Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020)
IEEE Xplore Part Number:CFP20F70-ART; ISBN:978-1-7281-4685-0

Image Processing based Vehicle Identification and Speed Measurement

Supriya Kamoji, Dipali Koshti, Alphaeus Dmonte, Solomon Jose George, Clayton Sohan Pereira
Department of Computer Engineering, University of Mumbai
Fr. Conceicao Rodrigues College of Engineering, Bandra(W), Mumbai- 50, India.

Abstract - **Traffic Collisions are a major source of deaths, injuries and property damage every year. In many of the cases, the reason for the death is over speeding. The rapid recent advancements in the computation ability of everyday computers have made it possible to widely apply deep learning methods to the analysis of traffic surveillance videos. The vehicle speed detection system developed by using video processing techniques. In this, a video is captured through a fixed camera with various parameters noted. The video from the camera is processed frame by frame, in which the 1) Each frame is processed with Image Enhancement to improve the characteristics of the image 2) Vehicle is identified with a bounding box applied to the vehicle, to note the movement in each frame 3) Movement of the vehicle is noted with change in pixel 4) Calculating the speed using distance formula considering PPM.**

Keywords- PPM (Pixel Per Metre), Haar Classifier, Object Detection, Speed Tracking

I. INTRODUCTION

The approach proposed uses image processing techniques to detect the speed of the vehicles from the video footage of the moving vehicle. The video is captured using a camera equipment and relayed back to the computer system where the actual computation takes place. The technique proposed here makes use of computer vision based library functions for speed calculation.

The current speed detection system makes use of radar technology for speed detection of vehicles [1]. Radar technology works on the physics principle of Doppler Effect. The speed detection equipment which is normally a speed gun measures the change in frequency of the returned radar signal. The basic principle stated by Doppler Effect is that the frequency of the returned signal increases if the object's speed increases while moving towards the unit, and decreases if the object is receding [1]. The radar and other LIDAR systems currently in use have certain disadvantages which demands the need of a new system. Radar/LIDAR systems are expensive equipment as their retail cost as well as maintenance costs are significantly higher. This is one of the most important reasons for a new speed detection system which could be cost effectively deployed and maintained.

The technique proposed here achieves cost efficiency because the special hardware components used in traditional approach is now replaced by a more software based approach. Software techniques are applied on the input video footage by the computer setup for detecting and tracking the vehicle in the video footage. Subsequently, the speed of the tracked vehicle is calculated using the classical principle of velocity calculation. The speed result thus obtained is then embedded within the output video. The output result can then be used for black listing the violators of the enforced speed limit in that locality.

The proposed system can be implemented with little hardware requirements. Further modifications to the system can be done in order to incorporate license plate recognition

II. RELATED WORKS

One of the most important steps in measuring the speed of a vehicle is to correctly track it. For the system to be robust, there is a need to make sure the process is efficient, in order to achieve this various techniques have been developed that make use of classical computer vision and machine learning techniques to achieve object tracking. Kate et al. (2015) [2] utilized a classical optical-flow algorithm along with motion vector estimation to solve the object tracking problem. Significant details regarding the object movement can be obtained through optical flow even in the absence of computation of the quantitative parameters. The evaluation of the presence of the object from successive frames can be obtained using motion vector estimation. This aids in obtaining better results irrespective of the anomalies present like image blur or cluttered background, which directly boosts the algorithms accuracy. In order to deal with environments which are not stationary, Geist et al. (2009) [3] contributed a framework based on reinforcement learning along with a Kalman Filter. The paper explains that tracking the desired item within the video footage can be considered to be the problem of estimating the accurate position of the bounding box in individual frames for the item described. Faragher (2012) presented a simple and detailed explanation of Kalman Filters [4]. Kalman filter is derived from first principles considering a simple physical example exploiting a key property of the Gaussian distribution. It involves two stages, prediction and measurement update.

Various research studies have been carried out regarding the detection of speed of the vehicles from the video footage along with different techniques. Rad et al. (2010) [5] describes a machine learning technique for the detection of object which can process the images very quickly and thereby achieving greater detection rates as compared to other techniques. The proposed approach involving the comparison of the vehicle position between the current frame and the previous frame to predict traffic speed from digital video captured with a stationary camera. Geometric equations were used in order to calibrate the camera device. Wang [6] proposed a technique which is based on moving target detection in a sample video by mapping the relation between real distance and distance which is in terms of pixel units. The algorithm made use of three frame differencing and background differencing to obtain features from the vehicles which are moving in the video. Later the vehicle was tracked and positioned using the vehicle centroid feature extraction technique.

