Vehicle Speed Estimation using Object Detection for Intelligent Traffic Management

Atharva Hiwarekar, Swaroop Chavhan, Onkar Deshpande, Vedant Joshi

Pune Institute of Computer Technology

Corresponding author: Vedant Joshi, Email: joshivedant84@gmail.com

Effective and safe transport is an essential need of every individual, and it plays a vital role in every aspect of our life. Traffic Management has become a key challenge in today's developing world. Managing growing surveillance in today's world is a challenge to the administration, which can be solved by the innovations in technology. Over-speeding is one of the main factors that cause road accidents, followed by injuries and deaths. Thus there is a need to develop a solution that will detect traffic and calculate vehicle speed in real-time. This paper deals with solving traffic problems using Object Detection. We used YOLOv3 and Deep-SORT algorithm to detect and track vehicles from input surveillance video. And, we calculate speed of the vehicles present in the input surveillance video using the Frame difference method. We have shown that object detection can be used to identify and track vehicles in extreme weather conditions like rain and snow. Further it is shown that the frame difference method and linear perspective transformation can be used to calculate vehicle speed with RMSE as low as 4.568 km/hr.

Keywords: Machine learning, Deep Learning, Computer Vision, Object Detection

2023. In Saroj Hiranwal & Garima Mathur (eds.), *Artificial Intelligence and Communication Technologies*, 677–685. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-955020-5-9-64

1. Introduction

Traffic accidents are a leading cause of deaths, injuries, and damage to the world. It is responsible for the death of approximately 1.3 million people each year. Road accident report of 2019 [14] conducted by the Ministry of Road Transport and Highways of India shows that a total of 449,002 accidents took place in the country during the year 2019, leading to 151,113 deaths and 451,361 injuries. The survey shows that 72.4% of accidents were caused due to overspeeding, 5.5% due to driving on the wrong side of the lane, and 2.3% due to drunk driving. 67.3% of the deaths due to traffic accidents were associated with overspeeding, 6.1% due to driving on the wrong side of the lane, and 3.5% due to drunk driving. This indicates that overspeeding is the primary cause of road accidents in India.

To prevent accidents, many common measures are frequently implemented, including driving under a specific speed limit, avoiding drunk driving, use of helmets by two-wheeler drivers, use of seat belts and child restraints in cars, improving visibility, appropriate headlights, and road lighting, etc. Despite these measures, we witness an increasing number of accidents, most of which are caused due to overspeeding. The motivation behind our work is to leverage Object detection and Image processing to implement a system that calculates the speed of vehicles from a road traffic surveillance video. This system can be further extended to recognize vehicles that violate the speed limit restrictions on a particular road.

In our work, we used a deep learning based approach for object detection, object tracking, and vehicle speed calculation. Vehicle speed estimation involves three steps - Object detection, Object tracking, and vehicle speed calculation. The first task is to accurately detect, localize, and classify vehicles present in the input video. Object detection is one of the most challenging tasks in Computer Vision. It requires accurate localization and identification of the object present in the image. We use the YOLOv3 algorithm [3] for detecting vehicles in the input video. Object tracking involves assigning a unique ID to each object and tracking the object trajectories while maintaining the IDs. We use the Deep SORT algorithm to track vehicles from the input video. To calculate vehicle speed, we use the frame difference method and the linear perspective transformation. Regions with CNN features (R-CNN) [4] was introduced by Ross Girshik in 2014, followed by further improvements with the introduction of Fast R-CNN [5] and Faster R-CNN [11]. R-CNN is a Region Proposal based architecture that uses Selective search to generate 2000 region proposals for each image. A CNN module is utilized to extract 4096-dimensional feature vectors. The features obtained are used to predict bounding boxes using Regression methods and classify images using SVM (Support Vector Machines).

You Only Look Once [9] is a method that treats object detection as a single regression problem. A single Convolutional Neural Network model predicts class probabilities and bounding box coordinates in one iteration. YOLO is much faster than Region-based methods like R-CNN [4], Fast R-CNN [5], and Faster R-CNN [11]. We use version 3 of YOLO (YOLOv3) in our work to detect and localize vehicle instances in the input video.

There are many approaches available in the literature on vehicle speed detection, Danang Wahyu Wicaksono [12] used Image processing using Morphology operation for object localization and Euclidean distance for vehicle speed calculation. P. Devi Mahalakshmi, Dr. M. Babu [8] used the Haar cascade classifier algorithm for vehicle identification and used Euclidean distance for vehicle speed calculation. Nagaratna M Raikar and Dr. Megha P Arakeri [9] use gradient edge detection algorithm and normalized cross-correlation method for object detection. Tingting Huang [6] uses Faster R-CNN for object detection, histogram-based tracking algorithm for object tracking, and image warping with linear perspective transformation for vehicle speed calculation.

