Mini Project report on

"TRAFFIC SIGNAL VIOLATION DETECTION SYSTEM"

A mini project dissertation submitted in partial fulfilment of the requirement for the award of degree

MASTER OF COMPUTER APPLICATIONS

by

RANJAN KISHOR, 1BY21MC040

SHARAT HEGDE, 1BY21MC049

Under the Guidance of

Mr. DWARAKANATH G V

Assistant Professor

Department of MCA
BMSIT&M
Bengaluru-560064

Department of Master of Computer Applications

BMS Institute of Technology and Management

(An autonomous Institution under VTU, Belagavi)

Bengaluru – 560064

September-2022

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An autonomous Institution under VTU, Belagavi)

Bengaluru – 560064

SEPTEMBER-2022

Department of MCA

(Accredited by NBA, New Delhi)

CERTIFICATE

This is to certify that **Mr.RANJAN KISHOR and Mr.SHARAT HEGDE** bearing **1BY21MC040 and 1BY21MC049** has successfully completed the VTU prescribed **Mini Project Work (21MCA210)** titled **TRAFFIC SIGNAL VIOLATION DETECTION SYSTEM** at **MCA R&D Centre, BMS Institute of Technology and Management, Bengaluru** under the guidance of **Mr.DWARAKANATH G V. Assistant Professor, Department of MCA** during the period from **July to September-2022**.

Signature of the Guide
Dwarakanth G V
Assistant Professor
Department of MCA
BMSIT&M
Bengaluru.

Signature of the R&D Centre Head
Dr. P. Ganesh
Professor & Head
Department of MCA
BMSIT&M
Bengaluru.

DECLARATION

We, **RANJAN KISHOR and SHARAT HEGDE** Student of 2^{nd} semester of MCA, **BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT** Yelahanka, Bangalore 560064, bearing **1BY21MC040 and 1BY21MC049,** hereby declare that the project entitled **"TRAFFIC SIGNAL VIOLATION DETECTION SYSTEM"** has been carried out by us under the supervision of guide Prof. **DWARAKANATH G V** submitted in partial fulfillment of the requirements for the Completion of 2nd semester of the Degree of Master of Computer Applications by the Visvesvaraya Technological University during the academic year 2021-2022. This report has not been submitted to any other Organization/University for any award of degree or certificate.

Name:

Date:

Signature:

ACKNOWLEDGEMENT

With immense please, We, **RANJAN KISHOR** and **SHARAT HEGDE** presenting project report of "**Traffic Signal Violation Detection System**" as part of the curriculum "System Development Project". We wish to thank all the people who gave us unending support. Here by we convey our sincere thanks for all of them.

We take this opportunity to express my gratitude to people who had been instrumental in the successful completion of this project. We are thankful to our management trustee for providing us an opportunity to work and complete the project successfully.

We wish to express our thanks to our Principal **Dr. Mohan Babu G N** for his support to the project work. We would like to acknowledge my gratitude to our HOD of MCA. **Dr. P. Ganesh** for his encouragement and support. Without his encouragement and guidance this project would not have materialized. The guidance and support received from our guides **Prof**. **Dwarakanath G V** who contributed to this project, was vital for the success of the project. We are grateful for their constant support and help.

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An autonomous Institution under VTU, Belagavi)

Bengaluru – 560064

Department of MCA

VISION

To develop quality professionals in Computer Applications who can provide sustainable solutions to the societal and industrial needs.

MISSION

Facilitate effective learning environment through quality education, state-of-the-art facilities, and orientation towards research and entrepreneurial skills.

Programme Educational Objectives (PEOs)

PEO 1: Develop innovative IT applications to meet industrial and societal needs.

PEO 2: Adapt themselves to changing IT requirements through life-long learning.

PEO 3: Exhibit leadership skills and advance in their chosen career.

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An autonomous Institution under VTU, Belagavi)

Bengaluru – 560064

Department of MCA

Programme Outcomes (POs)

PO 1: Apply knowledge of computing fundamentals, computing specialization, mathematics and domain knowledge to provide IT solutions.

PO 2: Identify, analyse and solve IT problems using fundamental principles of mathematics and computing sciences.

PO 3: Design, Develop and evaluate software solutions to meet societal and environmental concerns.

PO 4: Conduct investigations of complex problems using research-based knowledge and methods to provide valid conclusions.

PO 5: Select and apply appropriate techniques and modern tools for complex computing activities.

PO 6: Understand professional ethics, cyber regulations and responsibilities.

PO 7: Involve in life-long learning for continual development as an IT professional.

PO 8: Apply and demonstrate computing and management principles to manage projects in multidisciplinary environments by involving in different roles.

PO 9: Comprehend & write effective reports and make quality presentations.

PO 10: Understand the impact of IT solutions on socio-environmental issues.

PO 11: Work collaboratively as a member or leader in multidisciplinary teams.

PO 12: Identify potential business opportunities and innovate to create value for the society and seize that opportunity.

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An autonomous Institution under VTU, Belagavi)

Bengaluru – 560064

Department of MCA

Course Outcomes (COs)

CO 1: Analyse the given requirements.

CO 2: Design a suitable system model.

CO 3: Develop the solution using appropriate tools.

CO 4: Prepare effective documentation.

CO 5: Involve in team work.

ABSTRACT

The number of new vehicles on the road is increasing rapidly, which in turn causes highly congested roads and serving as a reason to break traffic rules by violating them. This leads to a high number of road accidents. The number of road accidents and even death rate is increasing. As per the report, about 300,000 accidents take place yearly, and over 80,000 people die due to accidents. Hence due to this problem, we decide to develop this project, our project monitors the vehicles and fines the culprit vehicle whenever the signal is violated Traffic violation detection systems using computer vision are a very efficient tool to reduce traffic violations. Traffic violation detection systems using computer vision are a very efficient tool to reduce traffic violations by tracking and Penalizing. The proposed system was implemented using YOLOV3 object detection for traffic violation detections such as signal jump, vehicle speed, and the number of vehicles. Further, the system is optimized in terms of accuracy. Using the Region of interest and location of the vehicle in the duration of frames, determining signal jump, When the traffic signal is red cars should stop beyond that limit line if they failed it is considered as a traffic violation. It checks if the signal is red and using camera module checks if any car crosses the limit line it records the video of the vehicle that how it violating the signal.