III. Proposed System

In this section, the proposed approach to detect the speed of the vehicle is discussed. This part first describes the overall approach of the proposed system and then subsequently gives the detailed information of individual components involved in the system.

3.1 Overall Description of Proposed Approach

The system consists of a computer connected to camera hardware. The camera is to be installed on a public infrastructure such that a proper view of the vehicles can be obtained for speed detection. The video footage captured by the camera is continuously relayed back to the computer. The vehicle tracking and speed detection software running on the computer takes the video footage as the input and provides a video playback of the result. The speed result calculated by the system is embedded in the video playback, which can be monitored.

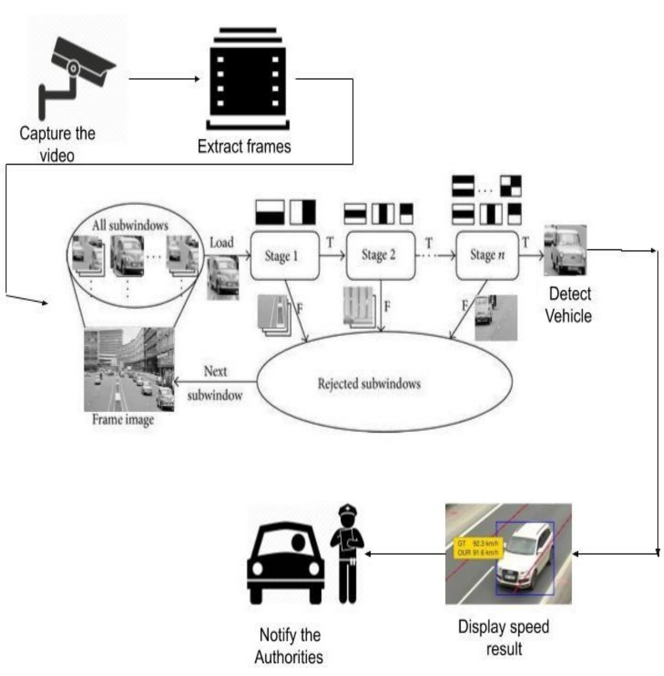

Figure 1: The architecture of the System

3.2 Detailed Description of Proposed Approach

The approach can be described using individual execution phases. In this part, the flow of execution is described with detailed description of individual phases involved.

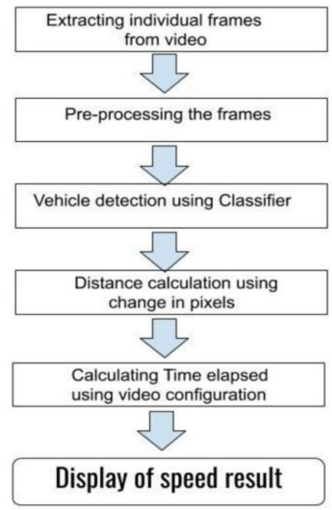

Figure 2: Flowchart depicting the phases involved in the system

3.2.1 Input to the System

The video footage of the vehicles on the road is acquired using an un-calibrated camera. The video acquired is relayed back to the computer system which extracts the individual frames out of the video. The camera's frame rate in terms of fps is already known to the computer. In this approach the video frames obtained are pre-processed using a cross platform and open source library, OpenCV [7]. The frames are resized to the desired dimension using the library function. OpenCV functions are also used to convert the frame into its grayscale equivalent which reduces the complexity required to carry out further processing on the input video frames[8].

3.2.2 Frame Extraction

The vehicles in the frames need to be detected out of the background scene. Given approach uses the Haar Classifier for detecting the vehicles from the frames. It is a feature based cascade classifier which is a method proposed in the paper by Paul Viola and Michael Jones [9]. Haar classifier provides a very significant speed advantage in calculating the features of the images due to its integral images based approach. Adaboost a popular boosting technique is used to improve the performance of the vehicle detection. Several weak classifiers are combined to create a strong classifier which is able to successfully detect the vehicles. The output of the classifier is the coordinates of the rectangle which encloses the detected vehicle. The OpenCV library function is then used to draw a bounding rectangle for the vehicle.

3.2.3 Vehicle Tracking

The proposed approach uses the dlib library for tracking the vehicles which is based on the paper published by Danelljan in 2014 [10]. The video footage may contain one or more vehicles which may be present in the video frame. The correlation tracker function of the dlib library is used which can track multiple vehicles in the video. Vehicle id is assigned to individual vehicles in the footage in order to keep track of the vehicle in subsequent frames of the video. This step is very essential as the speed estimation of a particular vehicle should not be confused with some other vehicles speed estimation.

