# Traffic Signal Violation Detection System

By,
**Abu Noman Md. Sakib**
**Roll: 1607058**
&
**Pias Roy**
**Roll: 1607071**

**Department of Computer Science and Engineering**

**Khulna University of Engineering & Technology, Khulna**

# Acknowledgement

With immense please, We, **Abu Noman Md. Sakib** and **Pias Roy** presenting project report of **“Traffic Signal Violation Detection System”** as part of the curriculum of **“CSE 3200: System Development Project”**. We wish to thank all the people who gave us unending support.

We express our profound thanks to our supervisor **Mahmudul Hasan Shauqi** sir who have directly guided us and helped us in the development of this system project.

**Abu Noman Md. Sakib**<br>
**Roll: 1607058**<br>
**&**<br>
**Pias Roy**<br>
**Roll: 1607071**

# TABLE OF CONTENTS

# 1 Introduction

The increasing number of cars in cities can cause high volume of traffic, and implies that traffic violations become more critical nowadays in Bangladesh and also around the 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 traffic light violation in real-time. 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.

# 2 Objectives

The goal of the project is to automate the traffic signal violation detection system and make it easy 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.

# 3 Related Work

Presented in [1] is 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.

# 4 System Overview

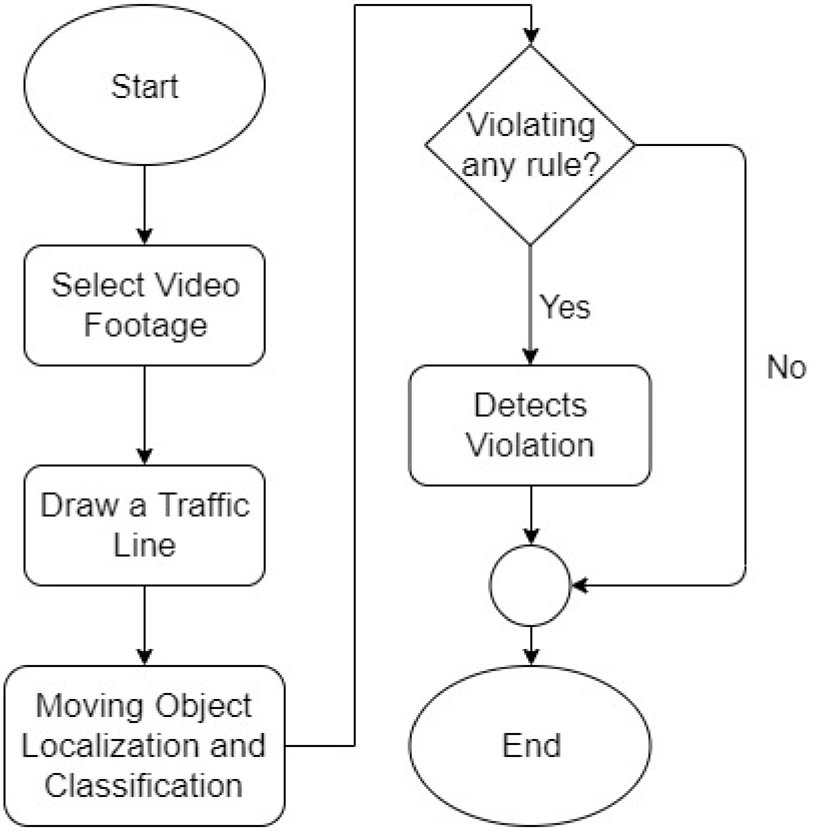


Figure-1: Flow diagram of traffic signal violation detection system.

The System consists of two main components -

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

First the video footage from the road side is sent to the system. Vehicles are detected from the footage. Tracking the activity of vehicles, system determines if there is any violation or not. Figure 1 shows how the system works.