CONTENT TABLE

1. INTRODUCTION

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

1.1 Objective of the project

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.

1.2 Scope of the project

- 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.
- It enables the wireless transmission of data and its data can be accessed anywhere at any time.
- Detects violations faster than humans.

1.3 Methodology

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.

2. LITERATURE SURVEY

2.1 Literature Review

The number of new vehicles on the road is increasing rapidly, which in turn causes highly congested roads and serving as a reason to break traffic rules by violating them. This leads to a high number of road accidents. Traffic violation detection systems using computer vision are a very efficient tool to reduce traffic violations by tracking and Penalizing. The proposed system was implemented using YOLOV3 object detection for traffic violation detections such as signal jump, vehicle speed, and the number of vehicles. Further, the system is optimized in terms of accuracy. Using the Region of interest and location of the vehicle in the duration of frames, determining signal jump. This implementation obtained an accuracy of 97.67% for vehicle count detection and an accuracy of 89.24% for speed violation detection [1].

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].

The paper speaks about traffic violation detection, which is the most happening topic where the existed system is being automated, or we can say that machines do all the work which includes automatically detecting the vehicle and their violation. Recording of the traffic will be collected through CCTV footages and then violation is detected by the system. Then that clip's image will be displayed showing the violation. This paper discusses detection of the violation that is specifically done by the algorithm that is, Genetic Algorithm. Genetic algorithm is used to optimize input given to machine and that could be the records set gathered from the CCTV footage. These inputs can be similarly transformed into frames (it is far one of the many nonetheless photos which composes the entire transferring picture).

Next step is the background subtraction which allows to take the ones frames as inputs and offer pictures foreground to be extracted for similarly processing [3].

Genetic set of rules can be used for the following step to be able to assist us to validate whether a contravention has occurred or not. The most important purpose is to perform via the advent of Genetic set of rules, that complements the enter set of rules with using genetic ideas so one can produce the “fittest” set of rules. Before the detection, background subtraction is done to get the frames from the video, then detecting the vehicles is important so the Haar technology is used, which is mostly used to get the xml files to use in the system [4].

Traffic rules violations have been so normal that we don't even consider them a problem. Most of us just pay a bit amount to the police officer whenever we get caught in action, and it's not something big. But not everyone is turning a blind eye toward these concerns. One of the major cities of India has implemented a smart traffic violation detection system that can detect the speed of the Two-wheeler and four-wheelers and sends them an overspeeding e-memo. Along with that, the system also detects the 2-wheeler rider driving without a helmet and the 4-wheeler driver driving without a seatbelt accurately. All the drivers of Ahmedabad city who violate speeding or safety rules while driving will get the e-memo automatically without any human interference. It will minimize the chances of road accidents and enhance the riders' safety [5].

3. PROJECT REQUIREMENTS

3.1 Software Specification

- Operating System : Windows
- Editor : Anaconda (Prompt)
- Language : Python

4. SOFTWARE REQUIREMENT SPECIFICATION

4.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements

- YOLO v3
- Traffic Signal Video
- Python 3
- Ide/Prompt which supports python (VS Code/Anaconda/Py Charm)
- Tkinter module for GUI interface
- Open CV
- Video Player

4.2 Non-Functional Requirements:

Non-functional requirements are required, which specify criteria that can be used to judge the operation of a system, rather than specification behaviors.

- Availability: The amount of time that it is operational and available for use, system I designed with a maximum uptime, so the system is highly available.
- Flexibility: System is planned in such way that it must be flexibility in terms of functionality of the software thought some additional features are added in future
- Usability: Simple is the key here. The system must be simple that people like to use it, but not so complex that people avoid using it. The user must be familiar with the user interfaces and should not have problems in migrating to a new system with a new environment.
- Reliability: The system should be trustworthy and reliable in providing the functionalities. Once a user has made some changes, the changes must be made visible by the system. The changes made by the Programmer should be visible both to the Project leader as well as the Test engineer.

- Scalability: The system should be scalable enough to add new functionalities at a later stage. There should be a common channel, which can accommodate the new functionalities.
- Maintainability: The system monitoring and maintenance should be simple and objective in its approach. There should not be too many jobs running on different machines such that it gets difficult to monitor whether the jobs are running without errors.

4. ANALYSIS AND DESIGN

5.1 System Design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as " the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

