Configuration Manual

MSc Research Project
MSc. In Cyber Security

Sachet Satish Karkera
Student ID: X21224838

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sachet Karkera
Student ID:	X21224838
Programme:	MSc. In Cyber Security **Year:** 2023/24
Module:	Research Internship
Supervisor:	Khadija Hafeez
Submission Due Date:	31/01/2024
Project Title:	Anomaly Detection for Identifying Cheating Behaviours and Techniques in Online Gaming Using AI
Word Count:	306 **Page Count: 10**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sachet Karkera
Date:	30/01/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sachet Karkera
Student ID:
X21224838

1. Hardware Setup
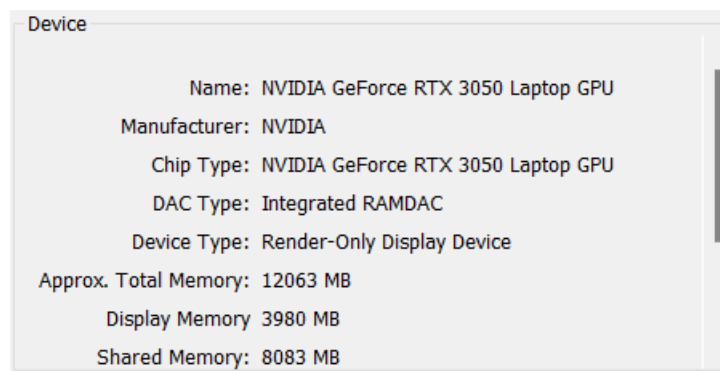

Figure 1: System Specification

Figure 2: GPU Information

2. Package/ Software Requirements and Installation

2.1 Software Requirements for YOLO model training, testing

The Following are the Software requirements –
How to use: **pip install -r requirements.txt**

Core Dependencies:
certifi==2023.7.22

click==8.1.7
colorama==0.4.6
filelock==3.12.4
six==1.16.0

Data Manipulation and Analysis:
numpy==1.26.1
Pillow==10.1.0
matplotlib==3.8.0

Machine Learning and Deep Learning:
torch==2 .1.0+cu121
torchaudio==2 .1.0+cu121
torchvision==0 .16.0+cu121
ultralytics==8.0.200

Web and Network Requests:
requests==2.31.0
urllib3==2.0.7

Development and Packaging Tools:
build==1.0.3
pip-tools==7.3.0

Text and Data Processing:
cycler==0 .12.1
fsspec==2023.10.0
idna==3.4
pyparsing==3.1.1
typing_extensions==4.8.0

Other Utilities:
contourpy==1.1.1
fonttools==4.43.1
mpmath==1.3.0
networkx==3.2
psutil==5.9.6
PyYAML==6.0.1
tqdm==4.66.1
tzdata==2023.3

2.1.1 Environmental Setup

A.   Anaconda Environment Setup

Figure 3: Environment Setup

B.   Anaconda Environment Activation with Ultralytics [1]Installation For YOLO which installs pytorch CPU version.

Figure 4: Activate Environment

Figure 5: Install Ultralytics

C.   Installation of pytorch GPU version[2], using the following command "conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -c nvidia"

---

[1] [1] https://docs.ultralytics.com/

[2] https://pytorch.org/get-started/locally/

Figure 6: Install Pytorch GPU

Figure 7: Check Pytorch GPU version

2.1.2. Executing Training of Model & Detection.

A. Training of model over 100 epochs using the following command.

Figure 8: Train Model

6

Figure 9: Start of Training

Figure 10: End of training

B. Detection of Suspected Player video by using the following command.

Figure 11: Detection

2.2 Software Requirements for Aimbot Web Application
The Following are the Software requirements –
How to use: **pip install -r requirements.txt**

Flask==3.0.0
numpy==1.26.1
scipy==1.11.3

2.2.1. Code
A. Below is the app.py

```python
from flask import Flask, render_template, request, redirect, url_for
import numpy as np
from scipy import stats

app = Flask(__name__)

def is_valid_number_list(s):
    try:
        numbers = list(map(int, s.split(',')))
        return all(isinstance(num, int) for num in numbers)
    except ValueError:
        return False

@app.route('/', methods=['GET', 'POST'])
def index():
    result = None
    error_message = None

    if request.method == 'POST':
        pitch_data = request.form['pitch']
        yaw_data = request.form['yaw']

        if not is_valid_number_list(pitch_data) or not is_valid_number_list(yaw_data):
            error_message = "Please enter valid comma-separated numbers."

        else:
            pitch_data = np.array(list(map(int, pitch_data.split(','))))
            yaw_data = np.array(list(map(int, yaw_data.split(','))))

            min_pitch, max_pitch = -89, 89
            min_yaw, max_yaw = -180, 180

            pitch_zscores = np.abs(stats.zscore(pitch_data))
            yaw_zscores = np.abs(stats.zscore(yaw_data))

            zscore_threshold = 2

            unnatural_movements = np.logical_or(pitch_zscores > zscore_threshold, yaw_zscores > zscore_threshold)

            frames_with_unnatural_movements = np.where(unnatural_movements)[0] + 1

            if len(frames_with_unnatural_movements) > 0:
                result = f"Aimbot detected in frames: {frames_with_unnatural_movements}"
            else:
                result = "No Aimbot detected."

    return render_template('index.html', result=result, error_message=error_message)

if __name__ == '__main__':
    app.run(debug=True)
```

