Vision Based Vehicle Speed Estimation

Shripad Bhatlawande[1], Sanmit Patil[2], Pushkar Pawar[3], Vipul Pisal[4]
[1,2,3,4]Dept of Electronics & Telecommunication Engineering
Vishwakarma Institute of Technology Pune

[1]shripad.bhatlawande@vit.edu, [2]sanmit.patil19@vit.edu, [3]pushkar.pawar19@vit.edu, [4]vipul.pisal19@vit.edu

Abstract - *This paper presents a system to measure the speed of vehicles on roads. Traffic law violations are controlled by detecting over speeding vehicles. Vehicles in a video are detected, tracked and their speed is estimated. Speed estimation is performed based on camera input without any extra sensors making it cost effective. Fast Retina Key-point (FREAK) and Features from Accelerated Segment Test (FAST) algorithms are used for feature extraction. FAST and FREAK provide results rapidly and are helpful in real time applications. Time calculation for speed estimation is independent of the processor being used. Voting based classifiers are used for detecting vehicles. Seven different classifiers are used in it. Random Forest provided the highest accuracy of 88.4% and F1 score of 88.5%. Proposed system provided speed estimation with an approximate error of 2 km/hr. The average percentage error obtained for the estimated speed is 9.22%.*

Keywords : *Driver Assistance System, Machine Learning, Computer Vision, Speed Estimation, Vehicle Detection.*

1. Introduction

Advanced driver-assistance systems (ADAS) are a necessary safety feature in today's vehicles [1]. Statistics of accidental deaths have shown an upward trend in India [2]. The National Crime Records Bureau reported 60% of all road accidents occurred due to over speeding. ADAS can be used for speed estimation and then it can be used to avoid over speeding. Speed estimation can be an important part of traffic safety systems [3]. Slow traffic speed can be detected and traffic jams can be avoided using speed estimation.

2. Literature Survey

LiDAR sensors, RS-LiDAR-32 and Velodyne-VLP-16 are used for scanning the environment [4]. Kalman filter is implemented for vehicle tracking. Lidar sensors being used presently increase the cost of the system. Deep learning techniques such as YOLOv2 and SSD are used in [5]-[6] for vehicle and license plate detection. Perceptive transformation is implemented by calculating the calibration matrix. In [7] vehicle detection is done using Faster R-CNN and no tracking is performed. SORT tracker is implemented along with Faster R-CNN detector in [8] on images from surveillance cameras. Faster-SCNN model is used for vehicle detection [9]. Requirement of significant amounts of data and time are the limitations of using deep learning models. Speed estimation is based on license plate detection [10] where its centre is considered as the reference. Observation of the number plate is carried out with the help of inductor loop detectors [11]. Speed estimation using inductive loops requires high maintenance and the data received regarding traffic is inadequate.

HOG is implemented in [12] along with several versions of SVM classifiers for vehicle detection. The solution is found to be ineffective on roads with heavy traffic. Background removal techniques, especially the Gaussian mixture model [13] along with the Maximum likelihood method, are applied for speed estimation. DBSCAN clustering is implemented along with the Kalman filter for motion detection [14]. Morphological processes are used for monitoring vehicle centroid and speed [15]. The proposed solutions implemented with Gaussian filtering need improvement for low light situations. Haar feature extractor is used on frontal vehicle images [16]. The Kalman filter and Munkres assignment algorithm are used for motion detection. The proposed methodology is effective only for vehicles' frontal views [16].

IP cameras along with Raspberry Pi 3 boards are used [17]. Absolute differencing for background separation and blob detection for centroid creation is performed. Improvements must be made for night time images. Vehicle detection and tracking is implemented using the three-frame difference method [18]. Vehicle speed estimation in [19] uses Pearson's correlation coefficient analysis. Five kinds of speed violations classes based on the magnitude of the speed violation rates [20]. Three-stage method is used for calibrating roadside cameras and tracking vehicles [21]. The algorithm determines the camera location relative to the highway. Distance between cars is calculated using vision-based metrics of headlights and taillights [22].

3. Methodology

The proposed system detects and tracks vehicles. Speed estimation is performed on the detected vehicles. A camera provides visual input of the surroundings. The processor-based system returns the speed of the vehicle in the form of text displayed on the screen.

