Detect Cheater in Online Gaming using AI

MSc Research Project
MSc. In Cyber Security

Sparsh Bajaj
Student ID: X0228392

School of Computing
National College of Ireland

Supervisor: Dr. Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sparsh Bajaj		
Student ID:	X0228392		
Programme:	MSc. In Cyber Security	**Year:**	2022
Module:	Research Internship		
Supervisor:	Ross Spelman		
Submission Due Date:	15/10/2022		
Project Title:	Detect Cheater in Online Gaming using AI		
Word Count:	4500	**Page Count:**	20

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: Sparsh Bajaj

Date: 14/12/2022

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.	□

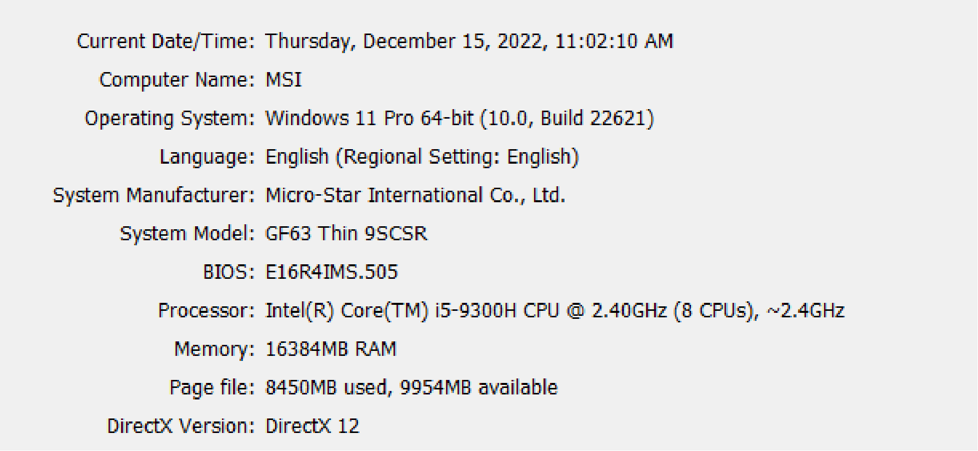
Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

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

Configuration Manual

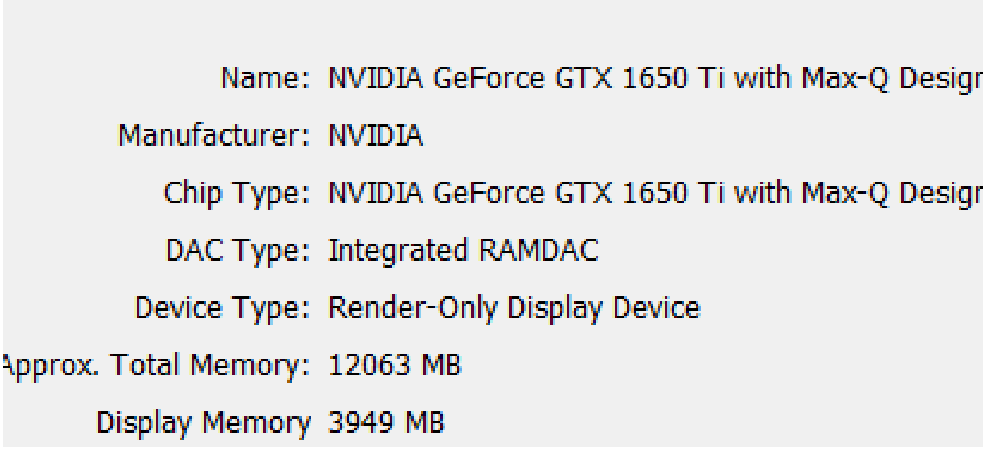
Forename Surname
Student ID:

1 Hardware Setup



Current Date/Time: Thursday, December 15, 2022, 11:02:10 AM
Computer Name: MSI
Operating System: Windows 11 Pro 64-bit (10.0, Build 22621)
Language: English (Regional Setting: English)
System Manufacturer: Micro-Star International Co., Ltd.
System Model: GF63 Thin 9SCSR
BIOS: E16R4IMS.505
Processor: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz (8 CPUs), ~2.4GHz
Memory: 16384MB RAM
Page file: 8450MB used, 9954MB available
DirectX Version: DirectX 12

