Configuration Manual

MSc Research Project
Data Analytics

Santosh Kumar Reddy Lekkala
Student ID: x22125736

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Santosh Kumar Reddy Lekkala
Student ID:	x22125736
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Christian Horn
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	730
Page Count:	5

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:	
Date:	31st January 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

Santosh Kumar Reddy Lekkala
x22125736

1 Introduction

This manual is intended to provide a comprehensive guide to installing and configuring the software application known as Pyton, Google Collab, Roboflow and Jupyter. This guide will walk you through the entire process from start to finish, covering topics such as system requirements, installation, configuration, and troubleshooting. Additionally, it will provide helpful tips to ensure that your installation and configuration are successful.

2 Hardware Details

Hardware is the physical components of a computer system, such as the central processing unit (CPU), memory, storage devices, network cards, motherboards, and other components. Below are the details used in this project in figure 1. Local environment has been used in the initial phase of the project to sort the data and its visualization if any.

Device name	Reacher
Processor	AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz
Installed RAM	16.0 GB (15.4 GB usable)
Device ID	6370235E-6167-45DA-9B29-89325FBE29F1
Product ID	00342-42610-27152-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Figure 1: Hardware Details

3 Software Details

We have made use of Google Colab and Roboflow software in this research. Other tools used will be described below.

- Programming Language - Python 3.10.12
- Cloud IDE - Google Colab
- Local IDE - Jupyter Notebook 6.4.8
- Annotation Tool - Roboflow[1]

3.1 Google Colab

1. Sign-in to your Google Account and go to Google Colab.
2. Create a new notebook or open an existing notebook that has been created earlier.
3. In the Google Collab Toolbar, as shown in figure 2, select Runtime -¿ Change Runtime type and choose the type of instance you require.

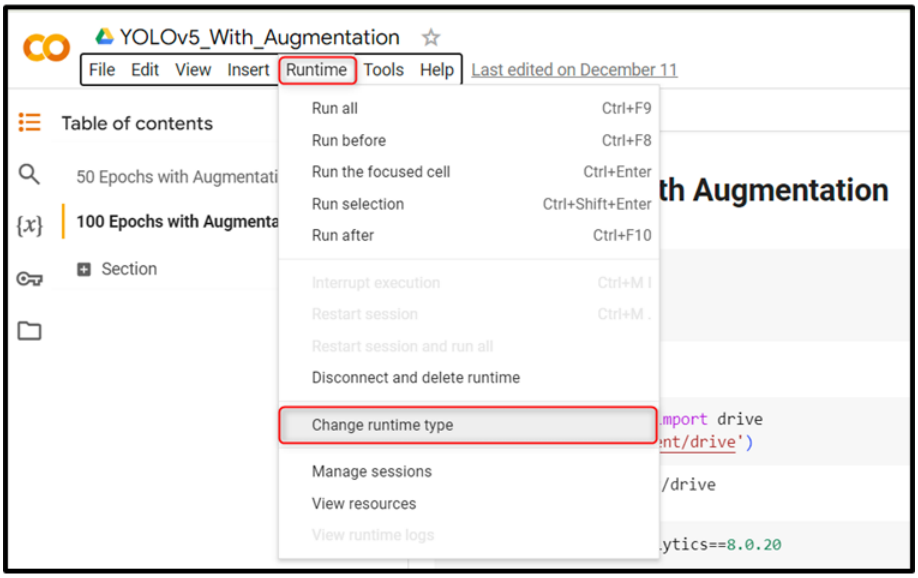


Figure 2: Changing Runtime

4. Select T4 GPU and click save, as shown in figure 3.
5. We need to import few packages that will help in the changing directories as and when required, as shown in figure 4.
6. Next we need to connect our Google Drive to the Google Colab application, which can be done using the code in figure 5.
7. We have to install Ultralytics Package to train our YOLOv8 Model in Colab environment and some checks need to be done to ensure all dependent packages have been installed, as can be seen figure 6

