Real-Time Automated Overspeeding Detection and Identification System

Lakshitaa Sehgal[1], Anshal Aggarwal[2], Sarthak Sood[3], Aryan Aggarwal[4]

[1]University Institute of Engineering and Technology Chandigarh, Panjab University, Chandigarh, India
[2]University Institute of Engineering and Technology Chandigarh, Panjab University, Chandigarh, India
[3]National Institute of Technology, Kurukshetra, India
[4]Chitkara University, Punjab, India

---***---

Abstract - *In today's world, overspeeding is a big issue and causes a great number of road accidents. A lot of people get severely injured and some also lose their lives due to this. So it becomes essential for traffic police authorities to prevent it and keep a strict check on people who break permissive speed limits. So, an automated system is necessary which detects whether a car is speeding or not. This paper proposes a real-time automated over speeding detection and Identification system which checks whether a car is overspeeding or not using OpenCV and Python. By using YOLOv3 for object detection, the vehicle number can then be determined if the vehicle is overspeeding. Then using Optical Character Recognition (OCR), the text extracted is passed through the government database system to retrieve information about the owner.*

***Key Words*: Speed detection, OpenCV, Python, Object Detection, Person Identification**

1. INTRODUCTION

In recent times, where road accidents are increasing day by day, it has become important to cater to them and keep a strict check to reduce them immensely. Overspeeding constitutes approximately 60% of road accidents (as per 2020 reports). Other dangers associated with overspeeding is that it reduces the effectiveness of airbags, seat belts and other safety equipment. It can pose a life threat to pedestrians also.

To check for this, this paper proposes an end-to-end system using the Cascade Classifier[1] to capture vehicles on the road and then using OpenCV[2] and Python to detect their speeds. Using YOLOv3[3], if it is determined that the vehicle is overspeeding, a license plate detection is done, which is then converted into text using Optical Character Recognition (OCR)[4], and this text is passed to the government database for identification of the vehicle owner.

2. PROPOSED SYSTEM

This section represents the fully automated and end-to-end proposed system for overspeeding detection and Identification.

2.1 SYSTEM SETUP

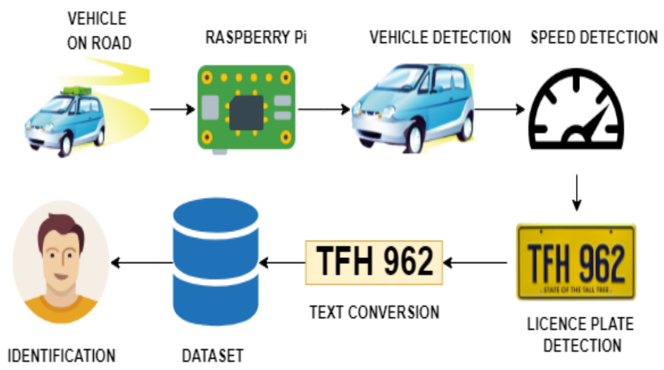

Fig -1: System Setup

The Raspberry Pi 4[5] and a CCTV camera with NVR for IP cameras and DVR for analogue cameras or webcam with attached system memory are used to capture real-time videos of vehicles on the road. Along with it, an additional HC-05 Bluetooth Module is used. The hardware system setting allows for the capture of real-time videos of vehicles, which are then passed for speed estimation.

2.2 VEHICLE DETECTION

To detect vehicles, the CascadeClassifier function[1] is used. It's a machine-learning approach in which a cascade function is trained using a large number of positive and negative photos. After that, it's used to find items in other photos. Along with this, a pre-trained XML file (vech.xml)[6] is used. In this case, multiple objects, such as cars, need to be identified at the same time, so detectMultiScale is also used.