Figure 1:System Configuration



Name: NVIDIA GeForce GTX 1650 Ti with Max-Q Desigr
Manufacturer: NVIDIA
Chip Type: NVIDIA GeForce GTX 1650 Ti with Max-Q Desigr
DAC Type: Integrated RAMDAC
Device Type: Render-Only Display Device
Approx. Total Memory: 12063 MB
Display Memory 3949 MB

Figure 2: GPU Memory

2 Package/ Software Requirements and Installation

The requirements are neatly stored in an txt file for local training for AI but it would be faster to use google collab notebook which already handles requirements.
Link to google collab –

If not these are the Pre requirements –

Usage: pip install -r requirements.txt

- gitpython
- ipython
- matplotlib>=3.2.2
- numpy>=1.18.5
- opencv-python>=4.1.1
- Pillow>=7.1.2
- psutil
- PyYAML>=5.3.1
- requests>=2.23.0
- scipy>=1.4.1
- thop>=0.1.1
- torch>=1.7.0
- torchvision>=0.8.1
- tqdm>=4.64.0
- **protobuf<=3.20.1**

Logging ---

- tensorboard>=2.4.1
- clearml>=1.2.0
- comet

Plotting --

- **pandas>=1.1.4**
- **seaborn>=0.11.0**

Export --

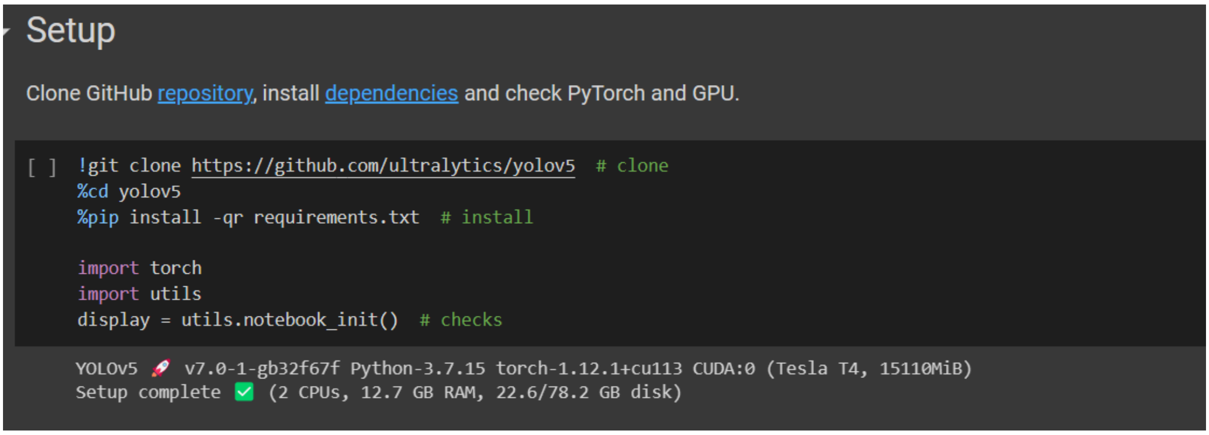
- coremltools>=6.0
- onnx>=1.9.0
- onnx-simplifier>=0.4.1
- nvidia-pyindex
- nvidia-tensorrt
- scikit-learn<=1.1.2
- tensorflow>=2.4.1
- tensorflowjs>=3.9.0
- openvino-dev

Deploy --

- **tritonclient[all]~=2.24.0**

Link to google collab notebook – https://colab.research.google.com/drive/1cCOsm0ANsqyF5xzd9pidZfEbJw3bBk3O?usp=sharing

