Configuration Manual

MSc Research Project
Programme Name

Ankit Verma
Student ID: x22145290

School of Computing
National College of Ireland

Supervisor: Anh Duong Trinh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ankit Verma
Student ID:	22145290
Programme:	MSC in Artificial Intelligence
Year:	2023
Module:	MSc Research Project
Supervisor:	Anh Duong Trinh
Submission Due Date:	31st January 2024
Project Title:	Configuration Manual
Word Count:	639
Page Count:	7

Signature:	Ankit Verma
Date:	31st January 2024

Configuration Manual

Ankit Verma
22145290

1 Abstract

This document describes the Flask backend and real-time video stream utilized by the web-based object identification application that makes use of YOLOv8 for object detection. Users can use their webcam to identify objects in real time with the aid of this program. The project's goal is to improve the usability of object detection in video streams such that the findings may be viewed on a webpage.

2 System Specification

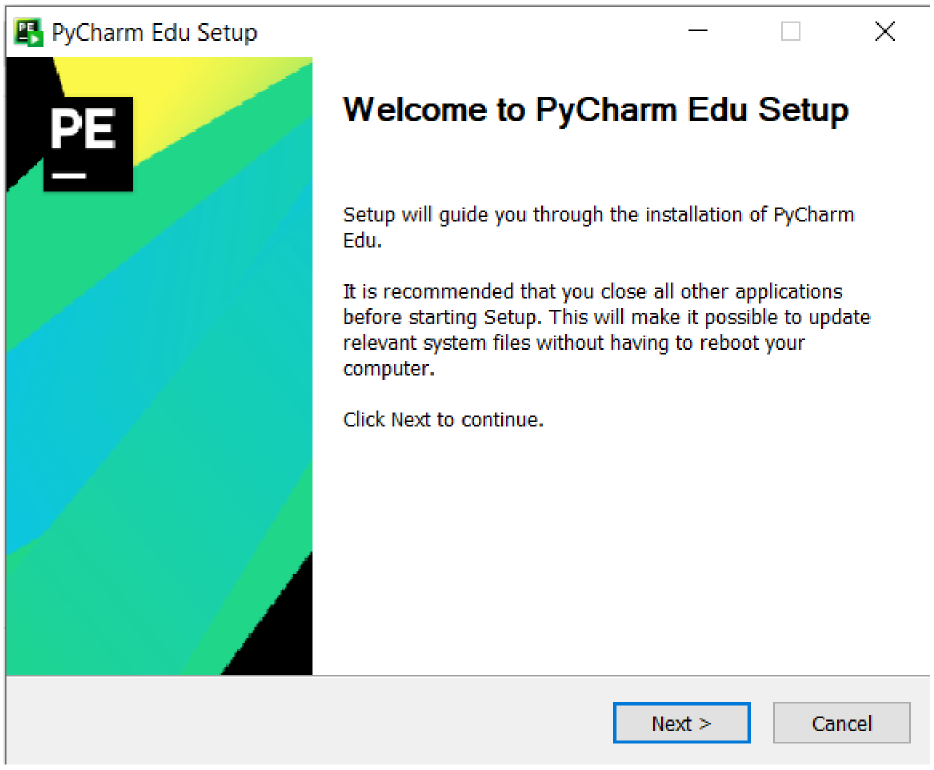
The Specification of the System are given below:-

- CPU : AMD Ryzen 5 5600H with Radeon Graphics 3.30GHz
- RAM : 24.0 GB 3200 MHz
- GPU : NVIDIA GeForce GTX 1650 4GB
- Operating System : Windows 10
- System type : 64-bit operating system, x64-based processor

3 Software Specifications

This section will examine the software needs needed to put this concept into practice. Python is the primary programming language used in the implementation, and the Anaconda prompt serves as the command-line interface. More libraries and packages have been installed in order to produce results that are accurate and well-organized. To implement and execute the system there are several software are required which are given below:-

- Python.
- Pycharm.
- WebBrowser.
- Vscode Text editor.
- Flask


PyCharm Edu Setup

Welcome to PyCharm Edu Setup

Setup will guide you through the installation of PyCharm Edu.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

