INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 14-06-2023

Accepted: 27-11-2023

Published: 20-02-2024

Citation: Siddiqua A, Saher A, Sumera (2024) Vehicle Speed Detection using Haar Cascade Classifier and Correlation Tracking. Indian Journal of Science and Technology 17(8): 741-750. https://doi.org/10.17485/IJST/v17i8.1458

* **Corresponding author**.

ayesha.siddi10@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Vehicle Speed Detection using Haar Cascade Classifier and Correlation Tracking

Ayesha Siddiqua[1]*, Amena Saher[2], Sumera[3]

1 M.S. Student, Department of Computer Science, Missouri University of Science and Technology, Rolla, 65409, MO, USA
2 B.E. Student, Department of Computer Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, Telangana, India
3 Assistant Professor, Department of Computer Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, Telangana, India

Abstract

Objectives: The aim of this study is to develop an efficient and cost-effective solution for predicting vehicle speeds using recorded video data. **Methods:** The proposed system employs a combination of image processing techniques and computer vision to calibrate cameras for traffic simulation, enabling the extraction of information on average vehicle speeds. It utilizes the Haar Cascade Classifier for object detection, followed by a correlation tracker for vehicle tracking. Speed estimation is achieved through the frame differencing method. The dataset comprises 90 minutes of recorded data from highway cameras, showcasing diverse traffic scenarios with various vehicle types (trucks, trailers, cars, buses, and bikes) at varying speeds. Predicted values are compared with ground truth data obtained from a GPS-equipped car, using Mean Absolute Error (MAE) as the evaluation metric. **Findings:** The algorithm's performance is evaluated, resulting in an average error rate of 1.72 km/h (2.07%). These findings are compared with state-of-the-art data. **Novelty:** This study introduces a novel system that combines the Haar Cascade Classifier, correlation tracker, and frame differencing method to track vehicle positions, incorporating bike detection into the analysis, and calculate their moving speeds. A relative analysis underscores the system's performance, emphasizing its effectiveness in real-world applications and demonstrating refinement in accuracy assessment.

Keywords: Image processing; Vehicle speed estimation; Haar Cascade Classifier; Correlation tracker; Error rate calculation; Computer vision

1 Introduction

The rapid urbanization of our cities has led to a significant increase in traffic congestion, thereby necessitating the implementation of advanced traffic surveillance systems. While traditional radar speed measurement tools have been widely employed for this purpose, they have limitations in detecting smaller vehicles with weaker echoes and swiftly changing vehicle speeds. Therefore, there is a need for a better technique to detect the speed of the moving vehicles. Unmanned aerial vehicles are used to not only monitor traffic but also to manage situations such as accidents and traffic congestion, and they are also used in the calculation of different traffic parameters, such as traffic density and vehicle speed. These developments give rise to an interdisciplinary field encompassing computer vision, image processing, artificial intelligence, and numerous other disciplines, which is of great importance for study in academic research[1,2].

Researchers in the field of computer vision and image processing have proposed numerous approaches that contribute to the task of speed detection of vehicles. [3] proposed a solution for roadside LIDAR (Light Detection and Ranging) object detection using Range and Intensity Background Subtraction. The method retrieves a mobile object from a definite image and the retrieved object is resulted as the threshold of image differencing. However, the results are affected in poor lighting or bad climatic conditions and acts as a drawback to this method. Additionally, this approach for vehicle tracking generates false positives by identifying pedestrians.

A traffic surveillance system for obtaining comprehensive vehicle information is proposed based on instance segmentation which is realized by Mask R-CNN (Mask Region-based Convolutional Neural Network). The results identified from multi-frames are obtained by means of the tracking method SORT for processing[4]. However, SORT (Simple Online and Real-time Tracking) does not handle occlusions (objects temporarily blocking each other) as effectively as correlation tracking methods. When objects overlap or occlude each other, SORT has difficulty correctly associating objects with their respective tracks and requires processing each detection in every frame to perform data association.

Feature based modelling is a technique used for identifying image displacements by detecting and tracking features such as edges and corners in a series of two or more images. These features are restricted to a two-dimensional space. This is usually carried out in two steps. Features are located in a series of two or more images in the first step. The second step involves matching these features between frames to determine the motion vectors. Research has been done to focus on improving feature detectors to increase their precision and reliability. However, one challenge is the potential for ambiguity in the matching process. Unless it is known beforehand that the image displacement is less than the distance between features, there can be multiple possible matches.

This paper proposes a method for detecting vehicles and also tracking them through the use of cascade classifier and centroid tracking[5], while incorporating georeferencing and coregistration of acquired images and then proceeding on to extract lanes. These methods focusing on tracking the centroid of the region representing the vehicle, which can be considered as the contour (either convex hull or concave), or some model such as the convex hull of the contour, an ellipse or the bounding box fitted to the detected blob. The use of the centroid as a representative state of the vehicle is very unreliable. On the one hand, most of the blob detection approaches are not very accurate and the vehicle contour may vary in the sequence due to multiple factors (shadows, overlaps, close objects etc.). This makes the centroid rarely representing the same point of the vehicle in time.