[1] https://app.roboflow.com/

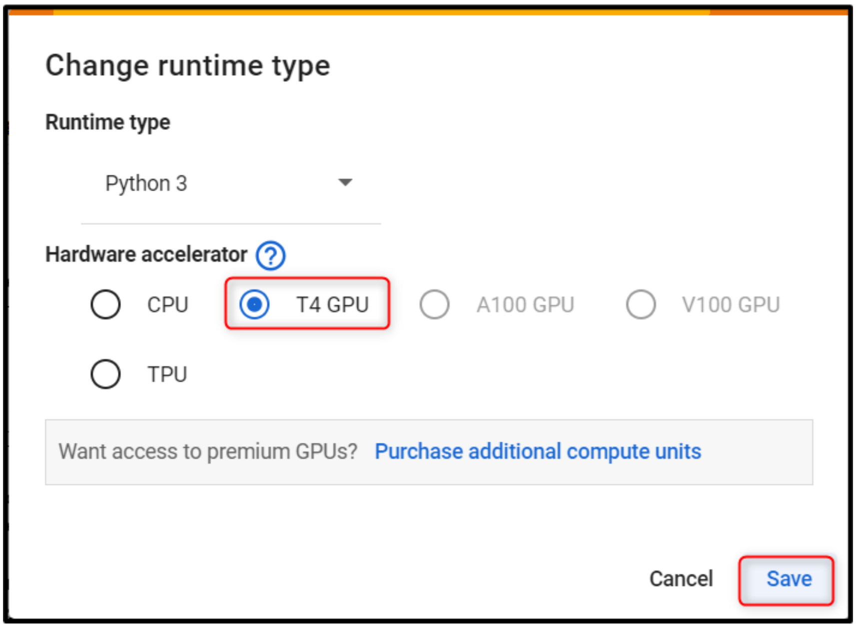


Figure 3: Selecting T4 GPU

```
import os
HOME = os.getcwd()
print(HOME)

/content
```

Figure 4: Directory Package

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

Figure 5: Google Drive Connection

```
[ ] !pip install ultralytics==8.0.20

    from IPython import display
    display.clear_output()

    import ultralytics
    ultralytics.checks()

    Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
    Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.9/78.2 GB disk)
```

Figure 6: Ultralytics Package installation

8. This will complete the initial setup of the environment required for running our notebook.

4 Dataset

The Dataset has been downloaded from Kaggle site and it was manually cleaned to remove unwanted images. The link for dataset is Kaggle[2]. The dataset consists of a total of 12GB of data where images are around 12000 and the infomation about them 12000 excel files which have then been pre-processed before augmenting them.

4.1 Dataset Pre-Processing

The images collected have been then annotated using a tool known as Roboflow[3]. The image to be annotated has to be uploaded on their webapp and can be seen in the figure 7. We then have to create bounding box as required covering the whole of the object being annotated, in this case a fighter plane. Next, the appropriate class has to be selected and classes created can be seen in8

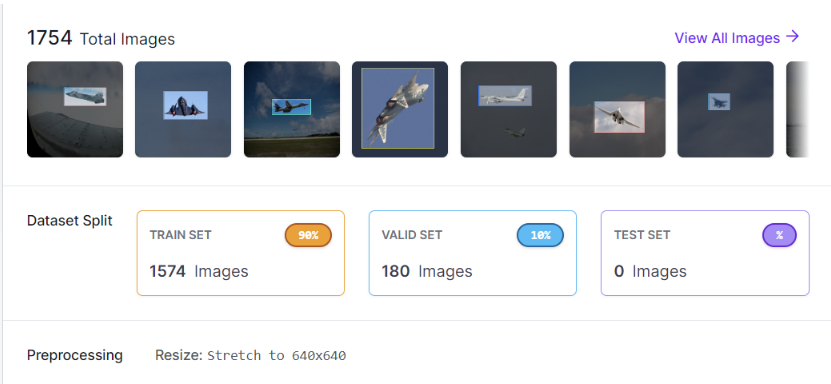


Figure 7: Annotated Images

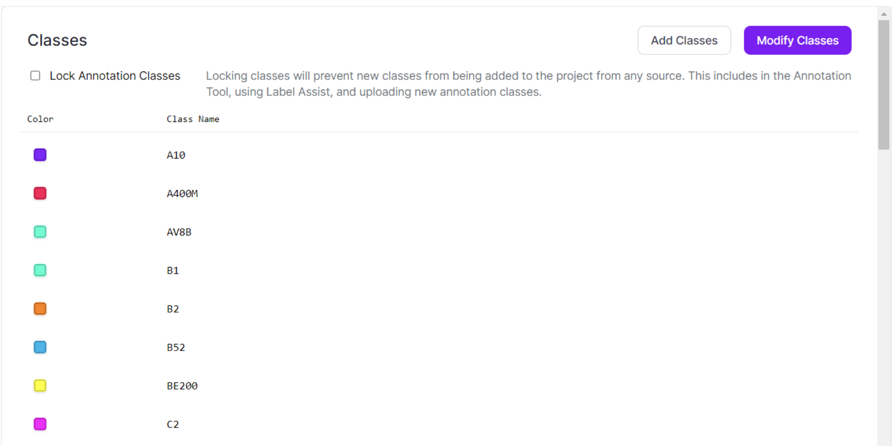


Figure 8: Different classes While Annotating images

After all images are annotated we will then split the images in random into training, validation and testing sets. Their ratios are 70:20:10respectively.

