A

Major Project

On

An AI-Enabled Traffic Management System

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

M.Sathvik(187R1A05G5)

V.Guru Pavani(187R1A05H7)

V.Surya Sagar(187R1A05H9)

Under the guidance of

Dr. T. S. Mastan Rao

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**An AI-Enabled Traffic Management System**" is being submitted by **M.SATHVIK(187R1A05G5), V. GURUPAVANI(187R1A05H7) &V.SURYA SAGAR(187R1A0H9)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. T. S. MASTAN RAO
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _______________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Dr. T. S. Mastan Rao, Associate Professor for his exemplary guidance, monitoring, and constant encouragement throughout the project work. The Blessing, help, and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. M. Varaprasad Rao, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao,** and **Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, and Mrs. G. Latha** for their cordial support, valuable information, and guidance, which helped us complete this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support were received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

M.SATHVIK (187R1A05G5)
V.GURU PAVANI (187R1A05H7)
V.SURYA SAGAR (187R1A05H9)

ABSTRACT

Due to the growing population and people's need for comfort, more automobiles are being purchased, particularly in urban areas. This can result in heavy traffic, indicating that traffic violations are becoming more dangerous in every corner of the world. As a result, people's awareness decreases, and there are more accidents, which may result in the loss of many lives. The existing system has less accuracy and slow detection of violations, here we are using YOLO and OCR algorithms for object and number plate detection, these algorithms can detect the violation at high speed with good accuracy. The proposed system can detect the most common types of traffic violations in real-time through computer vision techniques and it also leverages good results with an accuracy of 88.3%. The proposed traffic violation detector can identify signal violations, and the individuals are informed that they will be apprehended if they break a traffic law. The proposed system is faster and more efficient than human, as known already traffic police is the one who captures the image of individuals violating traffic rules but the traffic police will not be able to capture more than one violation simultaneously. When compared to other algorithms YOLO is found to be more advantageous and has higher efficiency and accuracy.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS Page No.

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

[1] This project is titled "Improvised Traffic Light Control System using Deep Learning" Bikes are an extremely mainstream method of transportation in India. However, there is a high risk involved due to a lack of protection. To decrease the involved risk, it is highly desirable for motorcycle riders to use helmets. That's why the government has made it a punishable offense to ride a bike without a helmet. [2]The drawback of the current method where human intervention is required can be solved by our proposed method. Nowadays everyone is moving towards automation and many countries have implemented automatic traffic surveillance systems. Here we are developing a system in which we are using a more efficient way which helps us in getting better results.

1.2 PROJECT PURPOSE

This has been developed to facilitate in order to ensure safety measures on roads of India, the identification of traffic rule violators is a highly desirable but challenging job due to numerous difficulties such as occlusion, and illumination, etc. In this paper, we propose an end-to-end framework for the detection of violations and notifying violators.

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. [3]The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once the analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A traffic violation detection system must be realized in real-time as the authorities track the roads all the time.[4] 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 the most common three types of traffic violations in real-time which are a signal violation, parking violation, and wrong direction violation.[5] 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.

2.2 EXISTING SYSTEM

The existing system is where we use traffic police officials to manage and maintain the traffic rules system manually.[6] As it is a manual process we have less efficiency in the system to detect large numbers of vehicles moving on the road.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

1. The cost of acquiring a large number of traffic police officials
2. Less efficiency as compared to the automatic process.

2.3 PROPOSED SYSTEM

[7]In this system we are going to implement an automatic traffic rule system. This system can detect the most common three types of traffic violations in real-time which are a 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.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- As it is an automated process the efficiency of detecting more moving vehicles will be increased.
- cost-effective as compared to the manual process where a team of traffic officials is maintained.
- Fast process as we are using automated systems to detect and raise a ticket for the violator.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. [8]During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- TechnicalFeasibility
- SocialFeasibility
- EconomicFeasibility
-

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give the best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during the preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, which gives an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIOURAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The Following are some hardware requirements.

- Processor : Intel Dual Core@ CPU 2.90GHz.
- Harddisk : 16GB.
- RAM : 4 GB.
- Monitor : 5 inches or above.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specify the logical characteristics of each interface and software component of the system. The following are some software requirements,

- Operating System : Windows 8,10
- Languages : Python
- IDE : Pycharm

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECTARCITECTURE

This project architecture shows the flow of extraction from the user to the website to the user again. Here it also explains the access rights of different individuals such as user and admin.

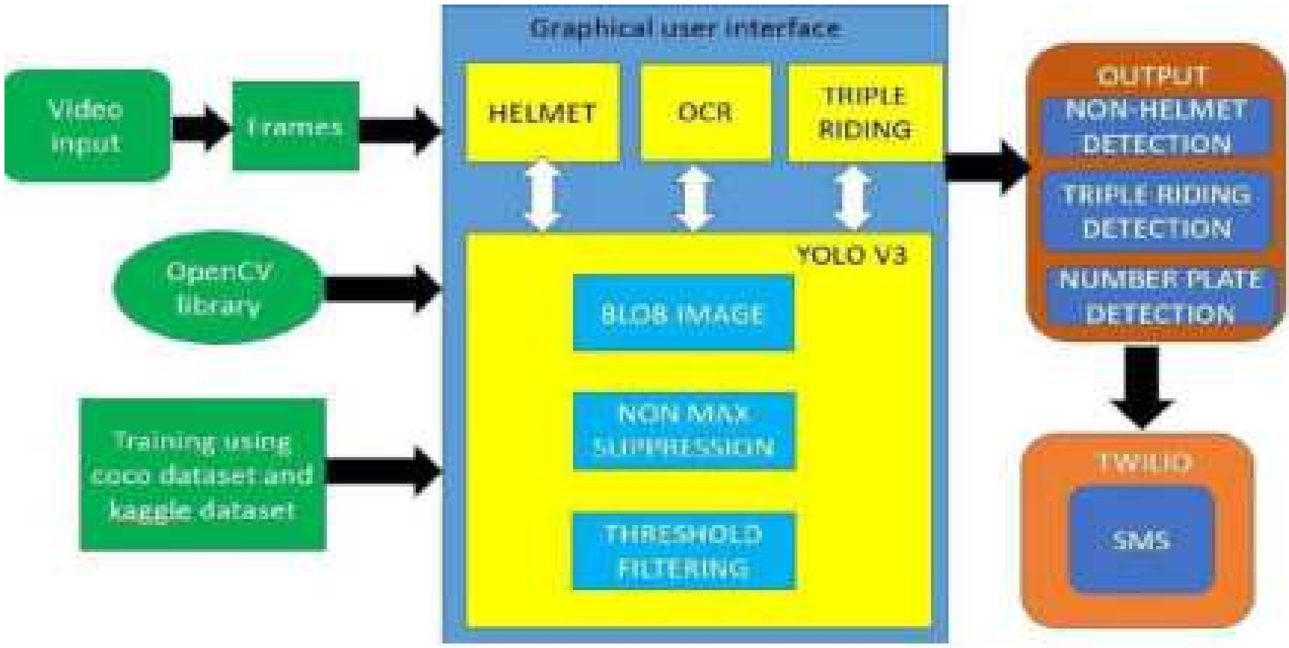


Figure 3.1: Project Architecture of Automated Traffic rules violation system

3.2 DESCRIPTION

Video input: The video input is the extracted video from the real world traffic and given to the system which divides it into frames.

OpenCV: OpenCV libraries are used to perform different machine learning algorithms on the video input.

Training dataset: The training data set is the images on which the machine learning system gets trained to identify the traffic rules violation.

Graphical user interface: The graphical user interface is used to operate different traffic violations by the administrator, yolo is the algorithm used for detection.

Output: It is where the information about the violated vehicle using the input video.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the administrator. The user has the right to log in, access resources, and view ticket details. Whereas the administrator has the login, access to resources of the users, and also the right to update and remove the tickets raised.

