis. (Salehinejad et al., 2018) evaluated improvements in RNNs, emphasizing their ability to learn long-term dependencies and its use in a variety of disciplines, including vehicle speed estimation. Similarly, (Bengio et al., 1994) examined the challenges and benefits of employing gradient descent to train RNNs, emphasizing the significance of capturing long-term dependencies in sequential data.

The availability of high-quality information is critical for designing and evaluating vehicle speed prediction models. (Djukanović et. al, 2022) developed a dataset for estimating vehicle speed using audio-video signals. This dataset contains recordings of 13 different cars traveling at established speeds, which serve as a useful baseline for researchers. Both conventional and deep learning-based approaches now in use have drawbacks when it comes to managing situations involving many vehicles, occlusions, or environmental changes. These issues are successfully addressed by the YOLO and LSTM combo. While LSTM is perfect for speed estimation based on temporal data because it can capture sequential dependencies, YOLO is excellent at real-time object detection. Especially in single-vehicle situations, this innovative method guarantees resilience in a variety of traffic situations and establishes the groundwork for expansion to multi-vehicle settings.

2. COMPUTATIONAL AND THEORETICAL DETAILS

The deep learning approach for estimating vehicle speed is divided into five important steps, as shown in Figure 2: data collection, data preprocessing, model selection, model training, and model evaluation. Each phase is critical to the entire vehicle speed estimation process:

Data collection: The VS13 dataset (Radovic, 2022), with its 400 video recordings, demonstrates a thoughtful design for speed estimation tasks by including diverse vehicle types and speeds. The training-testing split (80% for training and 20% for testing) follows standard practices to ensure effective model generalization.

The distribution of videos across car models as shown in Figure 1, ranging from 30 to 35 recordings per model, ensures balance, which is critical for mitigating bias in training and evaluation. This uniformity allows models to learn features representative of a wide range of vehicle types, enhancing the robustness of predictions.

Data preprocessing: It involves extracting frames from each video and using YOLOv5