Figure 12: Application Code

B. Below is the HTML for the same:

Figure 13: Code for web page

Figure 14: Code for web page

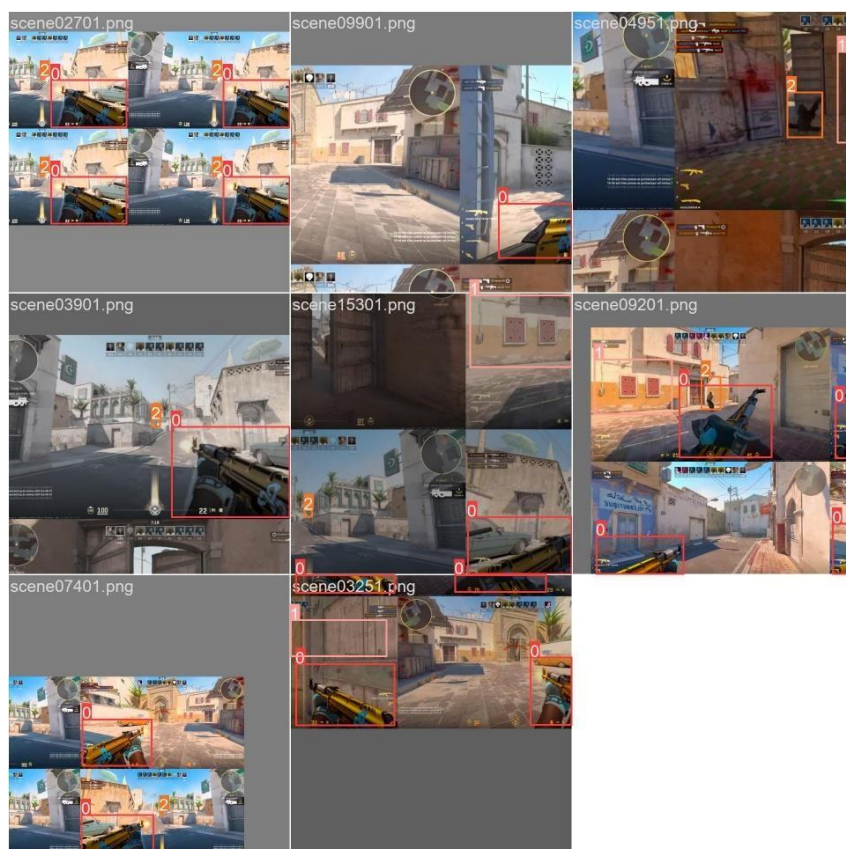# 3. Results.

