1. Introduction

The Increasing number of cars in cities can cause high volume of traffic, & implies that traffic violation become more critical now a days in world. This causes severe destruction of property and more accidents that may endanger the lives of the people. To solve the alarming problem and prevent such unfathomable consequences, traffic violation detection systems are needed. For which the system enforces proper traffic regulations at all times, and apprehend those who does not comply. A traffic violation detection system must be realized in real-time as the authorities track the roads all the time. Hence, traffic enforcers will not only be at ease in implementing safe roads accurately, but also efficiently; as the traffic detection system detects violations faster than humans. This system can detect most common three types of traffic violation in real- time which are signal violation, parking violation and wrong direction violation.

A user friendly graphical interface is associated with the system to make it simple for the user to operate the system, monitor traffic and take action against the violations of traffic rules Detection system, the proposed architecture of surveillance system with intelligent detection and tracking of multiple vehicles from the surveillance input video using mobile net v1 as an object detection algorithm. This is done through a neural network and an object detection model which are used in the classification of the moving objects into different respective classes, thus achieving vehicle classification. Next, from the same given video footage, traffic lights, zebra crossing, different lanes, and traffic signs are classified this comes under environment awareness. Combining these two, now violations are detected based on violations are then detected these can occur on the road which are signal jump, speed detection, and vehicle count. The main objective is to detect multiple vehicle violation detection and it gives a more detailed picture of concepts and technology involved in creating a traffic violation detection system using computer vision.

The application of genetic algorithm for traffic violation detection system was not that widespread with researchers, as the typical technique used in such system is image processing related topics. Yet, there were quite few papers that used genetic algorithm for image matching like Ke's, and Li's research [5], which could be correlated with the traffic violation detection system. This paper employs a machine vision for traffic violation detection system using genetic algorithm. The said system is assumed to have predefined parameters to acquire consistent results; hence a real-time processing is attained in the system. In addition, the violation detection system realized in this paper is limited in detecting swerving and blocking the pedestrian lane violations. The proceeding chapter discusses the process of the system implementation and the analysis of the results.

It is very difficult to identify moving vehicles, then to track and classify them in real time within a complex environment. There are a variety of approaches to vehicle detection in video streams, including background difference, inter-frame difference, inter-frame corresponding, and edge detection methods.

The background difference method is commonly used in video processing. A background image without vehicles is firstly obtained. This image is then subtracted from the current input image, and the difference image is obtained. One can determine whether vehicles exist in input image by binarization of the difference image. This

method is computationally fast. However, it needs to update the background image in real time when the environment changes.

Problem Statement :

This project aims to develop an Traffic Signal Violation Detection System with the goal so that it is very easy to monitor traffic and take action against those who violates the traffic rules.

2. Related Work

Presented in the implementation of real-time traffic violation detection in a monitoring stream which utilized simultaneous video stream from different cameras using parallel computing techniques. Another approach of implementing real-time traffic violation detection was seen in [2], as they used video-based traffic detection through an improved Background updating algorithm, thereafter track the moving vehicles by feature based tracking method.

This project is inspired by above project but it is implemented using a self-developed approach. Conventionally vehicle detection is referred as an object detection problem. To detect moving vehicle objects from the road, YOLOv3 model is used which uses Darknet-53. After detecting vehicles, violation conditions are checked.

MOVING DETECTION:

The background difference method is commonly used in video processing. A background image without vehicles is firstly obtained. This image is then subtracted from the current input image, and the difference image is obtained. One can determine whether vehicles exist in input image by binarization of the difference image. This method is computationally fast. However, it needs to update the background image in real time when the environment changes.

A. Background Difference Method:

The inter-frame difference method subtracts the current frame from the previous frame, and then we can find the changing area with setting the threshold value. When the target is in motion, there will be residual image in moving direction which the motive vehicles can be detected by.

B. Inter-Frame Difference Method:

The edge detection method applies edge detection on the input image, and then applies image denoising on the resulting image. The denoised image is matched with a template image. If they match each other, there is a vehicle. This method has few impacts on the environment. It outperforms the background difference method. However it has high computational complexity for vehicle model matching. Thus, this method is not suitable for real-time processing.

C. Edge Detection Method:

The optical flow method [6] is an effective way to detect moving targets. It detects moving objects by the change in the time domain of pixel intensity of an image sequence, and the relationship between the structure of objects and movement. This method is, however, computationally intensive and is susceptible to noise.

D. Optical Flow Method:

The optical flow method [6] is an effective way to detect moving targets. It detects moving objects by the change in the time domain of pixel intensity of an image sequence, and the relationship between the structure of objects and movement. This method is, however, computationally intensive and is susceptible to noise.

E. Block Matching Method:

The block matching algorithm (BMA) is a video detection method based on motion vectors. It splits an image

into M × N macro blocks. We can get the motion vectors by searching for optimal matching of a macro block of the current frame in the next frame. A moving vehicle is composed of many macro blocks which perform the same movement. This method is often used in motion estimation.