[2]https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset/data
[3]https://app.roboflow.com

4.2 Data Augmentation

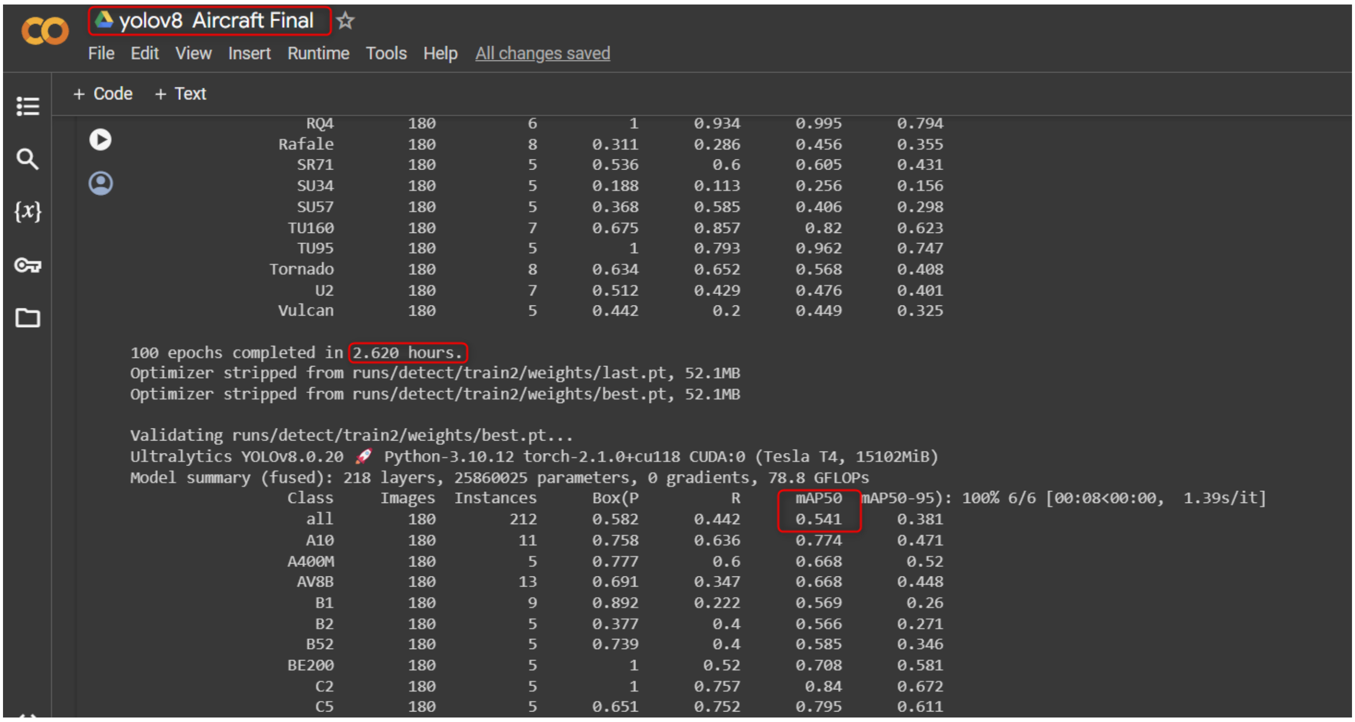
We have augmented our data in our notebook which can be seen in figure 9. As observed, we have used Adam optimizer, batch size of 32, the images have been rotated by 90 degrees and flipped vertically and horizontally. The number of epochs have been changed for different iterations for comparisons.

```
!yolo task=detect mode=train model=yolov8m.pt data=data.yaml plots=True imgsz=640 lr0=0.001
 optimizer='Adam' batch=32 epochs=100 degrees=90 scale=0.2 shear=0.2 flipud=0.25 fliplr=0.25 workers=10
```

Figure 9: Augmentation used during training

5 Modelling

This research required multiple iterations of training our model using different and epochs and with or without augmentation. In figure 10, we can see that a 100 epoch run without augmentation for a Yolov8 model required around 2.6 hour.




```
                   RQ4        180          6          1      0.934      0.995      0.794
                Rafale        180          8      0.311      0.286      0.456      0.355
                  SR71        180          5      0.536        0.6      0.605      0.431
                  SU34        180          5      0.188      0.113      0.256      0.156
                  SU57        180          5      0.368      0.585      0.406      0.298
                 TU160        180          7      0.675      0.857       0.82      0.623
                  TU95        180          5          1      0.793      0.962      0.747
               Tornado        180          8      0.634      0.652      0.568      0.408
                    U2        180          7      0.512      0.429      0.476      0.401
                Vulcan        180          5      0.442        0.2      0.449      0.325

100 epochs completed in 2.620 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 52.1MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 52.1MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25860025 parameters, 0 gradients, 78.8 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:08<00:00,  1.39s/it]
                   all        180        212      0.582      0.442      0.541      0.381
                   A10        180         11      0.758      0.636      0.774      0.471
                 A400M        180          5      0.777        0.6      0.668       0.52
                  AV8B        180         13      0.691      0.347      0.668      0.448
                    B1        180          9      0.892      0.222      0.569       0.26
                    B2        180          5      0.377        0.4      0.566      0.271
                   B52        180          5      0.739        0.4      0.585      0.346
                 BE200        180          5          1       0.52      0.708      0.581
                    C2        180          5          1      0.757       0.84      0.672
                    C5        180          5      0.651      0.752      0.795      0.611
```

Figure 10: Time required for a full run of 100 epochs