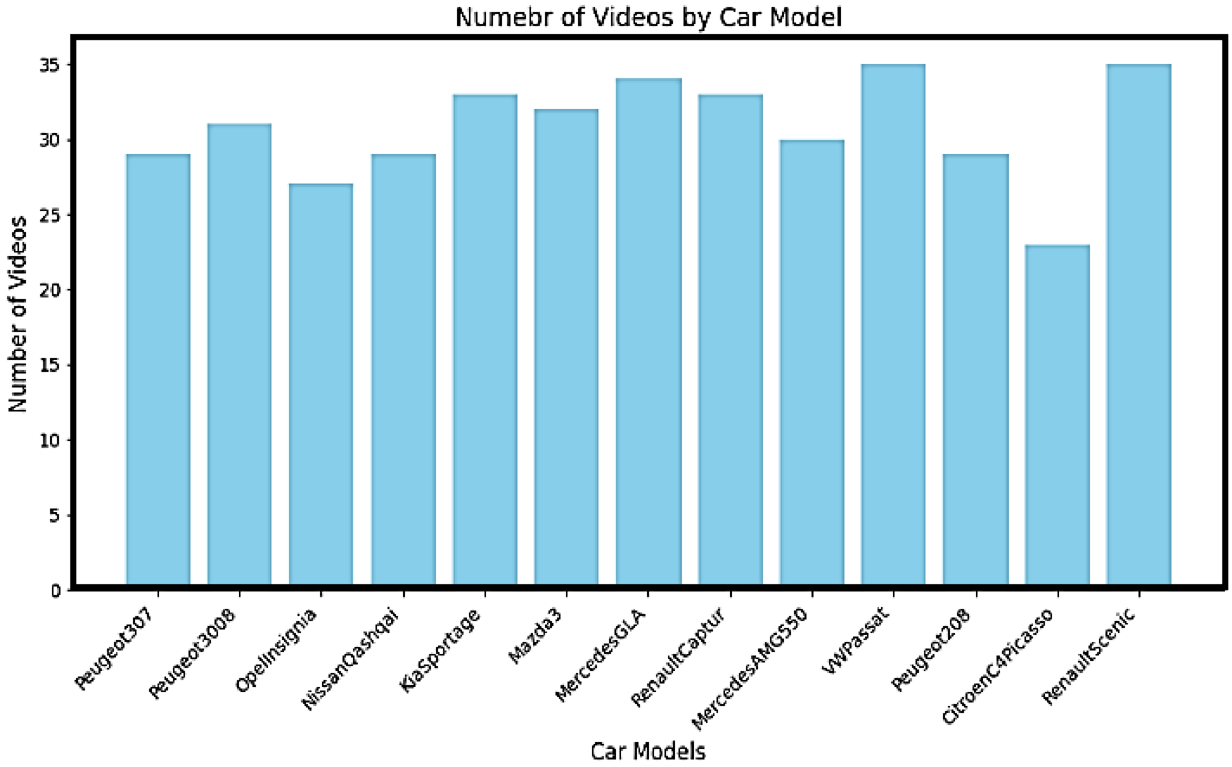

Figure 1. Dataset Distribution

(You Only Look Once, version 5) to construct bounding boxes for vehicle recognition. The bounding box region in each frame is the primary feature for estimating the vehicle's speed. The retrieved data is then standardized to ensure compatibility with the model and subsequent processing.

Model Selection: YOLOv5 was chosen for vehicle recognition since it has better real-time object detection capabilities than previous versions of YOLO and related algorithms. YOLOv5 offers efficient, precise detection and is ideal for real-time applications such as traffic monitoring.

Model Training: After establishing the bounding boxes and preprocessing the data, the model is trained for vehicle recognition using YOLOv5. The training process ensures that the model can reliably identify and track vehicles in real time.

Model Evaluation: During this phase, the model's performance is evaluated by running the system through a collection of unseen data (the 80 reserved videos). The evaluation is based on how well the model estimates vehicle speed by examining the temporal dependencies between consecutive frames acquired by the LSTM (Long Short-Term Memory) network.

Since LSTM is excellent at managing sequential data and capturing long-term relationships, it is essential for speed estimation. This allows for the accurate calculation of the vehicle's speed over time. More precise and reliable vehicle speed prediction is possible when YOLOv5 for real-time object identification and LSTM for temporal analysis are combined, particularly in dynamic, real-world traffic situations. Furthermore, YOLOv5's sophisticated architecture detects objects in complicated settings, making it suitable for a wide range of traffic scenarios.

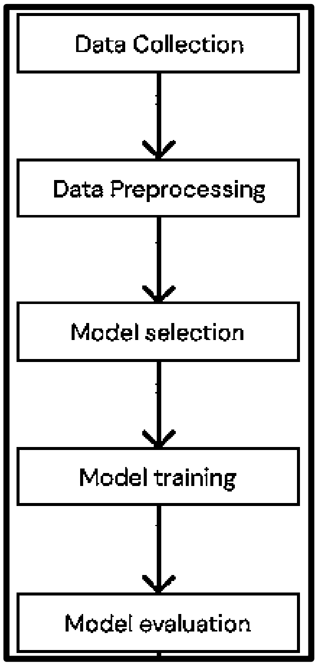

Figure 2. Flow diagram

Figure 3 represents an overview of YOLOv5 architecture. YOLOv5 (You Only Look Once version 5) is a popular real-time object detection model that retains the architecture of the original YOLO framework while including new design modifications to boost accuracy and speed. Here's an overview of its architecture:

- **Backbone:** YOLOv5 employs the CSPDarknet53 backbone, a modified version of Darknet53. The backbone extracts feature maps from the input image. CSP (Cross Stage Partial) networks are used to improve learning by separating the feature map into two halves, which reduces computation while increasing model accuracy.
- **Neck:** The neck of YOLOv5 is made up of PANet (Path Aggregation Network), which helps the model gather input from