D. Red Light Violation Detection System:

Red Light Violation Detection System is a mass surveillance system that automatically captures the Image of vehicle violating the traffic rules. RLVD system automatically captures images of vehicle from backside with license plate numbers and performs OCR (optical character recognition) on images to read the license plates on vehicles. RLVD continuously monitors the traffic signal, and camera is itself triggered by any vehicle passing over the sensors at specified time after signal has turned red. Alerts are typically sent by mail or SMS to the owners of violating vehicles, based on review of photographic evidence. RLVD can be used for following purposes such as Automatic vehicle identification system, Vehicle surveillance - vehicle and speed checks, stolen vehicle detection, Monitoring known offenders, criminals, Border patrol / enforcement, Cargo container logistic management system.

3. Objective

The goal of the project is to automate the traffic rules violation detection system and make it ease for the traffic police department to monitor the traffic and take action against the violated vehicle owner in a fast and efficient way. Detecting and tracking the vehicle and their activities accurately is the main priority of the system.

Detection system, the proposed architecture of surveillance system with intelligent detection and tracking of multiple vehicles from the surveillance input video using YOLOv3 as an object detection algorithm. This is done through a neural network and an object detection model which are used in the classification of the moving objects into different respective classes, thus achieving vehicle classification. Next, from the same given video footage, traffic lights, zebra crossing, different lanes, and traffic signs are classified this comes under environment awareness. Combining these two, now violations are detected based on violations are then detected these can occur on the road which are signal jump, speed detection, and vehicle count. The main objective is to detect multiple vehicle violation detection and it gives a more detailed picture of concepts and technology involved in creating a traffic violation detection system using computer vision. It also aims to throw light on some of the applications and the latest developments being made in the said field.

Because of violating traffic rules so many accidents are happening in day today life, So many accidents are happening because of violating speed limit technologies are available today like RADAR GUN and GASTO camera but they are failing to control speed violation because they are used in only few places only so driver can violet speed limit in places where these equipments are not there.

4. System Overview

The System consists of two main components -

- Vehicle detection model and
- A graphical user interface (GUI)

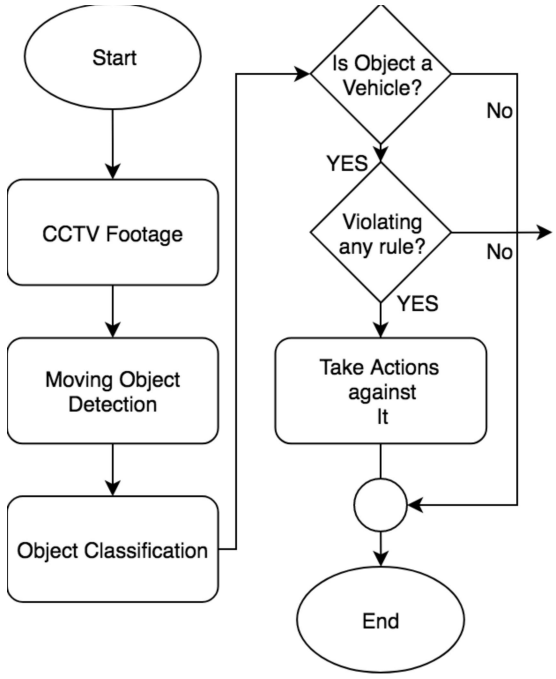

Fig. 4.1 System Flowchart

First the CCTV camera footage from the road side is sent to the system. Vehicles are detected from the footage. Tracking the activity of vehicles system determines if their is any violation or not. Different types of violations have different algorithms to determine the violation. A system flowchart 1 shows how the system works.

The Graphical User Interface (GUI) makes the system interactive for user to use. User can monitor the traffic footage and get the alert of violation with the captured vehicle image. User can take further action using the GUI.

Modules Used In :

1.PyQt5: Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development. PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android. PyQt5 may also be embedded in C++ based applications to allow users of those applications to configure or enhance the functionality of those applications.

2.QDarkStyle: The most complete dark/light style sheet for Qt applications (Qt4, Qt5, PySide, PySide2, PyQt4, PyQt5, QtPy, PyQtGraph, Qt.Py) for Python 2/3 and C++. Python 2, as well as Qt4 (PyQt4 and PySide), will not be supported anymore. They still there as it is, but no back-compatibility, fixes, nor features will be implemented. We

still preparing the portability to Qt6 since we need changes in QtPy dependency project. Check the documentation to see how to set the desirable theme palette. This module provides a function to load the stylesheets transparently with the right resources file.

3.PyQtTimer: If an operation is performed periodically in the application, such as periodically detecting the CPU value of the host, then the QTimer timer is needed.When the window's control receives a Timeout signal, it stops this timer.

