Cognitive Vehicle Behavior Detector with Real-Time Analytics and Implementation

Mummareddy Yogendra Sai, Suri Kavya, Sravya Bhimavarapu, Mona Mudaliar, and Om Prakash Sahu

Abstract In this contemporary world, with the surge in automobiles, traffic has been a major issue. This project is going to deal with the violation of the vehicles in the traffic signals. By considering the surveillance footage, frames are achieved by using the open-source platform computer vision. Further, by applying image processing tools, details of the violated vehicles are generated in the form of tickets. By considering these tickets, analysis of data is performed so that we can visualize the traffic violations. The whole programming is performed using Python language, ELK stack includes Elastic search, Logstash, and Kibana are used for data analytics. Tickets generated are stored in a single file and loaded through Logstash. Then it is uploaded into the Kibana and visualization is performed.

Keywords Violation · Data analytics · Image processing

1 Introduction

The expanding number of vehicles in urban areas can cause high volume of traffic and suggests that petty criminal offenses become progressively basic these days around the globe. This causes serious annihilation of property and more mishaps that may imperil the lives of the individuals. Indeed the driving task is a very complicated one, involving numerous simultaneous actions [1].

In order to assist this task, several driver assistant systems have been suggested in the past years using either database information (i.e., learned geographic information systems) or on-vehicle sensors (i.e., laser, camera, etc.) to provide various environment information such as traffic signs, speed limits, traffic lights, crosswalks, or any other information like pedestrian or obstacles [3]. To take care of the disturbing issue

M. Y. Sai · S. Kavya · S. Bhimavarapu · M. Mudaliar
Electronics and Communication Department, Koneru Lakshmaiah University, Guntur, India

O. P. Sahu (✉)
SENSE, Vellore Institute of Technology, Chennai, India
e-mail: omprakash.sahu@vit.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

A. Sivasubramanian et al. (eds.), *Futuristic Communication and Network Technologies*,
Lecture Notes in Electrical Engineering 792,
https://doi.org/10.1007/978-981-16-4625-6_26

and forestall such inconceivable outcomes, petty criminal offense recognition frameworks are required. .This framework can distinguish most regular three sorts of petty criminal offense continuously which are signal infringement, stopping infringement and misguided course infringement. An easy-to-use graphical interface is related with the framework to make it basic for the client to work the framework, screen traffic and make a move against the infringement of traffic rules. To solve the alarming problem and prevent such unfathomable consequences, traffic violation detection systems are needed [3].

The objective of the project is to automate the traffic rules violation identification framework and make it simple for the traffic police office to screen the traffic and make a move against the disregarded vehicle proprietor in a quick and effective manner. Identifying and following the vehicle and their exercises precisely is the principle need of the framework.

2 Literature Survey

The literature survey of this project started with the announcement of the government to enhance the traffic violations which is available in the Web site [4]. The main concept of the idea for this project is taken from the Web site where the constraints of the project along with the dataset are provided.

Further, looking at the constraints for the project, it is clear that machine learning model along with Python would fit for this [5]. After having glance at the types of machine learning, reinforcement type of learning will be suited for this project.

Python libraries such as TensorFlow, Computervision and other libraries related to the video surveillance are used [6]. For extracting the video into frames and converting the video into respective frames, Computervision is used along with the TensorFlow. All the basic concepts of Computervision and TensorFlow are stated clearly in this Web site.

Further idea was to implement the data analytics by using the Elastic cloud where the analytics is performed [7]. Elastic is a specialized cloud where ingestion of the data and visualization is performed in a single platform called Elastic stack where working of Elasticsearch, Logstash and Kibana are the clearly stated in this Web site.

Image processing techniques such as gray scaling and blurring, binary thresholding, dilution, and finding the counter are the predefined libraries [8]. All the libraries for image processing are clearly stated in this Web site.

3 Components of the System

The framework comprises of two parts which incorporates:

- Vehicle detection model and Graphical UI.

First the CCTV camera footage from the street side is sent to the framework. Vehicles are recognized from the recording. Following the action of vehicles decides whether there is any infringement or not. Various kinds of violations have various calculations to decide the violation. A flowchart as shown in Fig. 2 shows how the framework functions. The Graphical UI (GUI) makes the framework intelligent for client to utilize. User can screen the traffic film and get the alert of violation with the caught vehicle picture. Client can make further move utilizing the GUI.