Another tracking method proposed by[6,7] combines DaSiamRPN (Distractor-aware Siamese Region Proposal Network) and HOG-SVM (Histogram of Oriented Gradients-Support Vector Machine) with a key frame is used for detecting and tracking the athletes in races. Before the key frame is found, the DaSiamRPN is applied as the tracking method. In each frame after the key frame, the detection window is enlarged based on the bounding box generated by the DaSiamRPN tracker. However, here the proposed system asks for a cost-effective solution and the DaSiamRPN due to its more involved architecture, does not perform well on resource-constrained devices.

Monocular-based distance estimation is an approach used for speed estimation. First, based on intrusion or augmented lines or regions. These approaches do not require calibration of the camera system but measure the real distance between two or multiple virtual lines on the road, or the actual size of a road region (see Figure 1). Then, the problem of distance estimation is posed as a detection problem in which all vehicles are detected at the same distances whenever they cross the predefined virtual lines or regions. Since the virtual lines or regions are placed on the road, accurate distance estimation involves the accurate location of the contact point of some part of the vehicle. This part of the vehicle should be the same at the second location to obtain a coherent estimation of the speed. The limitation of the current system is that it needs human supervision for defining the region of interest. The user has to define an imaginary line where the centroid of the contours intersects for the counting of vehicles; hence, the accuracy is dependent on the judgment of the human supervisor.[2]

An approach presented by[8] introduces the concept of an Adaptive Traffic Light Control System (ATLCS) that employs Vehicular Density Value (VDV) for traffic assessment and the Statistical Block Matching Approach (SBMA) for object tracking and focuses on an ATLCS that adapts traffic signal timings based on real-time traffic conditions. This paper utilizes VDV for assessing vehicular density within specific Areas of Interest (AOIs). VDV calculates the average pixel values in the selected region to determine traffic density. Additionally, the paper employs the Statistical Block Matching Approach (SBMA) for object tracking, which involves dividing images into blocks and tracking objects' movements over time. The limitation of the current system is that it may encounter challenges related to computational intensity, real-time adaptability in complex urban traffic scenarios, and potential limitations in the accuracy of Vehicular Density Value (VDV) calculations under varying lighting conditions.

Another paper[9] introduces a real-time vehicle detection and Vehicle Speed Measurement (VSM) system using image processing techniques, specifically morphology operations and binary logical processes. The process begins with capturing an image from a video sensor and selecting Regions of Interest (ROI) through a dual-line approach where the two lines are separated by a measurable distance[10]. Subsequently, morphology operations are applied, involving the selection of a structuring element (S.E.). To enhance object tracking in unplanned traffic scenarios, Kalman filters are integrated into the system, contributing to effective vehicle tracking under diverse conditions. The proposed method exhibits an accuracy of 0.87. However, a significant limitation of this system is the use of a dual-line approach to define the ROI, which may present constraints in certain scenarios.

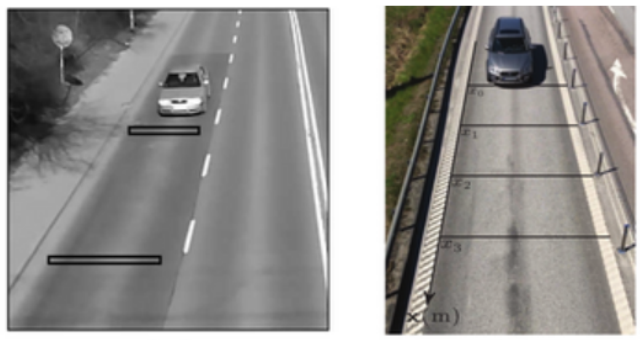

Fig 1. Monocular-based distance estimation

A method proposed by[11] for calculation of the speed, after the accurate tracking of the vehicle, involves detecting the vehicle with the same vehicle ID between two frames at two different time intervals. Intrusion lines on the frame perpendicular to the vehicle moving direction are drawn. The entry point of the frame is observed, and the number of frames elapsed is determined as the vehicle leaves the exit point[12].

Speed = N*d/FPS (Frames Per Second) is the equation used to compute speed. This method requires the actual on-site distance between the entry point and exit point in meters. As a result, it can only be used for one particular type of input, because the distance in pixels would vary for every type of input.