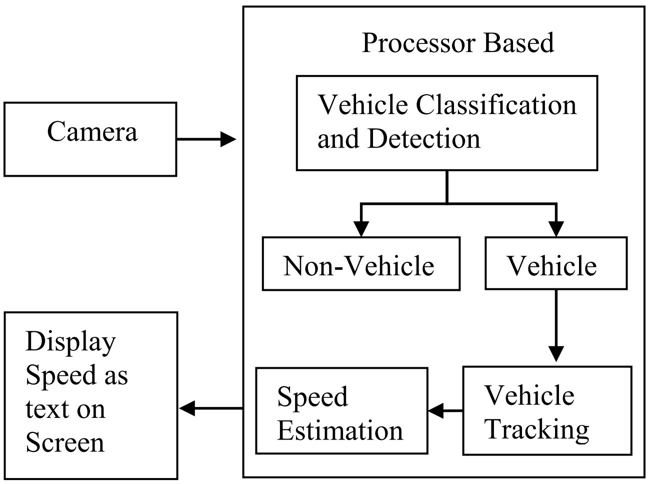

Fig. 1. Block diagram of the system for vehicle detection, tracking and speed estimation.

3.2 Data Description

A total of 6400 images are present in the compiled dataset for vehicle classification and detection. Out of them, 4800 images are used for training and 1600 unseen images are used for testing. Dataset images are obtained from the internet. Dataset images are divided into two classes, positive and negative. Positive class consists of images of cars. Negative class consists of non-vehicle images of empty roads, road dividers, people, trees. The overall distribution of images in the dataset is presented in Table 1.

Table 1. Dataset Distribution

Sr. No.	Class	Training/Testing	Number of Images
1	Positive (Cars)	Training	2400
2	Negative (Non-Vehicles)	Training	2400
3	Positive (Cars)	Testing	800
4	Negative (Non-Vehicles)	Testing	800
Total Images			6400

Fig. 2. Sample positive and negative images in the dataset

Images from positive class are cropped to ensure the car occupies a major part of the image. All images are then resized to 198x120 pixels. Convolution is performed using this dataset with a low pass average filter, followed by a high pass filter with kernel size of 3x3. The low pass average filter blurs the image to remove any existing noise. High pass filter enhances the edges and removes most of the background information. Power law transformation is applied with gamma value equal to 0.5, to brighten the image.

3.3 Classification and Detection of Vehicle

FAST and FREAK are used to extract features. FAST feature detector is a corner detection algorithm, and it detects key points for an image. Faster speed is obtained with it compared to Scale Invariant Feature Transform, Susan, and Harris detectors. High computational efficiency of the FAST detector is useful in real-time applications. Every surrounding pixel's intensity in the conventional four directions is compared with the selected pixel's intensity as shown in eq. (1). Three or more pixels must all be brighter or darker than the selected pixel for it to be considered as an interesting pixel. Same procedure is carried out for every pixel in the image.

$$S_{p\to x} = \begin{cases} darker, & I_{p\to x} \leq I_p - t \\ same, & I_p - t \leq I_{p\to x} < I_p + t \\ brighter, & I_p + t \leq I_{p\to x} \end{cases} \tag{1}$$

Where I_p is the intensity of the assumed interest pixel and $I_{p\to x}$ is the intensity of the surrounding pixel. Threshold set is t and $S_{p\to x}$ is the state of the surrounding pixel.

FREAK descriptor is used on the detected key points. It is inspired by the way the human retina works. The computational time and memory load requirement for it is less compared to other descriptors. Feature vectors generated by FREAK consist of 64 columns. Image intensities are calculated over retinal samplings. Binary string cascade string is created as shown in eq. (2).

$$B = \sum_{0 \le a < N} 2^a T(P_a) \qquad (2)$$

Where, B is the binary string formed and P_a is a pair of receptive fields. $T(P_a)$ is one bit difference of intensities of receptive fields and N is the descriptor size.

Feature vectors extracted using FAST and FREAK have a size of 3.5 million x 64. Feature scaling is performed on the computed feature vectors. Histogram values for images are divided by the number of its key points to normalize it. Standardization of the values is then performed. Dimension reduction is implemented to reduce the size of feature vectors yet maintain the amount of variation. A blend of K-Means and Principal Component Analysis (PCA) is used for dimension reduction. K-Means clustering is performed to divide the feature vectors into different clusters. The number of clusters is chosen as 19 using the elbow method. PCA merges the highly correlated features and converts it into a smaller number of linearly uncorrelated features. The number of columns is reduced from 19 to 17 using PCA.