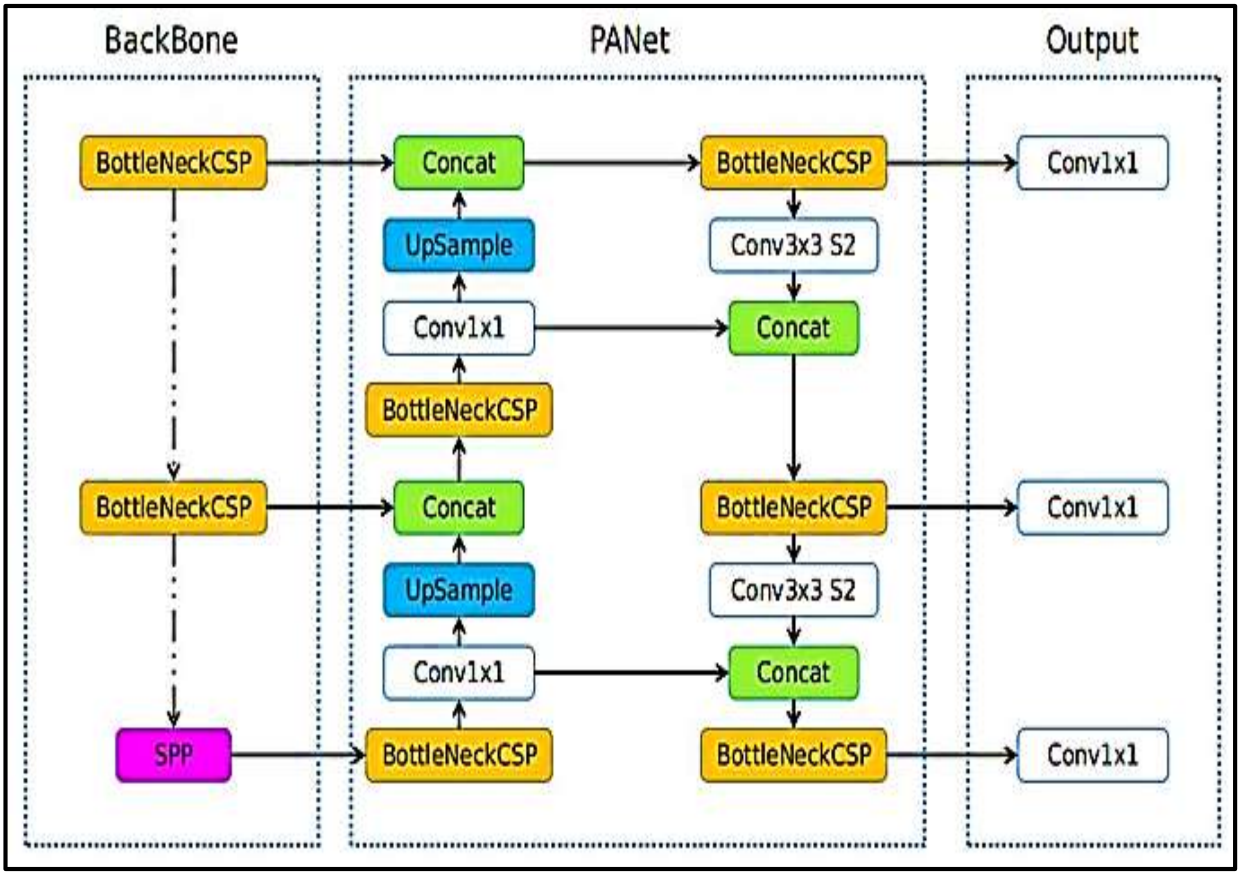

Figure 3. YOLOV5 Overview (Github, 2021)

many feature layers. PANet enables the merging of low-level and high-level characteristics, increasing object detection at various sizes.