In our approach, feature extraction is executed by discerning and selecting distinctive features from both positive and negative samples, employing a pre-trained Haar Cascade classifier designed for the detection of cars, trucks, and bikes. The detection process is carried out using the openCV (Open Source Computer Vision) library. Following the detection of vehicles, our subsequent objective involves vehicle tracking. To achieve this, we employ correlation tracking, which outperforms the traditional centroid tracking method[13,14]. Vehicle speed is then calculated using the focal points within the vehicle's region. Subsequently, we evaluate the algorithm's performance by comparing the estimated vehicle velocity with the ground truth.

For the objective of speed estimation of vehicles, there has been little information about comprehensive measurement of certain key metrics[6,15–17]. In this study, we contribute to the field by refining accuracy assessment and quantifying errors in vehicle speed estimation.

2 Methodology

Vehicle Speed Estimation involves three steps to find out speed of vehicles using cameras:

- Vehicle Detection
- Vehicle tracking
- Speed Calculation
- Monitor over-speeders.

Figure 2 demonstrates the entire workflow of the algorithm. Initially we have to convert video to frames and identify vehicles from frames by tracking the individual vehicles and assigning unique ID to each one, after which according to their positions on each frame, their speed is calculated with the help of speed formula. Lastly, we store the cropped images of each vehicle along with their ID's and speed's. Vehicles exceeding the speed limit are stored in a separate folder.

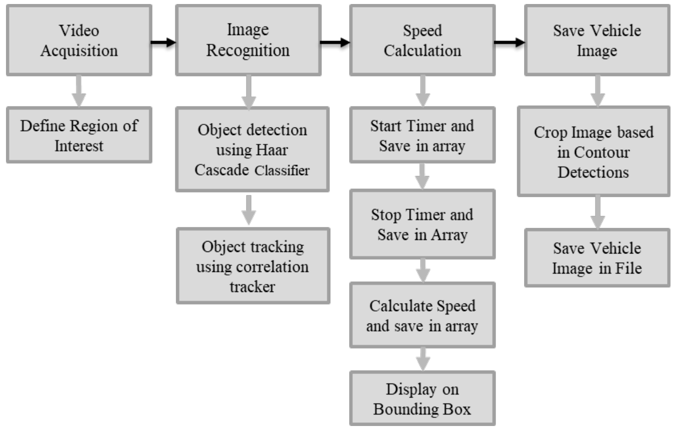

Fig 2. Architecture describing the workflow of the speed estimation algorithm

- **Vehicle Detection**

This paper leverages the Haar cascade classifier for the comprehensive detection of cars, bikes, and trucks. A dataset is curated with positive samples encompassing the target objects in various orientations, scales, and lighting conditions, along with negative samples for contrast. Haar-like features utilizing simple rectangular filters capturing intensity differences are extracted and processed through the Adaboost algorithm [18–20]. The resulting classifier is organized into a cascade of stages, efficiently rejecting non-object regions. In our project, we utilize a pre-trained Haar Cascade Classifier, eliminating the need for new training.

For car identification the classifier is tailored to detect size, shape, and symmetry patterns, ensuring precise localization. Similarly, the classifier is fine-tuned for bike recognition, focusing on frame shape, wheels, and handlebars, providing accurate identification in diverse conditions. In the case of trucks, the classifier recognizes larger structures and unique proportions associated with trucks, such as the box-like cargo area. Meticulous training enhances the classifier's discriminatory power, facilitating accurate and robust identification of cars, bikes, and trucks in the video data.

- **Vehicle tracking**

We utilized the dlib library to assign a unique ID to each car and store it [21]. This approach greatly facilitated the identification of vehicles in the video by referencing the assigned ids [22,23]. Additionally, using dlib, we were able to store the coordinates of each vehicle from the video. This stored information proved valuable for calculating the vehicle's speed and provided further insights for analysis.

- **Speed Calculation**

To calculate the speed of the tracked vehicle, we measure the distance it moves in one second, expressed in pixels [24]. By converting this distance to meters, we can obtain the speed of the vehicle. Initially, using the dlib tracker, we calculate two sets of coordinates for each car, denoted as [x1, y1, w1, h1] and [x2, y2, w2, h2]. Once the car crosses a specific point, we can generate the speed using the "d pixels" formula, which represents the distance between pixels [25,26].

$$Distance\ in\ pixels\ = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$$

$$Speed\ = \frac{distance\ in\ pixels}{pixel\ per\ meter} \times frames\ per\ second$$

Over-speeding vehicles: after the speed is calculated, we are able to store the id and speed of car along with the captured car image in a folder titled traffic record, i.e. The image of each car captured is labeled after its car ID and its speed. The car's which have crossed the specified speed limit, are to be saved in a separate folder titled 'exceeded'.