5. Methodology

5.1 Image Processing

Gray scaling and blurring: As the part of preprocessing the input frame got from the CCTV footage, the image is gray scaled and blurred with Gaussian Blur method.

Background Subtraction: Background subtraction method is used to subtract the current frame from the reference frame to get the desired object's area. Equation (1) shows the method. Dst(I) = saturate(| scr1(I) — scr2(I) |)

Binary Threshold: Binarization method is used to remove all the holes and noises from the frame and get the desired object area accurately. Equation (2) shows how the binary threshold works. Dst(x, y) = maxVa1 : if scr(x, y) › thresh e1se 0

Dilation and find the contour: After getting the threshold image, it is dilated to fill the holes and the contour is found from the image. Drawing rectangle box over the contours desired moving objects are taken.

5.2 Vehicle Classification

From the preprocessed image moving objects are extracted. A vehicle classification model is used to classify those moving objects into three class – Car, Moto bike and Non vehicle. The classifier model is built with mobile net v1 neural network architecture.

Table 5.1 Mobile Net v1 Process Values

Type / Stride	Filter Shape	Input Size
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Classifier	1 x 1 x 1000

Table 5.2 Training hyperparameters.

Parameter Name	Value
Learning Rate	0.01
Tranining Steps	100

Transfer learning approach is used to training the model with our dataset. The dataset consists of 500 images per class. The training parameters are mentioned in table (2).

Existing Features of This Methodology:

1.Bounding Box Predictions:

YOLOv3 is a single network the loss for objectiveness and classification needs to be calculated separately but from the same network. YOLOv3 predicts the objectiveness score using logistic regression where 1 means complete overlap of bounding box prior over the ground truth object. It will predict only 1 bonding box prior for one ground truth object and any error in this would incur for both classification as well as detection loss. There would also be other bounding box priors which would have objectiveness score more than the threshold but less than the best one. These errors will only incur for the detection loss and not for the classification loss.

2.Class Prediction:

YOLOv3 uses independent logistic classifiers for each class instead of a regular softmax layer. This is done to make the classification multi-label classification. Each box predicts the classes the bounding box may contain using multilabel classification.

3.Predictions across scales:

To support detection a varying scales YOLOv3 predicts boxes at 3 different scales. Then features are extracted from each scale by using a method similar to that of feature pyramid networks. YOLOv3 gains the ability to better predict at varying scales using the above method. The bounding box priors generated using dimension clusters are divided into 3 scales, so that there are 3 bounding box priors per scale and thus total 9 bounding box priors.

4.Feature Extractor:

YOLOv3 uses a new network- Darknet-53. Darknet-53 has 53 convolutional layers, its deeper than YOLOv2 and it also has residuals or shortcut connections. Its powerful than Darknet -19 and more efficient than ResNet-101 or ResNet-152.

Violation detection: After detecting the vehicles three violation cases arises-

Signal violation: If a vehicle crosses a predefined line on the road while there is red signal, it is detected as a signal violation.

Parking violation: If a vehicle stands still in no parking zone for a predefined time, it is detected as a parking violation.

Direction Violation: when a vehicle comes from a wrong direction, it is detected by tracking the vehicle. The direction of the vehicle is determined using its current position and previous few positions.

5.3 Database Structure

We have used SQLite database with python to manage the whole data of our application. Here, in the relational database we have used BCNF of 5 tables. The tables are:

1. Cars
2. Rules
3. Cameras
4. Violations
5. Groups

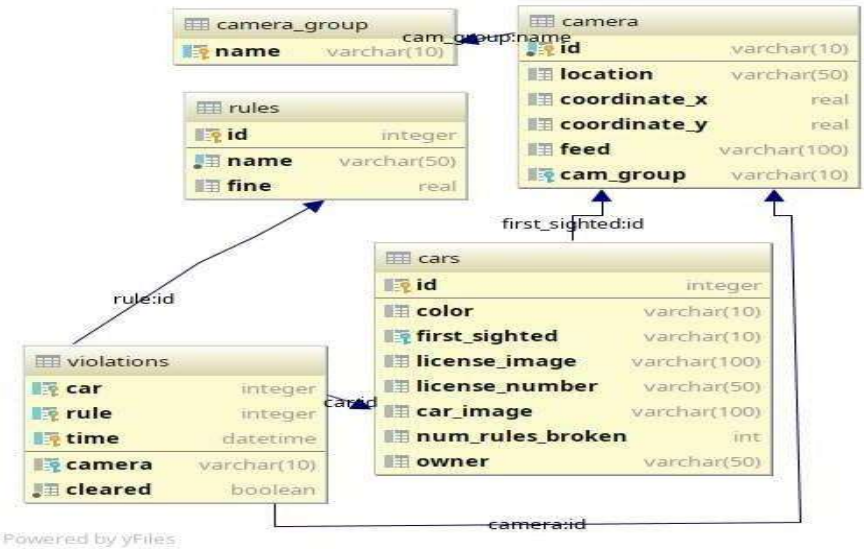

Fig: 5.3 Database tables

Here are the descriptions of each tables:

Cars:

This table will hold the recorded cars by the camera. A car entity is a car with a unique identifier(id), color(color), license-number of the car(license), where the car is first sighted (first_sighted), an image of the license number (license_image), an image of the car(car image), number of rules broken so far(num rules_broken) and the owner of the car (owner).

Rules:

This table holds all the rules, their description(name) and fine for breaking that rule (fine).

Camera:

Camera table holds a unique identifier for the camera(id), location description(location), the longitude(coordinate x) and the latitude(coordinate y) of the location of the camera, where the camera will feed its data video(feed) and in which group the camera is in(group).

Camera group:

This table simply holds the unique group names of the camera groups(name).

Violations:

This table takes all the ids of other tables as foreign key and creates a semantic record like this: A car with this id has broken that rule at this time, which is captured by this camera.

6. Implementation

6.1 Image Processing and Computer Vision

OpenCV computer vision library is used in Python for image processing purpose. For implementing the vehicle classifier with, Tensorflow machine learning framework is used.

6.2 Graphical User Interface (GUI)

The user interface has all the options needed for the administration and other debugging purpose so that, we do not need to edit code for any management. For example, if we need to add some sample cars or camera in the database, we can do it with the menu item.

Fig: 6.1 Overall user interface view

Primarily, for the start of the project usage, the administrator needs to add a camera with the menu item. In the way, the administrator can add the location of the camera, the feed file for the camera. Here the feed file is installed by the camera module over the internet. We have used Linux file sharing pattern for getting the video from the camera, where the camera will feed the given file to the server, and the server will take the feed file to process and detect violation. Also the X and Y coordinate(Fig: 2) of the camera location can be saved by the admin. This is done for future use, when we will try to use a map for locating the cameras with ease. Also the admin need to specify some rules with a JSON file for the camera. For example, the camera is used for cross road on red line violation, or is used for wrong place parking detection etc.

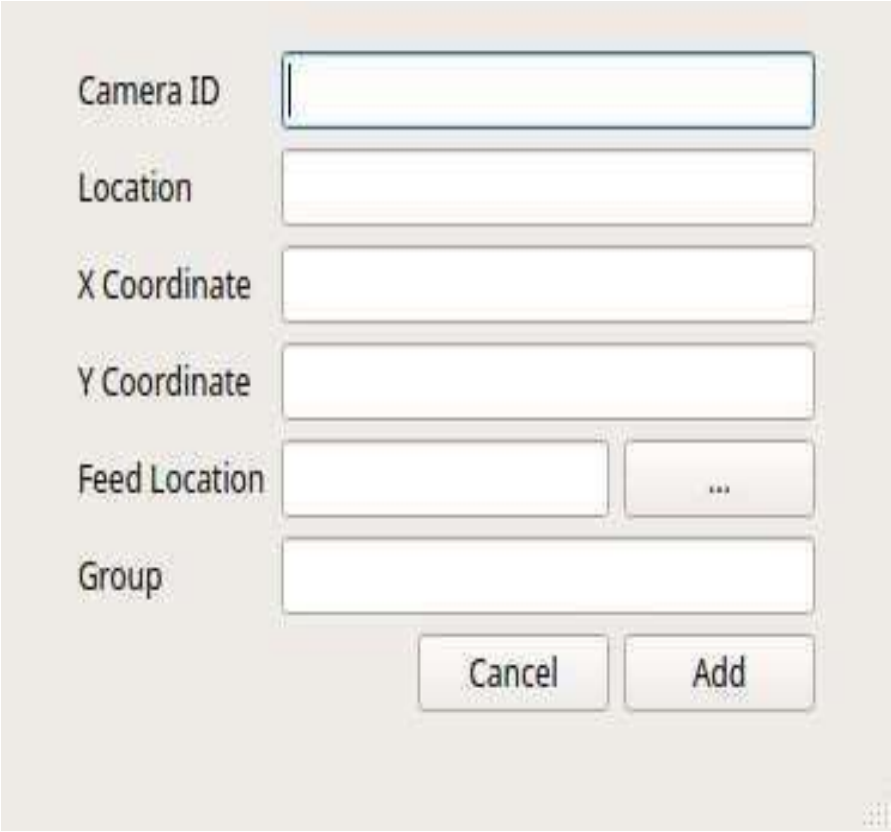

Fig: 6.2 Interface for adding camera entity

Actually this is all mainly needed for starting up the system, After adding the camera, the software will automatically start detecting violations of traffic rules, After this, opening the camera by selecting it with the drop down menu, will fill the detection rules violations.

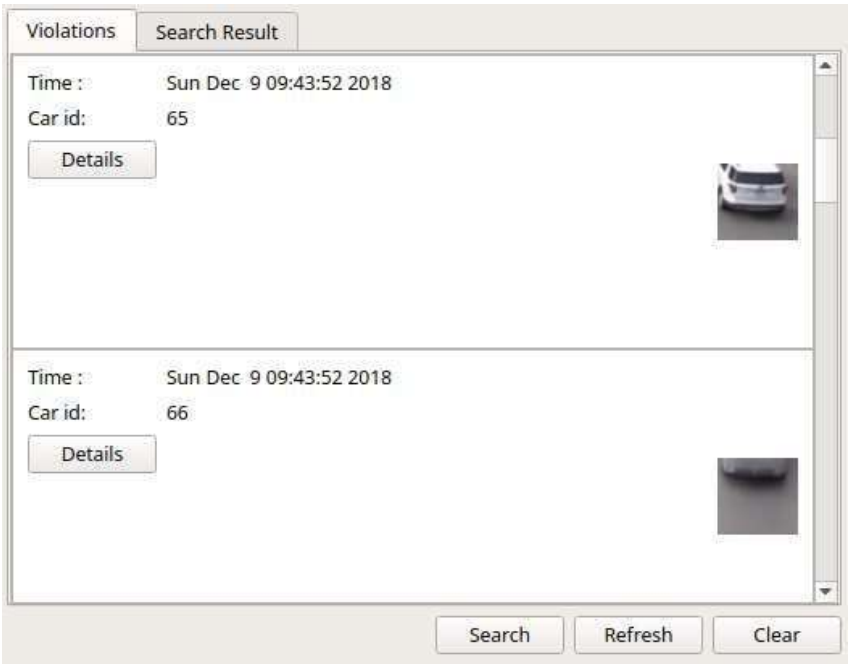

Fig: 6.3 List view of violation records

The user has many other objects to insert into the database. The admin can add the following entities in the graphical user interface:

1. Camera.
2. Car.
3. Rule.
4. Violation.
5. Groups

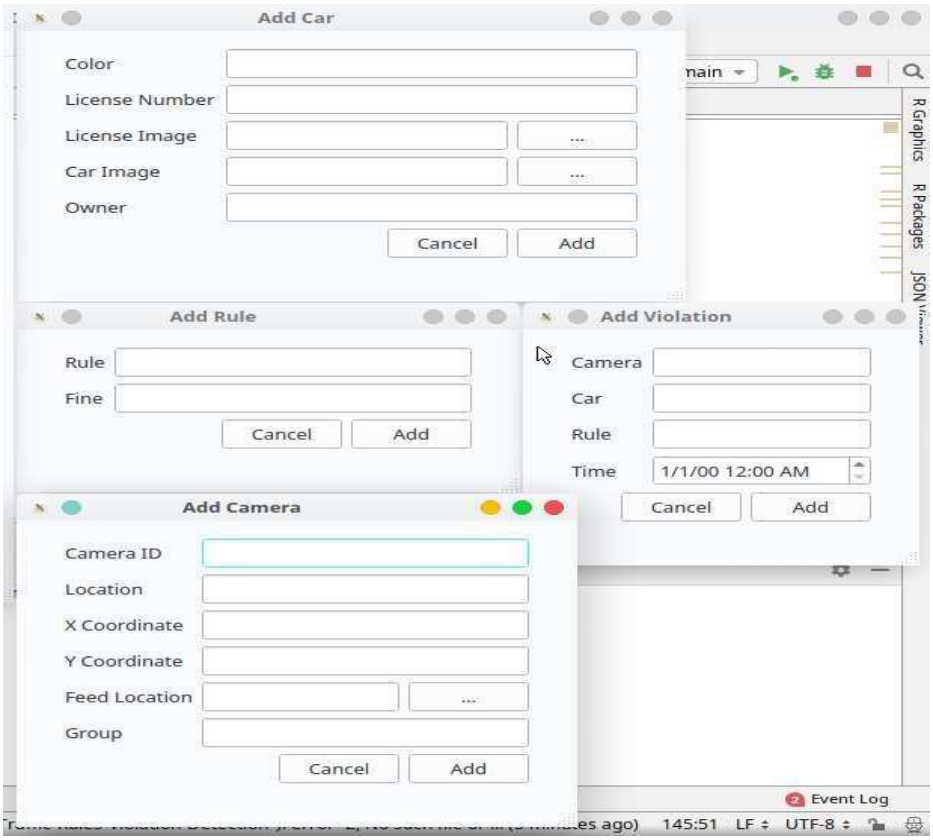

Fig: 6.4 Adding items interface

GUI is made mainly for this purpose that, there will always be a supervisor for a group of cameras. He can see the list of rule violations and can see details of the cars that violated the rules. If he clicks on the detail button, a new window will appear where the user will be able to file the report or send/print ticket for the car owner.

Also the admin/user can delete the records if he gets a false positive. But there will never a record deleted. The database has a marker of which file have been archived. If we want to retrieve a record from the deleted once, then the admin needs to go to the archive window. There he can restore any record he wants. The user can also search for a vehicle, with its license number, its color, or date of a rule violation. The license number has text prediction so the user will be sure while typing a license number that it exists.

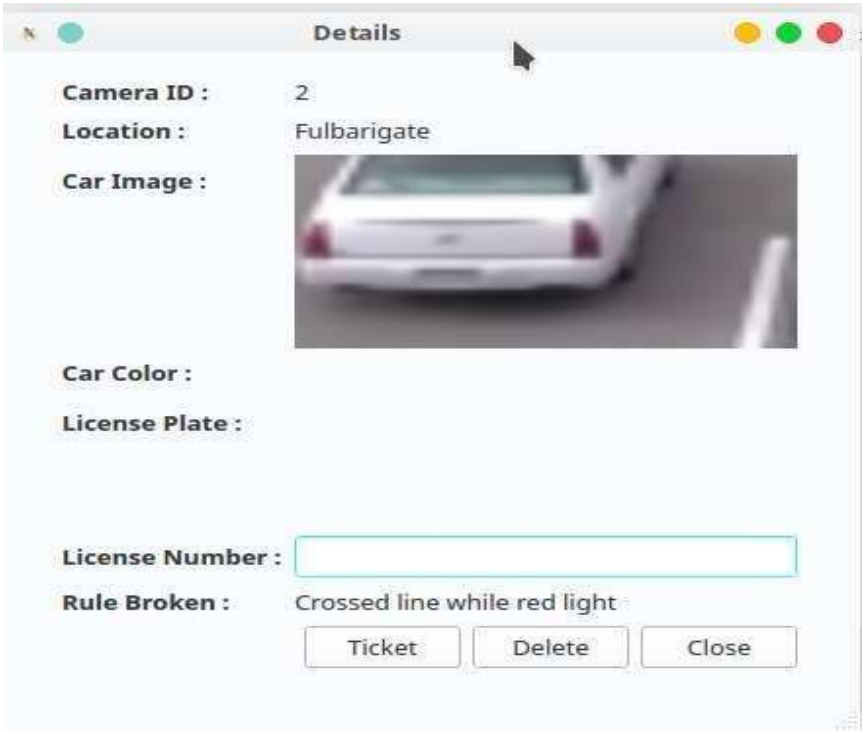

Fig: 6.5 details of rule violation

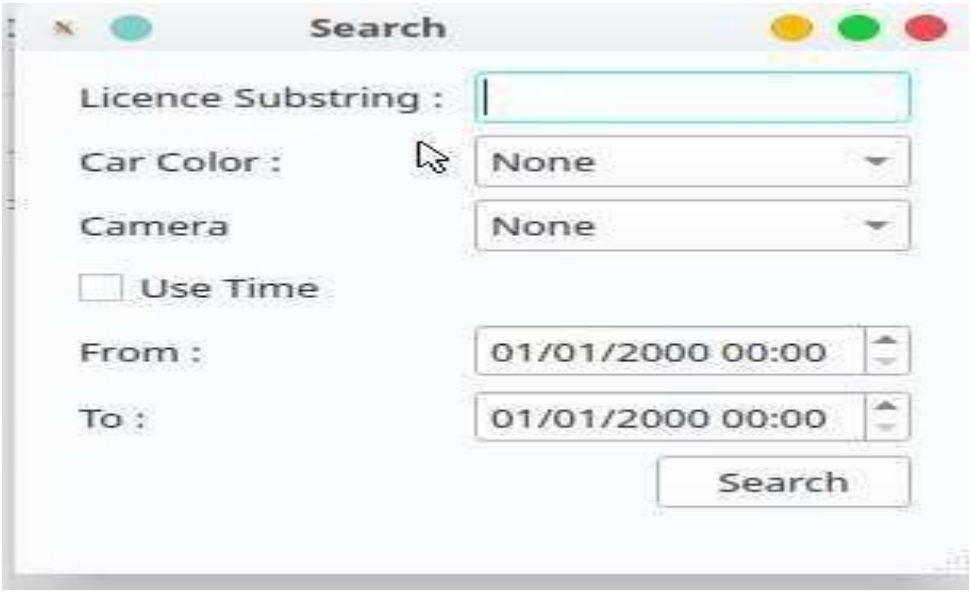

Fig: 6.6 Searching a car or rule violation

6.3 Rules violation video representation in UI

There are currently 3 rules we are concerned with.

1. Signal Violation.
2. Parking Violation.
3. Direction Violation.

For Signal Violation, We have used a straight line in the picture. When the traffic light is red and a car is crossing the straight line, a picture of that car is registered in the database along with some environmental values. The user can see in the live preview which car are being detected real time and tested if they are crossing the line.

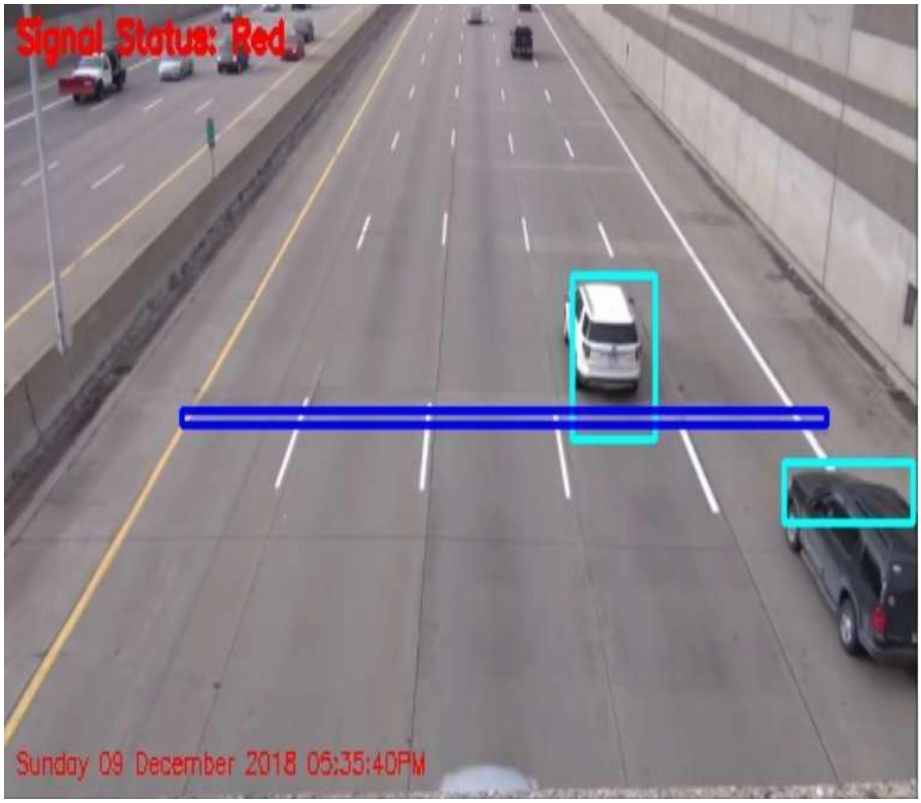

Fig: 6.7 Signal violation camera representation

For parking violation, we have prefigured a rectangle, which is the restricted area for car parking. If there is a vehicle inthe rectangle for more than a predefined time, then a image with other environmental values is being registered to the database.

For direction violation detection, some lines are drawn to divide into regions. Then when a car moves from one region to another, its direction is measured. If the direction is wrong, then it is registered as previous.

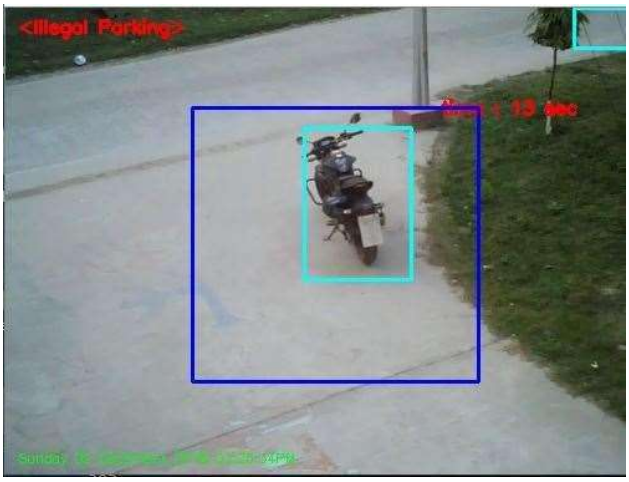

Fig: 6.8 Parking violation camera representation

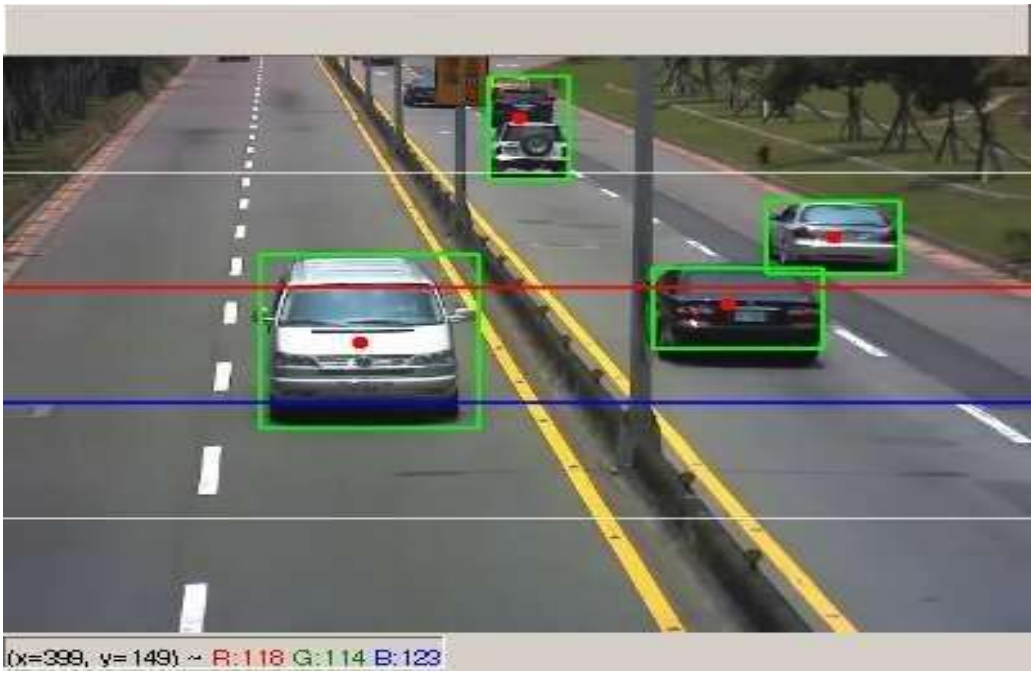

Fig: 6.9 Direction violation camera representation

Libraries used for graphical user interface:

1. PyQt5
2. QDarkStyle
3. PyQtTimer

7.Conclusion

The designed algorithm was effectively able to detect the type of violation specified on this project which are denying traffic signal. The convergence of detection for the traffic violation mentioned is dissimilar, since it has a different threshold condition. The system provides detection for traffic signal violation. Further, the system is able to process one data at a time. Also, the program runtime is somewhat slow, and can be improved by using a computer with high speed processor specifications or GPU.

Future research about the application of the designed algorithm for other advanced image processing techniques. Since, this may improve the program runtime of the system by neglecting other unnecessary steps done in a background difference method. A computer vision algorithm may be done instead to provide more intelligence in the system. Our future plan is to implement the number plate detection with OCR support to make this system more robust.

References

[1] G. Ou, Y. Gao and Y. Liu, "Real Time Vehicular Traffic Violation Detectionin Traffic Monitoring System," in 2012 IEEE/WIC/ACM, Beijing, China , 2012.

[2] X. Wang, L.-M. Meng, B. Zhang, J. Lu and K.-L.Du, "A Video-based Traffic Violation Detection System," in MEC, Shenyang, China, 2013.

[3] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement".

[4] Y. Artan, O. Bulan, R. P. Loce and P. Paul, "Passenger Compartment Violation Detection in HOV/HOT Lanes," in 2014 IEEE Winter Conference inApplications ofComputer Vision, Steamboat Springs, CO,USA, 2015.

[5] X. Zhang, Y. Tian, L. Liang, T. Huang and W. Gao, "MACRO-BLOCK-LEVEL SELECTIVE BACKGROUND DIFFERENCE CODING FOR SURVEILLANCE VIDEO," in 2012 IEEE International Conference on Multimedia and Expo, Beijing, China, 2012.

[6] G. Ou, Y. Gao and Y. Liu, "RealTimeVehicularTrafficViolationDetectioninTrafficMonitorin gStream," in 2012 IEEE/WIC/ACM , Beijing, China , 2012.

[7] X. Wang, L.-M. Meng, B. Zhang, J. Lu and K.-L.Du, "A Video-based Traffic Violation Detection System," in MEC, Shenyang, China, 2013.

[8] Y. Ke and D. L. Yuhao Li, "Image Matching using Genetic Algorithm on GPU," in IEEE, Tianjin, China, 2011.

[9] S. Sumathi, T. Hamasapriya and P. Surekha, Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab, Berlin: Springer, 2008.

[10] G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, A. A. Bandala and E. P. Dadios, "Blind Localization Method for Quadrotor-Unmanned Aerial Vehicle (QUAV) Utilizing Genetic Algortihm," in 7th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM), Puerto Princesa, Palawan, Philippines, 2013.

[11] S. C. Barrido and E. P. Dadios, "Online Robot Tracking Using Genetic Algorithms", in Intelligent Control, 2002 Proceedings of the 2002 IEEE International Symposium on Intelligent Control, Vancouver, Canada, 2002.

[12] M. A. Ferreira and H. K. Lee, "Multiscale Modeling: A Bayesian Perspective," Springer, Chicago, 2007.

[13] M. Cepin, Assessment of Power System Reliability: Methods and Applications, Chicago: Springer, 2011.

Links:

- PyQt: https://riverbankcomputing.com/software/pyqt/intro
- opencv: https://docs.opencv.org/2.4/index.html
- imutils: https://github.com/jrosebr1/imutil