5.2 Use Case Diagram

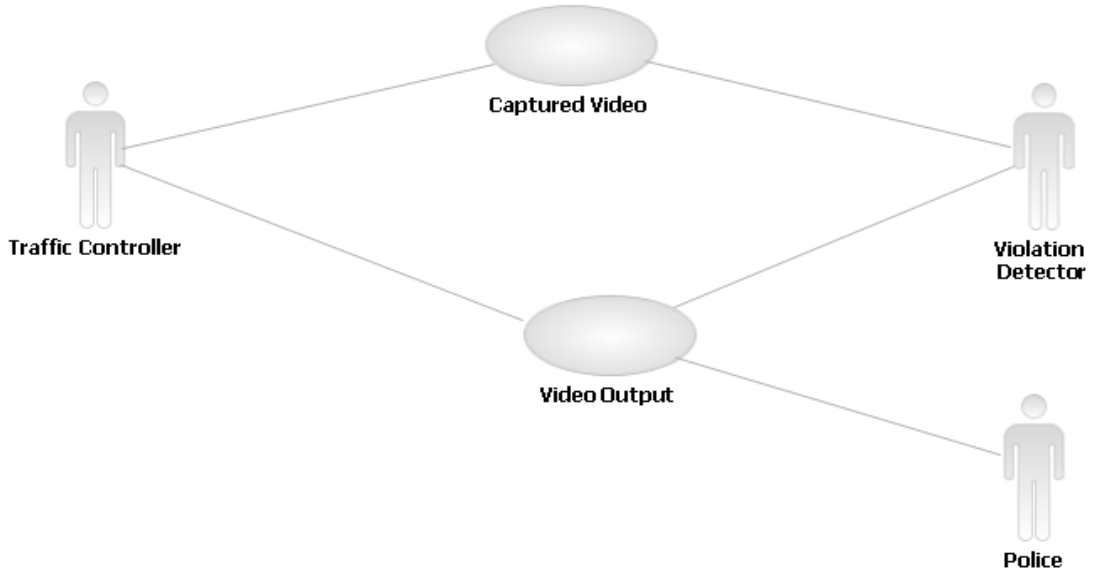
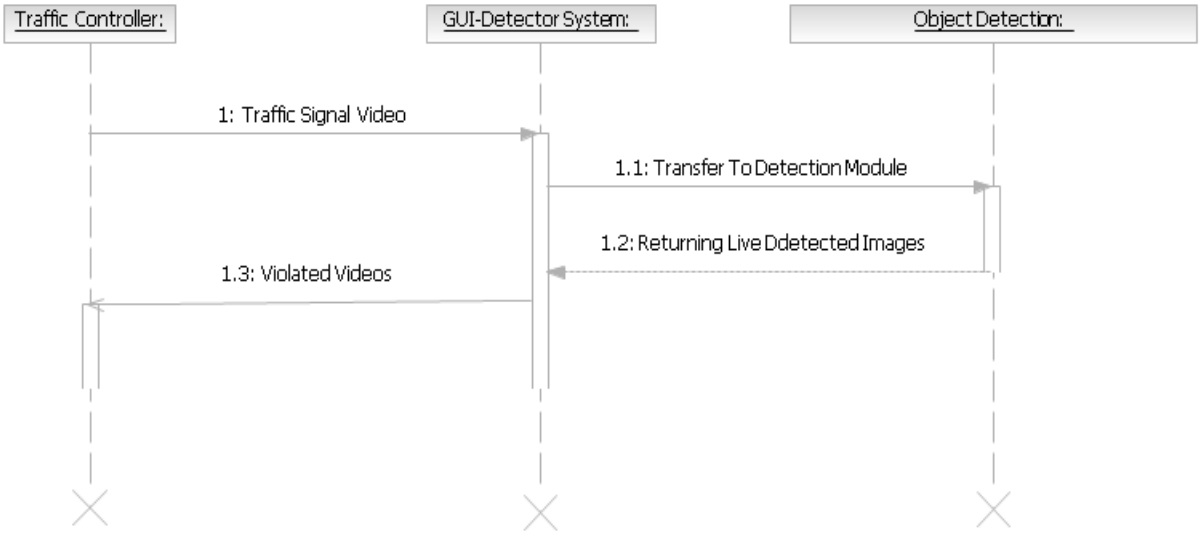
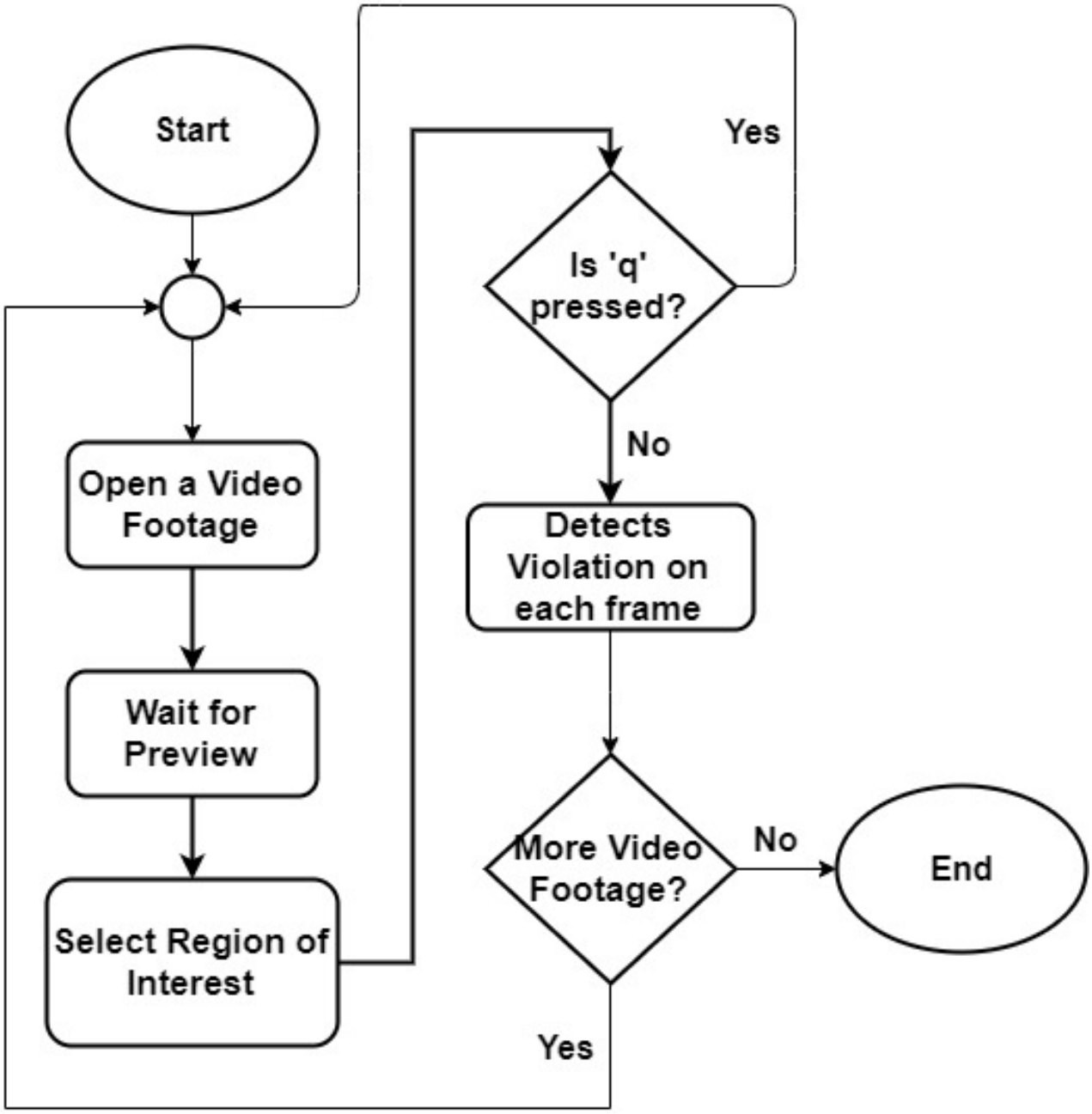


Fig.UseCase Diagram

5.3 Sequence Diagram



5.4 Data Flow Diagram



5. IMPLEMENTATION

6.1 Computer Vision

We humans have a sense of vision, due to which we can see and perceive different objects, we can traverse and man oeuvre an area, we can perform different tasks, we can take decisions, and what not? This makes us intelligent, autonomous and self-controlled. But let's take a machine, or a robot. They are man-made and not living things, and hence do not have a sense of vision in them, unless we impart it to them.