The dataset is divided into training and testing data in an 80:20 ratio. Classifiers are trained on the training data. Vehicle classification is implemented using seven classifiers: i) Decision Tree Classifier, ii) Random Forest Classifier, iii) K-Nearest Neighbor (KNN), iv) Support Vector Machine (SVM), v) Logistic Regression, vi) XgBoost and vii) Naïve Bayes. Random Forest classifier provided the highest accuracy of 88.4% amongst all mentioned classifiers. Precision of 91.5%, recall of 85.7% and F1 score of 88.5% is also obtained. Classifiers are combined to form a voting-based classifier. It takes predictions from each classifier and provides a single output based on a combination of individual predictions. Soft voting-based classification is performed to predict the probabilities of each class. The class with the highest sum of probabilities is decided as the output.

Classifiers are able to detect whether an image is of a vehicle or not. In particular video frame, to detect coordinates of a vehicle the sliding window approach is implemented. A window of 80x80 pixels is created and it slides all over the frames from the video. Predictions are made for each window. An all-black image of the same size as the original frame is created. The pixel values of the window are enhanced in the all-black image if prediction is positive. The pixel values are reduced in the all-black image if prediction is negative. Contour detection is implemented on the modified all-black image to get the coordinates of the vehicles.

3.3 Vehicle tracking

The Vehicles are tracked using the dlib library. The detected vehicles are tracked by assigning them a unique ID. A correlation tracker object is created for every vehicle detected. It locates the position of the vehicle in the upcoming frames by taking the coordinates of the detected vehicle in the initial frames. A correlation function is used for this purpose. It finds the most similar window in the next frame of the video compared to the current vehicle's bounding box. The window with the highest similarity is chosen as the bounding box for the current vehicle in the next frame. The normalized correlation function (C) is as shown in eq. (3). Its value lies between 0 and 1.

$$C_{ij} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{M-1} B(x,y)F_{ij}(x,y)}{\left[\sum_{x=0}^{M-1}\sum_{y=0}^{M-1} B^2(x,y)F_{ij}^{\,2}(x,y)\right]^{\frac{1}{2}}},\ 0 \le i,j \le N-M \qquad (3)$$

Where, B is the image of MxM size present inside the bounding box and F of size NxN is the next frame of the video. Pixel locations in the image are represented as x, y.

3.4 Speed estimation

Dependent Speed is estimated based on time required to cover the distance between two points in the video. Distance between the two_points in the real world is measured in meters. The frame number of the video when the vehicle crosses the starting point and ending point is stored. Number of frames required is calculated as the difference between them. It is divided by frames per second (fps) of video to calculate time as shown in eq. (4).

$$Time = (F_e - F_s)/fps \qquad (4)$$

Where, Fe is the frame number where the vehicle crosses the finish line. Fs is the frame number where the vehicle crosses the start line. Videos of 30 fps are used. The speed obtained is in meters per second (m/s) unit. Speed in kilometers per hour (km/hr) is calculated as shown in eq. (5).

$$Speed = (Distance/Time) \times 3.6 \qquad (5)$$

Algorithm 1 explains the process of vehicle detection, tracking and speed estimation in a video

Algorithm 1: Vehicle Detection, Tracking and Speed Estimation

Input: Video (720x480 pixels, 30 fps)
Output: Estimated Speed in text format

```
Read video (V)
For each frame F in V do
    B = all-black image
    Windows = SlidingWindow(F)
    For each window W in Windows do
        pred = Prediction(W)
        If pred=0 then
            B = Increase_pixel(W, B)
        If pred=1 then:
```

```
                    B = Decrease_pixel(W, B)
TABLE 1
          End For
          Coord = FindContours(B)
          Return Coord
     End For
     (x,y,w,h) = Track_vehicle(F, Coord)
     If y > starting point then
          Fs = frame number
     If y > finishing point then
          Fe = frame number
     Time = (Fe-Fs)/fps
     Speed = (Distance/Time) * 3.6
Return Speed
```

4. Results and Discussion

Vehicle classification is implemented using seven different classifiers. Performance evaluation metrics on training data are presented in Table 2.