Run the first block for setting up the environment –



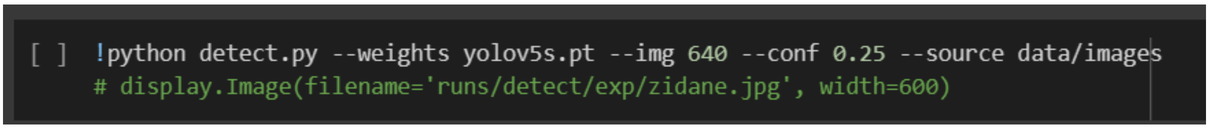
Setup

Clone GitHub repository, install dependencies and check PyTorch and GPU.

```
[ ] !git clone https://github.com/ultralytics/yolov5  # clone
    %cd yolov5
    %pip install -qr requirements.txt  # install

    import torch
    import utils
    display = utils.notebook_init()  # checks
```

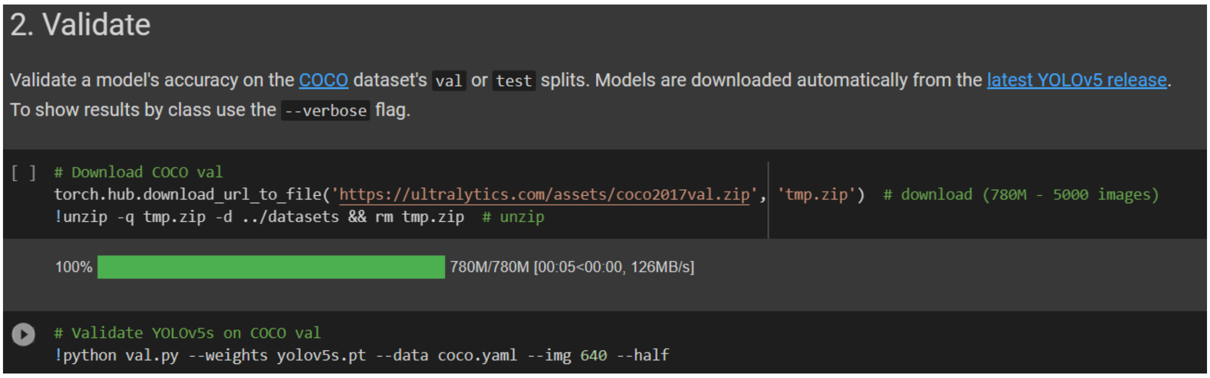
```
YOLOv5 🚀 v7.0-1-gb32f67f Python-3.7.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 22.6/78.2 GB disk)
```

Then uder detection run the second block –



```
[ ] !python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images
    # display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

Once it is done Run the validation block –



2. Validate

Validate a model's accuracy on the COCO dataset's `val` or `test` splits. Models are downloaded automatically from the latest YOLOv5 release. To show results by class use the `--verbose` flag.

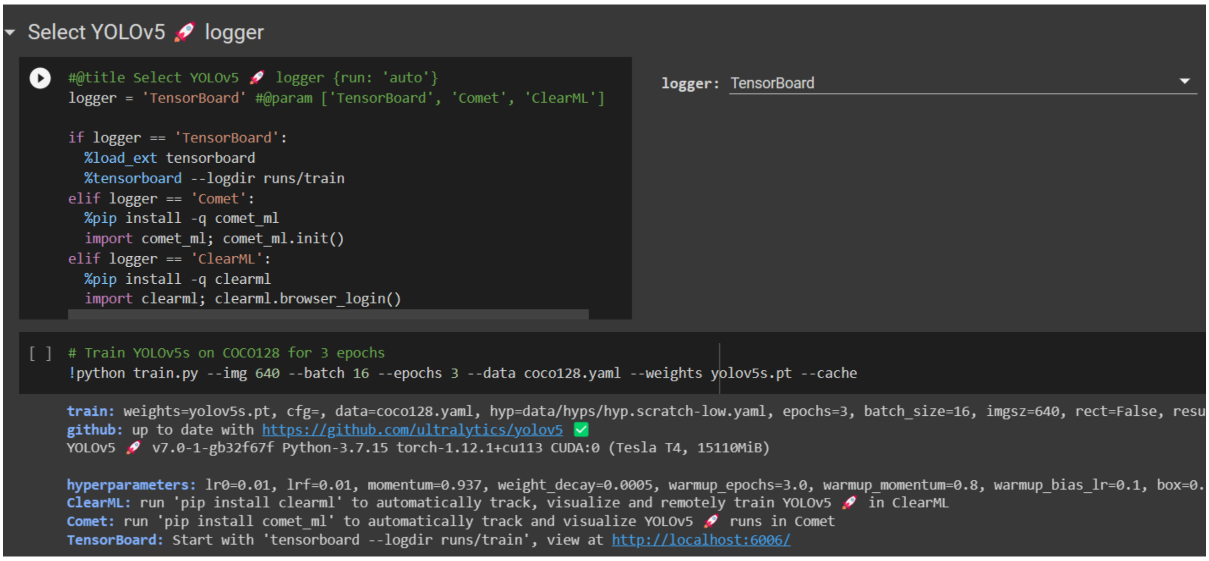
```
[ ] # Download COCO val
    torch.hub.download_url_to_file('https://ultralytics.com/assets/coco2017val.zip', 'tmp.zip')  # download (780M - 5000 images)
    !unzip -q tmp.zip -d ../datasets && rm tmp.zip  # unzip
```