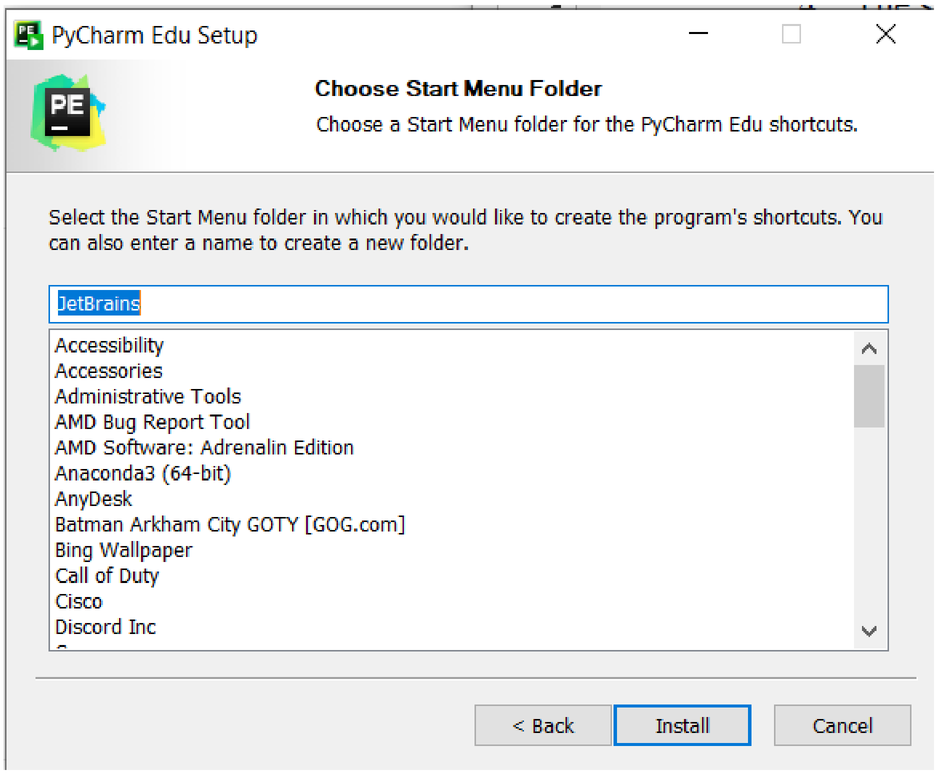
Figure 1: pycharm setup

- Flask-Cors
- Flask-SocketIO
- numpy
- opencv-python
- pandas
- Pillow
- torch
- ultralytics

4 Configuration Steps of Software installation

In this section steps are explain to install the tools.

1. Download the setup of Pycharm community edition from this link Pycharm download link
2. Once you download the setup then follow basic steps.
3. Double click the setup box which the open dailog box 1 click next
4. click on next until you get 2 this screen and click on install.
5. Download utility VScode from this link VSCode and install it.


PyCharm Edu Setup

Choose Start Menu Folder

Choose a Start Menu folder for the PyCharm Edu shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
AMD Bug Report Tool
AMD Software: Adrenalin Edition
Anaconda3 (64-bit)
AnyDesk
Batman Arkham City GOTY [GOG.com]
Bing Wallpaper
Call of Duty
Cisco
Discord Inc