3 Results and Discussion

The dataset consists of approximately one hour of data collected from cameras monitoring a major highway with normal to heavy traffic. The dataset is composed of a wide range of vehicles, including trucks, trailers, cars, and buses, each traveling at various speeds. The input frame resolution is 1280 × 720 pixels, providing a clear and comprehensive view of the traffic situations. The data was acquired using smartphones as recording devices, capturing the traffic scenes at a frame rate of 30 frames per second (fps). The dataset effectively captures the dynamic traffic flow, reflecting the fast-paced nature of the vehicles moving along the highway.

The camera starts recording and detecting the vehicle moving on the road. The rectangle that surrounds the vehicle depicts the object (i.e. Vehicle) moving. After the tracking period is over, the speed is calculated and displayed around the rectangle frame. Multiple cars are being detected and their speeds are being calculated simultaneously.

Fig 3. Bounding box and estimated speed displayed

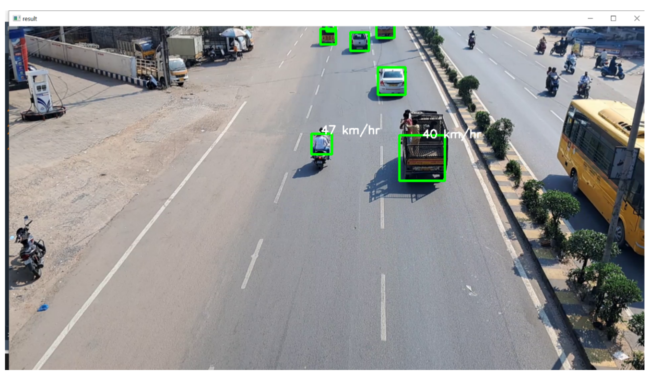

Fig 4. Recognition of trucks and bikes

The algorithm can be applied to any input video and is not specific to only one type of input. In Figures 3 and 4, the speed of the vehicles is calculated and displayed (vehicles include cars, bikes, and trucks).

All the vehicles being tracked are cropped appropriately, and are saved in the traffic record folder (Figure 5). Once the vehicle moving on the road is found overspending its image is captured and stored in a folder named 'exceeded' as shown in Figure 6.

In Figure 7, successful detection of cars, bikes, and trucks is evident, with their speeds accurately estimated. However, Figure 8 highlights a challenge, particularly with bike detection, where a bike is going undetected. This observation can be attributed to various factors inherent in real-world scenarios that introduce complexities and variations in object appearances. Real-world objects, such as bikes, may exhibit significant variations in lighting conditions, scale, rotation, occlusion, and pose. Haar cascades, being pattern-based classifiers, may face difficulties in detecting objects that deviate substantially from the trained examples, leading to instances of false negatives or missed detection. In the case of bike detection, when multiple bikes appear in a frame, the challenges are exacerbated, and some bikes may go undetected due to the limited adaptability of the Haar cascade model to diverse bike appearances. To address this, future improvements could involve augmenting the training dataset with a more diverse set of bike examples, fine-tuning the Haar cascade model parameters to better accommodate variations,

