National College of Ireland

Configuration Manual

MSc Research Project
MSc in Artificial Intelligence

Abhishek Goyal
Student ID: 23152851

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Abhishek Goyal
Student ID:	23152851
Programme:	MSc in Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Victor Del Rosal
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	650
Page Count:	6

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:	Abhishek Goyal
Date:	12th December 2024

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

Abhishek Goyal
23152851

1 Requirements

It is advised to follow all the steps of this manual in order to execute the code which is used in the documentation for a research project that supports the details of this configeration manual. This document will Practically, they will discuss the finite minima of the necessity of the supporting hardware as well as software reproduce this code.

2 Machine Hardware

Before implementing the code below the following machine hardware for the project requirements will have to be met to work. The machine that preformed this research configuration is: 16gb RAM, 12th Gen Intel(R) Core(TM) i7-12650H 2.70 GHz, 64-bit Operating System, window 11 .

3 Machine Software

The only software requirements required to run the code are as follows and these are the ones that have been used. Google Colaboratory is utilized to execute the research code of the current study. Python was utilized in this project and thank you for using version 3.7. Google drive/Gmail An account is then used to connect to notebook. Jupyter notebook is required, Microsoft excel is required as data stored as csv file. Microsoft word was used to type the research project report and this manual.

4 Environment set up

This is where the setting up of the colab environment is done, steps followed below permit code executing for the purpose of our research study. They are also illustrated by images: As within the other parts of the book, the individual steps are also supplemented with images. gained a better view of the processes followed.

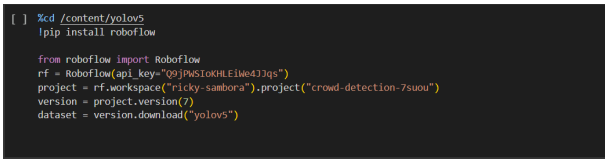

Figure 3: Dataset dowloading code

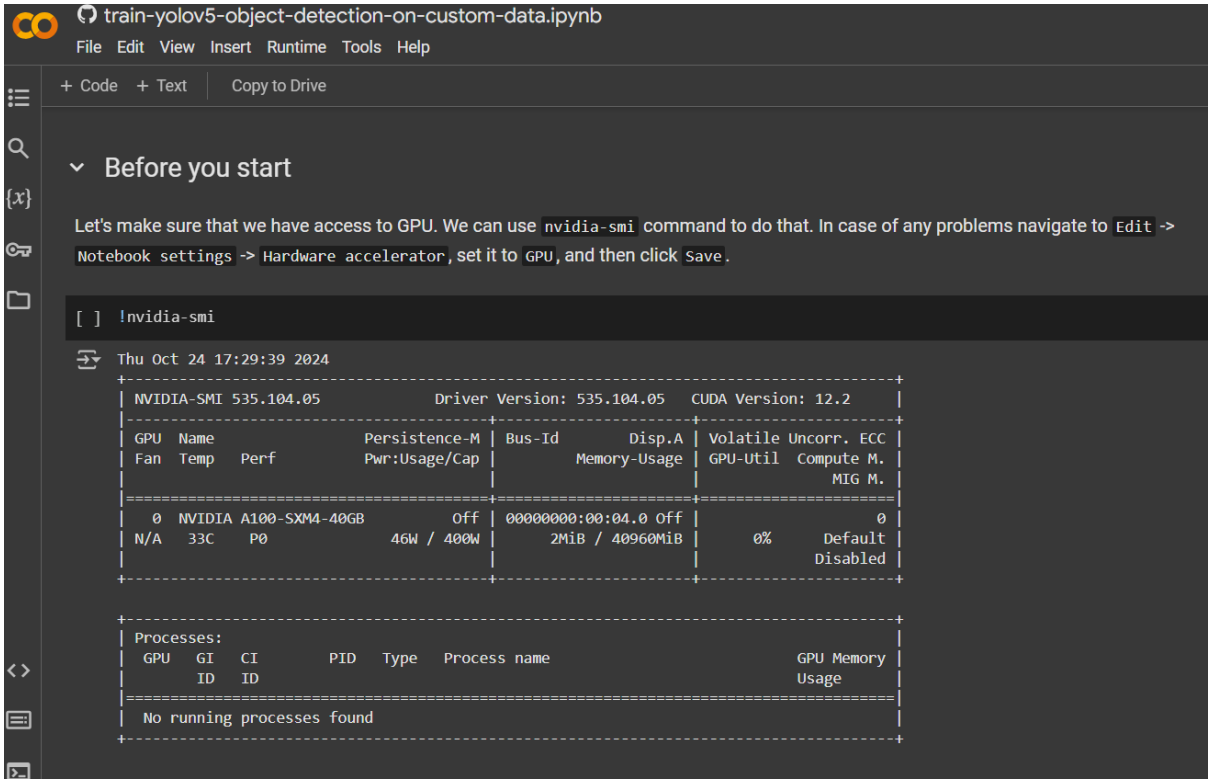

Figure 1: Environment setup

5 Data selection process

I have downloaded dataset from kaggle and done labelling and annotation from roboflow which is of 5730 images.

```
[ ] %cd /content/yolov5
    !pip install roboflow

    from roboflow import Roboflow
    rf = Roboflow(api_key="Q9jPWSIoKHLEiWe4JJqs")
    project = rf.workspace("ricky-sambora").project("crowd-detection-7suou")
    version = project.version(7)
    dataset = version.download("yolov5")
```