Figure 2: Installation

5 Package Installation

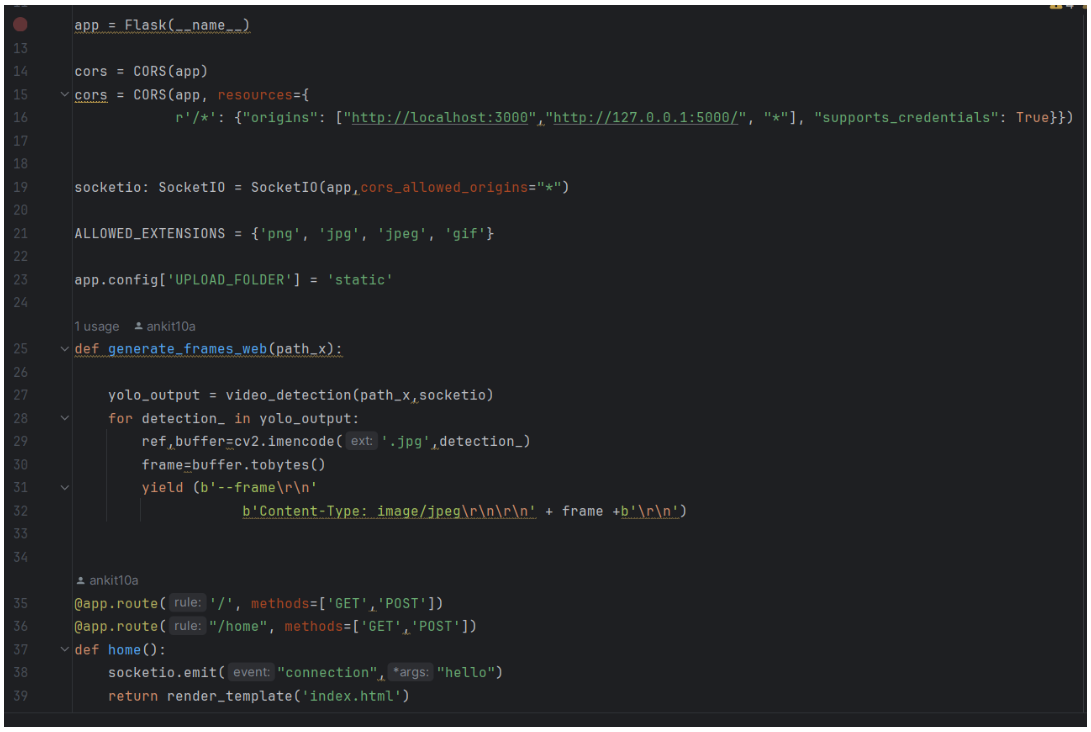
Make sure that before following these instructions, Python and pip are installed on your computer. Try verifying that Python is being executed in the proper environment or virtual environment and looking up package names if installation is proving difficult.

```
pip install -r requirements.txt
```

The package names and versions that you can use this command to install in your Python environment are listed in the requirements.txt file. Verify that this command and the requirements.txt file are located in the same directory.