Inanimate objects like machines and robots are made artificially, some are manually controlled, whereas some are autonomous and take decisions on their own. Unlike we human beings, who have a very complex neural network in our brain which gives us complex decision-making power, robots are completely blank. It's we who need to construct and program these robots or computers in such a way that they can 'see'. And this is what we call as Artificial Intelligence, commonly known as AI. Computer Vision is, unsurprisingly, the way and method by which we impart vision capabilities to machines and robots. As per Wikipedia, Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information. Most of the concepts remains the same in artificial intelligence as well, and where there is vision their intelligence and automation is possible as well.

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.

COMPUTER VISION CODE

```
cv2.line(image, line[0], line[1], (255, 0, 0), 3)

        if tf:

            cv2.rectangle(image, (box.xmin,box.ymin), (box.xmax,box.ymax), (255,0,0), 3)

            cimg = image[box.ymin:box.ymax, box.xmin:box.xmax]

            cv2.imshow("violation", cimg)

            cv2.waitKey(5)

            cv2.imwrite("G:/Traffic Violation Detection/Traffic Signal Violation Detection
```

```
            System/Detected Images/violation_"+str(dcnt)+".jpg", cimg)
            dcnt = dcnt+1
        else:
            cv2.rectangle(image, (box.xmin,box.ymin), (box.xmax,box.ymax), (0,255,0), 3)
        cv2.putText(image,
                    label_str + ' ' + str(round(box.get_score(), 2)),  (box.xmin, box.ymin - 13),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1e-3 * image.shape[0],
                    (0,255,0), 2)
    return image
```

TENSORFLOW:

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.

TENSORFLOW CODE:

```
if conv['stride'] > 1: x = ZeroPadding2D(((1,0),(1,0)))(x) # peculiar padding as darknet prefer left and top
        x = Conv2D(conv['filter'],
                   conv['kernel'],
                   strides=conv['stride'],
                   padding='valid' if conv['stride'] > 1 else 'same', # peculiar padding as darknet prefer left and top
                   name='conv_' + str(conv['layer_idx']),
                   use bias=False if conv['bnorm'] else True) (x)
```

```
if conv['bnorm']: x = Batch Normalization(epsilon=0.001, name='bnorm_' +  str(conv['layer_idx']))(x)
    if conv['leaky']: x = LeakyReLU(alpha=0.1, name='leaky_' + str(conv['layer_idx']))(x)
```

6.2 GRAPHICAL USER INTERFACE

There are a variety of university-level computer-human interaction programs. Although a few offers breath and diversity, many students graduate from universities that offer only one or two courses. As such, most students have a limited background in the various areas. This article offers a general overview in one area, graphical user interfaces (GUI). A GUI allows a computer user to move from application to application. A good GUI makes an application easy. As well as its play an important role in the digital world as they allow users to interact with electronic devices in many different sectors (such as smartphones, home appliances, medical devices, vehicles) through design elements such as icons and menus. Their increasing importance and value as strategic assets for companies, and the relevance of design rights for protecting the visual appearance of GUIs, is reflected by the strong growth in GUI design applications worldwide in recent years.

Both a practical tool for business and a source of policy recommendations, the ICC report helps businesses to identify issues to consider when developing a GUI design filing strategy, and suggests issues which policy makers should address to make design protection for GUIs more effective and efficient. Drawing from the contributions of experts from different regions, the report reviews the requirements and practices in 24 jurisdictions, with information on both prosecution and enforcement aspects of GUI designs.

Improving procedures and requirements for design protection for GUIs will have a positive effect on this increasingly important field of design and also on the protection of future generations of digital designs.