Figure 2: Dataset preperation

After adding the roboflow code it will automatically download the data

6 Installing Libraries

The following libraries have been imported and used to data preprocess, visualize, train , evaluate the models.

1. roboflow,
2. pandas
3. numpy
4. matplotlib
5. ultralytrics
6. seaborn
7. tensorflow
8. torchvision
9. os
10. pickle
11. git

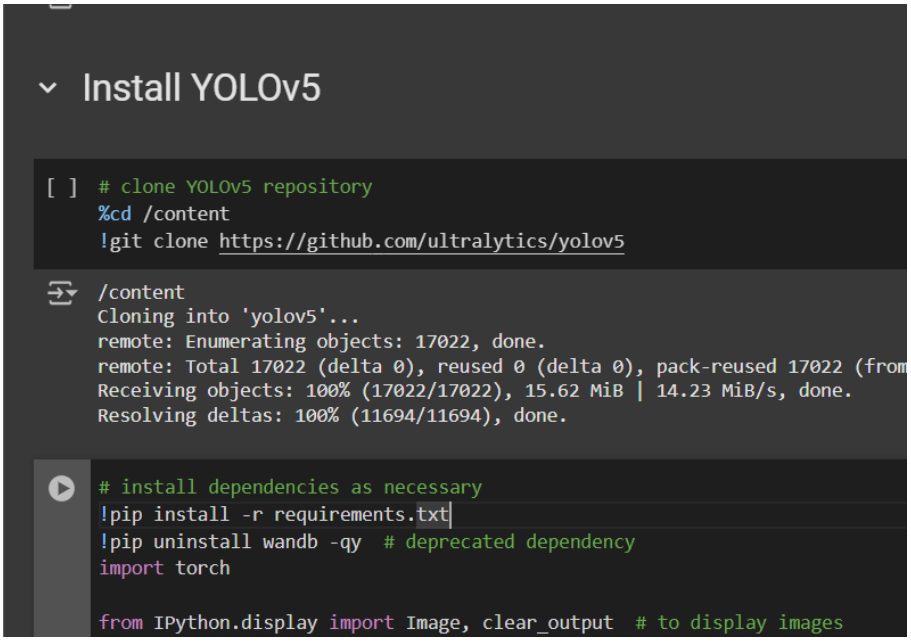

Figure 4: Installing framework

7 Implementation and using the code files.

7.0.1 Uploading images

Next, add the data to your newly created project. You can do it via API or through our web interface.

If you drag and drop a directory with a dataset in a supported format, the Roboflow dashboard will automatically read the images and annotations together.

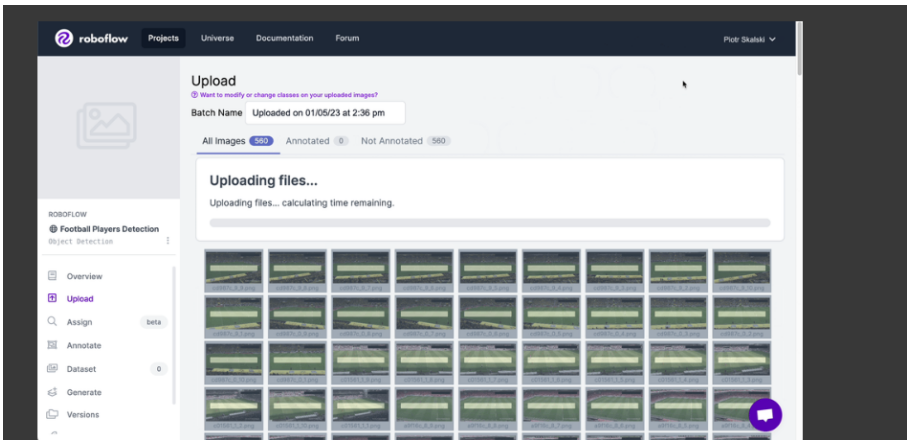

Figure 5: Data Labelling

8 Train Custom YOLOv5 YOLOv8 Detector

8.0.1 Next, we'll fire off training!

Here, we are able to pass a number of arguments:

- **img:** define input image size
- **batch:** determine batch size
- **epochs:** define the number of training epochs. (Note: often, 3000+ are common here!)
- **data:** set the path to our yaml file
- **cfg:** specify our model configuration
- **weights:** specify a custom path to weights. (Note: you can download weights from the Ultralytics Google Drive folder)
- **name:** result names
- **cache:** cache images for faster training

```
# train yolov5s on custom data for 25 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py \
  --img 416 \
  --batch 16 \
  --epochs 25 \
  --data {dataset.location}/data.yaml \
  --weights yolov5s.pt \
  --name yolov5s_results  \
  --cache
```

Figure 6: Model Training

9 Evaluate Custom YOLOv5 Detector Performance

You can view the training graphs associated with a training job in the `\href{https://colab.research.` folder.

Training losses and performance metrics are also saved to Tensorboard and also to a logfile defined above with the –**name** flag when we train. In our case, we named this `yolov5s_results`.

Note from Glenn: Partially completed `results.txt` files can be plotted with `from utils.utils imp`

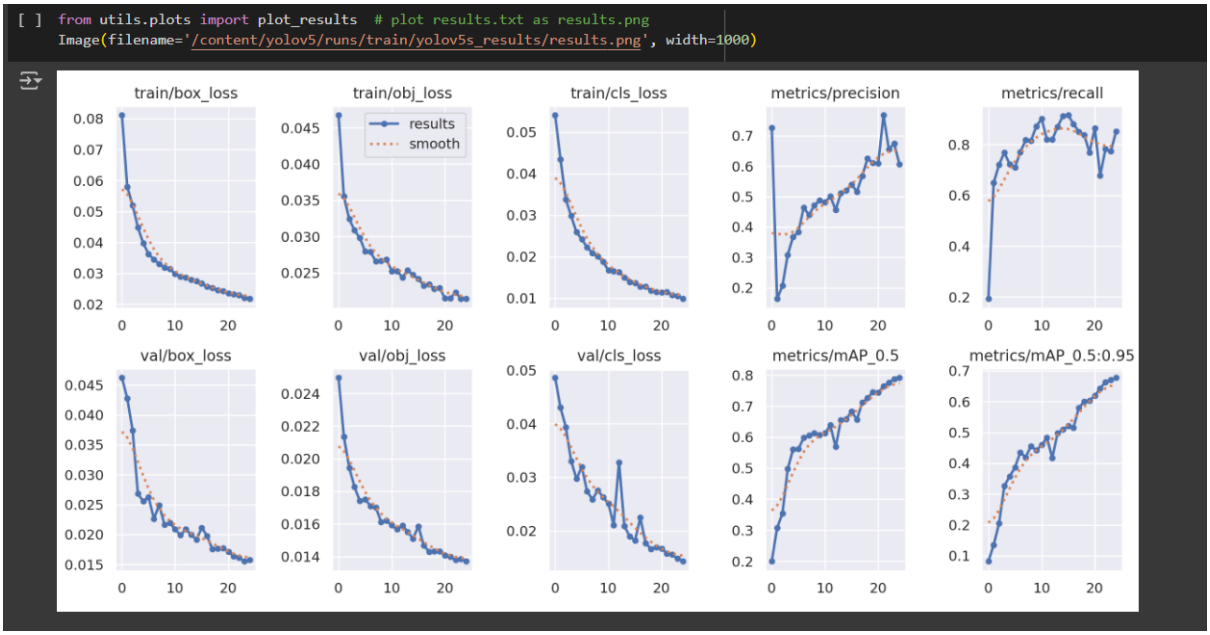

Figure 7: Evaluation and Results

9.0.1 Inference with trained weights

Next, we can run inference with a pretrained checkpoint on all images in the `test/images` folder to understand how our model performs on our test set.

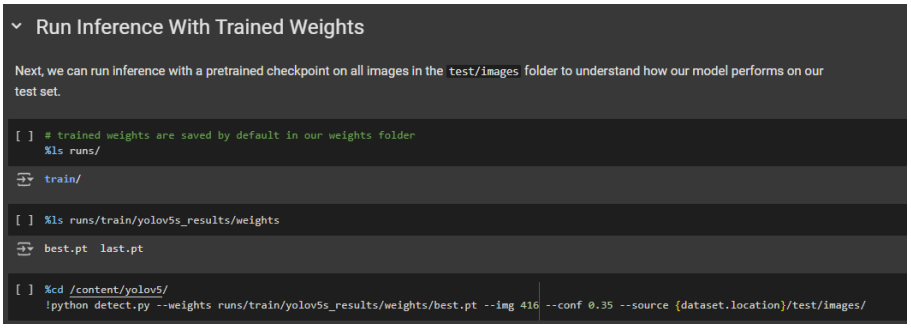

Figure 8: Inference

References