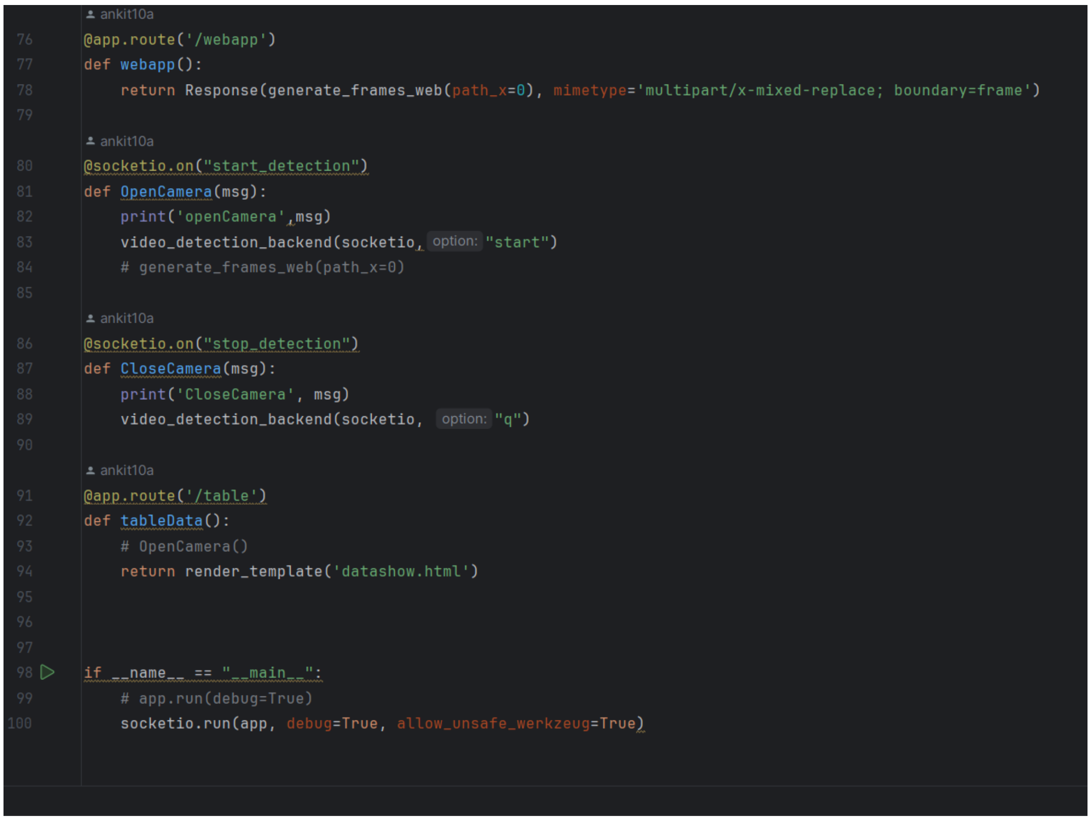
6 Code execution

- 3 shows the flask frame setup of the backend in line 12.
- In 3 line 14 shows the CORS (IP whitelisting for request)
- In 3 line 19 shows the socket setup for real time data transfer.
- In 4 show the socket, routing and page rendering logic.
- In 5 have main function which have intialize the yolo model and open camera for frame capturing.
- In 6 contains the logic of each frame class detection.



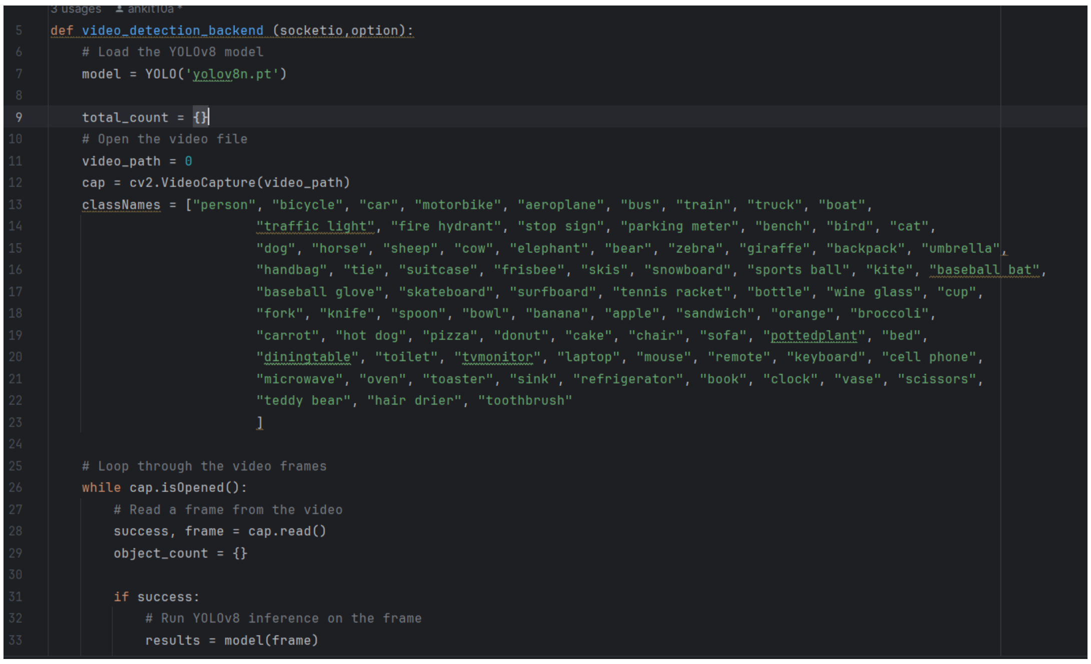
```python
app = Flask(__name__)

cors = CORS(app)
cors = CORS(app, resources={
            r'/*': {"origins": ["http://localhost:3000","http://127.0.0.1:5000/", "*"], "supports_credentials": True}})


socketio: SocketIO = SocketIO(app,cors_allowed_origins="*")

ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}

app.config['UPLOAD_FOLDER'] = 'static'

1 usage   ankit10a
def generate_frames_web(path_x):

    yolo_output = video_detection(path_x,socketio)
    for detection_ in yolo_output:
        ref,buffer=cv2.imencode( ext: '.jpg',detection_)
        frame=buffer.tobytes()
        yield (b'--frame\r\n'
                    b'Content-Type: image/jpeg\r\n\r\n' + frame +b'\r\n')


ankit10a
@app.route( rule: '/', methods=['GET','POST'])
@app.route( rule: "/home", methods=['GET','POST'])
def home():
    socketio.emit( event: "connection", *args: "hello")
    return render_template('index.html')
```