2. Data Description

In our work, we have used the AAU RainSnow traffic surveillance dataset [1]. It consists of 22 traffic surveillance videos collected from seven intersections in the Danish cities of Aalborg and Viborg. Each location has multiple videos captured in different weather and light conditions such as rain, snow, haze, and fog. The resolution of cameras used is 640 x 480 pixels, and the frame rate is 20 frames per second. We also use Vehicle speed measurement on urban highways [7] dataset. This dataset is divided into five sets with a total of 20 videos captured by a 5-megapixel CMOS image sensor. The videos have a frame resolution of 1920 x 1080 pixels, at 30.15 frames per second.

3. Methods

This section deals with the steps involved in vehicle speed detection such as object detection, object tracking, and frame difference method with linear perspective transformation using mathematical formulations.

3.1 Object Detection

YOLOv3 [3] model with DarkNet-53 as a backbone is trained on ImageNet 1000-class competition dataset processes images in real-time and outputs Bounding boxes and their corresponding Class Probabilities. YOLO algorithm [10] is best suited for real-time, fast, and robust object detection.

Bounding Box prediction : YOLOv3 predicts four coordinates for every bounding box – t_x, t_y, t_w, t_h. The bounding box predictions are represented by the following equations:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{tw} \tag{3}$$

$$b_h = p_h e^{th} \tag{4}$$

Where p_w and p_h are prior width and prior height of the bounding box and c_x and c_y are the cell offsets from top left corner of the image.

Class Prediction : Every bounding box predicts the class of the object using multi-label classification. YOLOv3 uses binary cross-entropy loss for the class predictions. Frames extracted from traffic surveillance videos are used as input, and object detection is applied to them. False Detections are filtered by using an Intersection over Union(IoU) Threshold of 0.5 on bounding box dimensions and a Confidence threshold of 0.5 on the class probability.

Fig. 1. Object Detection using YOLOv3.

After applying YOLOv3 to frames from videos for different locations, the final detections are shown in Figure 1.

3.2 Multiple Object Tracking

We have used Deep SORT [13], which is an extension to Simple Online Realtime Tracking (SORT) [2] for Multiple Object Tracking. SORT uses Kalman Filtering for Object Tracking and frame-by-frame data association using the Hungarian method to measure bounding box overlap. It uses appearance as a metric for Object Tracking. SORT suffers from a relatively high number of identity switches. Deep SORT overcomes the issue by using a more informed metric that combines motion and appearance information. Deep SORT uses a pretrained Convolutional Neural Network (CNN) model to generate features for the objects, which increases its robustness against misses and occlusions. Figure 2. shows output of object tracking using Deep SORT.

Fig. 2. Object tracking using Deep SORT

3.3 Vehicle Speed Detection

Vehicle speed calculation involves two crucial steps:
1) Calculating speed in pixels per second
2) Conversion of speed from pixels per second to meters per second
Firstly we mark two checkpoints on the lane in the video. We then calculate the vehicle speed in pixels per second by calculating the time taken by that vehicle in traveling from one checkpoint to the other checkpoint. Time taken is measured by calculating the difference in frame numbers of the video when the vehicle reaches the first to the second checkpoint.

$$t = f / FPS \quad (5)$$

$$speed = d \times FPS / f \quad (6)$$

Where f is the frame difference, FPS is the frames per second value of the input video, and d is the pixel distance between two checkpoints (see Fig. 3).
To convert speed from pixels per second to meters per second, we calculate the conversion factor by mapping the pixel distance to meter distance using Homography for linear perspective transformation (see Fig. 4).

Fig 3. Checkpoints for the location 1

Location 1

Location 2

Location 3

Fig. 4. Images before and after Linear Perspective transformation

The calculation of the conversion factor is done by using a known metric such as lane width for calculating the ratio between pixel and real-world distances. For instance, location 1 is a two-lane road, and in Denmark, its width is 7 meters, and the pixel distance for the transformed image is 320 px, and the conversion ratio is 0.0218.

4. Results and Discussion

We have used two performance measurement metrics: Detection rate and Root mean squared error, which is discussed below.

4.1 Performance Measurement

Detection Rate: Detection rate is a measure for evaluating the performance of the object detection algorithm. It is the average ratio of the number of vehicles accurately detected to the total number of vehicles present in the frame for each frame in the entire surveillance video.

$$DR = \Sigma_o^n (v_i / t_i) \quad (7)$$

Where DR denotes detection rate, v_i denotes the number of vehicles detected in the i^{th} frame, and t_i denotes total vehicles present in the i^{th} frame.
Root Mean Squared Error (RMSE): Root mean squared error is the standard deviation of the error in vehicle speed calculation. RMSE is a measure for evaluating the performance of the vehicle speed calculation module. Error in vehicle speed calculation is the absolute value of the difference between the predicted and actual speed of vehicles.

$$\mathrm{RMSE} = [\, \Sigma_o^n (v1_i - v2_i)^2 / n]^{1/2} \quad (8)$$

Where v1 and v2 are predicted speed and actual speed of the i^{th} vehicle and n is the total number of vehicles.