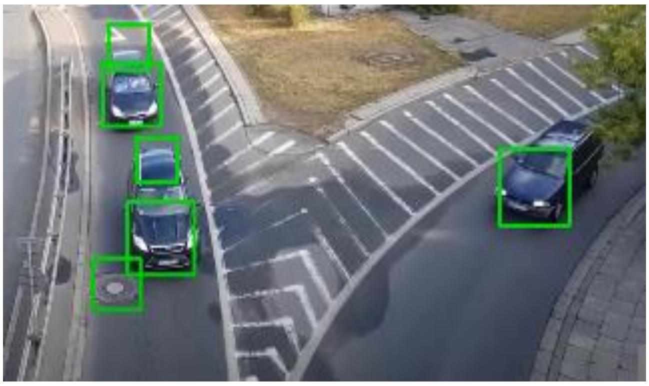

Fig -2: Vehicle Detection

2.3. SPEED ESTIMATION AND DETECTION

For speed detection, OpenCV[2] and Python are used. The basic approach for detecting is to locate two points on the road and calculate the distance between them. Time is estimated by calculating the time taken by a vehicle to cover the distance between those two demarcated points. Speed is then calculated by applying the distance-time formula as illustrated below.

Let (x1, y1) and (x2, y2) be the two points where (x_2, y_2) and (x_1, y_1) represents the current frame's and previous frame's centroid of the vehicle respectively.

Time taken by vehicle to cover distance between two points = T seconds

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

Speed = Distance/Time

Speed = $\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]} / T$

Here, distance is measured in kilometers

Time is measured in hours

So, speed is detected in kilometers/hours

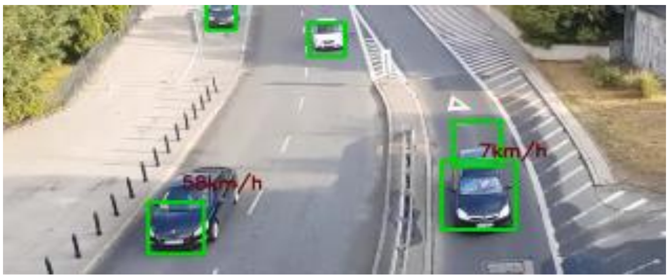

Fig -3: Speed Detection using OpenCV and Python

The above model is tested on some random videos available online for speed detection and optimum results have been obtained from this model.

If the vehicle is found overspeeding, then the license plate of the vehicle is detected.

2.4 LICENSE PLATE OBJECT DETECTION

If a vehicle is found to be overspeeding, an object detection system records the vehicle's license plate. Here is YOLOv3 (You Only Look Once, Version 3)[3]. Video, live feeds, and images can all be analyzed by a real-time object detection algorithm called YOLOv3[3]. It uses features learned by a deep convolutional neural network to detect an object.

The model using YOLOv3[3] for object detection is trained and tested on the Car Number Plate Detection dataset[7] which consists of 931 car images and gives the best accuracy of approximately 85.4%

Fig -3: Before Object Detection

Fig -4: After Object Detection

Fig -5: Before Object Detection

Fig -6: After Object Detection

2.5 OPTICAL CHARACTER RECOGNITION (OCR)

To convert the identified object into text, we use pytesseract OCR[4] library in python. Tesseract searches pixels, letters, phrases, and sentences for templates. It employs a two-step process known as adaptive recognition. It requires one data step for character identification, followed by a second stage to fill in any missing letters with letters that match the word or phrase context.

Fig -7: Image to Text Conversion

Fig -8: Image to Text Conversion

After the text conversion, the text is passed through the government database and identification of the car owner can be identified from the registered car license plate.

3. CONCLUSION

A robust end-to-end Real-Time Automated Overspeeding Detection and Identification System has been proposed here which can be used in the future to keep strict on overspeeding vehicles. This proposed system is more cost-efficient as no expensive speed guns, heavy speed detection devices or extra traffic police staff are required to manually detect speed. This is an end-to-end automated speed detection and identification system.

REFERENCES

[1] Soo, Sander. "Object detection using Haar-cascade Classifier." (2014).

[2] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek,"A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1725-1730.

[3] https://arxiv.org/abs/1804.02767

[4] Sargur N. Srihari, Ajay Shekhawat, and Stephen W. Lam. 2003. Optical character recognition (OCR). Encyclopedia of Computer Science. John Wiley and Sons Ltd., GBR, 1326–1333

[5] Maksimovic, Mirjana & Vujovic, Vladimir & Davidovi´c, Nikola & Milosevic, Vladimir & Perisic, Branko. (2014). Raspberry Pi as Internet of Things hardware: Performances and Constraints.

[6] https://docs.cryengine.com/display/CEMANUAL/Vehicle+XML+Reference

[7] https://www.kaggle.com/elysian01/car-number-plate-detection