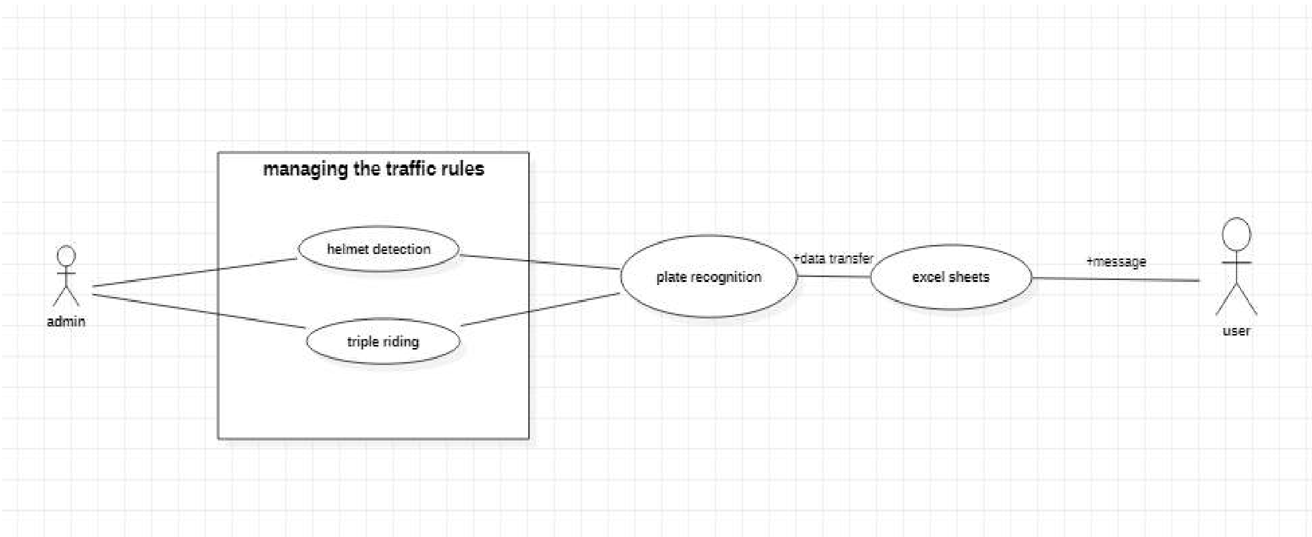


Figure 3.2: Use Case Diagram for an Automated Traffic rules violation system

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects

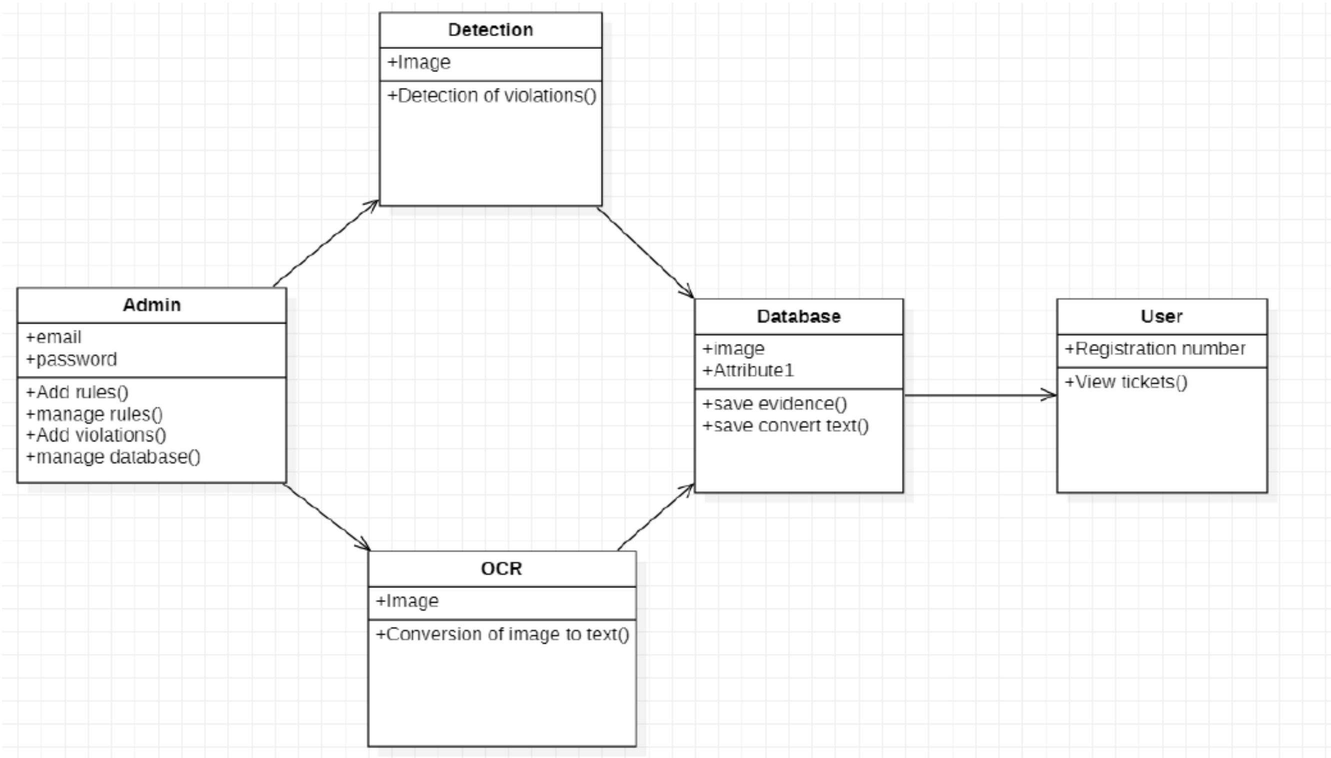


Figure 3.3: Class Diagram for an Automated Traffic rules violation system

3.5 SEQUENCE DIAGRAM

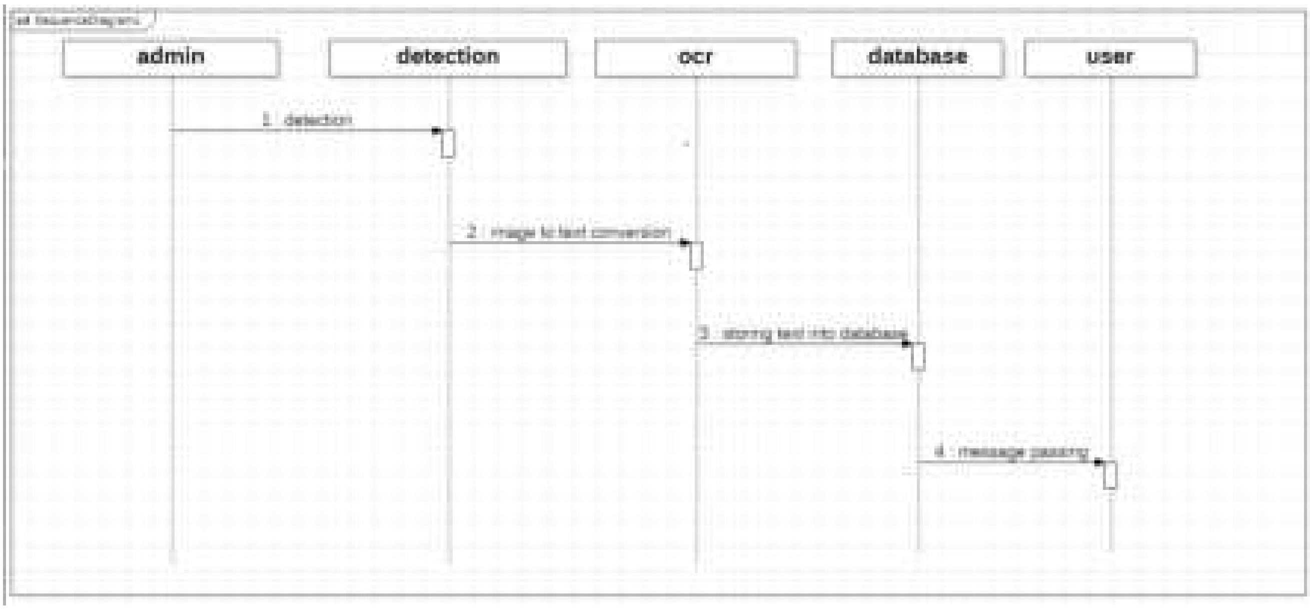


Figure 3.4: Sequence Diagram for an Automated Traffic rules violation system

3.6 ACTIVITY DIAGRAM

It describes the flow of activity states.