```
100% 780M/780M [00:05<00:00, 126MB/s]
```

```
# Validate YOLOv5s on COCO val
!python val.py --weights yolov5s.pt --data coco.yaml --img 640 --half
```

Don’t forget to upload the training data.

For traing run the third block –



Select YOLOv5 🚀 logger

```
#@title Select YOLOv5 🚀 logger {run: 'auto'}
logger = 'TensorBoard' #@param ['TensorBoard', 'Comet', 'ClearML']

if logger == 'TensorBoard':
  %load_ext tensorboard
  %tensorboard --logdir runs/train
elif logger == 'Comet':
  %pip install -q comet_ml
  import comet_ml; comet_ml.init()
elif logger == 'ClearML':
  %pip install -q clearml
  import clearml; clearml.browser_login()
```

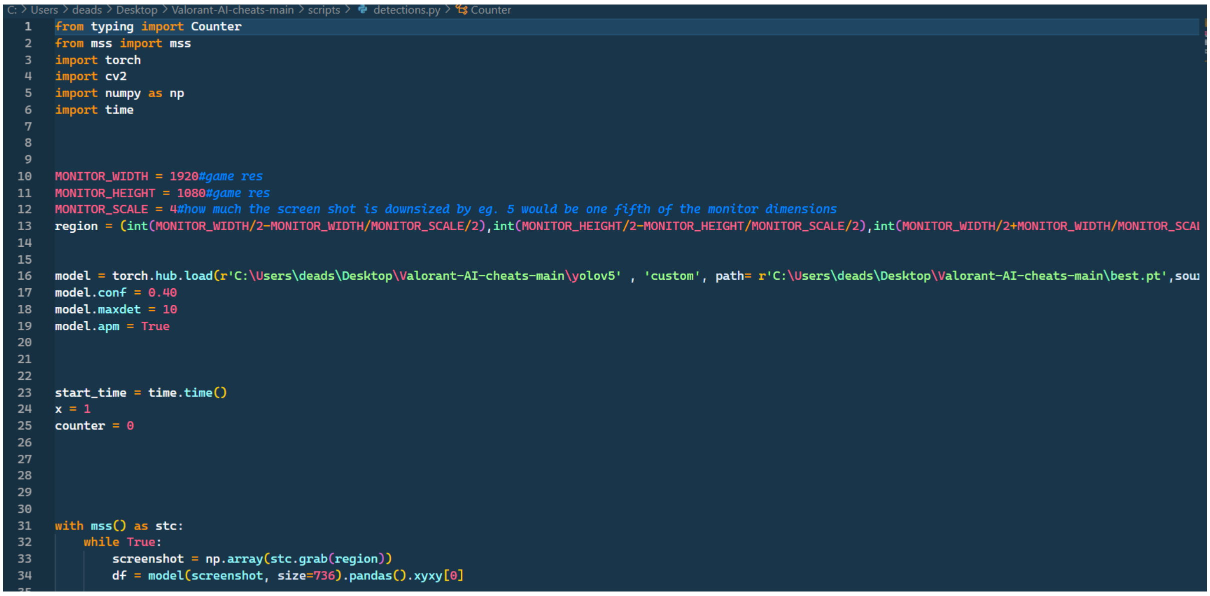
logger: TensorBoard