3.2.4 Speed Detection

The position of the vehicle is obtained from the above phases. This identifies the position of the vehicle in terms of pixel units. Speed detection requires distance calculation as well as the time taken to cover the distance. Initially the change in pixels is calculated using the coordinates obtained. The distance covered is in terms of pixel units which are converted in terms of SI units of distance. A pixel per metre (ppm) value is used to convert the distance in terms of metre units. Ppm value denotes the number of pixels covered in the video for one metre distance in real world scenario. Time elapsed is obtained by the camera's frame rate setting.

Distance calculation using ppm value

dist_pixels = math_sqrt (math_power (position2[0] - position1[0] , 2) + math_power (position2[1] - position1[1] , 2)) ... (1)

dist_metres = dist_pixels / ppm ... (2)

Speed calculation in terms of metres

speed = dist_metres * FPS * 3.6 ... (3)

math_sqrt denotes the mathematical square root
math_power denotes the mathematical power function
ppm denotes the pixels per metre value
FPS denotes the frames per second value

The ppm and fps values are provided to the system before the speed detection is carried out on the input video.

3.2.5 Reporting

The output reporting of the system is in the format of a video playback depicting the detected vehicle in a bounding box. The speed detected is embedded beside the bounding box itself for the vehicle. Multiple vehicles in the video are enclosed using separate bounding boxes with the speed results embedded beside it.

IV. Experimental Results

For the testing of the model, a single car was used. The entire video was first divided in frames. Each frame was then converted to grayscale image, and these frames were then passed through the model. Using the detection algorithm, it was found that the model could detect the car with 95-96% accuracy. For the testing scenario, only one vehicle was used.

Using the ‘dlib’ library, the vehicle was tracked. Here also it was able to track the car correctly throughout the video. The model estimates the vehicle speed, while the vehicle is in a particular region, also called the region of interest. The actual speed of the car at different instances, while in the region of interest was observed. For speed calculation, 10 frames were used. Using the model, the speed of the car was estimated when it was in the region of interest. After comparing the actual speed from the one that was estimated, it was observed that there was a similarity of around 98%.

Figure 3: Speed estimation of a single vehicle

A random traffic surveillance video was then used for testing.. Here, also the vehicle detection accuracy was between 90-95%. The model was also able to accurately track the vehicles, from the moment it entered the scene, till it exits. Also comparing the speed estimate with the data already available, the model was able to estimate the speed with great accuracy.

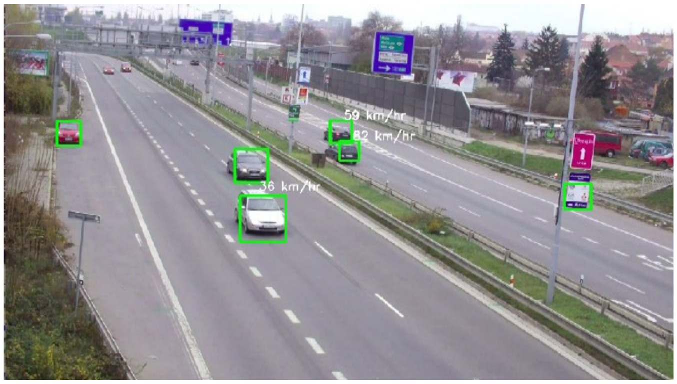

Figure 4: Multiple vehicle speed estimation

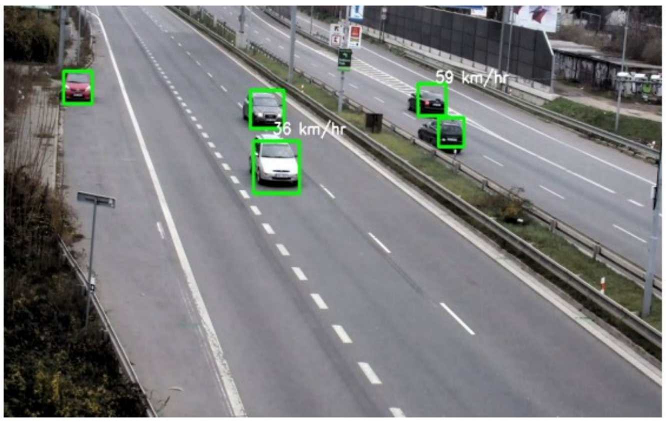

Figure 5: Speed estimation of multiple vehicles

The model was also tested on a video with multiple vehicles in a particular frame at a given time, and as taken from a different angle than other videos. It can be seen that the vehicles are detected throughout the image, but the vehicle speed is estimated only when the vehicle is in the particular region, as seen from the image below.