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

# 5 Methodology

## 5.1 Vehicle Classification

From the given video footage, moving objects are detected. An object detection model YOLOv3 is used to classify those moving objects into respective classes. YOLOv3 is the third object detection algorithm in YOLO (You Only Look Once) family. It improved the accuracy with many tricks and is more capable of detecting objects. The classifier model is built with *Darknet-53* architecture. Table-1 shows how the neural network architecture is designed.

### Features:

**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:**

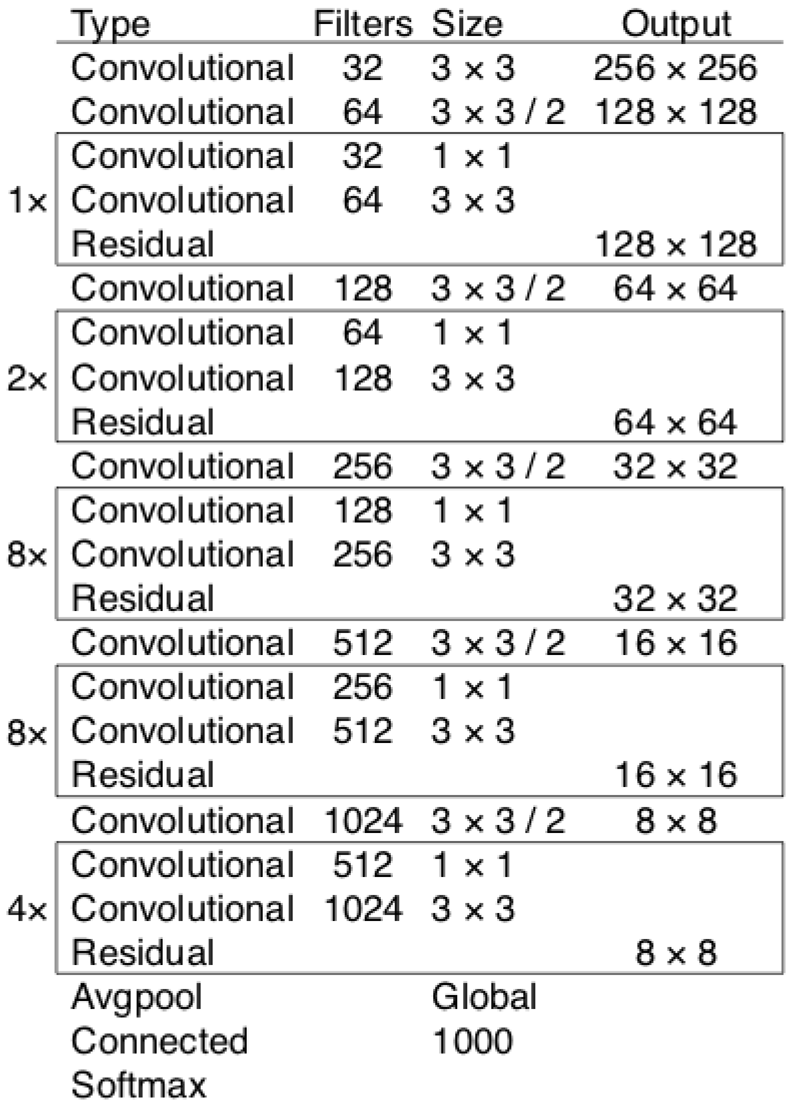
YOLOv3 uses independent logistic classifiers for each class instead of a regular softmax layer. This is done to make the classification multi-label classification. Each box predicts the classes the bounding box may contain using multilabel classification.