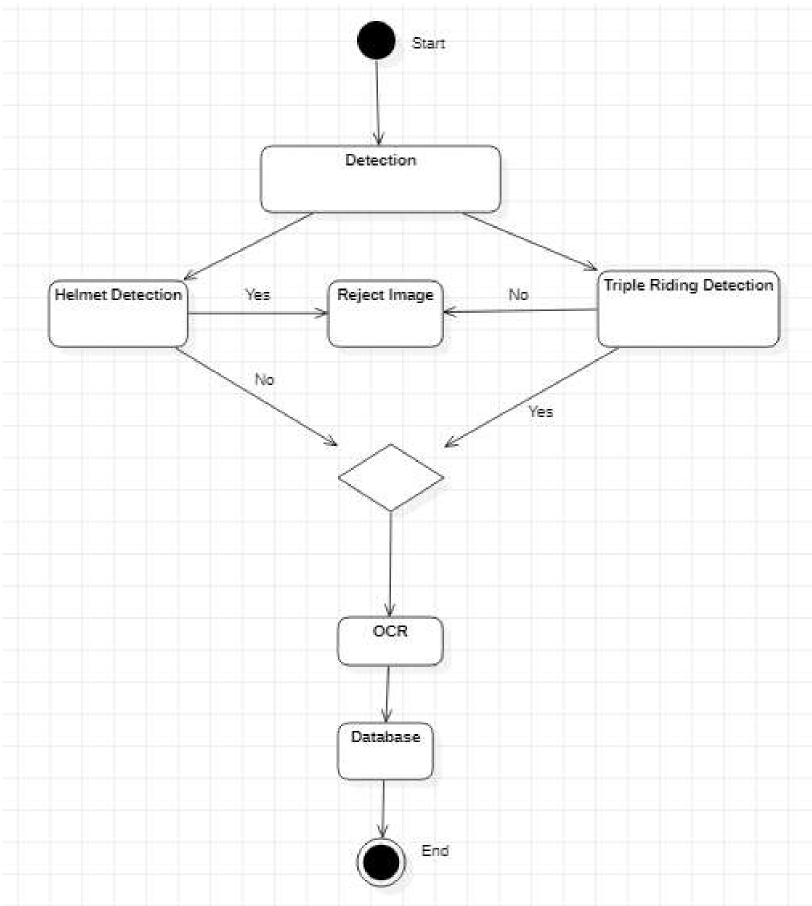


Figure 3.5: Activity Diagram for an Automated Traffic rules violation system

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import *
from tkinter import messagebox
top = Tk()

C = Canvas(top, bg="blue", height=250, width=300)
filename = PhotoImage(file = "resized_test.png")
background_label = Label(top, image=filename)
background_label.place(x=0, y=0, relwidth=1, relheight=1)

C.pack()

from tkinter import *
window=top
root=window
window.title("project1")
window.geometry("3500x900")
window.configure(background="#E4287C")
lbl=Label(window, text="Triple Ride and Helmet Detection" ,fg="black"  ,width=25
,height=1,font=('times', 30, 'italic bold underline'))
lbl.place(x=400,y=10)
import os
def time():
    os.startfile("yolo_detection_webcam1.py")
def time1():
    os.startfile("Helmet_detection_YOLOV3.py")
def time2():
    os.startfile("yolo_detection_images.py")
def time3():
    os.startfile("yolo_detection_webcam.py")
def time4():
    os.startfile("helmet.py")
def time5():
    os.startfile("live1.py")
def time6():
    os.startfile("yolo_detection_images4.py")
def time8():
    try:
        import numpy as np
        import cv2
        import cv2 as cv

        confidenceThreshold = 0.0
        NMSThreshold = 0.6

        modelConfiguration = 'cfg/yolov3.cfg'
        modelWeights = 'yolov3.weights'
```

```
labelsPath = 'coco.names'
labels = open(labelsPath).read().strip().split('\n')
np.random.seed(10)
COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")

net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

outputLayer = net.getLayerNames()
outputLayer = [outputLayer[i - 1] for i in net.getUnconnectedOutLayers()]

video_capture = cv2.VideoCapture("om1.mp4")

(W, H) = (None, None)
count = 0
while True:
    ret, frame = video_capture.read()
    fram1=frame
    #frame = cv2.flip(frame, 1)
    if W is None or H is None:
        (H,W) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB = True, crop = False)
    net.setInput(blob)
    layersOutputs = net.forward(outputLayer)

    boxes = []
    confidences = []
    classIDs = []

    for output in layersOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > confidenceThreshold:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY,  width, height) = box.astype('int')
                x = int(centerX - (width/2))
                y = int(centerY - (height/2))

                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)


    #Apply Non Maxima Suppression
    detectionNMS = cv2.dnn.NMSBoxes(boxes, confidences, confidenceThreshold,
NMSThreshold)
    if(len(detectionNMS) > 0):
```

```
            for i in detectionNMS.flatten():
                (x, y) = (boxes[i][0], boxes[i][1])

                (w, h) = (boxes[i][2], boxes[i][3])

                color = [int(c) for c in COLORS[classIDs[i]]]

                print(labels[classIDs[i]])
                if labels[classIDs[i]]=="motorbike": #since my detector only has 1 class
                    cv2.imwrite("tripleride//framet%d.jpg" % count, frame[y-200:y+h, x-20:x+w])
                    count = count + 1
            cv2.imshow('Output', frame)
            if(cv.waitKey(1) & 0xFF == ord('q')):
                break

        #Finally when video capture is over, release the video capture and destroyAllWindows
        video_capture.release()
        cv2.destroyAllWindows()
    except:
        print("completed video")

def time9():
    os.startfile("yolo_detection_images6.py")
def time7():
    import requests
    import base64
    import json
    from glob import glob
    import pandas as pd
    import time
    import os
    def ocr(IMAGE_PATH):
        SECRET_KEY = 'sk_fa7d3dcec0363bdfb6ac3e06'
        with open(IMAGE_PATH, 'rb') as image_file:
            img_base64 = base64.b64encode(image_file.read())
        url =
'https://api.openalpr.com/v2/recognize_bytes?recognize_vehicle=1&country=ind&secret_key=%s' %
(SECRET_KEY)  #Replace 'ind' with  your country code
        r = requests.post(url, data = img_base64)
        try:
            return(r.json()['results'][0]['plate'])
        except:
            print("No number plate found")
    l=[]
    c=0
    for fn in glob('LP/*.jpg'):
        print("processing",c)
        c+=1
        l.append(ocr(fn))
        if(c==3):
```

```
            break
    l=set(l)

    print(l)
    for text in l:
        raw_data = {'date':[time.asctime( time.localtime(time.time()))],'':[text]}
        #raw_data = [time.asctime( time.localtime(time.time()))],[text]
        df = pd.DataFrame(raw_data)
        df.to_csv('data.csv',mode='a')
    os.startfile('data.csv')



btn1=Button(window, text="Helmet input Video", command=time  ,fg="blue"  ,bg="orange"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=0,y=150)

btn1=Button(window, text="Detect Persons", command=time2  ,fg="blue"  ,bg="orange"  ,width=30
,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=500,y=150)

btn1=Button(window, text="Detect Helmet", command=time1  ,fg="blue"  ,bg="orange"  ,width=30
,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=980,y=150)

btn1=Button(window, text="TripleRide input Video", command=time8  ,fg="blue"  ,bg="White"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=150,y=280)

btn1=Button(window, text="Detect Persons", command=time9  ,fg="blue"  ,bg="White"  ,width=30
,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=900,y=280)


btn1=Button(window, text="Detect NumberPlate", command=time6  ,fg="blue"  ,bg="White"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=150,y=450)

btn1=Button(window, text="Detect Text NumberPlate", command=time7  ,fg="blue"  ,bg="White"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=900,y=450)


btn1=Button(window, text="Detect Tripleride video", command=time3  ,fg="blue"  ,bg="#00ff00"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=0,y=580)

btn1=Button(window, text="Detect helmet video", command=time4  ,fg="blue"  ,bg="#00ff00"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
```

```
btn1.place(x=500,y=580)


btn1=Button(window, text="traffic Detection", command=time5  ,fg="blue"  ,bg="#00ff00"
,width=30  ,height=3 ,activebackground = "white" ,font=('times', 18, ' bold '))
btn1.place(x=980,y=580)


window.mainloop()
```

Helmet detection code :

```
import numpy as np
import cv2
import cv2 as cv

confidenceThreshold = 0.5
NMSThreshold = 0.3

modelConfiguration = 'yolov3-obj.cfg'
modelWeights = 'yolov3-obj_2400.weights'

labelsPath = 'obj.names'
labels = open(labelsPath).read().strip().split('\n')

np.random.seed(10)
COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")

net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

outputLayer = net.getLayerNames()
outputLayer = [outputLayer[i - 1] for i in net.getUnconnectedOutLayers()]

video_capture = cv2.VideoCapture("t15.mp4")

(W, H) = (None, None)
count = 0
while True:
    ret, frame = video_capture.read()
    fram1=frame
    #frame = cv2.flip(frame, 1)
    if W is None or H is None:
        (H,W) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB = True, crop = False)
    net.setInput(blob)
    layersOutputs = net.forward(outputLayer)
```