Initial predictions exhibited a high number of false negatives for the voting-based classifier. Recall of 0.85 is obtained. The threshold confidence for negative predictions is increased to reduce the number of false negatives. The threshold is increased from 0.50 to 0.55 for negative predictions. The number of false positives is now reduced. The recall obtained now increased to 0.88. A fine balance is now obtained between positive and negative predictions. The voting-based classifier model is saved and used to make predictions on the 1600 unseen images of the testing dataset. The results obtained on the testing dataset are presented in Table 3. An accuracy and F1 score of 0.88 is obtained with the voting-based classifier.

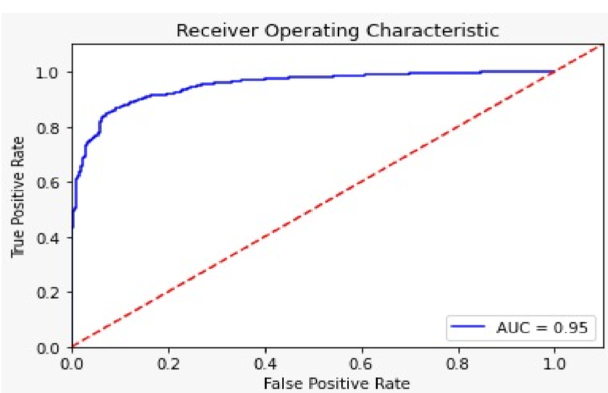

Fig. 3. Receiver Operating Characteristic curve for voting based classifier based on training data

Table 2. Performance evaluation metrics on training dataset

Models	Performance Evaluation Metrics				
	Accuracy		Precision	Recall	F-1 Score
	Train	Test			
Decision Tree	0.987	0.816	0.836	0.803	0.819
Random Forest	1.0	0.884	0.915	0.857	0.885
K Nearest Neigbours	0.899	0.884	0.901	0.873	0.887
SVM RBF Kernel	0.919	0.878	0.892	0.871	0.875
XGBoost	1.0	0.879	0.889	0.865	0.877
Logistic Regression	0.885	0.876	0.871	0.881	0.876
Naive Bayes	0.867	0.880	0.898	0.857	0.877
Voting Based Classifier	0.971	0.880	0.896	0.854	0.874

Table 3. Performance evaluation metrics on training dataset

Models	Performance Evaluation Metrics			
	Accuracy	Precision	Recall	F-1 Score
Voting Based Classifier	0.883	0.883	0.884	0.884

Tests are conducted on a distance of nine meters for speed estimation. Frame number of the video is stored when the vehicle crosses the starting and ending point. The video footage used is of 30 fps. The tests are conducted for a speed range of 10-30 km/hr. The comparison between actual and estimated speeds is presented in Table 4.

Table 4. Speed estimation evaluation

Sr. No.	Actual Speed S (km/hr)	Estimated Speed S' (km/hr)	Percentage Error |S-S'|/S (%)
1.	13	15	15.38
2.	15	18	20.0
3.	15	17	13.33
4.	16	18	12.5
5.	18	19	5.55
6.	20	20	0
7.	20	22	10.00
8.	20	23	15.00
9.	23	25	8.69
10.	24	27	12.50
11.	25	26	4.00
12.	25	26	4.00
13.	28	30	7.14
14.	28	29	3.57
15.	30	32	6.66
16.	Average Percentage Error		9.22

5. Conclusion

Over speeding leads to accidents and traffic law violations. There is a need for creation and implementation of systems to estimate vehicle speeds accurately. The presented system uses computer vision and machine learning to detect, track and estimate vehicle speed. It requires less computing power compared to deep learning approaches. A monocular camera being the only requirement makes it cost effective as well. Furthermore, the simplicity of the methodology leads to obtaining results quickly. Experimental results, confirm the measurement error for speed is within +2/-3 kilometers per hour. The proposed system works more efficiently on cars. In the future, more vehicle classes will be added to improve the systems performance.

References

[1] V. K. Kukkala, J. Tunnell, S. Pasricha and T. Bradley: Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles. In: IEEE Consumer Electronics Magazine, vol. 7, no. 5, pp. 18-25 (2018).

[2] S. Gothane and M. V. Sarode: Analyzing Factors, Construction of Dataset, Estimating Importance of Factor, and Generation of Association Rules for Indian Road Accident. In: IEEE 6th International Conference on Advanced Computing, pp. 15-18 (2016).

[3] Wicaksono, Danang Wahyu, and Budi Setiyono: Speed estimation on moving vehicle based on digital image processing. In: International Journal of Computing Science and Applied Mathematics, pp. 21 (2017).