```
[ ] # Train YOLOv5s on COCO128 for 3 epochs
    !python train.py --img 640 --batch 16 --epochs 3 --data coco128.yaml --weights yolov5s.pt --cache
```

```
train: weights=yolov5s.pt, cfg=, data=coco128.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=3, batch_size=16, imgsz=640, rect=False, resu
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-1-gb32f67f Python-3.7.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 🚀 in ClearML
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 🚀 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
```

This would save all the test resluts to your google drive. And our training dataset would be completed to run on local.

3 Running AI on local to test with the game

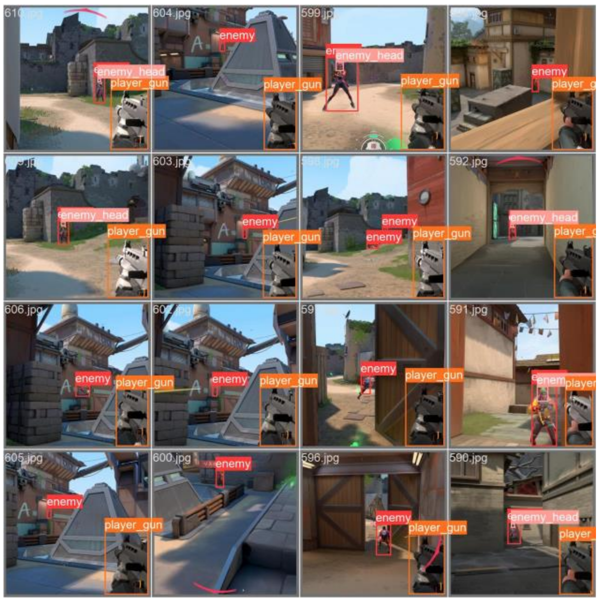

C: > Users > deads > Desktop > Valorant-AI-cheats-main > scripts > detections.py > Counter

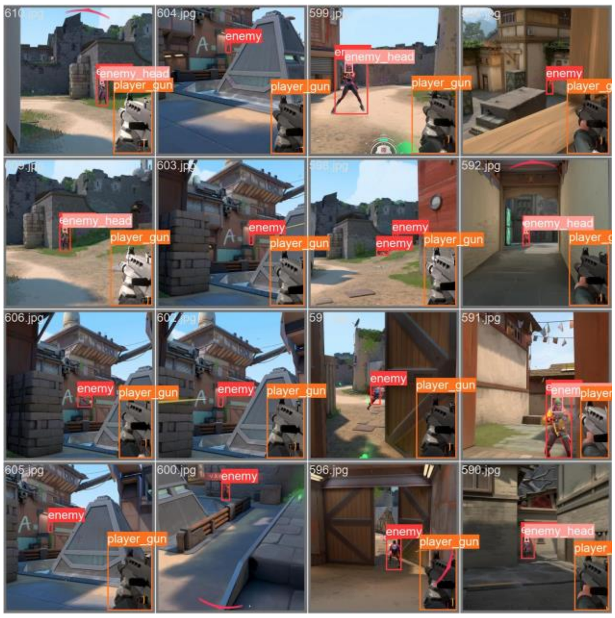
```
from typing import Counter
from mss import mss
import torch
import cv2
import numpy as np
import time



MONITOR_WIDTH = 1920#game res
MONITOR_HEIGHT = 1080#game res
MONITOR_SCALE = 4#how much the screen shot is downsized by eg. 5 would be one fifth of the monitor dimensions
region = (int(MONITOR_WIDTH/2-MONITOR_WIDTH/MONITOR_SCALE/2),int(MONITOR_HEIGHT/2-MONITOR_HEIGHT/MONITOR_SCALE/2),int(MONITOR_WIDTH/2+MONITOR_WIDTH/MONITOR_SCAl


model = torch.hub.load(r'C:\Users\deads\Desktop\Valorant-AI-cheats-main\yolov5' , 'custom', path= r'C:\Users\deads\Desktop\Valorant-AI-cheats-main\best.pt',sou
model.conf = 0.40
model.maxdet = 10
model.apm = True