```
    boxes = []
    confidences = []
    classIDs = []

    for output in layersOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > confidenceThreshold:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY,  width, height) = box.astype('int')
                x = int(centerX - (width/2))
                y = int(centerY - (height/2))

                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)


    #Apply Non Maxima Suppression
    detectionNMS = cv2.dnn.NMSBoxes(boxes, confidences, confidenceThreshold, NMSThreshold)
    if(len(detectionNMS) > 0):
        for i in detectionNMS.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])

            color = [int(c) for c in COLORS[classIDs[i]]]
            cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
            text = '{}: {:.4f}'.format(labels[classIDs[i]], confidences[i])
            print(labels[classIDs[i]])

            if labels[classIDs[i]]=="motorbike": #since my detector only has 1 class
                cv2.imwrite("dev//frame%d.jpg" % count, frame[y:y+h+25, x:x+w])
                count = count + 1

            cv2.putText(frame, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    cv2.imshow('Output', frame)
    if(cv.waitKey(1) & 0xFF == ord('q')):
        break

#Finally when video capture is over, release the video capture and destroyAllWindows
video_capture.release()
cv2.destroyAllWindow
s()
```

Live:

```
import numpy as np
import cv2
import cv2 as cv

confidenceThreshold = 0.0
NMSThreshold = 0.0

modelConfiguration = 'yolov33.cfg'
modelWeights = 'yolov33.weights'

labelsPath = 'obj1.names'
labels = open(labelsPath).read().strip().split('\n')

np.random.seed(10)
COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")

net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

outputLayer = net.getLayerNames()
outputLayer = [outputLayer[i - 1] for i in net.getUnconnectedOutLayers()]

video_capture = cv2.VideoCapture(0)
#cv2.VideoCapture('http://192.168.100.104:8080/video')
(W, H) = (None, None)
count = 0
while True:
    ret, frame = video_capture.read()
    fram1=frame
    #frame = cv2.flip(frame, 1)
    if W is None or H is None:
        (H,W) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB = True, crop = False)
    net.setInput(blob)
    layersOutputs = net.forward(outputLayer)

    boxes = []
    confidences = []
    classIDs = []

    for output in layersOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > confidenceThreshold:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY,  width, height) = box.astype('int')
```

```
                x = int(centerX - (width/2))
                y = int(centerY - (height/2))
                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)


    #Apply Non Maxima Suppression
    detectionNMS = cv2.dnn.NMSBoxes(boxes, confidences, confidenceThreshold, NMSThreshold)
    if(len(detectionNMS) > 0):
        for i in detectionNMS.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])

            color = [int(c) for c in COLORS[classIDs[i]]]
            cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
            text = '{}: {:.4f}'.format(labels[classIDs[i]], confidences[i])
            print(labels[classIDs[i]])
            if labels[classIDs[i]]=="motorbike":
                pass
                #since my detector only has 1 class
                #cv2.imwrite("dev//frame%d.jpg" % count, frame[y-55:y+h, x-20:x+w])




                count = count + 1

            cv2.putText(frame, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    cv2.imshow('Output', frame)
    if(cv.waitKey(1) & 0xFF == ord('q')):
        break

#Finally when video capture is over, release the video capture and destroyAllWindows
video_capture.release()
cv2.destroyAllWindows()
```

Yolo image detection:

```
import numpy as np
import cv2
from glob import glob

confidenceThreshold = 0.5
NMSThreshold = 0.3

modelConfiguration = 'cfg/yolov3.cfg'
```

```
modelWeights = 'yolov3.weights'

labelsPath = 'coco.names'
labels = open(labelsPath).read().strip().split('\n')

np.random.seed(10)
COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")
count=0
net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
for fn in glob('dev/*.jpg'):
    image = cv2.imread(fn)
    (H, W) = image.shape[:2]

    #Determine output layer names
    layerName = net.getLayerNames()
    layerName = [layerName[i - 1] for i in net.getUnconnectedOutLayers()]

    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB = True, crop = False)
    net.setInput(blob)
    layersOutputs = net.forward(layerName)

    boxes = []
    confidences = []
    classIDs = []

    for output in layersOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > confidenceThreshold:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY,  width, height) = box.astype('int')
                x = int(centerX - (width/2))
                y = int(centerY - (height/2))

                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)

    #Apply Non Maxima Suppression
    detectionNMS = cv2.dnn.NMSBoxes(boxes, confidences, confidenceThreshold, NMSThreshold)
```