Figure 3: flask setup



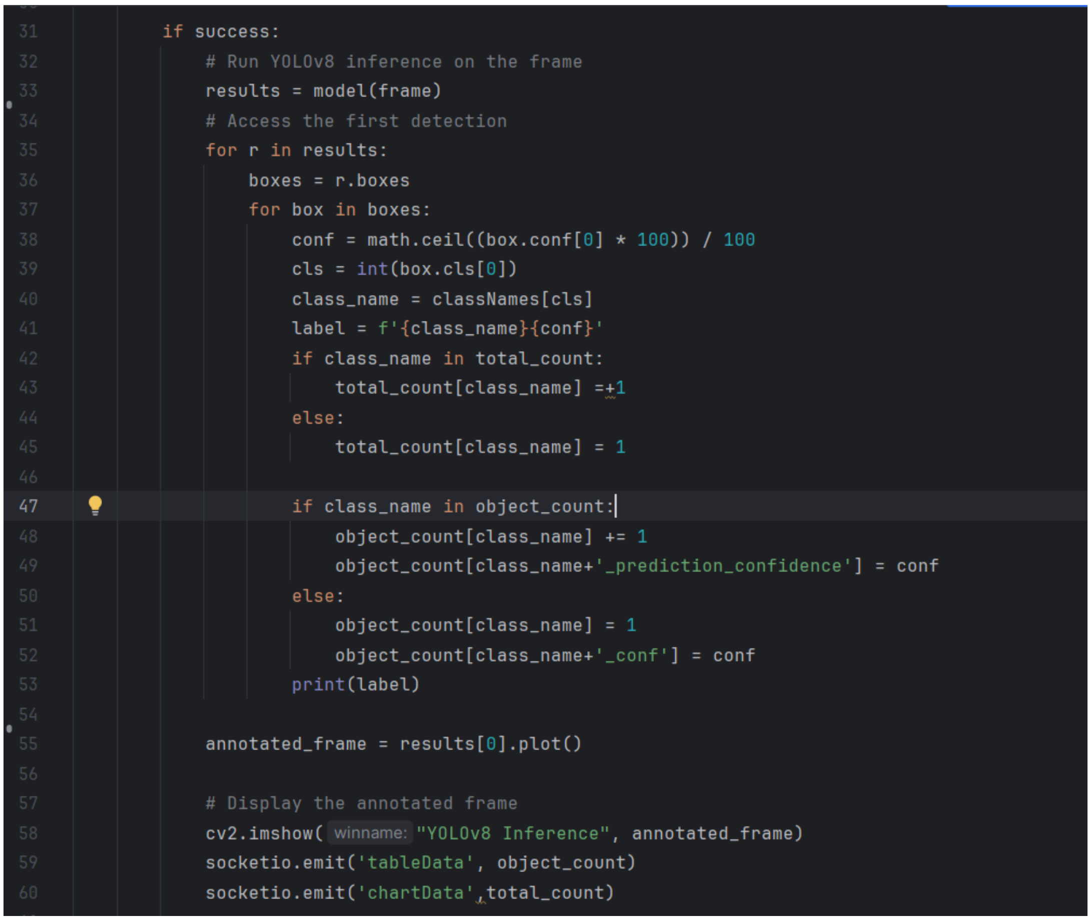
```python
ankit10a
@app.route('/webapp')
def webapp():
    return Response(generate_frames_web(path_x=0), mimetype='multipart/x-mixed-replace; boundary=frame')

ankit10a
@socketio.on("start_detection")
def OpenCamera(msg):
    print('openCamera',msg)
    video_detection_backend(socketio, option: "start")
    # generate_frames_web(path_x=0)

ankit10a
@socketio.on("stop_detection")
def CloseCamera(msg):
    print('CloseCamera', msg)
    video_detection_backend(socketio,  option: "q")

ankit10a
@app.route('/table')
def tableData():
    # OpenCamera()
    return render_template('datashow.html')



if __name__ == "__main__":
    # app.run(debug=True)
    socketio.run(app, debug=True, allow_unsafe_werkzeug=True)
```