start_time = time.time()
x = 1
counter = 0





with mss() as stc:
    while True:
        screenshot = np.array(stc.grab(region))
        df = model(screenshot, size=736).pandas().xyxy[0]
```

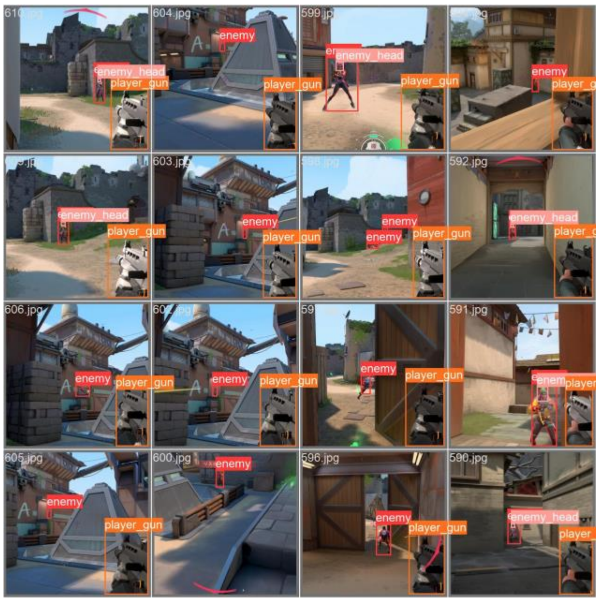
Run detections.py file with appropriate location for best.pt file (trained model, best case). This will open a small screen capute window with live detections while you run the game (valorant).

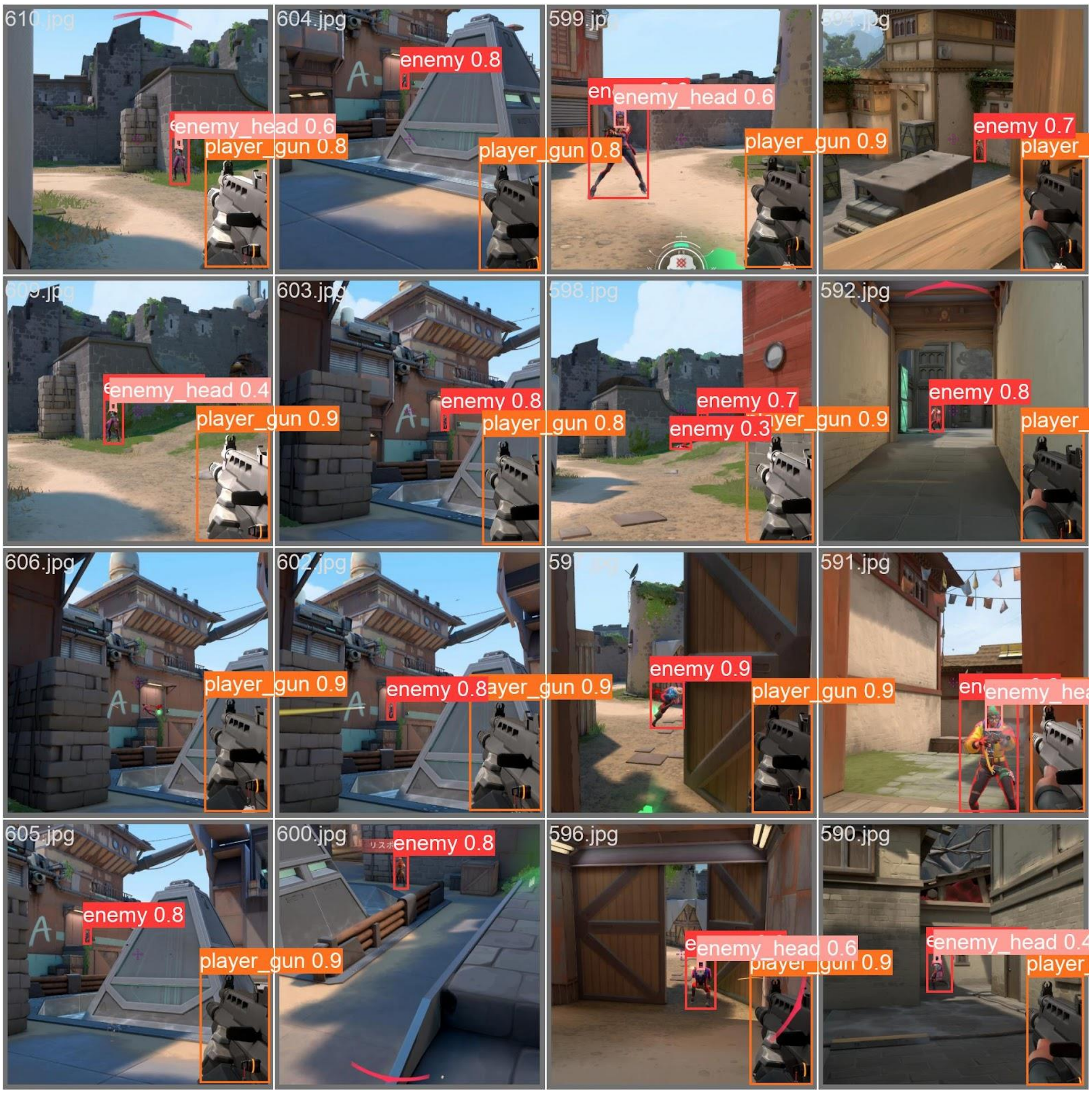
Note – There may be some dependencies required to install, these can be installed based on errors using pip install commands.

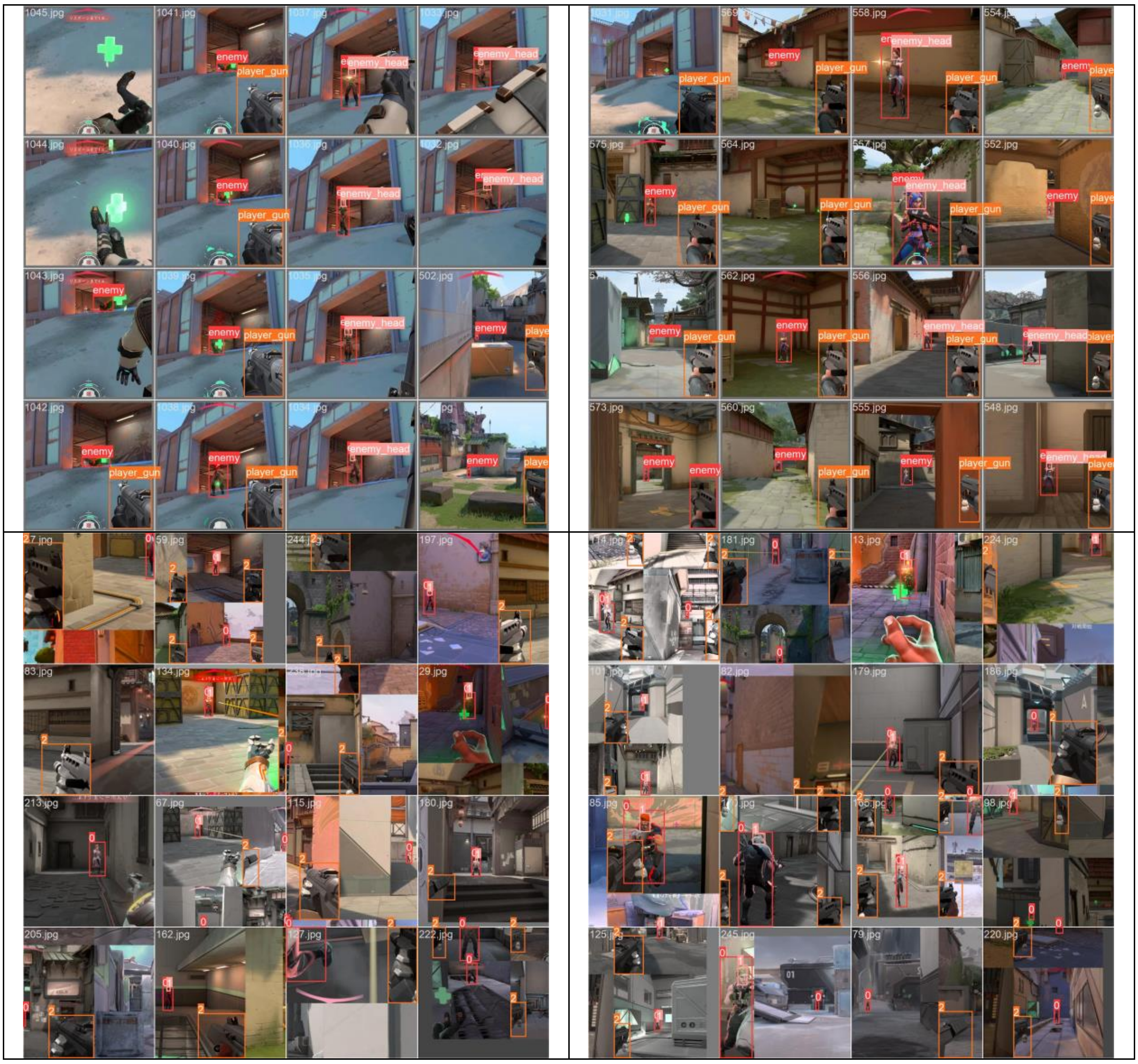
4 Outputs for trained model




610.jpg
enemy_head
player_gun
604.jpg
enemy
player_gun
599.jpg
enemy_head
player_gun
enemy
player_g
603.jpg