3.1 Database Structure

Making use of the Python programming language and SQL, whole data of the application is managed as we can observe in Fig. 1. Moreover, relational database such as Boyce–Codd normal form is also used for the main tables which includes (Fig. 2):

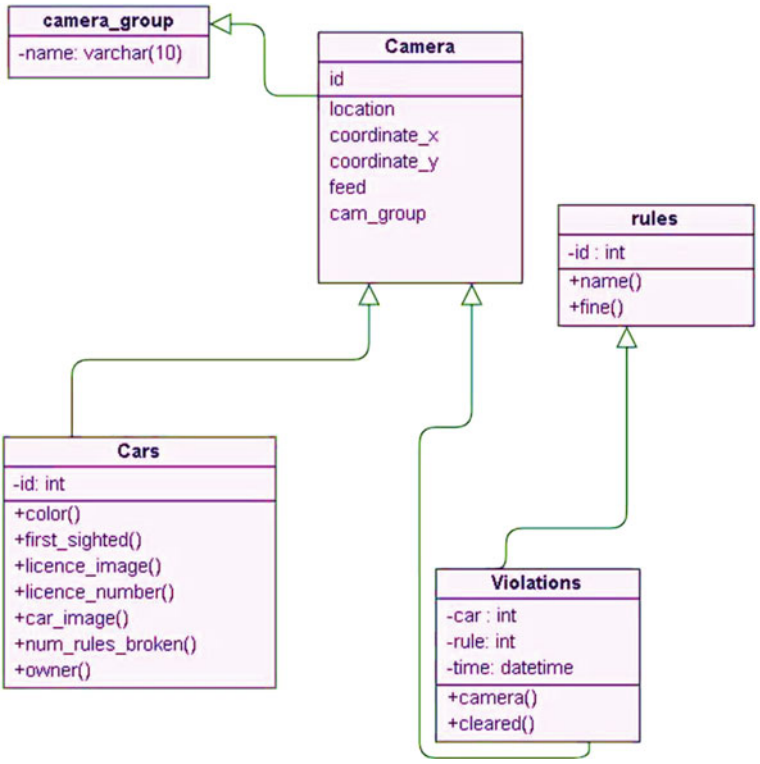

Fig. 1 Class diagram

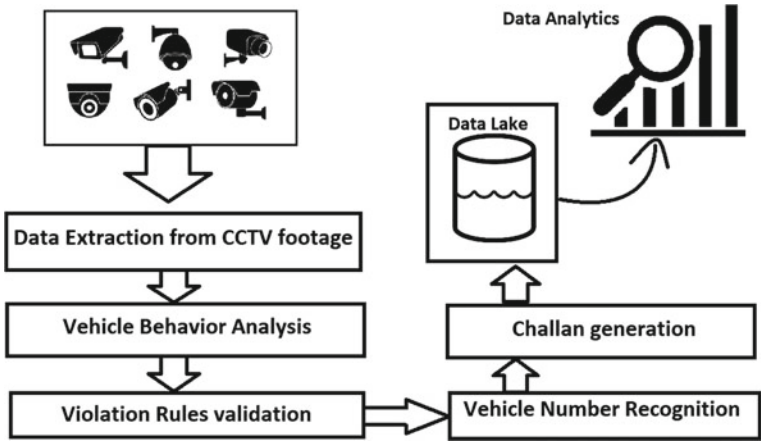

Fig. 2 Flow diagram of the project

- Cars
- Rules
- Cameras
- Violations
- Groups.