GRAPHICAL USER INTERFACE CODE:

```
class Window(Frame):
  def __init__(self, master=None):
    Frame.__init__(self, master)
    self.master = master
```

```
        self.pos = []
        self.line = []
        self.rect = []
        self.master.title("GUI")
        self.pack(fill=BOTH, expand=1)
        self.counter = 0
        menu = Menu(self.master)
        self.master.config(menu=menu)
        file = Menu(menu)
        file.add_command(label="Open", command=self.open_file)
        file.add_command(label="Exit", command=self.client_exit)
        menu.add_cascade(label="File", menu=file)
        analyze = Menu(menu)
        analyze.add_command(label="Region of Interest", command=self.regionOfInterest)
        menu.add_cascade(label="Analyze", menu=analyze)
        self.filename = "Images/home.jpg"
        self.imgSize = Image.open(self.filename)
        self.tkimage =  ImageTk.PhotoImage(self.imgSize)
        self.w, self.h = (1366, 768)
        self.canvas = Canvas(master = root, width = self.w, height = self.h)
        self.canvas.create_image(20, 20, image=self.tkimage, anchor='nw')
        self.canvas.pack()
    def open_file(self):
        self.filename = filedialog.askopenfilename()
        cap = cv2.VideoCapture(self.filename)
        reader = imageio.get_reader(self.filename)
        fps = reader.get_meta_data()['fps']
        ret, image = cap.read()
        cv2.imwrite('C:/project/Traffic-Signal-Violation-Detection-System-master/Images/preview.jpg',
image)
        self.show_image('C:/project/Traffic-Signal-Violation-Detection-System-
master/Images/preview.jpg')
```

```
    def show_image(self, frame):
        self.imgSize = Image.open(frame)
        self.tkimage =  ImageTk.PhotoImage(self.imgSize)
        self.w, self.h = (1366, 768)
        self.canvas.destroy()
        self.canvas = Canvas(master = root, width = self.w, height = self.h)
        self.canvas.create_image(0, 0, image=self.tkimage, anchor='nw')
        self.canvas.pack()
    def regionOfInterest(self):
        root.config(cursor="plus")
        self.canvas.bind("<Button-1>", self.imgClick)
    def client_exit(self):
        exit()
    def imgClick(self, event):
        if self.counter < 2:
            x = int(self.canvas.canvasx(event.x))
            y = int(self.canvas.canvasy(event.y))
            self.line.append((x, y))
            self.pos.append(self.canvas.create_line(x - 5, y, x + 5, y, fill="red", tags="crosshair"))
            self.pos.append(self.canvas.create_line(x, y - 5, x, y + 5, fill="red", tags="crosshair"))
            self.counter += 1

        if self.counter == 2:
            #unbinding action with mouse-click
            self.canvas.unbind("<Button-1>")
            root.config(cursor="arrow")
            self.counter = 0
            #show created virtual line
            print(self.line)
            print(self.rect)
            img = cv2.imread('C:/project/Traffic-Signal-Violation-Detection-System-
master/Images/preview.jpg')
```

```
            cv2.line(img, self.line[0], self.line[1], (0, 255, 0), 3)
            cv2.imwrite('C:/project/Traffic-Signal-Violation-Detection-System-master/copy.jpg', img)
            self.show_image('C:/project/Traffic-Signal-Violation-Detection-System/copy.jpg')
        self.main_process()
            print("Executed Successfully!!!")
            #clearing things
            self.line.clear()
            self.rect.clear()
            for i in self.pos:
                self.canvas.delete(i)
        def intersection(self, p, q, r, t):
            print(p, q, r, t)
            (x1, y1) = p
            (x2, y2) = q

            (x3, y3) = r
            (x4, y4) = t

            a1 = y1-y2
            b1 = x2-x1
            c1 = x1*y2-x2*y1

            a2 = y3-y4
            b2 = x4-x3
            c2 = x3*y4-x4*y3

            if(a1*b2-a2*b1 == 0):
                return False
            print((a1, b1, c1), (a2, b2, c2))
            x = (b1*c2 - b2*c1) / (a1*b2 - a2*b1)
            y = (a2*c1 - a1*c2) / (a1*b2 - a2*b1)
            print((x, y))
```

```
        if x1 > x2:
            tmp = x1
            x1 = x2
            x2 = tmp
        if y1 > y2:
            tmp = y1
            y1 = y2
            y2 = tmp
        if x3 > x4:
            tmp = x3
            x3 = x4
            x4 = tmp
        if y3 > y4:
            tmp = y3
            y3 = y4
            y4 = tmp

        if x >= x1 and x <= x2 and y >= y1 and y <= y2 and x >= x3 and x <= x4 and y >= y3 and y <= y4:
            return True
        else:
            return False

    def main_process(self):

        video_src = self.filename

        cap = cv2.VideoCapture(video_src)

        reader = imageio.get_reader(video_src)
        fps = reader.get_meta_data()['fps']
        writer = imageio.get_writer('C:/project/Traffic-Signal-Violation-Detection-System-
master/Resources/output/output.mp4', fps = fps)
```

```
        j = 1
        while True:
            ret, image = cap.read()

            if (type(image) == type(None)):
                writer.close()
                break

            image_h, image_w, _ = image.shape
            new_image = od.preprocess_input(image, od.net_h, od.net_w)

            # run the prediction
            yolos = od.yolov3.predict(new_image)
            boxes = []

            for i in range(len(yolos)):
                # decode the output of the network
                boxes += od.decode_netout(yolos[i][0], od.anchors[i], od.obj_thresh, od.nms_thresh,
od.net_h, od.net_w)

            # correct the sizes of the bounding boxes
            od.correct_yolo_boxes(boxes, image_h, image_w, od.net_h, od.net_w)

            # suppress non-maximal boxes
            od.do_nms(boxes, od.nms_thresh)

            # draw bounding boxes on the image using labels
            image2 = od.draw_boxes(image, boxes, self.line, od.labels, od.obj_thresh, j)

            writer.append_data(image2)

            # cv2.imwrite('E:/Virtual Traffic Light Violation Detection System/Images/frame'+str(j)+'.jpg',
```

```
image2)
        # self.show_image('E:/Virtual Traffic Light Violation Detection
System/Images/frame'+str(j)+'.jpg')

        cv2.imshow('Traffic Violation', image2)
        print(j)
        if cv2.waitKey(10) & 0xFF == ord('q'):
            writer.close()
            break
        j = j+1
    cv2.destroyAllWindows()
root = Tk()
app = Window(root)
root.geometry("%dx%d"%(535, 380))
root.title("Traffic Violation")
root.mainloop()
```

Libraries used for Graphical User Interface:

TKINTER:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. Although Tkinter is considered the de facto Python GUI framework, it's not without criticism. One notable criticism is that GUIs built with Tkinter look outdated. If you want a shiny, modern interface, then Tkinter may not be what you're looking for. However, Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern sheen is unnecessary, and the top priority is to quickly build something that's functional and cross-platform.

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-1: 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-1). The administrator can open any video footage from the storage files (Figure-2).