**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.



| | Type | Filters | Size | Output |
|---|---|---|---|---|
| | Convolutional | 32 | 3 × 3 | 256 × 256 |
| | Convolutional | 64 | 3 × 3 / 2 | 128 × 128 |
| 1× | Convolutional | 32 | 1 × 1 | |
| | Convolutional | 64 | 3 × 3 | |
| | Residual | | | 128 × 128 |
| | Convolutional | 128 | 3 × 3 / 2 | 64 × 64 |
| 2× | Convolutional | 64 | 1 × 1 | |
| | Convolutional | 128 | 3 × 3 | |
| | Residual | | | 64 × 64 |
| | Convolutional | 256 | 3 × 3 / 2 | 32 × 32 |
| 8× | Convolutional | 128 | 1 × 1 | |
| | Convolutional | 256 | 3 × 3 | |
| | Residual | | | 32 × 32 |
| | Convolutional | 512 | 3 × 3 / 2 | 16 × 16 |
| 8× | Convolutional | 256 | 1 × 1 | |
| | Convolutional | 512 | 3 × 3 | |
| | Residual | | | 16 × 16 |
| | Convolutional | 1024 | 3 × 3 / 2 | 8 × 8 |
| 4× | Convolutional | 512 | 1 × 1 | |
| | Convolutional | 1024 | 3 × 3 | |
| | Residual | | | 8 × 8 |
| | Avgpool | | Global | |
| | Connected | | 1000 | |
| | Softmax | | | |

Table 1. **Darknet-53.**

## 5.2 Violation Detection

The vehicles are detected using YOLOv3 model. After detecting the vehicles, violation cases are checked. A traffic line is drawn over the road in the preview of the given video footage by the user. The line specifies that the traffic light is red. Violation happens if any vehicle crosses the traffic line in red state.

The detected objects have a green bounding box. If any vehicle passes the traffic light in red state, violation happens. After detecting violation, the bounding box around the vehicle becomes red.

# 6 Implementation

## 6.1 Computer Vision

OpenCV is an open source computer vision and machine learning software library which is used in this project for image processing purpose. Tensorflow is used for implementing the vehicle classifier with *darknet-53*.

## 6.2 Graphical User Interface (GUI)

The graphical user interface has all the options needed for the software. The software serves administration and other debugging purposes. We don’t need to edit code for any management. For example, if we need to open any video footage, we can do it with the Open item (Figure-2).