4 Methodology

4.1 Image Processing

The video consists of frames which can be extracted by performing image processing. Many steps such as

- **Gray scaling and blurring**: In the preprocessing stage, input frame which is received from CCTV footage is grayscaled and blurred using Gaussian blur method.
- **Background subtraction:** By subtracting the current frame with reference frame, objects area can be determined.

$$\text{dst(I)} = \text{saturate}(|\text{scr1(I)} - \text{scr2(I)}|)$$

- **Binary thresholding:** To obtain the accurate object area by removing all the noise and holes frame binarization method is used by providing threshold to the image.

$$\text{dst}(x, y) = \text{maxVal is scr}(x, y) > \text{thresh else } 0$$

Table 1 MobileNet body architecture

Type/stride	Filter shape	Input size
Conv/s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw/s1	3 × 3 × 32 dw	112 × 112 × 32
Conv/s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw/s2	3 × 3 × 64 dw	112 × 112 × 64
Conv/s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw/s1	3 × 3 × 128 dw	56 × 56 × 128
Conv/s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw/s2	3 × 3 × 128 dw	56 × 56 × 128
Conv/s1	1 × 1 × 128 × 128	28 × 28 × 128
Conv dw/s1	3 × 3 × 256 dw	28 × 28 × 256
Conv/s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw/s2	3 × 3 × 256 dw	28 × 28 × 256
Conv/s1	1 × 1 × 256 × 512	14 × 14 × 256
Conv dw/s1	3 × 3 × 512 dw	14 × 14 × 512
Conv/s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw/s2	3 × 3 × 512 dw	14 × 14 × 512
Conv/s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw/s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv/s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool/s1	Pool 7 × 7	7 × 7 × 1024
FC/s1	1000 × 1024	1 × 1 × 1024
Softmax/s1	Classifier	1 × 1 × 1024

- **Dilation and finding the contour:** Image is diluted to fill the holes and contour is found from the image by drawing the rectangle box over desired object (Table 1).

Transfer learning approach is accustomed to prepare the model with our dataset. The dataset comprises of 500 pictures for every class. The preparation boundaries are referenced in table (Table 2).

Table 2 Training hyperparameters

Parameter Name	Value
Learning rate	0.01
Training steps	100

5 Implementation

OpenCV is the library for the picture handling where inbuilt libraries are utilized. Tensor stream AI structure is utilized for actualizing vehicle classifier. User can include menu things, for example, camera, area, the feed record for the camera. Here the feed record is introduced by the camera module over the Web. We have utilized Linux record sharing example for getting the video from the camera, where the camera will take care of the offered document to the server, and the server will take the feed record to process and identify violation. X and Y directions as shown in Fig. 3 of the camera area are utilized for possible future reason. JSON record is likewise required by determining a few standards for camera reason.

The client has numerous different articles to embed into the database. The administrator can include the accompanying elements as shown in Fig. 4 in the graphical UI:

- Camera
- Car
- Rule
- Violation.

A GUI is made for the purpose of gathering of cameras by the administrator consistently. He can see the rundown of rule violations and can see details of the vehicles that abused the standards. In the event that he taps on the detail button, another window will show up where the client will have the option to document the report or send/print ticket for the vehicle proprietor.

Likewise the administrator/client can erase the records for a false positive. However, there will be never a record erased. The database has a marker of which