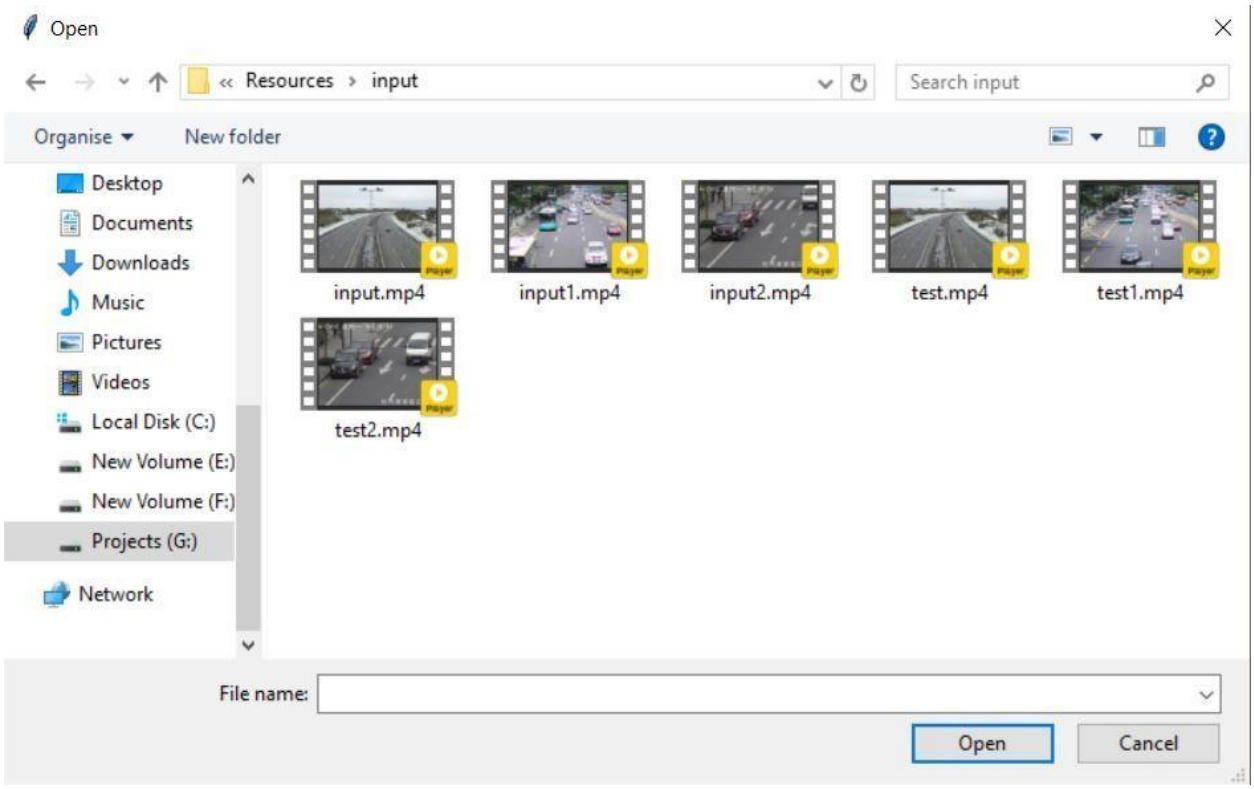


Figure-2 : Opening a video footage from storage.

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 eed to select two points to draw a line that specifies traffic signal.

Figure-3: Region of Interest (Drawing signal line)

IMAGEIO:

Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images, volumetric data, and scientific formats. It is cross-platform, runs on Python 3.5+, and is easy to install. The maintainers of imageio and thousands of other packages are working with Tide lift to deliver commercial support and maintenance for the open-source dependencies you use to build your applications. Save time, reduce risk, and improve code health, while paying the maintainers of the exact dependencies you us

IMAGEIO CODE:

```
def open_file(self):
    self.filename = filedialog.askopenfilename()

    cap = cv2.VideoCapture(self.filename)

    reader = imageio.get_reader(self.filename)
```

```
        fps = reader.get_meta_data()['fps']

        ret, image = cap.read()
        cv2.imwrite('C:/project/Traffic-Signal-Violation-Detection-System-master/Images/preview.jpg',
image)

        self.show_image('C:/project/Traffic-Signal-Violation-Detection-System-
master/Images/preview.jpg')

    def show_image(self, frame):
        self.imgSize = Image.open(frame)
        self.tkimage =  ImageTk.PhotoImage(self.imgSize)
        self.w, self.h = (1366, 768)

        self.canvas.destroy()

        self.canvas = Canvas(master = root, width = self.w, height = self.h)
        self.canvas.create_image(0, 0, image=self.tkimage, anchor='nw')
        self.canvas.pack()

    def regionOfInterest(self):
        root.config(cursor="plus")
        self.canvas.bind("<Button-1>", self.imgClick)

    def client_exit(self):
        exit()

    def imgClick(self, event):

        if self.counter < 2:
            x = int(self.canvas.canvasx(event.x))
            y = int(self.canvas.canvasy(event.y))
            self.line.append((x, y))
```

```
            self.pos.append(self.canvas.create_line(x - 5, y, x + 5, y, fill="red", tags="crosshair"))
            self.pos.append(self.canvas.create_line(x, y - 5, x, y + 5, fill="red", tags="crosshair"))
            self.counter += 1



        if self.counter == 2:
            #unbinding action with mouse-click
            self.canvas.unbind("<Button-1>")
            root.config(cursor="arrow")
            self.counter = 0
            #show created virtual line
            print(self.line)
            print(self.rect)
            img = cv2.imread('C:/project/Traffic-Signal-Violation-Detection-System-
master/Images/preview.jpg')
            cv2.line(img, self.line[0], self.line[1], (0, 255, 0), 3)
            cv2.imwrite('C:/project/Traffic-Signal-Violation-Detection-System-master /copy.jpg', img)
            self.show_image('C:/project/Traffic-Signal-Violation-Detection-System-
master/Images/copy.jpg')
```

STRUCT:

This module performs conversions between Python values and C structs represented as Python bytes objects. This can be used in handling binary data stored in files or from network connections, among other sources. It uses Format Strings as compact descriptions of the layout of the C structs and the intended conversion to/from Python values.

STRUCT CODE:

```
class WeightReader:
    def __init__(self, weight_file):
        with open(weight_file, 'rb') as w_f:
            major,    = struct.unpack('i', w_f.read(4))
            minor,    = struct.unpack('i', w_f.read(4))
            revision, = struct.unpack('i', w_f.read(4))
```

NUMPY:

Array programming provides a powerful, compact and expressive syntax for accessing, manipulating and operating on data in vectors, matrices and higher-dimensional arrays. NumPy is the primary array programming library for the Python language. It has an essential role in research analysis pipelines in fields as diverse as physics, chemistry, astronomy, geoscience, biology, psychology, materials science, engineering, finance and economics. For example, in astronomy, NumPy was an important part of the software stack used in the discovery of gravitational waves1 and in the first imaging of a black hole2. Here we review how a few fundamental array concepts lead to a simple and powerful programming paradigm for organizing, exploring and analyzing scientific data. NumPy is the foundation upon which the scientific Python ecosystem is constructed. It is so pervasive that several projects, targeting audiences with specialized needs, have developed their own NumPy-like interfaces and array objects. Owing to its central position in the ecosystem, NumPy increasingly acts as an interoperability layer between such array computation libraries and, together with its application programming interface (API), provides a flexible framework to support the next decade of scientific and industrial analysis.