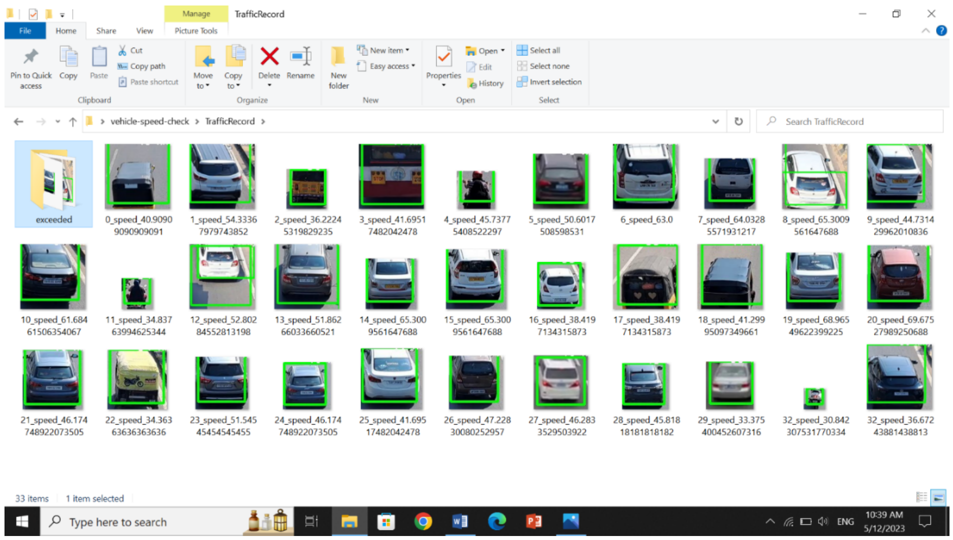

Fig 5. Images of vehicles tracked and saved with vehicle ID and speed

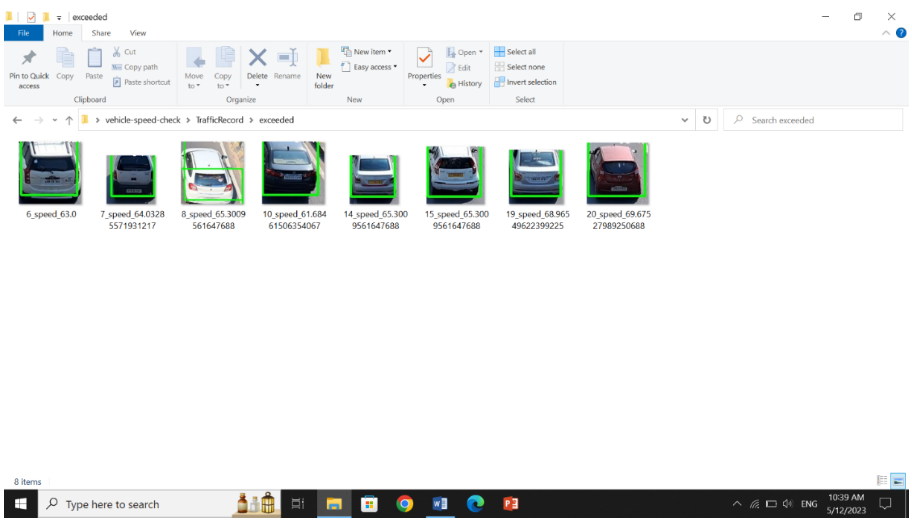

Fig 6. Over-speeding vehicles saved in folder named 'exceeded'

and exploring post-processing techniques to enhance detection robustness in the presence of multiple bikes and challenging environmental conditions. These considerations align with ongoing efforts to enhance the accuracy and reliability of object detection systems in dynamic real-world settings.

It is to be noted as they have complex shapes, textures, or appearance patterns that are challenging to capture accurately with Haar cascades. Another factor to consider is that the performance of Haar cascade detection can be influenced by various parameters, such as the scale factor, minimum object size, and minimum neighbor count. Suboptimal parameter settings cause certain objects to be missed or result in false detection, which can be seen in the case of detection of bikes specifically.

- **Accuracy estimation**

In order to evaluate the proposed method more accurately, we drove through the ROI with a GPS-equipped car and compared the detected speeds with the ones obtained via the GPS[27,28]. Several runs were made at different constant speeds, and the motion data extracted from the vehicle included the time, location, and speed, see Figure 8. The speeds recorded by the navigator device were deemed as the accurate measurements since the error range of a GPS used in similar atmospheric conditions is extremely minimal.