enemy_head
player_gun
enemy
player_gun
enemy
enemy
player_gun
592.jpg
enemy_head
player_g
606.jpg
enemy
player_gun
enemy
player_gun
enemy
player_gun
591.jpg
enem
player
605.jpg
enemy
player_gun
600.jpg
enemy
596.jpg
enemy
player_gun
590.jpg
enemy_head
player


610.jpg
604.jpg
enemy
599.jpg
enemy_head
player_gun
player_gun
enemy_head
player_gun
enemy
player_g
603.jpg
enemy_head
player_gun
enemy
player_gun
enemy
enemy
player_gun
592.jpg
enemy_head
player_g
606.jpg
enemy
player_gun
enemy
player_gun
enemy
player_gun
591.jpg
enem
player
605.jpg
enemy
player_gun
600.jpg
enemy
596.jpg
enemy
player_gun
590.jpg
enemy_head
player


610.jpg
enemy_head 0.6
player_gun 0.8
604.jpg
enemy 0.8
player_gun 0.8
599.jpg
en
enemy_head 0.6
player_gun 0.9
enemy 0.7
player_
enemy_head 0.4
player_gun 0.9
603.jpg
enemy 0.8
player_gun 0.8
598.jpg
enemy 0.7
enemy 0.3
yer_gun 0.9
592.jpg
enemy 0.8
player_
606.jpg
player_gun 0.9
enemy 0.8
ayer_gun 0.9
enemy 0.9
player_gun 0.9
591.jpg
en
enemy_hea
605.jpg
enemy 0.8
player_gun 0.9
600.jpg
enemy 0.8
596.jpg
enemy_head 0.6
player_gun 0.9
590.jpg
enemy_head 0.4
player_


1045.jpg
1041.jpg
1037.jpg
1033.jpg
1044.jpg
1040.jpg
1036.jpg
1032.jpg
1043.jpg
1039.jpg
1035.jpg
502.jpg
1042.jpg
1038.jpg
1034.jpg
enemy
enemy_head
player_gun
1031.jpg
558.jpg
575.jpg
564.jpg
567.jpg
552.jpg
562.jpg
556.jpg
573.jpg
560.jpg
555.jpg
548.jpg
27.jpg
59.jpg
197.jpg
83.jpg
134.jpg
29.jpg
213.jpg
67.jpg
115.jpg
180.jpg
205.jpg
162.jpg
127.jpg
222.jpg
181.jpg
13.jpg
224.jpg
82.jpg
179.jpg
186.jpg
85.jpg
98.jpg
125.jpg
245.jpg
79.jpg
220.jpg