NUMPY CODE:

```
def make_yolov3_model():
    input_image = Input(shape=(None, None, 3))

    # Layer  0 => 4
    x = _conv_block(input_image, [{'filter': 32, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True,
'layer_idx': 0},
                                  {'filter': 64, 'kernel': 3, 'stride': 2, 'bnorm': True, 'leaky': True, 'layer_idx': 1},
                                  {'filter': 32, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 2},
                                  {'filter': 64, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 3}])

    # Layer  5 => 8
    x = _conv_block(x, [{'filter': 128, 'kernel': 3, 'stride': 2, 'bnorm': True, 'leaky': True, 'layer_idx': 5},
                        {'filter':  64, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 6},
                        {'filter': 128, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 7}])
```

```
    # Layer  9 => 11
    x = _conv_block(x, [{'filter':  64, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 9},
                        {'filter': 128, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 10}])

    # Layer 12 => 15
    x = _conv_block(x, [{'filter': 256, 'kernel': 3, 'stride': 2, 'bnorm': True, 'leaky': True, 'layer_idx': 12},
                        {'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 13},
                        {'filter': 256, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 14}])

    # Layer 16 => 36
    for i in range(7):
        x = _conv_block(x, [{'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 16+i*3},
                            {'filter': 256, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 17+i*3}])

    skip_36 = x

    # Layer 37 => 40
    x = _conv_block(x, [{'filter': 512, 'kernel': 3, 'stride': 2, 'bnorm': True, 'leaky': True, 'layer_idx': 37},
                        {'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 38},
                        {'filter': 512, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 39}])

    # Layer 41 => 61
    for i in range(7):
        x = _conv_block(x, [{'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 41+i*3},
                            {'filter': 512, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 42+i*3}])

    skip_61 = x

    # Layer 62 => 65
    x = _conv_block(x, [{'filter': 1024, 'kernel': 3, 'stride': 2, 'bnorm': True, 'leaky': True, 'layer_idx': 62},
```

```
                {'filter':  512, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 63},
                {'filter': 1024, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 64}])

    # Layer 66 => 74
    for i in range(3):
        x = _conv_block(x, [{'filter':  512, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx':
66+i*3},
                            {'filter': 1024, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 67+i*3}])

    # Layer 75 => 79
    x = _conv_block(x, [{'filter':  512, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 75},
                {'filter': 1024, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 76},
                {'filter':  512, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 77},
                {'filter': 1024, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 78},
                {'filter':  512, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 79}],
skip=False)

    # Layer 80 => 82
    yolo_82 = _conv_block(x, [{'filter': 1024, 'kernel': 3, 'stride': 1, 'bnorm': True,  'leaky': True,
'layer_idx': 80},
                              {'filter':  255, 'kernel': 1, 'stride': 1, 'bnorm': False, 'leaky': False, 'layer_idx': 81}],
skip=False)

    # Layer 83 => 86
    x = _conv_block(x, [{'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 84}],
skip=False)
    x = UpSampling2D(2)(x)
    x = concatenate([x, skip_61])

    # Layer 87 => 91
    x = _conv_block(x, [{'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 87},
                {'filter': 512, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 88},
                {'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 89},
```

```
                {'filter': 512, 'kernel': 3, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 90},
                {'filter': 256, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True, 'layer_idx': 91}],
skip=False)

    # Layer 92 => 94
    yolo_94 = _conv_block(x, [{'filter': 512, 'kernel': 3, 'stride': 1, 'bnorm': True,  'leaky': True,
'layer_idx': 92},
                        {'filter': 255, 'kernel': 1, 'stride': 1, 'bnorm': False, 'leaky': False, 'layer_idx': 93}],
skip=False)

    # Layer 95 => 98
    x = _conv_block(x, [{'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True, 'leaky': True,   'layer_idx': 96}],
skip=False)
    x = UpSampling2D(2)(x)
    x = concatenate([x, skip_36])

    # Layer 99 => 106
    yolo_106 = _conv_block(x, [{'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True,  'leaky': True,
'layer_idx': 99},
                        {'filter': 256, 'kernel': 3, 'stride': 1, 'bnorm': True,  'leaky': True,  'layer_idx': 100},
                        {'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True,  'leaky': True,  'layer_idx': 101},
                        {'filter': 256, 'kernel': 3, 'stride': 1, 'bnorm': True,  'leaky': True,  'layer_idx': 102},
                        {'filter': 128, 'kernel': 1, 'stride': 1, 'bnorm': True,  'leaky': True,  'layer_idx': 103},
                        {'filter': 256, 'kernel': 3, 'stride': 1, 'bnorm': True,  'leaky': True,  'layer_idx': 104},
                        {'filter': 255, 'kernel': 1, 'stride': 1, 'bnorm': False, 'leaky': False, 'layer_idx': 105}],
skip=False)
```