```
        color = [int(c) for c in COLORS[classIDs[i]]]
        cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
        print(labels[classIDs[i]])
        if labels[classIDs[i]]=="person": #since my detector only has 1 class
           cv2.imwrite("dev1//frame%d.jpg" % count,image)
           count+=1
        text = '{}: {:.4f}'.format(labels[classIDs[i]], confidences[i])
        cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    cv2.imshow('Image', image)
    cv2.waitKey(500)
```

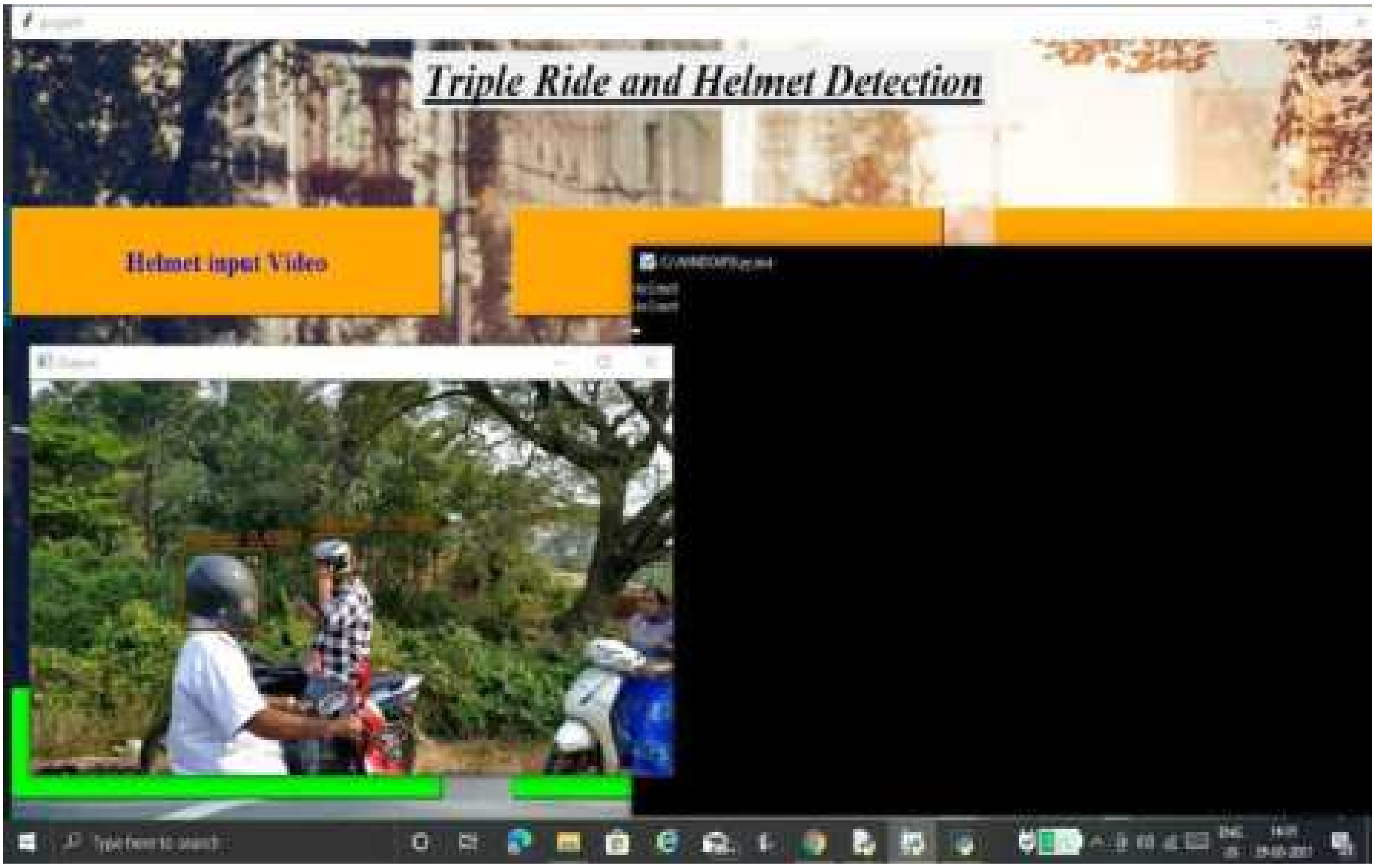
5. SCREENSHOTS

5.1 Framework for traffic rules management:

This UI is mainly designed to make traffic rules management easier for the traffic police who act as admin, here through a single interface we can do different operations such as helmet detection, triple riding detection, and number plate detection.

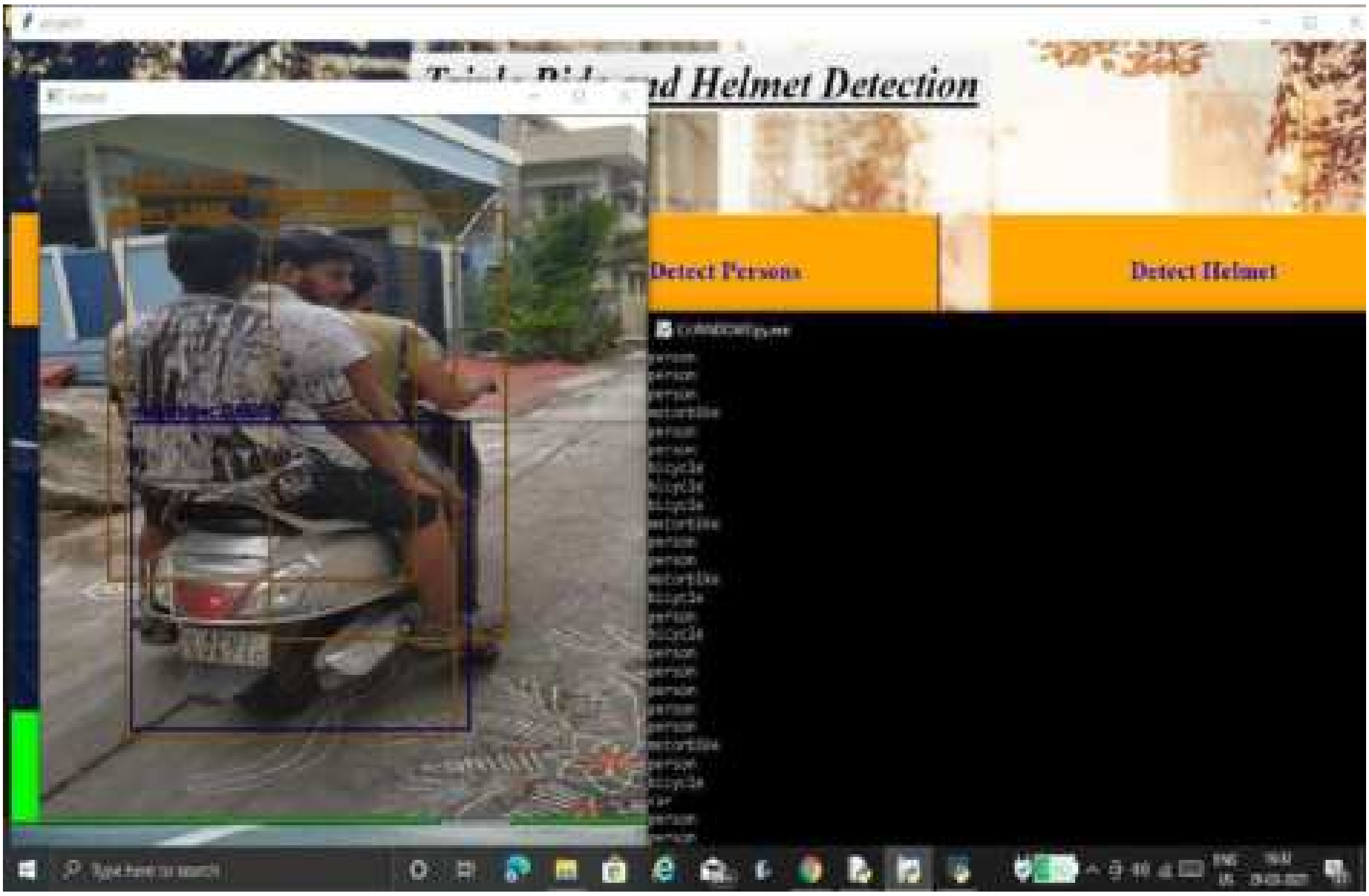
Tkinter -[10]It is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Hence we use Tkinter in our project For user interface for different applications.

5.2 HELMET DETECTION OUTPUT



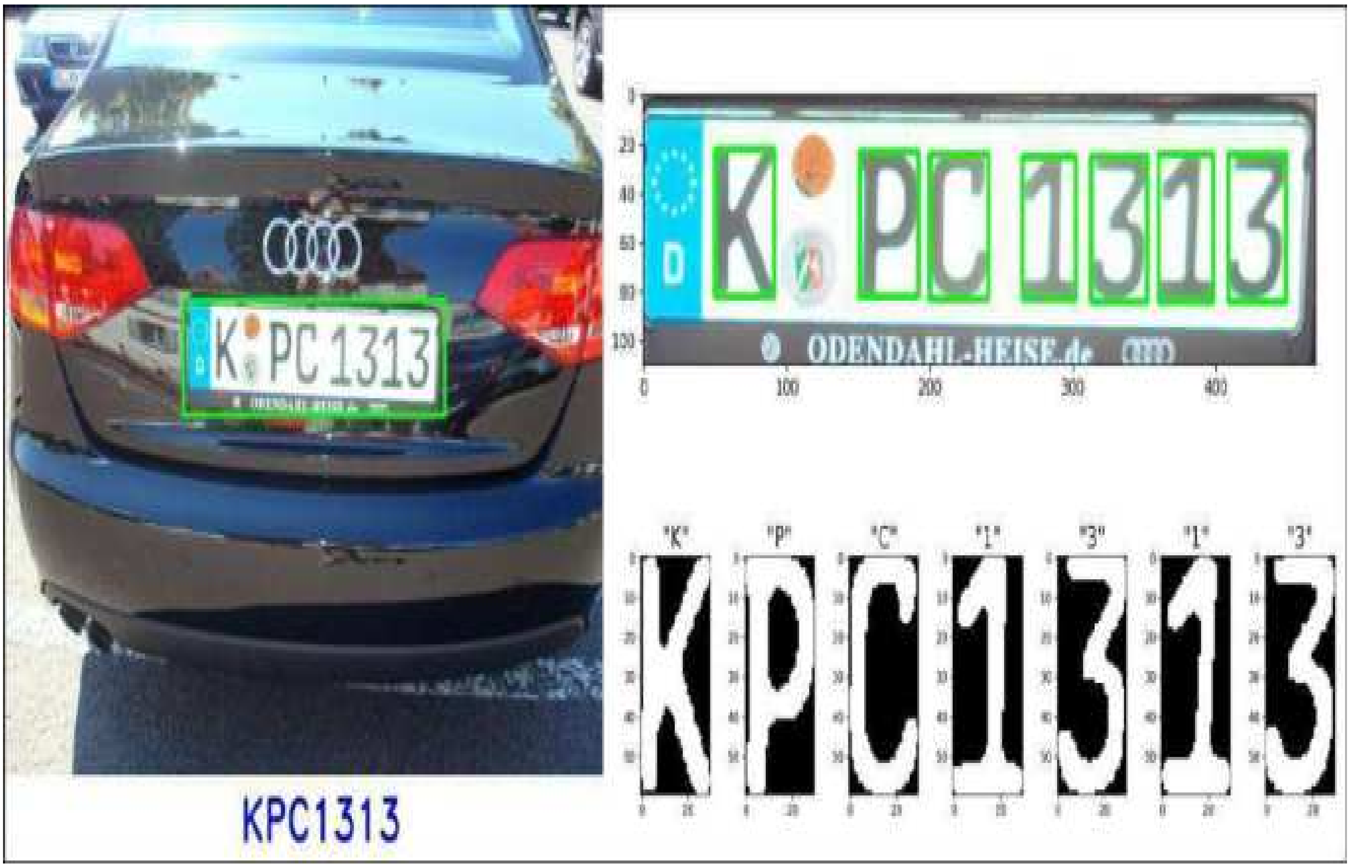
The output of the live video is detected and printed on the command prompt, in this case of helmet detection if the helmet is detected for the rider it prints helmet or else no helmet. The proposed approach first captures the real-time image of road traffic and then differentiates the two-wheelers from other vehicles on the road. [11]]It then processes to check whether the rider and pillion rider is wearing helmets or not using OpenCV. If any one of the riders and the pillion rider is found not wearing the helmet, their vehicle number plate is processed using optical character recognition (OCR). [12]After extracting the vehicle registration number, a challan will be generated against the respective vehicle and all the details of the challan will be sent via e-mail and SMS to the concerned person. A user interface (an app and a website) will also be provided to pay their challans.

5.3 TRIPLE RIDING DETECTION OUTPUT



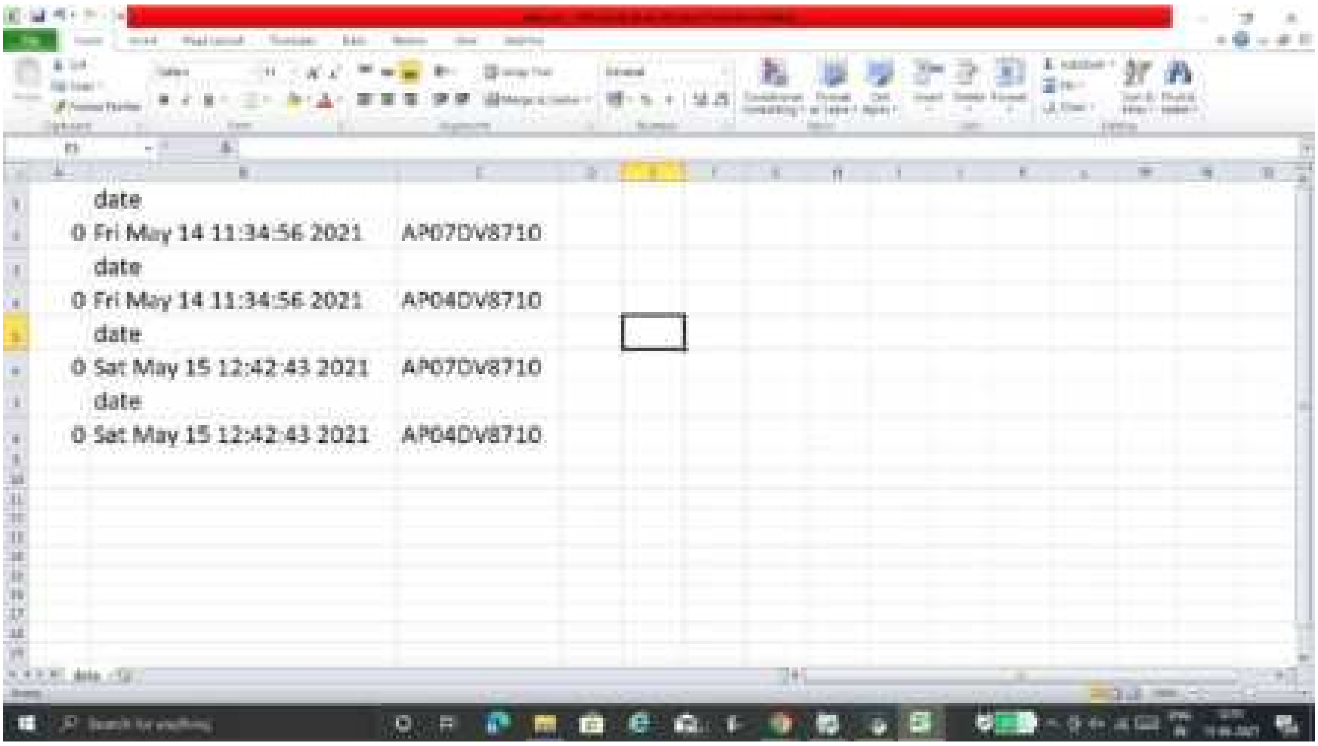
The output of the live video is detected and printed on the command prompt, in this case of helmet detection if the triple riding is detected for the rider it prints the person three times after the detecting bike or fewer print persons less then triple which is allowed. The main objective of this work is to identify the Triple Riding. [13]To detect the triple riders, the deep learning framework darknet is used, which in turn uses a type of convolutional neural network i.e. Deconvolutional neural network-based YOLO (You Only Look Once) algorithm for detection of the number of persons riding a bike, the system classifies the vehicle as to the rule-breach vehicle or not. The junctions act as the data collection center collects the data.

5.4 Number plate detection using OCR



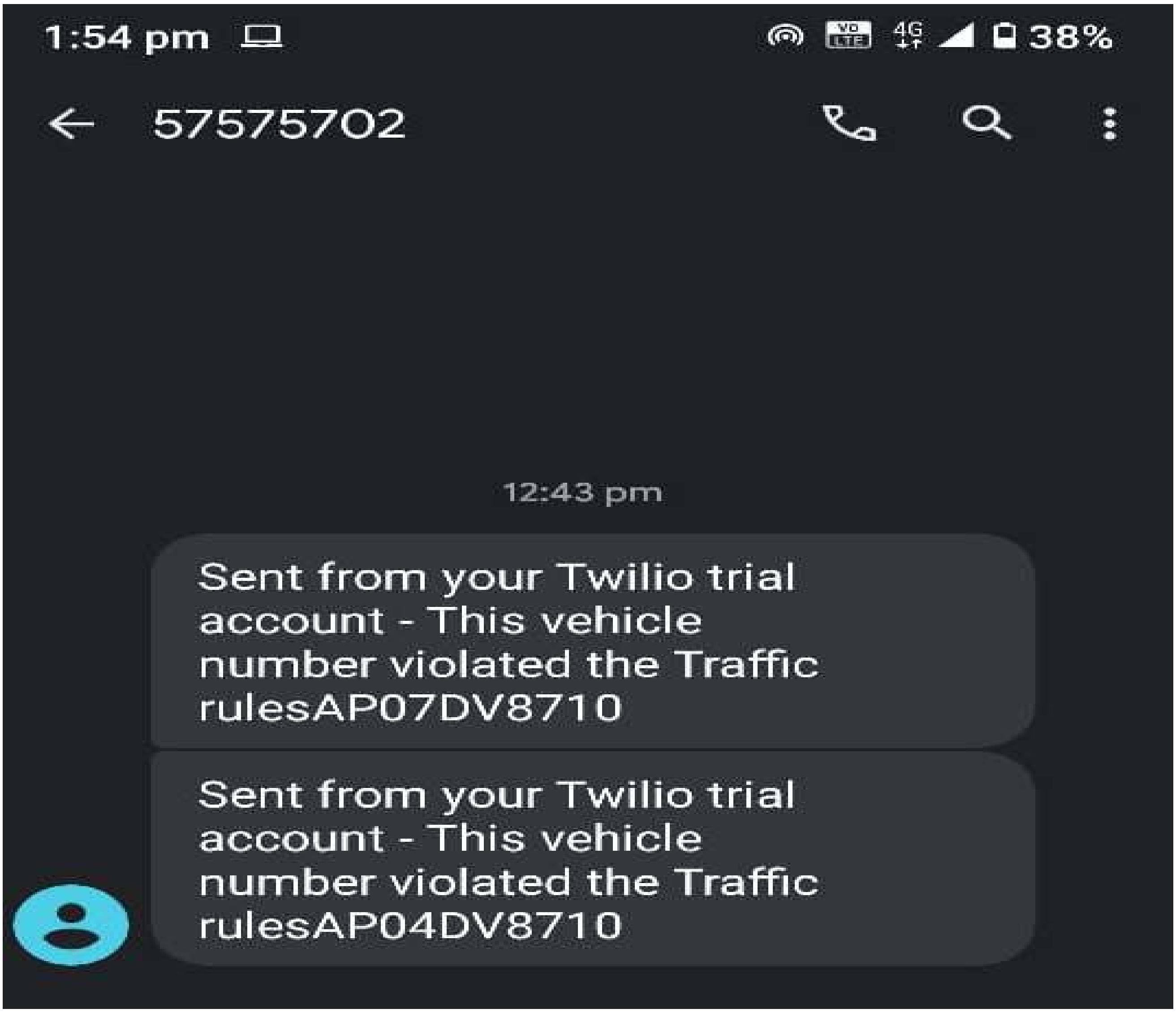
Optical character recognition (OCR) is the process of recognizing characters from images using computer vision and machine learning techniques.[14] This reference app demos how to use TensorFlow Lite to do OCR. It uses a combination of a text detection model and a text recognition model as an OCR pipeline to recognize text characters. OCR tasks are often broken down into 2 stages. First, we use a text detection model to detect the bounding boxes around possible texts. Second, we feed processed bounding boxes into a text recognition model to determine specific characters inside the bounding boxes (we also need to do Non-Maximal Suppression, perspective transformation and etc. before text recognition).[15] Takes an image of the car and searches for the number plate in the image. Once the probable number plate area is located it is given to OCR. If OCR doesn't recognize the characters from the image number plate area is searched again from the image. If characters are recognized then the number plate search is terminated.

5.5 OCR OUTPUT



Twilio Messaging is an API to send and receive SMS, MMS, and OTT messages globally. It uses intelligent sending features to ensure messages reliably reach end-users wherever they are. Twilio has SMS-enabled phone numbers available in more than 180 countries. Initially, we have to link our python code with the free online platform. So we will be giving the SID(SID stands for String Identifier. It's a unique key that is used to identify specific resources. At Twilio, each SID has 34 digits and you can identify the type of SID and the product it's associated with by the first two characters)in the program and hence integrating our project with Sms software. We have to provide a number in Twilio to which SMS has to be sent. The body of the text can be changed according to the user's perspective. So here we use to send the violated vehicle number to the respective authority. Once we trigger the button in the GUI an automatic message will be sent to the given number along with the violation message.

5.6 VIOLATION ALERT THROUGH SMS



By using Twilio we could send SMS alerts to the respected authority with the vehicle number. Twilio Messaging Service features tackle challenges before they make it to your error logs. Increase your message delivery rates, find the right phone number to pair with specific users, and scale the message volume.

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies, and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement

6.2 TYPES OF TESTING