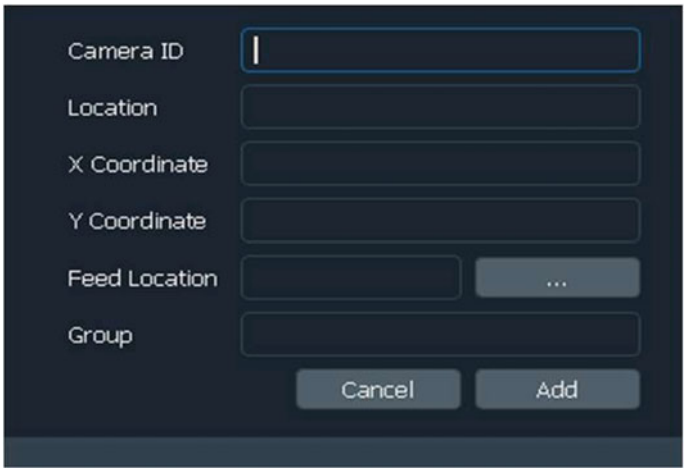

Fig. 3 Details of rule violation

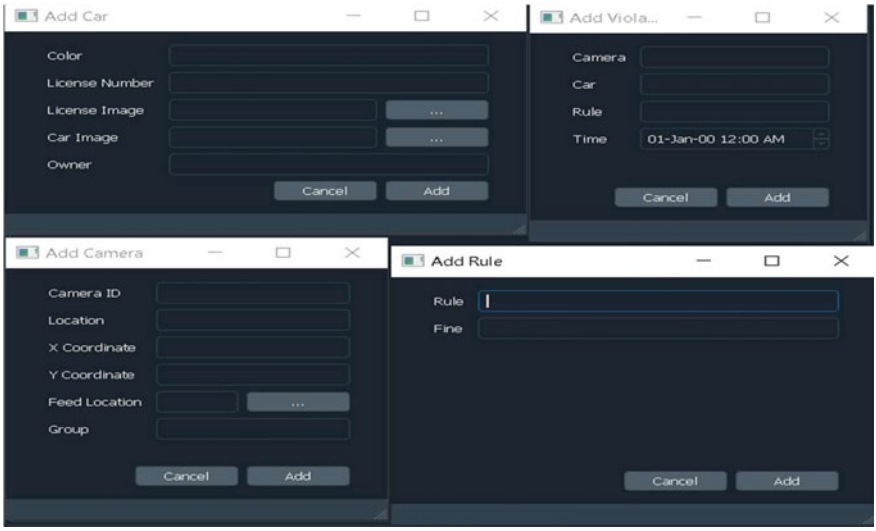

Fig. 4 Entity for adding windows

document have been chronicled. On the off chance if we need to recover a record from the erased once, at that point the administrator needs to go to the file window. There he can re-establish any record he needs.

The user can also look for a vehicle, with its license number, its color, or date of a rule violation as shown in Fig. 5. The permit number has content forecast so the client will be certain while composing a license number that it exists.

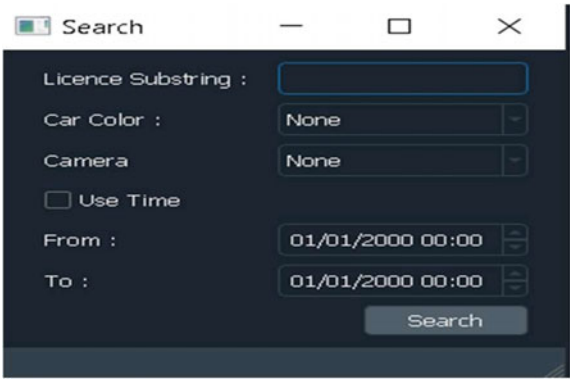

Fig. 5 Searching a car or rule violation

6 Rules Violation Video Representation in UI

There are as of now three guidelines we are worried about.

- Signal violation
- Parking violation
- Direction violation.

6.1 Signal Violation

We have utilized a straight line in the image. At the point when the traffic light is red and a vehicle is crossing the straight line, an image of that vehicle is enrolled in the database alongside some natural qualities. The client can find in the live review which vehicle is being identified continuous and tried in the event that they are disregarding.

6.2 Parking Violation

For parking violation, we have prefigured a square shape, which is the confined region for vehicle leaving. On the off chance that there is a vehicle in the square shape for more than a predefined time, at that point a picture with other ecological qualities is being enlisted to the database.

6.3 Direction Violation

For direction violation location as shown in Fig. 6, a few lines are attracted to isolate into districts. At that point when a vehicle moves starting with one area then onto the next, its heading is estimated. On the off chance that the course is not right, at that point it is enlisted as past.

7 Data Analytics

Now by considering each violation as a ticket, analytics is performed so that areas with high violations can be known .Data analytics is performed using Elastic stack which includes Elasticsearch, Logstash and Kibana [9].

Fig. 6 Direction violation

7.1 Elasticsearch

Elasticsearch is a dispersed, open-source search and analytics engine for a wide range of information, including printed, numerical, geospatial, organized, and unstructured. Elasticsearch is the focal point of the Elastic Stack, a lot of open-source instruments, enemy information ingestion, improvement, stockpiling, investigation, and perception. To begin Elasticsearch, open the order brief and type elasticsearch

7.2 Logstash

Logstash is an open-source, server-side information handling pipeline that ingests information from a huge number of sources, changes it, and then sends it to our necessary reserve. In this project, tickets are transformed into the config file where every details of the violations lies. In the conf file, details of the violations are present and to start the logstash, open the command prompt by name of file.