- **Head:** The head is responsible for making final predictions. YOLOv5 predicts bounding boxes, object confidence scores, and class probabilities using three distinct scales of anchor-based detection heads. This multi-scale method improves the ability to recognize objects of varied sizes.
- **Output:** YOLOv5 predicts objects using anchor boxes and uses non-max suppression (NMS) to reduce redundant overlapping boxes. It returns the final bounding boxes, confidence scores, and object class labels.

YOLOv5's architecture is lightweight and geared for speed and accuracy, making it efficient for real-time objects.

Similarly, another architecture of LSTM model used herewith is shown in Figure 4. LSTM (Long Short-Term Memory) models are a sort of recurrent neural network (RNN) that is ideal for time-series data, such as vehicle speed estimation. Here's a brief summary of how LSTMs are used:

- Input: Sequential data, such as previous vehicle speed, acceleration, and other time-dependent variables.
- Memory Cells: LSTM features memory cells that maintain essential patterns over time, which addresses the vanishing gradient problem found in ordinary RNNs. This helps to capture long-term dependencies in vehicle dynamics.

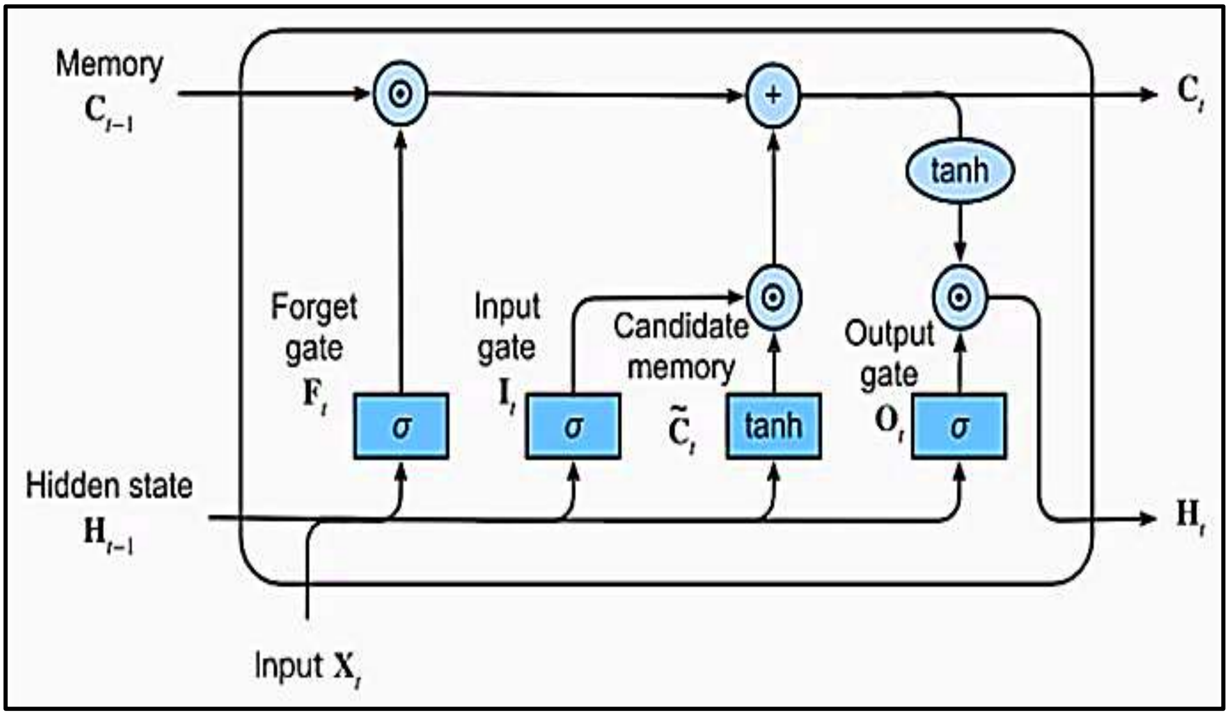

Figure 4. LSTM Model Architecture (Calzone, 2018)

- Output: The LSTM examines the sequence and predicts future vehicle speeds using previously learnt patterns.