6.2.1 UNITTESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually, run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that are specified by the business and technical requirements manuals.

Functional testing is centered on the following items:
functions tested are available as system documentation, and user

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.
The organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, and predefined processes.

6.3 TEST CASES

6.3.1 CHECKING SERVICES

Test case ID	Test case name	Purpose	Test Case	Output
1	Helmet detection	To detect helmet used while riding	Video of the rider with and without helmet	successfully detected the rider with and without helmet
2	Triple riding detection	To detect triple	video of the rider with triple riding and without triple riding	successfully detected the rider with and without triple riding
3	Number plate detection	To detect traffic rule violator vehicle plate number	The number plate should be scanned and stored in excel sheet	Display result on Excel sheet.
4	SMS notification	Send SMS and alert the violator	Sms services should be performed after detection of number plate using ocr	Successfully send Sms notifying the violation and fine need to be paid.

CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The project titled “An AI-Enabled traffic management system” is a machine learning-based application. We detect traffic violators on roads automatically using this application. As when compared to other algorithms YOLO is found to be more advantageous and has higher efficiency and accuracy. Hence, we use the same approach to identify triple riding. All modules in the system have been tested with valid data and invalid data and everything works successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The constraints are met and overcome successfully. The system is designed as it was decided in the design phase. The project gives a good idea of developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

7.2 FUTURE ENHANCEMENTS

Furthermore violations can be detected to improve the flexibility and requirements of traffic management authorities. Services provided can be made much easier and faster to perform and use in real life.

8. BIBLIOGRAPHY

8.1 GITHUB REPOSITORY LINK

https://github.com/modhisathvik7733/Artificial-intelligence-enabled-traffic-management-system

https://github.com/pavani015/an-ai-enabled-traffic-management-system/tree/main

https://github.com/suryasagar12/An-AI-Enabled-Traffic-Management-System

8.2 REFERENCES

* [1] Helmet and Triple Riding Detection on Motorcyclists using Image Processing Pradeep Maiya 1, Sharath Kumar R 1, Shreyas P V 1, Vishwa Prakash H V 1, Poornima M 2 U.G. Student, Department of Information Science and Engineering, SJB Institute of Technology, Bengaluru
* [2] Automatic Helmet Detection on Public Roads Maharshi Desai 1, Shubham Khandelwal 2, Lokesh Singh 3, Prof. Shilpa Gite 4
* [3] Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method Prajwal M. J., Tejas K. B., Varshad V., Mahesh Madivalappa Murgod, Shashidhar R.
* [4]Helmet detection on two-wheeler riders using machine learning, AMANDEEP RATHEE, KRISHNANGINI KALITA, MAHIMA SINGH DEO
* [5] Extraction and Segmentation of Helmets in Motor Cycles 1G.S.Gopika, 2R.Monisha and 3 S.Karthik.

* [6] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Fast R-CNN" (Submitted on 4 Jun 2015 (v1), last revised 6 Jan 2016 (this version, v3)).

* [7] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", the University of Washington, Allen Institute Of AI. [6] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", University of Washington, Allen Institute of AI

* [8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng – Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector".

* [9] Artificial Intelligence Enabled Traffic Monitoring System

Vishal Mand; Abdul Rashid Mussah RAND Research Report. 2018.

* [10] Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning

Ruben J Franklin; Mohana Mohana

* [11] artificial intelligence-enabled automatic traffic

V Mandal; Yaw Adu-Gyamfi

* [12] Transport and Telecommunication Vol. 22, no.2, 2021 INTELLIGENT TRAFFIC MANAGEMENT: A REVIEW OF CHALLENGES, SOLUTIONS, AND FUTURE PERSPECTIVES Roopa Ravish, Shanta Ranga Swamy

* [13] Xia and Shao, "Modelling of traffic flow and air pollutions ion with application to Hong Kong Island", Journal of Environmental Modelling & Software. Vol. 20, 2005. pp 1175-1188.

* [14] G.Sathya, Fathima Shameema S, Jyothi MolSebastian, Jemsya K S"Automatic Rescue System for Ambulance and Authoritative Vehicles, Vol.2 -Issue 4 April.

* [15] Md. Rokebul Islam*, Nafis Ibn Shahid*, Dewan Tanzim ul Karim*, Abdullah Al Mamun**, Dr. Md. Khalilur Rh "An Efficient Algorithm for Detecting Traffic Congestion

8.3 WEBSITES

1. https://www.academia.edu/49248807/TRAFFIC_SIGNAL_VIOLATION_DETECTION_USING_ARTIFICIAL_INTELLIGENCE_AND_DEEP_LEARNING

2. https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/72268/MandalVishalResearch.pdf?sequence=1&isAllowed=y

9. JOURNAL

An AI-Enabled Traffic Management System

1M.Sathvik, 2V.Guru Pavani, 3V.Surya Sagar, 4 Dr. T. S. Mastan Rao
123B.Tech Student, 4Associate Professor
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CMR TECHNICAL CAMPUS, Hyderabad-501401

ABSTRACT