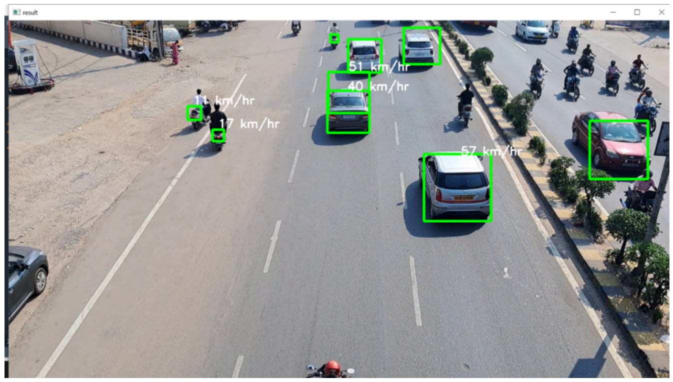

Fig 7. Bike detected

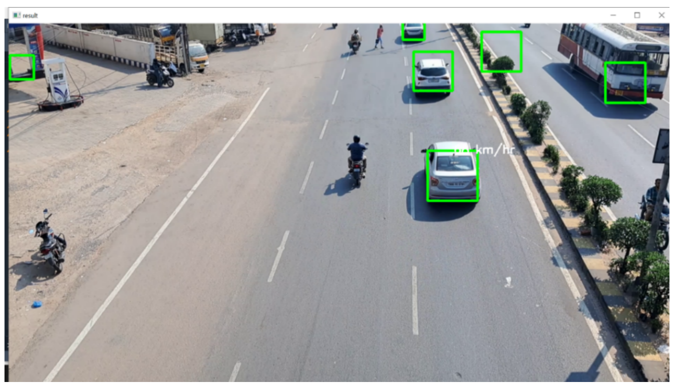

Fig 8. B ike undetected

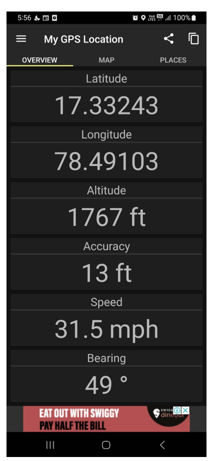

Fig 9. Screenshot of the mobile GPS

3.1 Ground truth

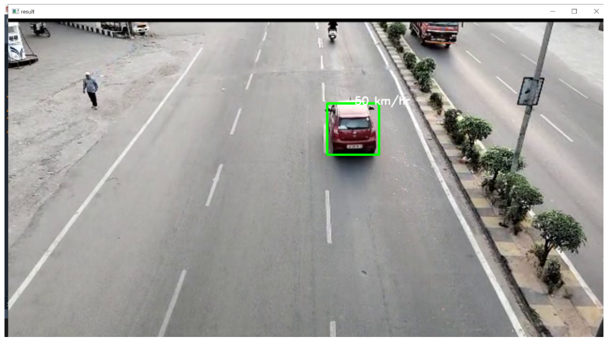

Fig 10. Test video result

In the above Figure 9 the screenshot depicts the exact location (longitude, longitude) and the speed of the vehicle in Figure 10 which is taken from the test video. The speed after convening from miles per hour to km/h is 49.88966 km/h. Which means this value is the actual value and the speed value around the bounding box in Figure 10 is the predicted value.

The algorithm accuracy is evaluated by comparing to the ground truth estimate of the individual vehicle speed.

Table 1. Error measurements of estimated ground truth and result

Test Cases	Estimated value	Ground truth	Error (km/h)
Test_video_1	50	49.88966	0.11034
Test_video_2	43	41.348	1.652
Test_video_3	52	45.345	2.411
Test_video_4	51	50.794	0.26
Test_video_5	47	47.1537	-0.1537

Table 1 presents the error measurements of the estimated ground truth and the results obtained from the algorithm. The error is calculated as the difference between the estimated and ground truth values. To further assess the algorithm's performance, we employed additional evaluation metrics: The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The RMSE, a measure of the average magnitude of the errors, is calculated to be 3.07 km/h, which indicates that the algorithm's predictions deviate from the actual speeds by approximately 3.07 km/h. While the MAE, representing the average absolute errors, is 1.76 km/h, which suggests that, on average, the algorithm's predictions have an absolute deviation of 1.76 km/h from the ground truth speeds. These metrics provide insights into the accuracy and precision of the algorithm's predictions.

The results in Table 1, along with the RMSE and MAE values, collectively indicate the algorithm's performance in predicting vehicle speeds. Despite achieving an average error rate of 1.72 km/h (2.07%), the RMSE and MAE offer a more detailed understanding of the algorithm's predictive capabilities. It's important to consider these metrics in conjunction with the earlier reported Mean Absolute Error (MAE) of 1.72 km/h, providing a comprehensive evaluation of the algorithm's accuracy and precision.

Table 2. Quantitative comparison results

Title/Method	Detection techniques	Tracking/speed measurement	Error Rate
Tracking and Evaluation of the Speed of Portable Vehicles based on video processing using Python [29]	Haar cascade	Centroid tracking	3km/h
Vehicle Speed Detection System using Motion Vector Interpolation [30]	Motion Vector (MV) interpolation	NA	3%
Motion-Vector Clustering for Traffic Speed Detection from UAV Video [31]	Interest point tracking	Velocity clustering process	11.69%
Estimation of Average Car Speed Using the Haar-Like Feature and Correlation Tracker Method [32]	Haar cascade	Correlation tracking	7.7km/h

Continued on next page

Table 2 continued

Vehicle speed measurement model for video-based systems [12]	Intrusion line technique	Monocular-based distance estimation	2.17% (at 30fps)
Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation [9]	Intrusion line technique	Kalman Filter	13%
Proposed Methodology	Haar cascade	Correlation tracking	2.07%

In Table 2 a comparative analysis is presented, where we have summarized techniques for traffic speed detection and vehicle tracking from video data. Each approach employs distinct methods, revealing varying error rates. Our methodology, utilizes Haar cascade detection and Correlation tracking, achieving a low error rate of 2.07%.

4 Conclusion

In conclusion, the study introduces a novel system for vehicle speed detection by integrating the Haar Cascade Classifier, correlation tracker, and frame differencing method. The inclusion of an error rate calculation provides a quantitative assessment of algorithm accuracy, with evaluation results showing an average error rate of 1.72 km/h, demonstrating the effectiveness of the proposed method.

Nevertheless, the methodology has inherent limitations. The dataset is predominantly composed of cars, lacking diversity in vehicle types such as trucks and buses due to resource constraints. Additionally, data collection is restricted to a specific bridge location, limiting the dataset's applicability to a broader range of traffic conditions.

Despite these constraints, the research contributes a cost-effective solution for vehicle tracking and speed estimation, with potential applications in road safety and traffic management. Addressing limitations by diversifying the dataset and expanding the geographical scope could enhance the generalizability of the findings, fostering advancements in the field of vehicle speed detection and promoting innovative methodologies for improved road safety.