Figure 4: api and socket setup



```python
def video_detection_backend (socketio,option):
    # Load the YOLOv8 model
    model = YOLO('yolov8n.pt')

    total_count = {}
    # Open the video file
    video_path = 0
    cap = cv2.VideoCapture(video_path)
    classNames = ["person", "bicycle", "car", "motorbike", "aeroplane", "bus", "train", "truck", "boat",
                  "traffic light", "fire hydrant", "stop sign", "parking meter", "bench", "bird", "cat",
                  "dog", "horse", "sheep", "cow", "elephant", "bear", "zebra", "giraffe", "backpack", "umbrella",
                  "handbag", "tie", "suitcase", "frisbee", "skis", "snowboard", "sports ball", "kite", "baseball bat",
                  "baseball glove", "skateboard", "surfboard", "tennis racket", "bottle", "wine glass", "cup",
                  "fork", "knife", "spoon", "bowl", "banana", "apple", "sandwich", "orange", "broccoli",
                  "carrot", "hot dog", "pizza", "donut", "cake", "chair", "sofa", "pottedplant", "bed",
                  "diningtable", "toilet", "tvmonitor", "laptop", "mouse", "remote", "keyboard", "cell phone",
                  "microwave", "oven", "toaster", "sink", "refrigerator", "book", "clock", "vase", "scissors",
                  "teddy bear", "hair drier", "toothbrush"
                  ]

    # Loop through the video frames
    while cap.isOpened():
        # Read a frame from the video
        success, frame = cap.read()
        object_count = {}

        if success:
            # Run YOLOv8 inference on the frame
            results = model(frame)
```

Figure 5: Frame object detection



```python
        if success:
            # Run YOLOv8 inference on the frame
            results = model(frame)
            # Access the first detection
            for r in results:
                boxes = r.boxes
                for box in boxes:
                    conf = math.ceil((box.conf[0] * 100)) / 100
                    cls = int(box.cls[0])
                    class_name = classNames[cls]
                    label = f'{class_name}{conf}'
                    if class_name in total_count:
                        total_count[class_name] =+1
                    else:
                        total_count[class_name] = 1

                    if class_name in object_count:
                        object_count[class_name] += 1
                        object_count[class_name+'_prediction_confidence'] = conf
                    else:
                        object_count[class_name] = 1
                        object_count[class_name+'_conf'] = conf
                    print(label)

            annotated_frame = results[0].plot()

            # Display the annotated frame
            cv2.imshow( winname: "YOLOv8 Inference", annotated_frame)
            socketio.emit('tableData', object_count)
            socketio.emit('chartData',total_count)
```

Figure 6: detected object

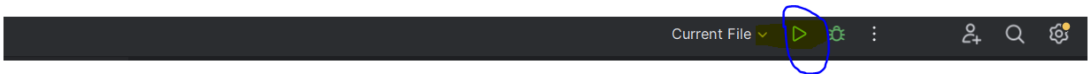


Figure 7: Start Application

Figure 8: Main Page

- In 7 open the main.py file on top of pycharm the toolbar see the highlighted run button which is show in the fig. To Start application just click on that button.
- In 8 show the landing page of application. Click on LiveWebcam on the menu will redirect in the page 9. To Redirect Analysis click on the Detection Analysis which show this page 10
- In 9 show the web live detection on web portal using webcamera
- In 10 show the analysis of the object. To start camera click on start Camera button which start web cam as a result detected object shows in table and chart as well. To stop camera click stop camera button.

References

https://www.jetbrains.com/edu-products/download/other-PCE.html
https://code.visualstudio.com/download