LSTMs are efficient in modeling the temporal dependencies required for accurate vehicle speed prediction.

Root Mean Square Error (RMSE) is a commonly used metric to measure the accuracy of a regression model. It represents the square root of the average squared differences between the predicted values and the actual values.

The formula for RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Where,

$\hat{y}_i$ = predicted value for the i-th data point.

y_i = actual value for the i-th data point.

n = total number of data points.

Since both the predicted values and the actual values are in km/hr , the RMSE is also expressed in km/hr.

3. RESULTS AND DISCUSSIONS

The LSTM model is trained over 600 epochs with the bounding box region of 80 consecutive frames as input. The model reduced the RMSE from 60 km/hr to approximately 4 km/hr during training.

The model was tested on the reserved set (around 80 single-vehicle videos). The average RMSE of 13 vehicle models was 5.7931 km/hr, with individual RMSEs ranging from 4.0602 to 7.6198 km/hr.

During the model's training, the RMSE is significantly reduced. Initially, the RMSE was in the 60 km/hr, but it quickly declined, reaching around 10 km/hr within the first 300 epochs. After 600 epochs, the RMSE stabilized at 4 km/hr, suggesting that the model had successfully learned to estimate

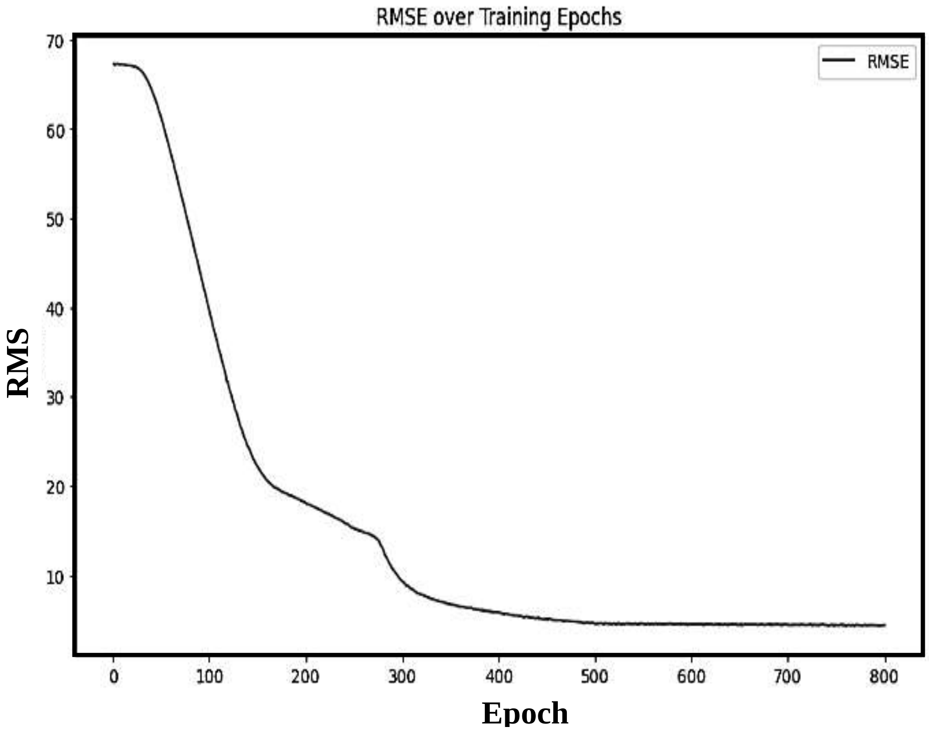

Figure 5. Model Training History

speed with high accuracy. Figure 5 illustrates the training curve. The RMSE values for individual vehicles are listed in the Table 1.