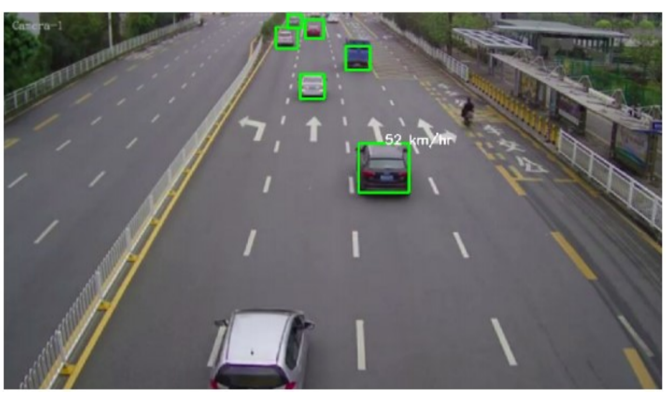

Figure 6: Vehicle speed estimated only when the vehicle is within a particular region.

The vehicle detection algorithm was also tried on the images that were taken at night as well as a few images taken during bad weather. It was observed that the model was able to detect the vehicle in these conditions as well with good accuracy.

The following table includes the number of cars, trucks, etc used for training as well as the number of negative images.

Image Name	**No. of images**
Cars	1645
Autorickshaw	16
Trucks	100
Bus	55
Background(trees,roads,etc)	2001

Table 1: No. of various images used

V. Conclusion

The model was created to achieve three main objectives. The first one is to detect the vehicle. As can be seen from our experimental results, this can be achieved with very good accuracy. The second objective was to track the vehicle, which is also achieved with the help of dlib library, which is a cross-platform library written in C++. The main module used from this library is the dlib correlation tracker. The third objective was speed estimation. From the results that were observed, it can be concluded that the model was able to do this as well, with a very high accuracy.

This model can be very useful for traffic surveillance, a factor that motivated us to create this model. It can be used in different weather conditions, and it still gives a very good accuracy, which is very important for the system to work. Different background conditions have a minimal effect on the working of the model.

The camera should be static and it works well on highways. The model can work on city roads as well . The proposed model also works on different types of vehicles like cars, buses, trucks, etc. which can be very beneficial.

VI. Future Enhancement

To incorporate a system for identifying the vehicle by automatically recognising the number plate. Using this, the speeding vehicles can be flagged. Using the automated system to monitor traffic and identifying rogue drivers.

VII. REFERENCES

[1] Su Myat Paing, Su Su Yi Mon, Hla Myo Tun. Design And Analysis Of Doppler Radar-Based Vehicle Speed Detection. IJSTR, VOLUME 5, ISSUE 06, JUNE 2016.

[2] K. Kale, S. Pawar, and P. Dhulekar. Moving object tracking using optical flow and motion vector estimation. In Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on, pages 1–6. IEEE, 2015.

[3] M. Geist, O. Pietquin, and G. Fricout. Tracking in reinforcement learning. In International Conference on Neural Information Processing, pages 502–511. Springer, 2009.

[4] R. Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. IEEE Signal Processing Magazine, 29(5):128–132, Sept 2012.

[5] A. G. Rad, A. Dehghani, and M. R. Karim. Vehicle speed detection in video image sequences using cvs method. International Journal of Physical Sciences, 5(17):2555–2563, 2010.

[6] J.-x. Wang. Research of vehicle speed detection algorithm in video surveillance. In Audio, Language and Image Processing (ICALIP), 2016 International Conference on, pages 349–352. IEEE, 2016.

[7] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov,Victor Eruhimov. Realtime Computer Vision with OpenCV. Volume 10 Issue 4, April 2012.

[8] S. Sri Harsha, K. R. Anne. A Highly Robust Vehicle Detection, Tracking and Speed Measurement Model for Intelligent Transport Systems. International Journal of Applied Engineering Research. ISSN 0973-4562 Volume 11, Number 5 (2016).

[9] Paul Viola, Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (IJCV). 2001

[10] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate Scale Estimation for Robust Visual Tracking. Proceedings of the British Machine Vision Conference. BMVA Press, September 2014.