5 Acknowledgement

A collective work done which emphasize the potential and effectiveness of the proposed system as a cost-effective solution for vehicle tracking and speed estimation. By leveraging video data and incorporating innovative methodologies, this system offers a valuable tool for traffic monitoring and management, contributing to enhanced road safety and efficient traffic control. This valuable information can be utilized by researchers to gain a deeper understanding of the method and potentially integrate it into their own work, contributing to the advancement of knowledge in the field.

References

1) Hossain M, Hossain MA, Sunny FA. A UAV-Based Traffic Monitoring System for Smart Cities. In: 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE. 2020;p. 1–6. Available from: https://doi.org/10.1109/STI47673.2019.9068088.
2) Lu S, Wang Y, Song H. A high accurate vehicle speed estimation method. *Soft Computing*. 2020;24(2):1283–1291. Available from: https://doi.org/10.1007/s00500-019-03965-w.
3) Zhang T, Jin PJ. Roadside LiDAR Vehicle Detection and Tracking Using Range and Intensity Background Subtraction. *Journal of Advanced Transportation*. 2022;2022:1–14. Available from: https://doi.org/10.1155/2022/2771085.
4) Zhang B, Zhang J. A Traffic Surveillance System for Obtaining Comprehensive Information of the Passing Vehicles Based on Instance Segmentation. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(11):7040–7055. Available from: https://doi.org/10.1109/TITS.2020.3001154.
5) Ali S, Hanzla M, Rafique AA. Vehicle Detection and Tracking from UAV Imagery via Cascade Classifier. In: 2022 24th International Multitopic Conference (INMIC). IEEE. 2022;p. 1–6. Available from: https://doi.org/10.1109/INMIC56986.2022.9972959.
6) Rakesh S, Chari MV, Manohar KB, Goud YVNA. Speed checker and reporting system. *International journal of health sciences*. 2022;6(S4):5189–5201. Available from: https://doi.org/10.53730/ijhs.v6nS4.9315.
7) Yang Y, Xing W, Zhang S, Yu Q, Guo X, Guo M. A Learning Frequency-Aware Feature Siamese Network for Real-Time Visual Tracking. *Electronics*. 2020;9(5):1–13. Available from: https://doi.org/10.3390/electronics9050854.
8) Trivedi JD, Devi MS, Dave DH. A Vision-Based Real-Time Adaptive Traffic Light Control System Using Vehicular Density Value and Statistical Block Matching Approach. *Transport and Telecommunication Journal*. 2021;22(1):87–97. Available from: https://doi.org/10.2478/ttj-2021-0007.
9) Trivedi JD, Mandalapu SD, Dave DH. Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation. *Journal of Industrial Information Integration*. 2022;27:100280. Available from: https://doi.org/10.1016/j.jii.2021.100280.
10) Chen Z, Guo H, Yang J, Jiao H, Feng Z, Chen L, et al. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement*. 2022;201:111655. Available from: https://doi.org/10.1016/j.measurement.2022.111655.
11) Sharma K. Survey on Various Techniques for Over-Speed Detection of Vehicles. *International Journal for Research in Applied Science and Engineering Technology*. 2021;9(5):1633–1640. Available from: https://www.ijraset.com/fileserve.php?FID=34578.
12) Javadi S, Dahl M, Pettersson MI. Vehicle speed measurement model for video-based systems. *Computers & Electrical Engineering*. 2019;76:238–248. Available from: https://doi.org/10.1016/j.compeleceng.2019.04.001.