3.1. Training Results From YOLOV8

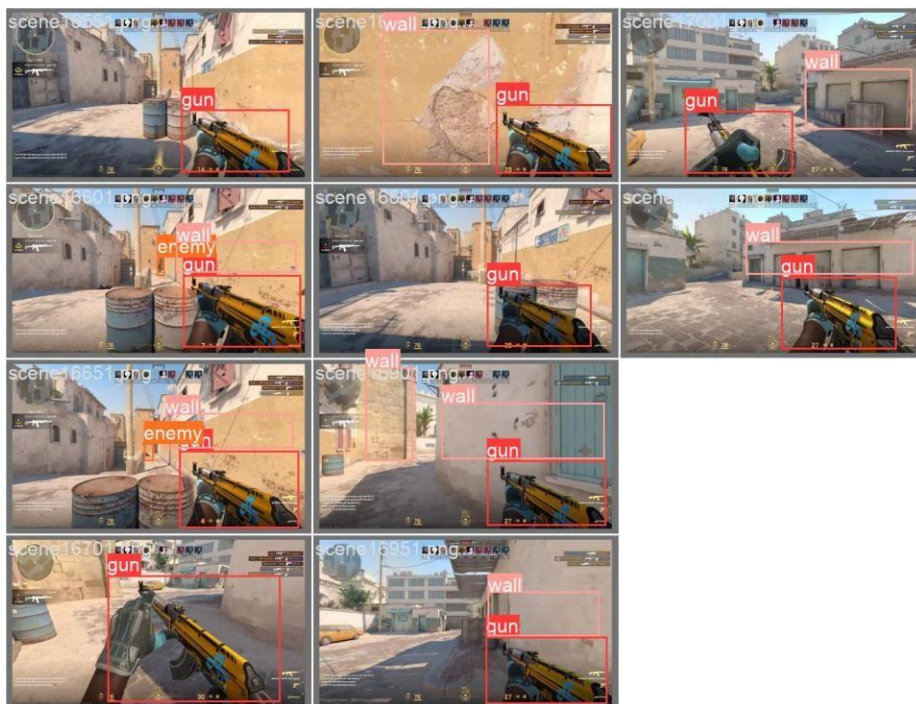**Figure 15: Detection Result 1**
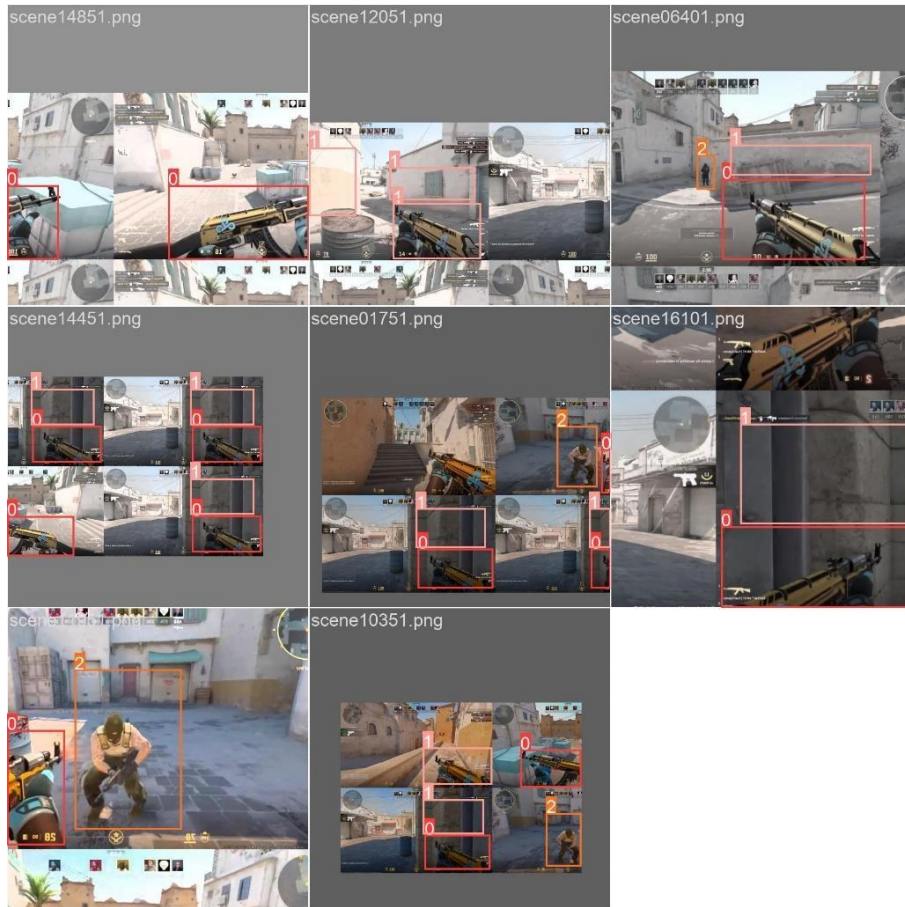
Figure 16: Detection Result 2
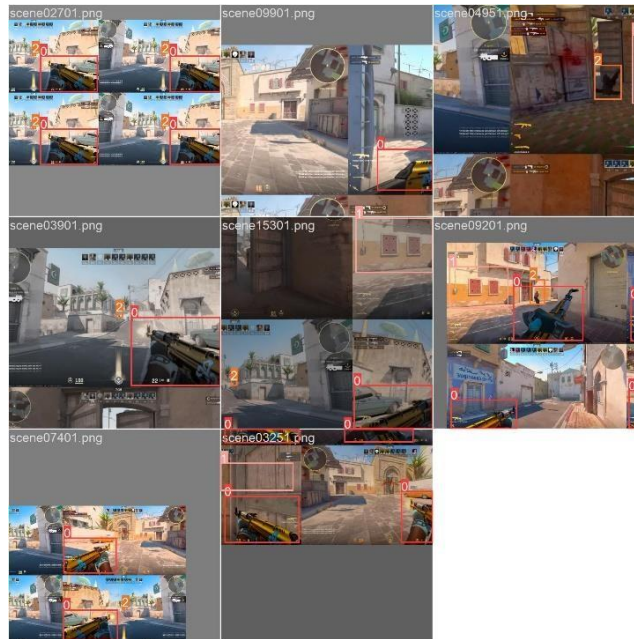

Figure 17: Detection Result 3

Figure 18: Detection Result 4

3.2.   Output:

Figure 19: Aimbot Detected

Figure 20: Aimbot Not Detected