[4] J. Zhang, W. Xiao, B. Coifman and J. P. Mills: Vehicle Tracking and Speed Estimation From Roadside Lidar. In: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 5597-5608 (2020).

[5] Thakurdesai, Hrishikesh and Aghav, Jagannath: Computer Vision Based Position and Speed Estimation for Accident Avoidance in Driverless Cars. In: ICT Systems and Sustainability, AISC, volume 1077, pp. 435-443 (2020).

[6] W. -P. Wu, Y. -C. Wu, C. -C. Hsu, J. -S. Leu and J. -T. Wang: Design and Implementation of Vehicle Speed Estimation Using Road Marking-based Perspective Transformation. In: IEEE 93rd Vehicular Technology Conference, pp. 1-5 (2021).

[7] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds and S. Atkinson: A Vision-Based Pipeline for Vehicle Counting, Speed Estimation, and Classification. In: IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 12, pp. 7547-7560 (2021).

[8] Grents, Alexander Varkentin, Vitalii and Goryaev, Nikolay: Determining vehicle speed based on video using convolutional neural network. In: Transportation Research Procedia. pp. 192-200 (2020).

[9] H. Kang and J. Lee: A Vision-based Forward Driving Vehicle Velocity Estimation Algorithm for Autonomous Vehicles. In: International Conference on Advanced Intelligent Mechatronics, pp. 492-497 (2021).

[10] M. Famouri, Z. Azimifar and A. Wong: A Novel Motion Plane-Based Approach to Vehicle Speed Estimation. In: IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 4, pp. 1237-1246 (2019).

[11] D. C. Luvizon, B. T. Nassu and R. Minetto: A Video-Based System for Vehicle Speed Measurement in Urban Roadways. In: IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 6, pp. 1393-1404 (2017).

[12] J. J. P. Belen, J. C. V. Caysido, A. B. Llena, E. J. O. Samonte, G. N. Vicente and E. A. Roxas: Vision based classification and speed estimation of vehicles using forward camera. In: IEEE 14th International Colloquium on Signal Processing and Its Applications, pp. 227-232 (2018).

[13] V. Mattioli, D. Alinovi and R. Raheli: A Maximum Likelihood Approach to Speed Estimation of Foreground Objects in Video Signals. In: 28th European Signal Processing Conference, pp. 715-719 (2021).

[14] J. Gerát, D. Sopiak, M. Oravec and J. Pavlovicová: Vehicle speed detection from camera stream using image processing methods," International Symposium, pp. 201-204 (2017).

[15] Gunawan, Alexander and Tanjung, Deasy and Gunawan, Fergyanto: Detection of Vehicle Position and Speed using Camera Calibration and Image Projection Methods. In: Procedia Computer Science, pp. 255-256 (2019).

[16] M. Jalalat, M. Nejati and A. Majidi: Vehicle detection and speed estimation using cascade classifier and sub-pixel stereo matching. In: 2nd International Conference of Signal Processing and Intelligent Systems, pp. 1-5 (2016).

[17] R. M. Waregaonkar, S. M. Deokar, S. S. Patankar and J. V. Kulkarni: Development of prototype for vehicle speed measurement. In: 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, pp. 945-949 (2017).

[18] J. Wang: Research of vehicle speed detection algorithm in video surveillance. In: International Conference on Audio, Language and Image Processing, pp. 349-352 (2016).

[19] K. -S. Shim, N. i. Park, J. -H. Kim, O. -Y. Jeon and H. Lee: Vehicle Speed Measurement Methodology Robust to Playback Speed-Manipulated Video File. In: IEEE Access, vol. 9, pp. 132862-132874 (2021).

[20] Kuşkapan, Emre and Çodur, Muhammed and Atalay, Ahmet: Speed violation analysis of heavy vehicles on highways using spatial analysis and machine learning algorithms" Accident Analysis and Prevention. pp. 155 (2021).

[21] Javadi, Saleh & Dahl, Mattias & Pettersson, Mats: Vehicle speed measurement model for video-based systems. In: Computers & Electrical Engineering. pp. 76 (2019).

[22] Shiranthika, C., et al.: Realtime Computer Vision-Based Accurate Vehicle Counting and Speed Estimation for Highways. In: International Conference on Intelligent Computing. Springer, Cham, pp. 583-592 (2019).