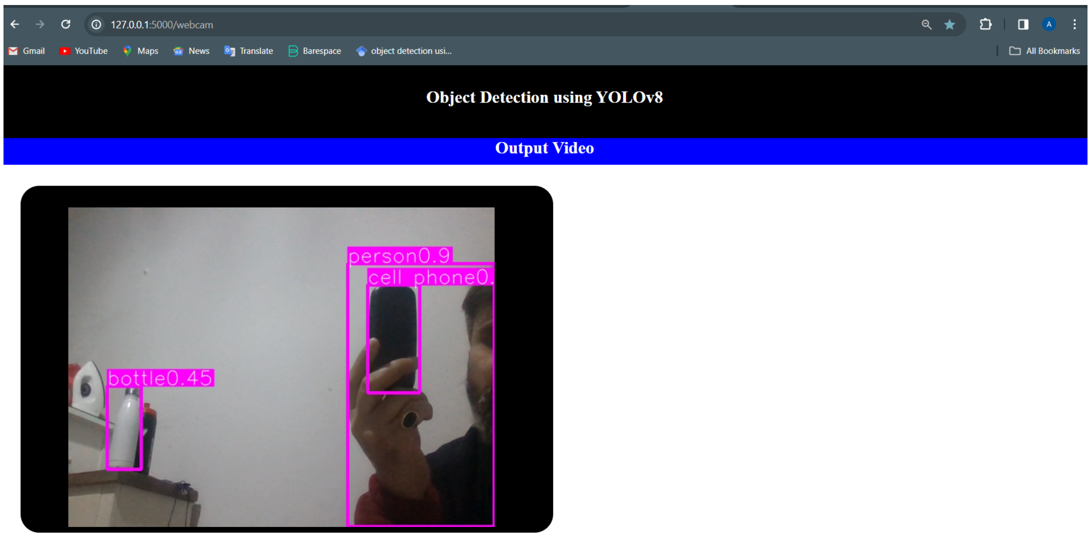
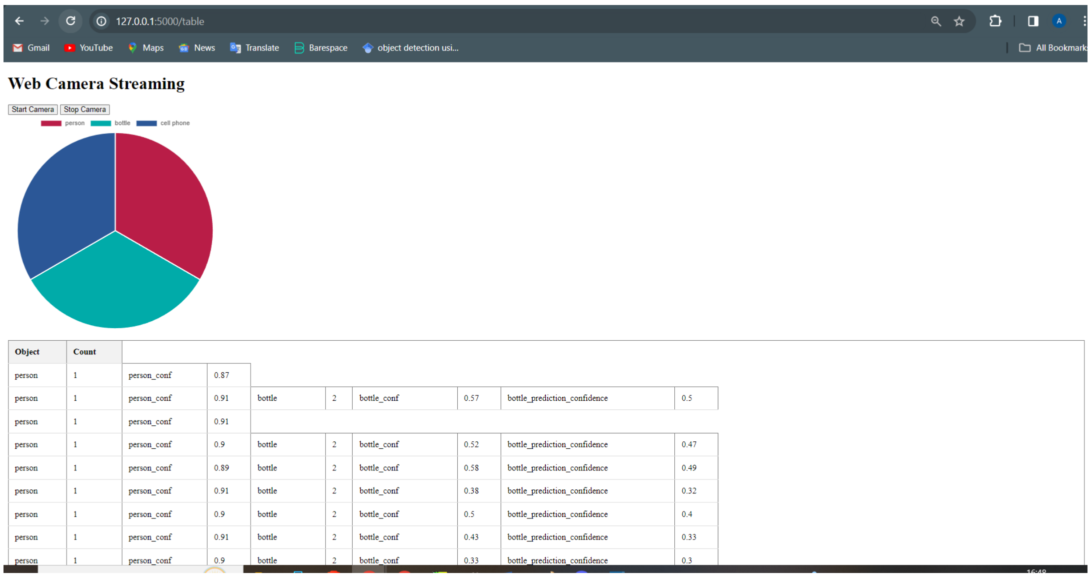


Figure 9: web protal object detection



Object	Count								
person	1	person_conf	0.87						
person	1	person_conf	0.91	bottle	2	bottle_conf	0.57	bottle_prediction_confidence	0.5
person	1	person_conf	0.91						
person	1	person_conf	0.9	bottle	2	bottle_conf	0.52	bottle_prediction_confidence	0.47
person	1	person_conf	0.89	bottle	2	bottle_conf	0.58	bottle_prediction_confidence	0.49
person	1	person_conf	0.91	bottle	2	bottle_conf	0.38	bottle_prediction_confidence	0.32
person	1	person_conf	0.9	bottle	2	bottle_conf	0.5	bottle_prediction_confidence	0.4
person	1	person_conf	0.91	bottle	2	bottle_conf	0.43	bottle_prediction_confidence	0.33
person	1	person_conf	0.9	bottle	2	bottle_conf	0.33	bottle_prediction_confidence	0.3

Figure 10: web protal object Analysis