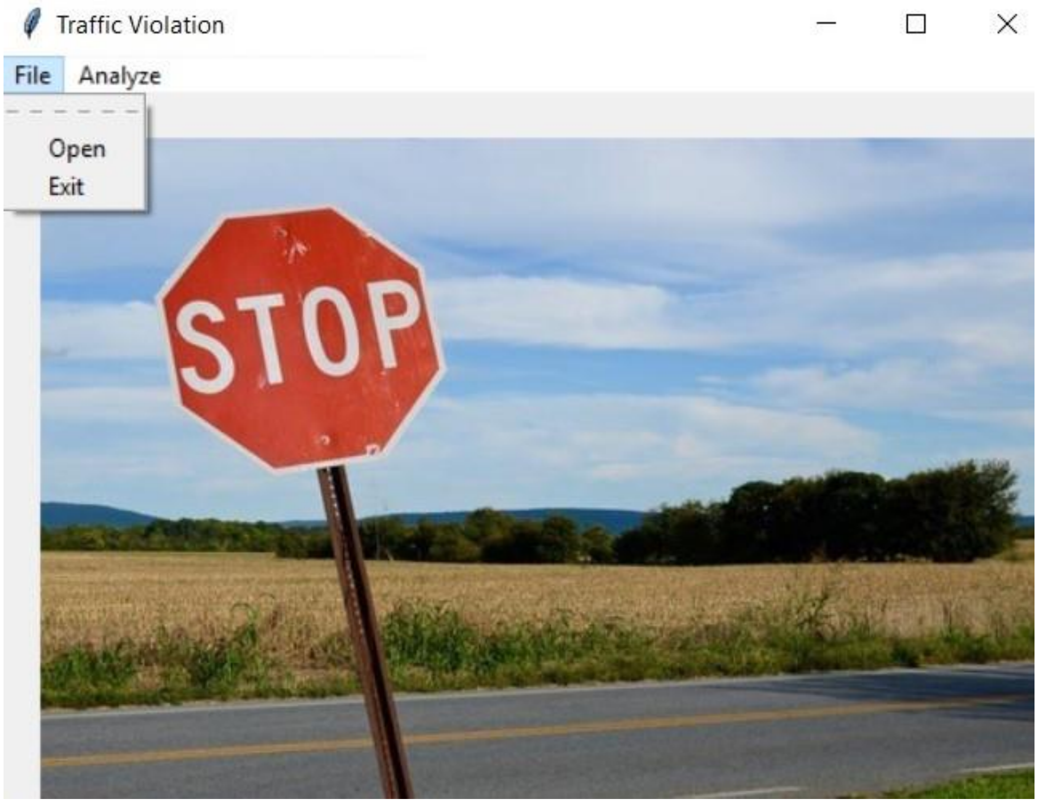


Figure-2: Initial user interface view.

Primarily, for the start of the project usage, the administrator needs to open a video footage using 'Open' item that can be found under 'File' (Figure-2). The administrator can open any video footage from the storage files (Figure-3).

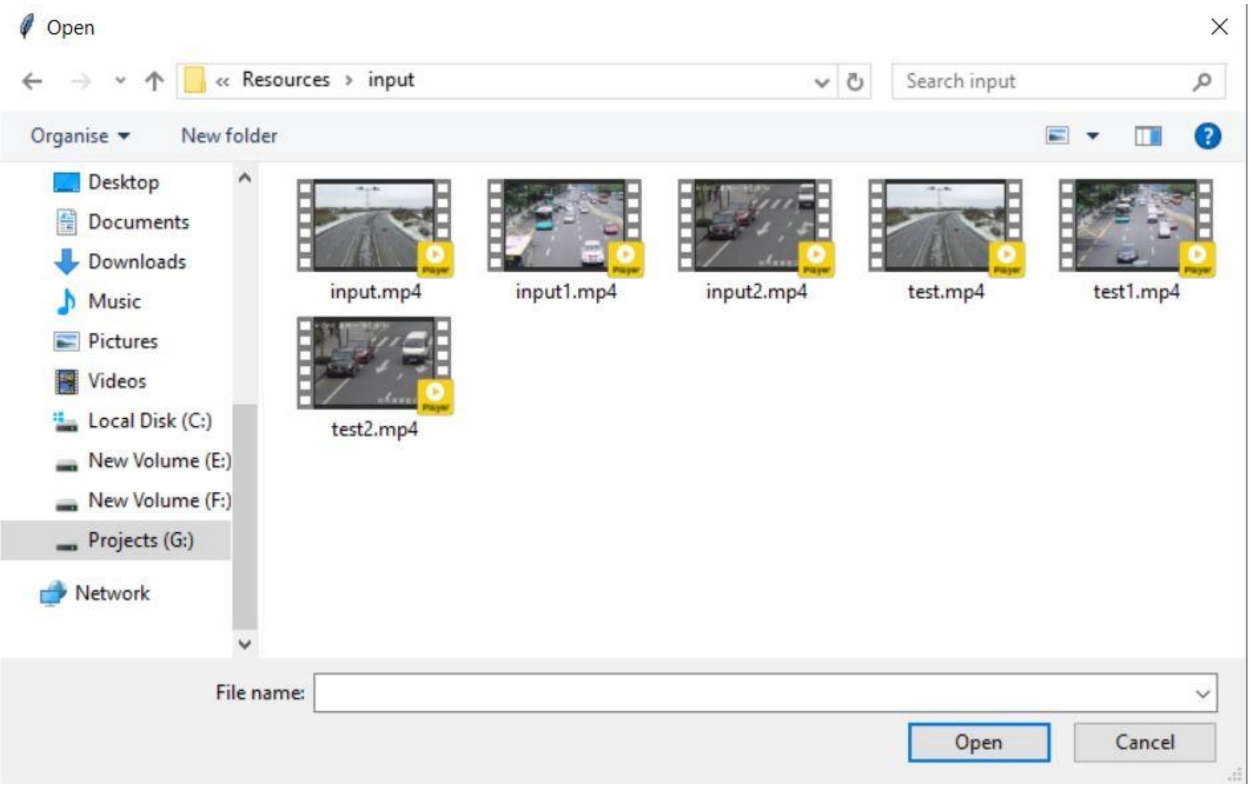


Figure-3: Opening a video footage from storage.

After opening a video footage from storage, the system will get a preview of the footage. The preview contains a frame from the given video footage. The preview is used to identify roads and draw a traffic line over the road. The traffic line drawn by administrator will act as a traffic signal line. To enable the line drawing feature, we need to select 'Region of interest' item from the 'Analyze' option (Figure-4). After that administrator will need to select two points to draw a line that specifies traffic signal.

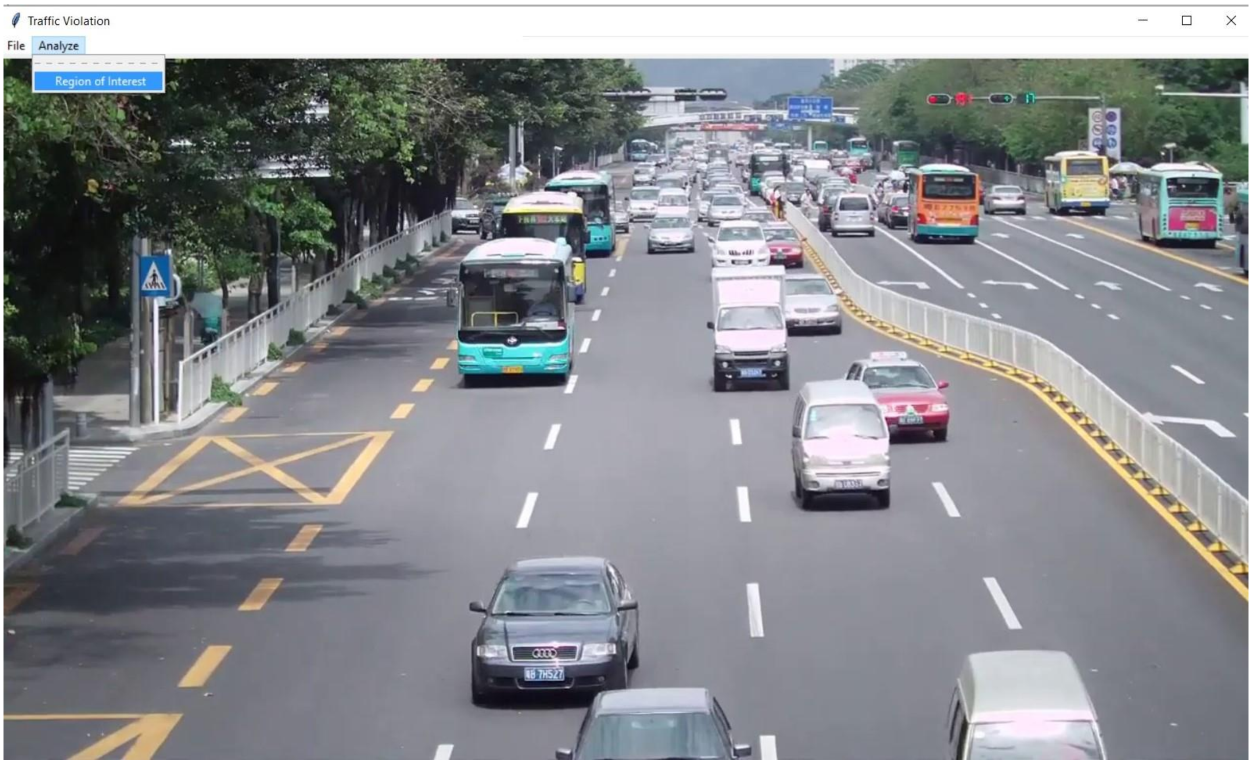
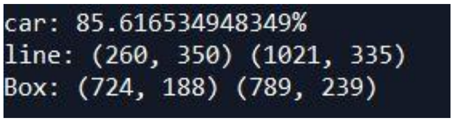


Figure-4: Region of Interest (Drawing signal line)

Selecting the region of interest will start violation detection system. The coordinates of the line drawn will be shown on console (Figure-5). The violation detection system will start immediately after the line is drawn. At first the weights will be loaded. Then the system will detect objects and check for violations. The output will be shown frame by frame from the GUI (Figure-6).



```
car: 85.616534948349%
line: (260, 350) (1021, 335)
Box: (724, 188) (789, 239)
```

Figure-5: Line Coordinates (from console)

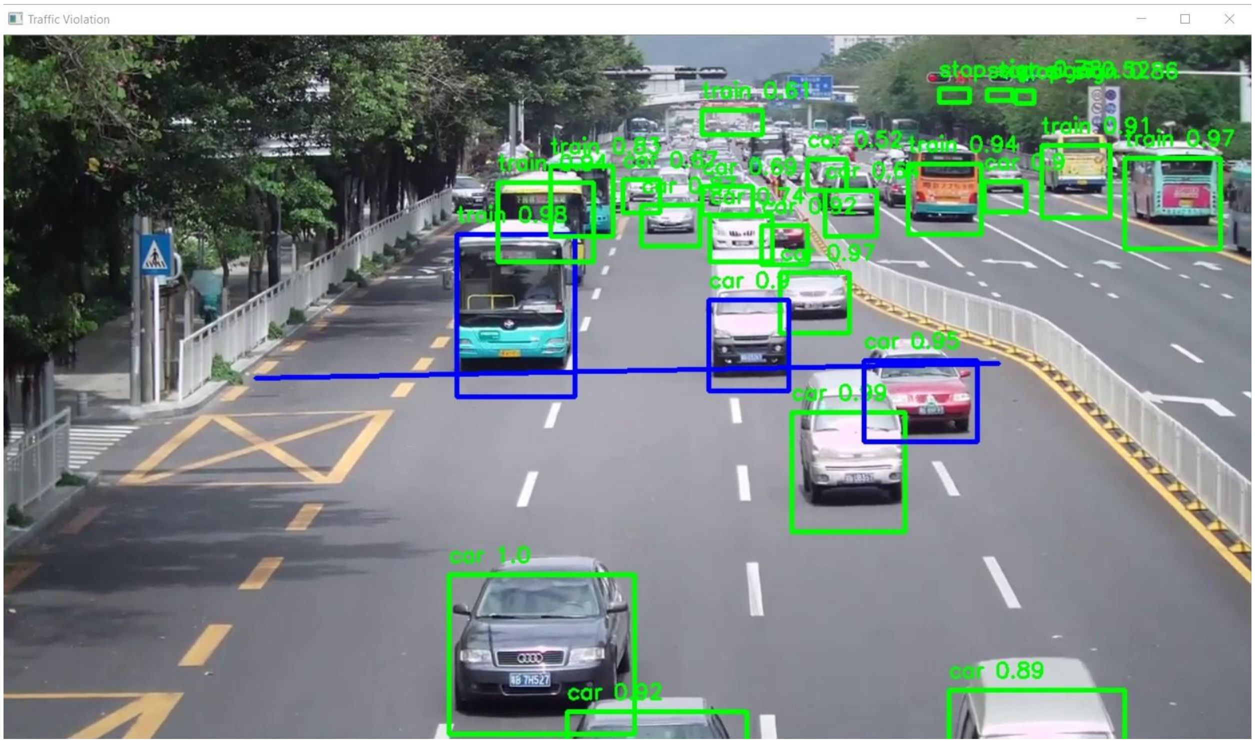


Figure-6: Final Output (on each frame)

The system will show output until the last frame of the footage. In background a ‘output.mp4’ will be generated. The file will be in ‘output’ folder of ‘Resources’. The process will be immediately terminated by clicking ‘q’.

After processing a video footage, the administrator can add another video footage from the initial file manager (Figure-2). If the work is complete the administrator can quit using ‘Exit’ item from File option.

**Libraries used for Graphical User Interface:**

- Tkinter

**The overall flow of the software:**

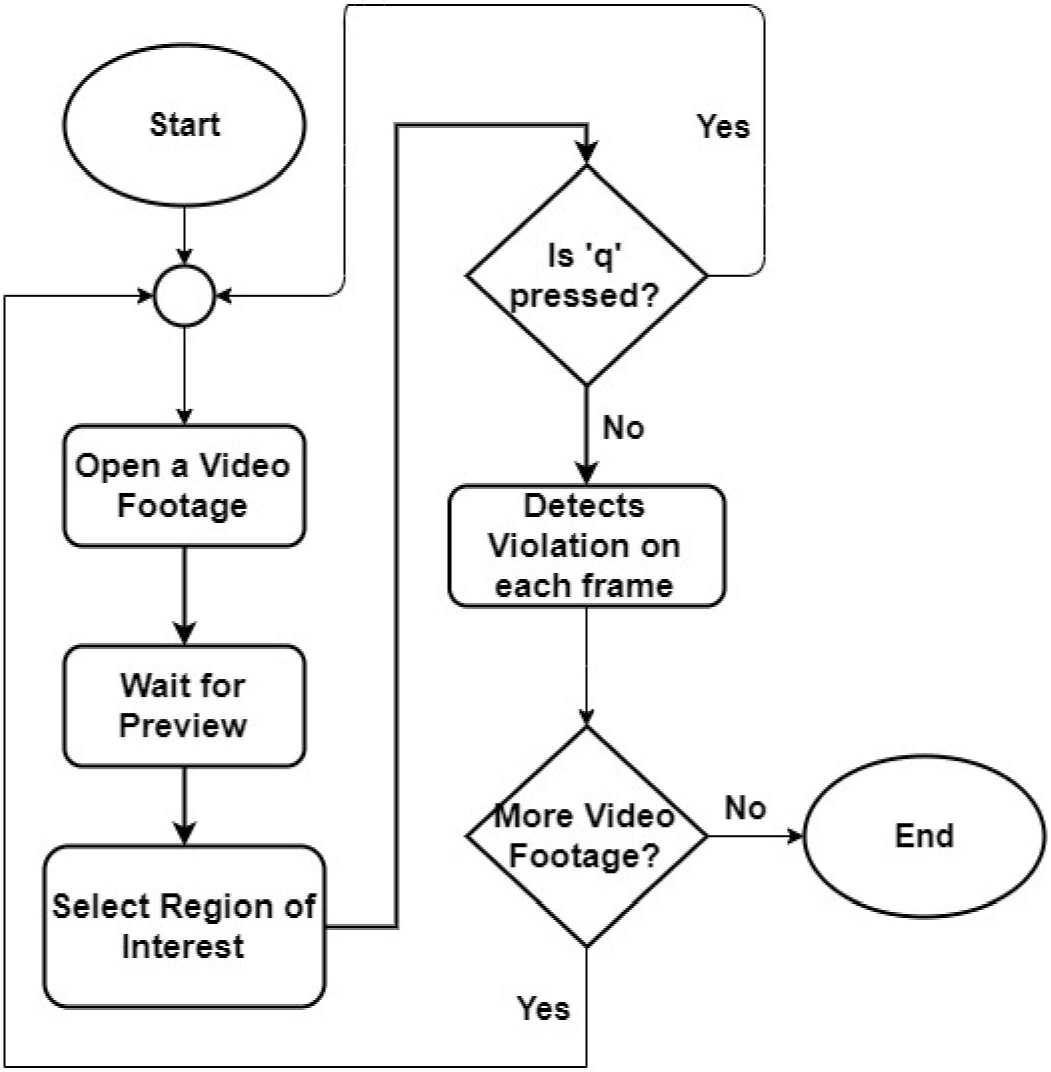


Figure-7: Overall flow of the software.

# 7 Conclusion & Recommendation

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.

# 8 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".