7.3 Kibana

Kibana lets you to visualize your elastic search data and navigate the elastic stack so that you can do anything from tracking query load to understand the way requests flow through required applications. In this project, visualization is performed for violated vehicles. Kibana can be accessed by typing kibana.bat in the command prompt.

8 Results

Violated vehicle details are considered as tickets and stored in a file using the programming in Python. Subsequently, analytics is performed by using the ELK stack. Storing, searching and analyzing of the data will be done by elasticsearch. By using the Logstash, ingestion of data from the respective file to the cloud is done. Final output of the project is visualized in Kibana where the overall details of the violations can be known.

According to the violations of the vehicles, in Fig. 7, by using various visualization metrics, counts of the tickets are shown with the help display meter in the first sub-parts of figure. In the following sub-figure, numerical value of count of the tickets are depicted. The registration number of the vehicle with highest violations are visualized by means of heat map which enlarges the license number with more number of violations and types of violations are depicted in the fourth sub-figure.

In Fig. 8, as the wrong route violations are higher compared to parking violation, it gets enlarged by heat map. In the following figure, count of the violation type is shown.

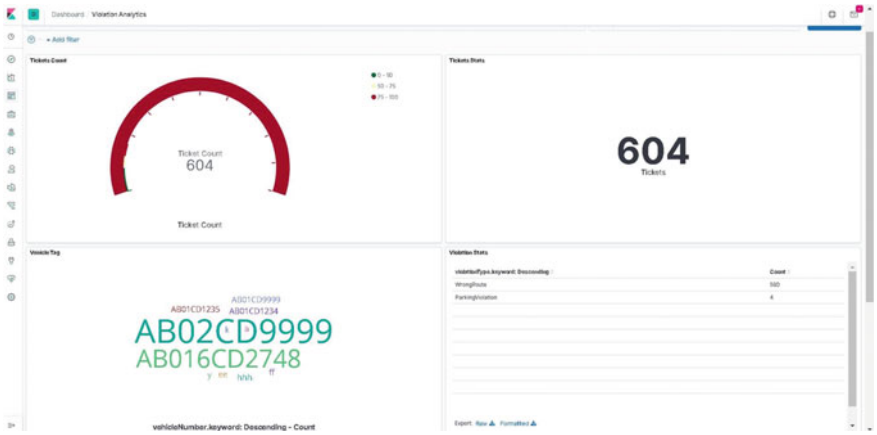

Fig. 7 Visualization in Kibana

Fig. 8 Violation tag and violation metrics

9 Conclusion

Vehicle challan can be generated by policeman by cross verifying the details generated by programming. Data visualization of the violated vehicles is done. In future, man power can be reduced at the traffic signals. Challan generations can be made fully automated with enhanced AI/ML models.

10 Discussion

The primary purpose of this study was to analyze the traffic and violations so that minimal man power can be used at the traffic signals. Initially, the video frames will be extracted and by using TensorFlow and image processing tools are applied where violations can be generated as a ticket.

Our results clearly proved the vehicles violation and analytics depict the type of violation from the specified location. From the analytics, while generating challan it will be clear for the person about the violation.

References

1. Rouzier B, Murakami T (2016) Hazard detection and cognition for an active driving assistance. In: 2016 IEEE 14th international workshop on advanced motion control (AMC)
2. de Charette R, Nashasihibi F Real time visual traffic lights recognition. In: World congress final paper template
3. Wang X, Meng L-M, Zhang B, Lu J, Du K-L (2013) A video-based traffic violation detection system. In: MEC, Shenyang, China, 2013
4. https://data.gov.in/ogpl_apis
5. https://www.analyticsvidhya.com/blog/2018/12/key-breakthroughs-ai-ml-2018-trends-2019/
6. https://www.geeksforgeeks.org/introduction-to-opencv/?ref=lbp
7. https://www.elastic.co/
8. https://www.geeksforgeeks.org/python-image-blurring-using-opencv/?ref=lbp
9. Elastic Stack and Product Documentation retrieved from https://www.elastic.co/guide