13) Bozcan I, Kayacan E. AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May 2020 - 31 August 2020, Paris, France. IEEE. 2020;p. 8504–8510. Available from: https://doi.org/10.1109/ICRA40945.2020.9196845.
14) Majumder M, Wilmot C. Automated Vehicle Counting from Pre-Recorded Video Using You Only Look Once (YOLO) Object Detection Model. *Journal of Imaging*. 2023;9(7):1–19. Available from: https://doi.org/10.3390/jimaging9070131.
15) Mahalakshmi PD, Babu M. Vehicle Speed Estimation using Haar Classifier Algorithm. *Published in International Journal of Trend in Scientfific Research and Development (ijtsrd)*. 2019;4(1):243–246. Available from: https://www.ijtsrd.com/papers/ijtsrd29482.pdf.
16) Kamoji S, Koshti D, Dmonte A, George SJ, Pereira CS. Image Processing based Vehicle Identification and Speed Measurement. In: 2020 International Conference on Inventive Computation Technologies (ICICT). IEEE. 2020;p. 523–527. Available from: https://doi.org/10.1109/ICICT48043.2020.9112419.
17) Kanavi P, Chaithra KB, Chaitra KT, Bhoomika MG, Lakshmi CT. Automatic Vehicle Over Speed Detection Alert and Controlling System on Highway. *International Journal of Engineering Research & Technology*. 2022;10(11):602–604. Available from: https://www.ijert.org/research/automatic-vehicle-over-speed-detection-alert-and-controlling-system-on-highway-IJERTCONV10IS11140.pdf.
18) Xie M, Su Q, Wang L, Zhang Z, Ding B, Xu X, et al. The research of traffic cones detection based on Haar-like features and adaboost classification. In: 9th International Symposium on Test Automation & Instrumentation (ISTAI 2022). Institution of Engineering and Technology. 2023. Available from: https://doi.org/10.1049/icp.2022.3263.
19) Gharge S, Patil A, Patel S, Shetty V, Mundhada N. Real-time Object Detection using Haar Cascade Classifier for Robot Cars. In: 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE. 2023;p. 64–70. Available from: https://doi.org/10.1109/ICESC57686.2023.10193401.
20) Dandashy T, Hassan M, El, Bitar A. Enhanced Face Detection Based on Haar-Like and MB-LBP Features. *International Journal of Engineering and Management Research*. 2019;9(4):117–124. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3479590.
21) Krishna DS, Vadamodula P, Jayasri K. Vehicle Speed Estimation Based On Videos Using Deep Learning Techniques. *The Seybold Report*. 2022;17(9):1320–1333. Available from: https://doi.org/10.5281/zenodo.7107001.
22) Gupta U, Kumar U, Kumar S, Shariq M, Kumar R. Vehicle speed detection system in highway. *International Research Journal of Modernization in Engineering Technology and Science*. 2022;4(5):406–411. Available from: https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/22249/final/fin_irjmets1651835089.pdf.
23) Neamah SB, Karim AA. Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8. *Aro-The Scientific Journal of Koya University*. 2023;11(2):137–150. Available from: https://aro.koyauniversity.org/index.php/aro/article/view/1327.
24) Zhang J, Xiao W, Coifman B, Mills JP. Vehicle Tracking and Speed Estimation From Roadside Lidar. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2020;13:5597–5608. Available from: https://doi.org/10.1109/JSTARS.2020.3024921.
25) Yu Y, Zhao J, Gong Q, Huang C, Zheng G, Ma J. Real-Time Underwater Maritime Object Detection in Side-Scan Sonar Images Based on Transformer-YOLOv5. *Remote Sensing*. 2021;13(18):1–28. Available from: https://doi.org/10.3390/rs13183555.
26) Hsu CM, Chen JY. Around View Monitoring-Based Vacant Parking Space Detection and Analysis. *Applied Sciences*. 2019;9(16):1–30. Available from: https://doi.org/10.3390/app9163403.
27) Budhkar AK, Asaithambi G, Maurya AK, Arkatkar SS. Emerging Traffic Data Collection Practices Under Mixed Traffic Conditions: Challenges and Solutions. In: Transportation Research in India . Springer Transactions in Civil and Environmental Engineering;Singapore. Springer. 2022;p. 101–129. Available from: https://doi.org/10.1007/978-981-16-9636-7_6.
28) Wu J, Zhang Y, Xu H. A novel skateboarder-related near-crash identification method with roadside LiDAR data. *Accident Analysis & Prevention*. 2020;137:105438. Available from: https://doi.org/10.1016/j.aap.2020.105438.
29) Ravi J, Pavan V, Sri VY, Srinu V, Kumar YG. Tracking and Evaluation of the Speed of Portable Vehicles Based on Video Processing using Python. *International Journal of Progressive Rese/arch in Engineering Management and Science (IJPREMS)*. 2023;3(3):7–13. Available from: https://www.ijprems.com/uploadedfiles/paper/issue_3_march_2023/30675/final/fin_ijprems1677758281.pdf.
30) Julina JKJ, Sharmila TS, Gladwin SJ. Vehicle Speed Detection System using Motion Vector Interpolation. In: 2019 Global Conference for Advancement in Technology (GCAT), 18-20 October 2019, Bangalore, India. IEEE. 2020. Available from: https://doi.org/10.1109/GCAT47503.2019.8978375.
31) Ke R, Kim S, Li Z, Wang Y. Motion-vector clustering for traffic speed detection from UAV video. In: 2015 IEEE First International Smart Cities Conference (ISC2). IEEE. 2015;p. 1–5. Available from: https://doi.org/10.1109/ISC2.2015.7366230.
32) Fauzi MD, Putra AE, Wahyono W. Estimation of Average Car Speed Using the Haar-Like Feature and Correlation Tracker Method. *Indonesian Journal of Computing Cybernetics Systems*. 2020;14(4):353–364. Available from: https://doi.org/10.22146/ijccs.57262.