5. Experimental setup and Results

We calculated the number of vehicles accurately detected and the total number of vehicles present in each frame for 2500 frames across the surveillance videos captured from three different locations in the AAU Rainsnow Traffic Surveillance dataset [1]. Using equation (7) we obtained an overall Object detection rate of 91.22 % across three locations.

Since the dataset provides information about the vehicles' speeds, we have used Vehicle Speed Measurement on the Urban Highways dataset [7] for RMSE calculation. We calculated the vehicle speed for 55 vehicles with the Frame difference method using the equations (5) and (6) and obtained an overall RMSE of 4.568 km/hr on the dataset.

The Fig. 5 shows a histogram that visualizes a distribution of percentage error in vehicle speed calculation for 55 vehicles in the input video sequence.

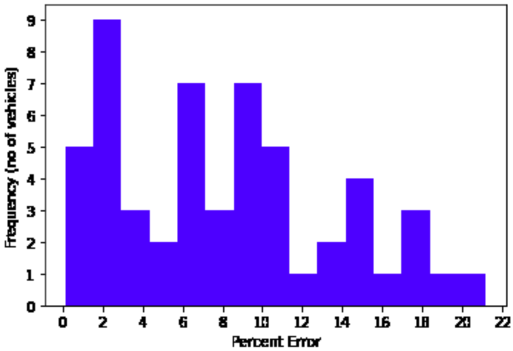

Fig 5. Histogram for error distribution

6. Conclusion

Vehicle speed detection is an important aspect of intelligent traffic surveillance and management systems. Our research shows that object detection can be used to identify and track vehicles in extreme weather conditions like rain and snow. It also shows that the frame difference method and linear perspective transformation can be used to calculate vehicle speed with RMSE as low as 4.568 km/hr. Our system can identify and track vehicles with an object detection rate of 91.22%. This work can be further extended to detect over-speeding vehicles on highways, which can potentially be a key factor in reducing the number of accidents due to over-speeding.

7. Future Scope

Our work can be further extended to uniquely identify each vehicle from input surveillance video using License Plate recognition by using Optical Character Recognition (OCR). This can be leveraged to uniquely identify and take action against vehicles that cross a specific speed limit on the highway.

8. Acknowledgment

The authors are thankful to Prof. S. P. Shintre, PICT, (SPPU Pune), for her valuable support and guidance.

References

[1] Bahnsen, Chris H. and Moeslund, Thomas B. (2018). Rain Removal in Traffic Surveillance: Does it Matter?. IEEE Transactions on Intelligent Transportation Systems, 1-18.

[2] Alex Bewley , Zongyuan Ge, Lionel Ott, Fabio Ramos and Ben Upcroft. (2016). Simple Online And Realtime Tracking. IEEE International Conference on Image Processing, 1-5.

[3] Ali Farhadi and Joseph Redmon. (2018). YOLOv3: An Incremental Improvement. IEEE Conference on Computer Vision and Pattern Recognition, 1-6.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. (2014). Rich feature hierarchies or accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580-587.

[5] Ross Girshick. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 1440-1448.

[6] Tingting Huang. (2018). Traffic Speed Estimation from Surveillance Video Data. IEEE Conference on Computer Vision and Pattern Recognition, 1-7.

[7] Diogo C. Luvizon, Bogdan Tomoyuki Nassu and Rodrigo Minetto. (2017). A Video-Based System for Vehicle Speed Measurement in Urban Roadways. IEEE Transactions on Intelligent Transportation Systems, 18:1-12.

[8] P. Devi Mahalakshmi and Dr. M. Babu. (2019). Vehicle Speed Estimation using Haar Classifier Algorithm. International Journal of Trend in Scientific Research and Development, 4:1-4.

[9] Nagaratna M Raikar and Dr. Megha P Arakeri. (2020). Development of Vehicle Speed Estimation Technique using Image Processing. International Journal of Engineering Research & Technology, 9:1-8.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. (2016). You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition, 1-10.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. (2017). Faster R-CNN: Towards Real-ime Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39:1-14.

[12] Danang Wahyu Wicaksono and Budi Setiyono. (2017). Speed Estimation On Moving Vehicle Based On Digital Image Processing. International Journal of Computing Science And Applied Mathematics, 3:1-7.

[13] Nicolai Wojke, Alex Bewley and Dietrich Paulus, (2017). Simple Online and Realtime Tracking with a Deep Association Metric. IEEE International Conference on Image Processing, 1-4.

[14] https://morth.nic.in/road-accident-in-india.