Due to the growing population and people's need for comfort, more automobiles are being purchased, particularly in urban areas. This can result in heavy traffic, indicating that traffic violations are becoming more dangerous in every corner of the world. As a result, people's awareness decreases, and there are more accidents, which may result in the loss of many lives. The existing system has less accuracy and slow detection of violations, here we are using YOLO and OCR algorithms for object and number plate detection, these algorithms can detect the violation at high speed with good accuracy. The proposed system can detect the most common types of traffic violations in real-time through computer vision techniques and it also leverages good results with an accuracy of 88.3%. The proposed traffic violation detector can identify signal violations, and the individuals are informed that they will be apprehended if they break a traffic law. The proposed system is faster and more efficient than human, as known already traffic police is the one who captures the image of individuals violating traffic rules but the traffic police will not be able to capture more than one violation simultaneously. When compared to other algorithms YOLO is found to be more advantageous and has higher efficiency and accuracy.

I. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled "Improvised Traffic Light Control System using Deep Learning" Bikes are an extremely mainstream method of transportation in India. However, there is a high risk involved due to a lack of protection. To decrease the involved risk, it is highly desirable for motorcycle riders to use helmets. That's why the government has made it a punishable offense to ride a bike without a helmet. The drawback of the current method where human intervention is required can be solved by our proposed method. Nowadays everyone is moving towards automation and many countries have implemented automatic traffic surveillance systems..Here we are developing a system in which we are using a more efficient way which helps us in getting better results.

1.2 PROJECT PURPOSE

This has been developed to facilitate in order to ensure safety measures on roads of India, the identification of traffic rule violators is a highly desirable but challenging job due to numerous difficulties such as occlusion, and illumination, etc. In this paper, we propose an end-to-end framework for the detection of violations and notifying violators.

II. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analysed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1PROBLEM DEFINITION

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 the most common three types of traffic violations in real-time which are a 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.

2.2 EXISTING SYSTEM

The existing system is where we use traffic police officials to manage and maintain the traffic rules system manually. As it is a manual process we have less efficiency in the system to detect large numbers of vehicles moving on the road.

LIMITATIONS OF EXISTING SYSTEM

1. The cost of acquiring a large number of traffic police officials.
2. Less efficiency as compared to the automatic process.

2.3 PROPOSED SYSTEM

In this system we are going to implement an automatic traffic rule system. This system can detect the most common three types of traffic violations in real-time which are a 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.

ADVANTAGES OF THE PROPOSED SYSTEM

- As it is an automated process the efficiency of detecting more moving vehicles will be increased.
- cost-effective as compared to the manual process where a team of traffic officials is maintained.
- Fast process as we are using automated systems to detect and raise a ticket for the violator.

III. ARCHITECTURE

3.1 PROJECTARCITECTURE

This project architecture shows the flow of extraction from the user to the website to the user again. Here it also explains the access rights of different individuals such as user and admin.

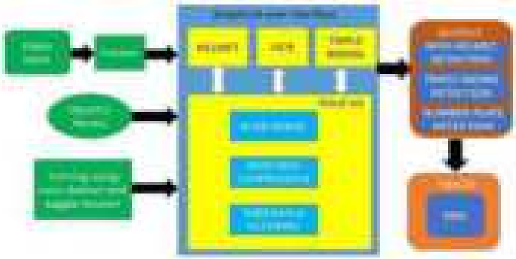


Figure 3.1: Project Architecture of Automated Traffic rules violation system

3.2 DESCRIPTION

Video input: The video input is the extracted video from the real world traffic and given to the system which divides it into frames.
OpenCV: OpenCV libraries are used to perform different machine learning algorithms on the video input.
Training dataset: The training data set is the images on which the machine learning system gets trained to identify the traffic rules violation.
Graphical user interface: The graphical user interface is used to operate different traffic violations by the administrator, yolo is the algorithm used for detection.
Output: It is where the information about the violated vehicle using the input video.

IV.SCREEN SHOTS

Framework for traffic rules management:

This UI is mainly designed to make traffic rules management easier for the traffic police who act as admin, here through a single interface we can do different operations such as helmet detection, triple riding detection, and number plate detection.

Tkinter -It is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Hence we use Tkinter in our project For user interface for different applications.

HELMET DETECTION OUTPUT

The output of the live video is detected and printed on the command prompt, in this case of helmet detection if the helmet is detected for the rider it prints helmet or else no helmet.

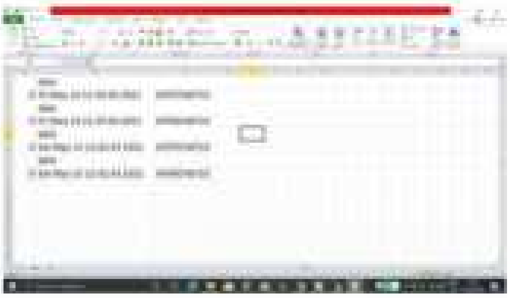
TRIPLE RIDING DETECTION OUTPUT

The output of the live video is detected and printed on the command promt, in this case of helmet detection if the triple riding is detected for the rider it prints person three time after detecting bike or less print persons less then triple which is allowed.