6.3 Screenshots

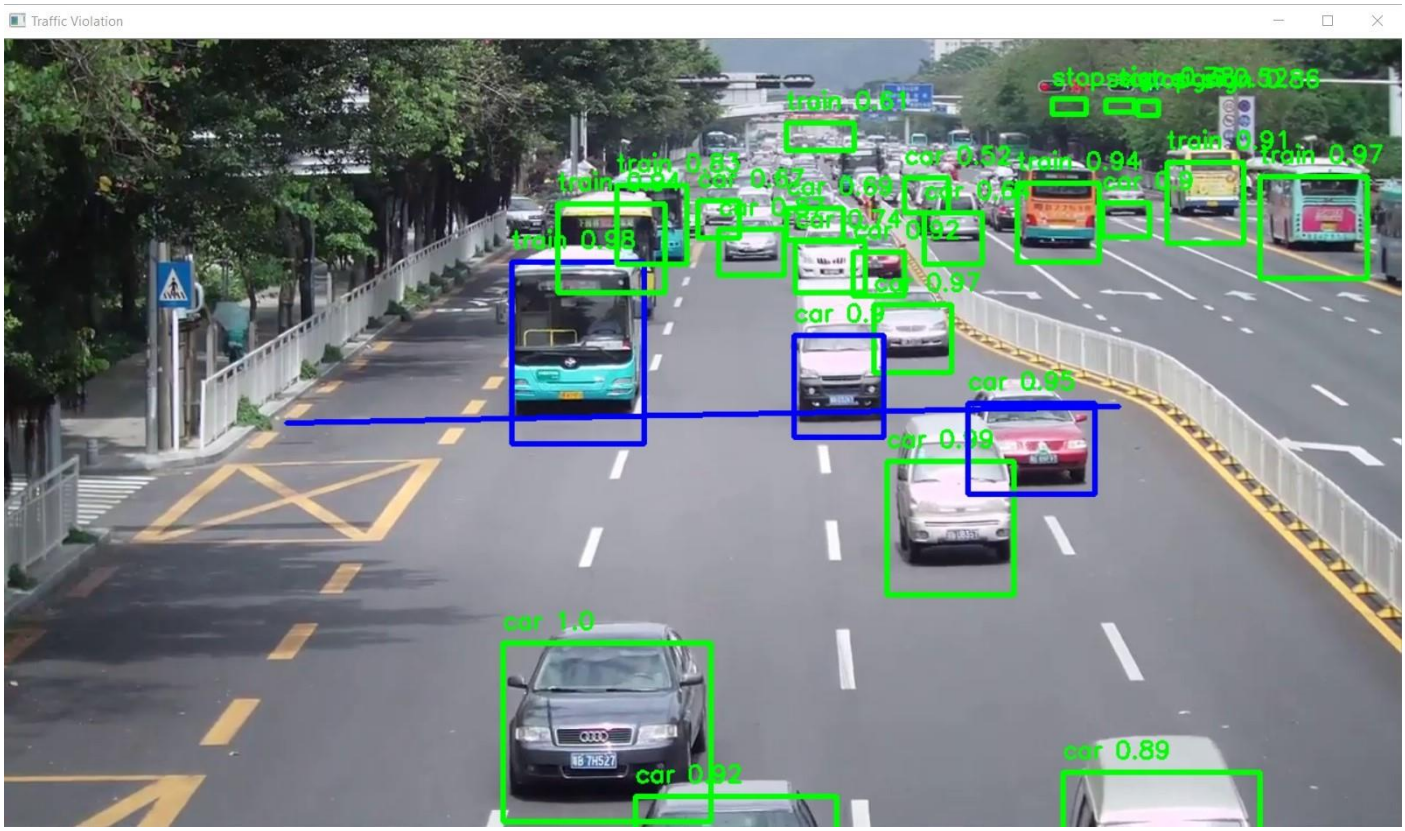


Fig.4. Violation Detecting in Frame

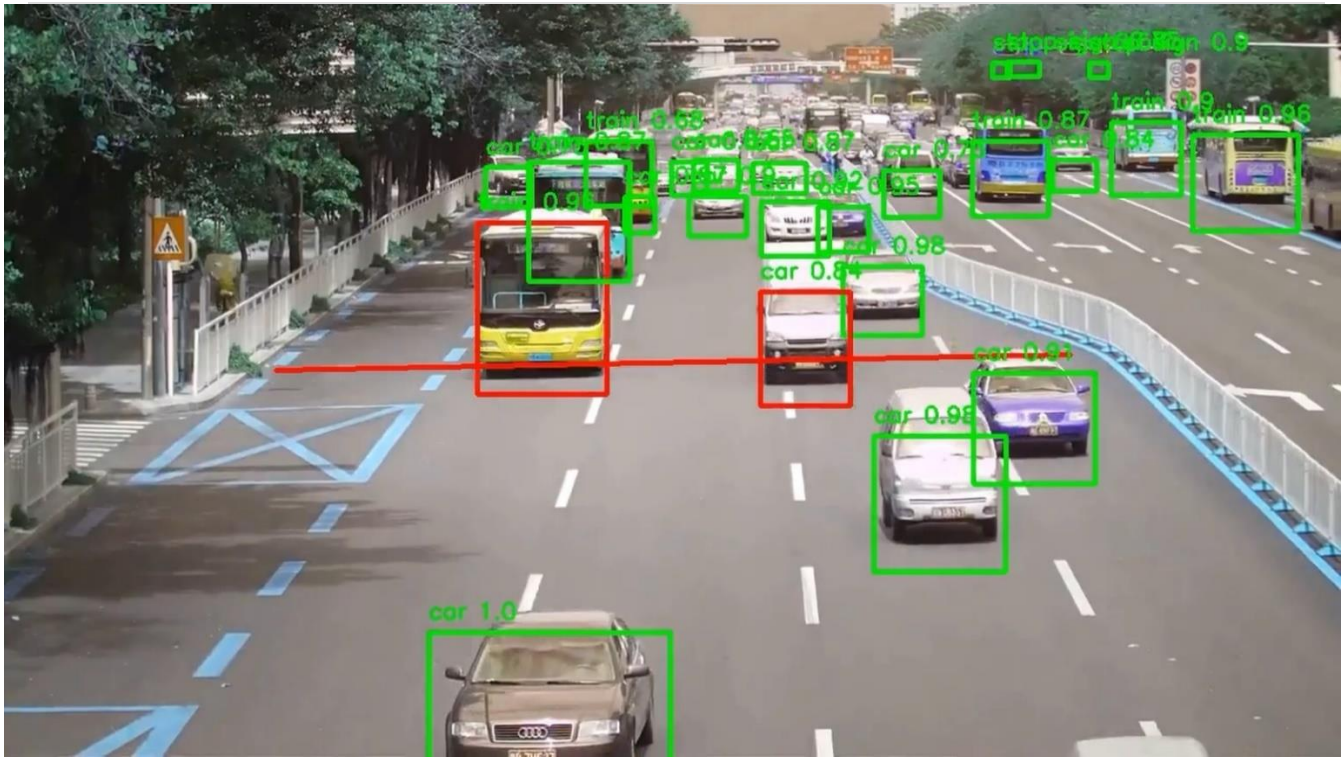


Fig.5. Violation Detected

Fig.6. Final Output

7. CONCLUSION AND FUTURE ENHANCEMENT

7.1 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.

7.2 Future Scope and Enhancement:

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. BIBLIOGRAPHY

[1] https://ieeexplore.ieee.org/document/9137873

[2] https://github.com/anmspro/Traffic-Signal-Violation-Detection-System

[3] https://www.sciencedirect.com/science/article/pii/S2666285X21000844

[4] https://www.researchgate.net/publication/261490939

[5] https://www.hyperlinkinfosystem.com/case/smart-traffic-rule-violation-detection-system