Table 1: Speed Detection Performance (RMSE) on Different Cars' Dataset

	Vehicle (Car)	RMSE (km/hr)
1	Peugeot 307	5.5319
2	Renault Captur	6.7812
3	Peugeot 208	4.3728
4	Nissan Qashqai	5.3187
5	MercedesAMG550	**4.0602**
6	Mercedes GLA	7.4322
7	CitroenC4Picasso	4.4368
8	Kia Sportage	5.4585
9	Renault Scenic	**7.6198**
10	Peugeot 3008	6.2481
11	Opel Insignia	6.0126
12	Mazda3	6.4512
13	VW Passat	5.5866
Average		5.7931

The LSTM model for speed estimate performs well during both training and testing.

- Training Performance: RMSE decreased from 60 to 4 km/hr across 600 epochs, with substantial progress in the first 100 and stabilization after 300 epochs.

- Individual Vehicle Test Performance: The RMSE ranged from 4.0602km/hr (Mercedes AMG550 whose performance is shown in Figure 6) to 7.6198 km/hr (Renault Scenic whose performance is shown in Figure 7), with an average of 5.7931 km/hr, indicating acceptable accuracy across thirteen automobiles.

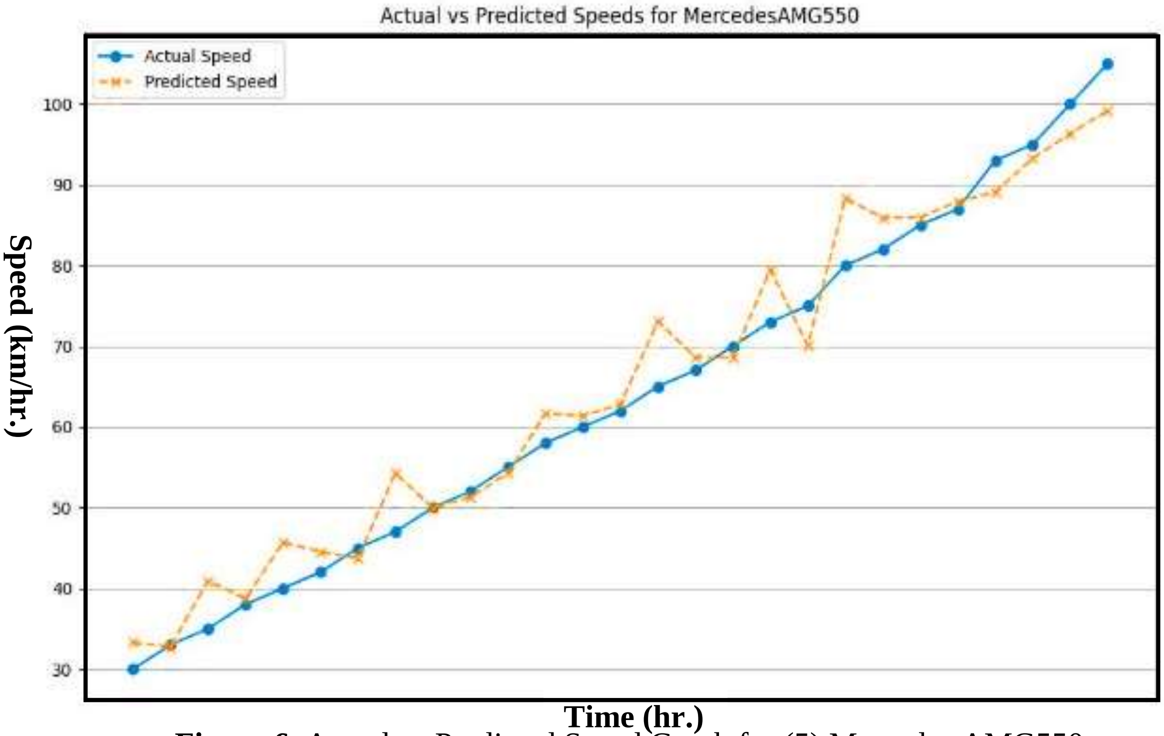

Figure 6. Actual vs Predicted Speed Graph for (5) Mercedes AMG550

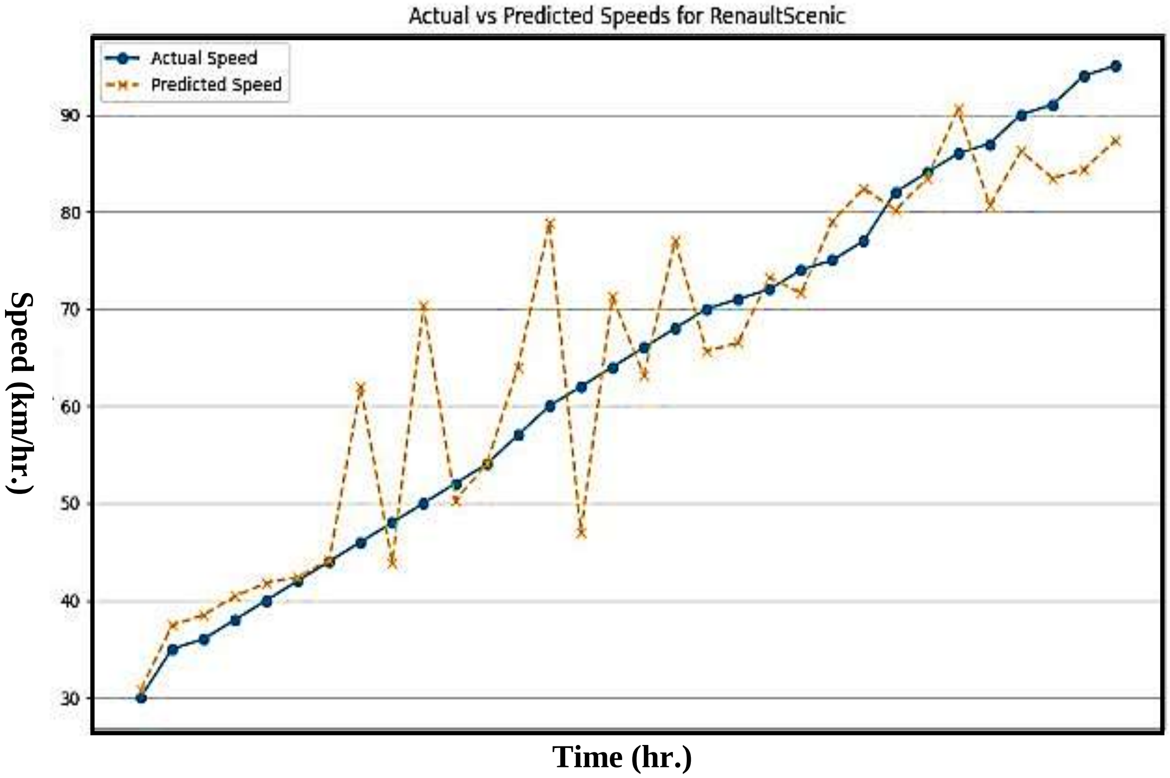

Figure 7. Actual vs Predicted Speed Graph for (9) Renault Scenic

According to the observations, the actual and anticipated speeds are closely aligned, suggesting that the model is effective at forecasting the vehicle's speed.

Minor variations can be detected at spots where the orange line deviates slightly from the blue line, suggesting cases of prediction error. Overall, both lines' trends are very comparable, indicating a high degree of agreement between actual and anticipated values.While the model generalizes well overall, vehicle-specific variances (e.g., Renault Scenic) indicate areas for improvement in some circumstances as shown in Figure 7.

The LSTM model for vehicle speed estimate outperforms both classic and deep learning approaches. Traditional approaches, such as those used by Sangsuwan and Ekpanyapong (2020), use pixel displacement and virtual lines to estimate speed, with an RMSE of 4.69 km/h. On the test set, the LSTM model has a larger RMSE of 5.79 km.hr, but it is still competitive because LSTMs capture sequential relationships better.

Deep learning has transformed speed estimation. Cvijetić et al. (2023) used YOLO with a 1D-CNN to achieve an error of 2.76 km/h by utilizing variations in bounding box area. Although their model exhibited higher accuracy, Salehinejad et al. stress that LSTM's ability to understand long-term dependencies, as demonstrated by reducing RMSE from 60 to 4 km/hr during training, remains robust for temporal data.

An important factor in the model's performance is the dataset. Similar to the current work, Djukanović et al. (2022) used a dataset with 13 cars. Even though the LSTM model's average RMSE was 5.79 km/hr, errors could still be decreased by improving the dataset or fine-tuning the architecture.

4. CONCLUSION

This study was driven by the need to precisely estimate vehicle speeds for applications such as traffic management, autonomous driving, and safety systems, which require exact temporal modeling. The choice of LSTM arises from its demonstrated ability to handle time-series data and sequential dependencies, which provides a significant advantage over older methods that frequently fail to account for such complexities. While CNN-based models and YOLO have demonstrated superior accuracy in previous studies, LSTMs were chosen for their capacity to capture time-dependent patterns in vehicle speed data.

The model showed an average RMSE of 5.79 km/hr over 13 vehicles, demonstrating its general robustness. However, larger error rates in some circumstances indicate the need for additional tuning or hybrid approaches. This performance grade emphasizes its suitability for real-world use, particularly in scenarios that require real-time speed predictions or adaptive learning systems.

In the future, the model can be expanded to include external elements such as road conditions, weather, and driver behavior to improve its accuracy and generalizability.

ACKNOWLEDGMENTS

The authors would like to thank the faculty members of department of computer and electronics engineering for their invaluable suggestions. The entire editorial and reviewer board members are acknowledged for their constructive reviews.

REFERENCES

1. Sangsuwan, K., and Ekpanyapong. (2020). Video-based vehicle speed estimation using speed measurement

metrics. *School of Engineering and Technology, Asian Institute of Technology*, Khlong Nueng, Thailand.

2. Llorca, D. F., Martínez, A. H., and Daza, I. G. (2021). Vision-based vehicle speed estimation: A survey. *[Online]*.
3. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). Las Vegas, NV, USA.
4. Cvijetić, A., Djukanović, S., and Peruničić, A. (2023). Deep learning-based vehicle speed estimation using the YOLO detector and 1D-CNN. In *Proceedings of the 27th International Conference on Information Technology (IT)* (pp. 1-4). Zabljak, Montenegro.
5. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2017). Recent advances in recurrent neural networks. Available: *https://arxiv.org/abs/1801.01078.*
6. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157-166.
7. Djukanović, S., Bulatović, N., and Čavor, I. (2022). A dataset for audio-video-based vehicle speed estimation. *Proceedings of the 30th Telecommunications Forum (TELFOR)* (pp. 1-4). Belgrade, Serbia.
8. Radovic, S. (2022). *VS13 Dataset for Vehicle Speed Estimation*. Available: *https://slobodan.ucg.ac.me/science/vs13/paper.pdf*
9. GitHub. (2021). YOLOv5 overview [Image]. Available: *https://github.com/ultralytics /yolov5/issues/280*
10. Calzone, O. (2018). An intuitive explanation of LSTM. Medium. Available: *https://medium.com/@ottaviocalzone/an-intuitive explanation-of-lstm-a035eb6ab42c*