Number plate detection using OCR

Takes an image of the car and searches for the number plate in the image. Once the probable number plate area is located it is given to OCR. If OCR doesn't recognize the characters from the image number plate area is searched again from the image. If characters are recognized then the number plate search is terminated.

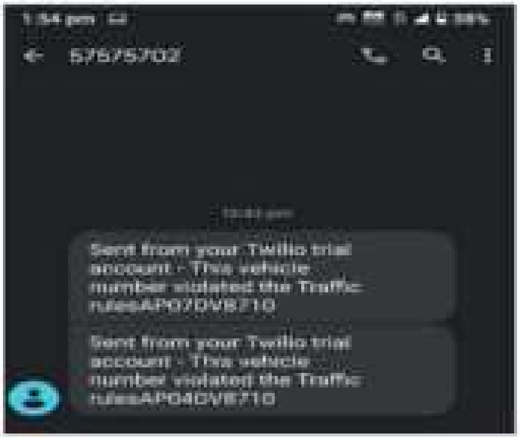
OCR OUTPUT



Twilio Messaging is an API to send and receive SMS, MMS, and OTT messages globally. It uses intelligent sending features to ensure messages reliably reach end-users wherever they are. Twilio has SMS-enabled phone numbers available in more than 180 countries. Initially, we have to link our python code with the free online platform. So we will be giving the SID(SID stands for String Identifier. It's a unique key that is used to identify specific resources. At Twilio, each SID has 34 digits and you can identify the type of SID and the product it's associated with by the first two characters)in the program and hence integrating our project with Sms software. We have to provide a number in Twilio to which SMS has to be sent. The body of the text can be changed according to the user's perspective. So here we use to send the violated vehicle number to the respective authority. Once we trigger the button in the GUI an automatic message will be sent to the given number along with the violation message.

VIOLATION ALERT THROUGH SMS

By using Twilio we could send SMS alert to the respected authority with the vehicle number



V.CONCLUSION & FUTURE SCOPE

PROJECT CONCLUSION

The project titled "An AI-Enabled traffic management system" is a machine learning-based application. We detect traffic violators on roads automatically using this application. As when compared to other algorithms YOLO is found to be more advantageous and has higher efficiency and accuracy. Hence, we use the same approach to identify triple riding. All modules in the system have been tested with valid data and invalid data and everything works successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The constraints are met and overcome successfully. The system is designed as it was decided in the design phase. The project gives a good idea of developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

FUTURE ENHANCEMENTS

Furthermore violations can be detected to improve the flexibility and requirements of traffic management authorities. Services provided can be made much easier and faster to perform and use in real life.

REFERENCES

* [1] Helmet and Triple Riding Detection on Motorcyclists using Image Processing Pradeep Maiya 1, Sharath Kumar R 1, Shreyas P V 1, Vishwa Prakash H V 1, Poornima M 2 U.G. Student, Department of Information Science and Engineering, SJB Institute of Technology, Bengaluru

* [2] Automatic Helmet Detection on Public Roads Maharshi Desai 1, ShubhamKhandelwal 2, Lokesh Singh 3, Prof.ShilpaGite 4

* [3] Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method Prajwal M. J., Tejas K. B., Varshad V., Mahesh MadivalappaMurgod, Shashidhar R.

* [4]Helmet detection on two-wheeler riders using machine learning, AMANDEEP RATHEE, KRISHNANGINI KALITA, MAHIMA SINGH DEO

* [5] Extraction and Segmentation of Helmets in Motor Cycles 1G.S.Gopika, 2R.Monisha and 3 S.Karthik.

* [6] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Fast R-CNN" (Submitted on 4 Jun 2015 (v1), last revised 6 Jan 2016 (this version, v3)).

* [7] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", the University of Washington, Allen Institute Of AI. [6] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", University of Washington, Allen Institute of AI

* [8] Wei Liu, DragomirAnguelov, DumitruErhan, Christian Szegedy, Scott Reed, Cheng – Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector".

* [9] Artificial Intelligence Enabled Traffic Monitoring System
Vishal Mand; Abdul Rashid Mussah
RAND Research Report. 2018.

* [10] Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning
Ruben J Franklin; MohanaMohana

Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1

www.journalstd.com, Mail : editorstdjournal@gmail.com

CERTIFICATE OF PUBLICATION

IMPACT FACTOR 6.1

DOI : 16.10089/STD

This is to certify that the paper entitled

"An AI-Enabled Traffic Management System "

Authored by

M.Sathvik

From

CMR TECHNICAL CAMPUS, Hyderabad-501401 .

Has been published in

STD JOURNAL, VOLUME XI, ISSUE VI, JUNE - 2022

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL

Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1
www.journalstd.com, Mail : editorstdjournal@gmail.com

IMPACT FACTOR 6.1

DOI : 16.10089/STD

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"An AI-Enabled Traffic Management System "

Authored by

V.Guru Pavani

From

CMR TECHNICAL CAMPUS, Hyderabad-501401 .

Has been published in

STD JOURNAL, VOLUME XI, ISSUE VI, JUNE - 2022

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL

Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1
www.journalstd.com, Mail : editorstdjournal@gmail.com

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"An AI-Enabled Traffic Management System "

Authored by

V.Surya Sagar

From

CMR TECHNICAL CAMPUS, Hyderabad-501401 .

Has been published in

STD JOURNAL, VOLUME XI, ISSUE VI, JUNE - 2022

IMPACT FACTOR 6.1

DOI : 16.10089/STD

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL