Configuration Manual

MSc Research Project
Artificial Intelligence

Hemaraju Padavala
Student ID: x22185011

School of Computing
National College of Ireland

Supervisor: Muslim Jameel Syed

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……..................hemaraju Padavala.............................

Student ID: ....x22185011......................................….

Programme: ....…...... Artificial Intelligence.....................…...... **Year:** ......2023.........

Module: ....MSc Research project...............................…......

Lecturer: .... Muslim Jameel Syed ....

Submission Due Date: ....31/01/2024...

Project Title: ... Performance Evaluation of Underwater Plastic Detection Model Under Diverse Environmental Conditions.........

Word Count: ....997...... **Page Count:** ....12.........…......

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: ....P.Hemaraju...

Date: ....31/01/2024...

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

Hemaraju Padavala
Student ID: x22185011

1 Introduction

Configuration Manual of "Comparison and Analysis of the Three Object Detection Model". You will find this guide as it helps you to compare three object detection models. This guide is intended for any person who intends to know the advantages and disadvantages of these models.

2 Prerequisites

1. A Google Drive account with at least 15GB free space to store data and models
2. Access to Google Colab to run Python code and GPU-accelerated workloads
3. Availability of the T4 GPU in Google Colab(Google, n.d.)which greatly speeds up training and evaluation

To check for T4 GPU availability in Colab:
1. Open a Colab notebook
2. Click Runtime > Change runtime type
3. Under Hardware accelerator, select GPU and choose T4 if available

With storage, Colab access, and a suitable GPU in place, you'll have the key prerequisites to configure and run your analysis comparing different object detection models.

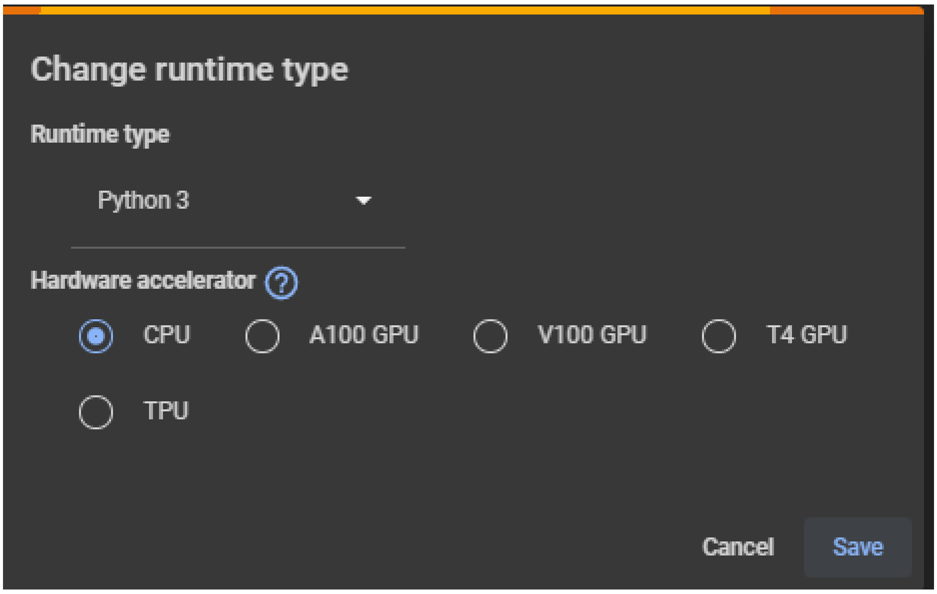


Figure 1 Hardware availability in colab

3 Hardware Requirement

- We don't need to set up any hardware on our end because the object detection experiments are being run in Google Colab.
- Google Colab gives you free access to powerful computing resources in the cloud, such as GPUs, for running machine learning workloads. In particular, we chose the T4 GPU option for the Colab runtime.
- The T4 GPU has tensor cores that are specifically designed for deep learning. It works very quickly and well for training neural networks and drawing conclusions from large datasets.
- Because we are using Colab's T4 GPU backend, we don't have to buy GPU servers in-house just for this analysis. Colab takes care of all the hardware infrastructure and setup that is needed to quickly train models.
- We only write and run the Python scripts in Colab notebooks; Google takes care of the GPU-enabled cloud infrastructure that is needed for a fast, responsive runtime.
- We can now focus on modeling and evaluating object detection without having to worry about getting, setting up, and maintaining hardware.

4 Setup to Run the Model

4.1 Faster RCNN and CenterNet

Step 1: Install Required Libraries

Open a Colab notebook and clone the TensorFlow Models repository(Tensorflow, n.d.)

by running "!git clone https://github.com/tensorflow/models". This will give access to the object detection code.

```
# clone the tensorflow models on the colab cloud vm
!git clone --q https://github.com/tensorflow/models.git

#navigate to /models/research folder to compile protos
%cd models/research

# Compile protos.
!protoc object_detection/protos/*.proto --python_out=.

# Install TensorFlow Object Detection API.
!cp object_detection/packages/tf2/setup.py .
!python -m pip install .
```

Figure 2 Tensorflow Model Libray installation

Next, install TensorFlow 2.13 with “! pip install tensorflow==2.13".

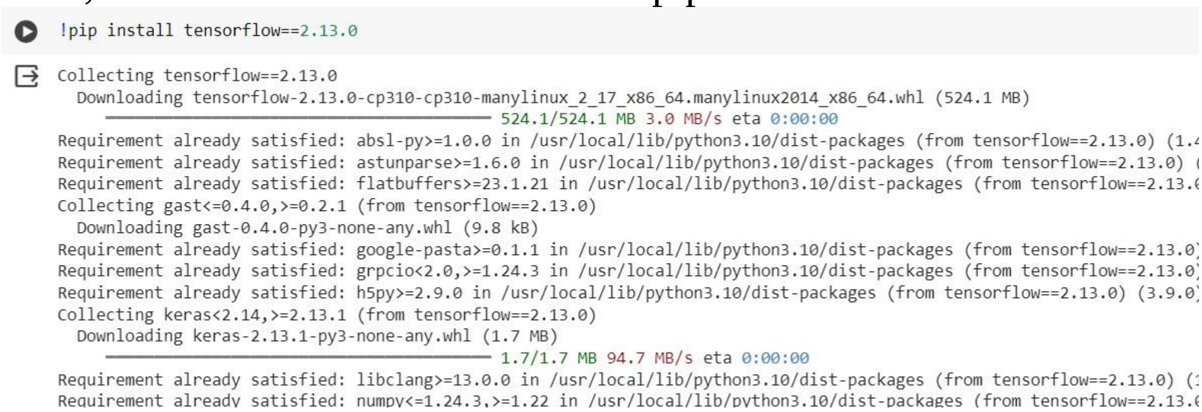


Figure 3 Tensorflow Installation

Step 2: Connect Google Drive Run "from google .colab import drive" and "drive. mount('/content/drive')" to mount Google Drive. This will prompt you to authenticate and provide access to Drive where data and models are stored.

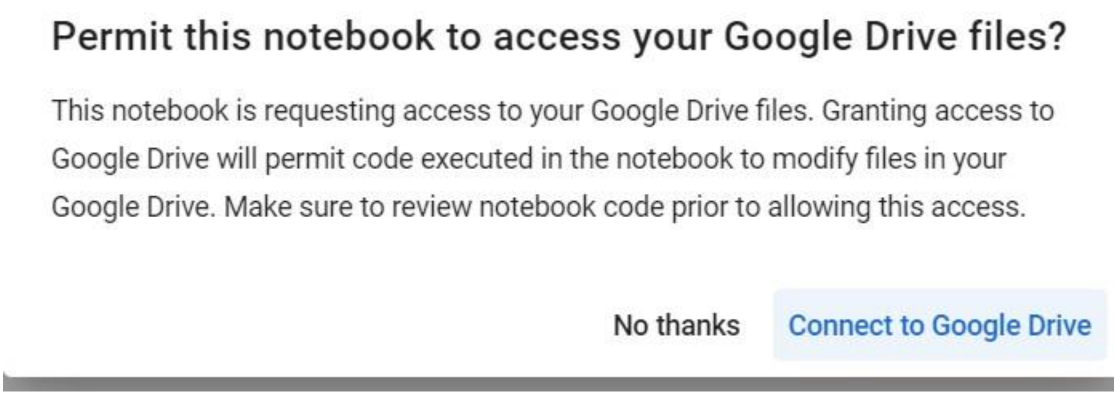


Figure 4 Connecting google drive in colab

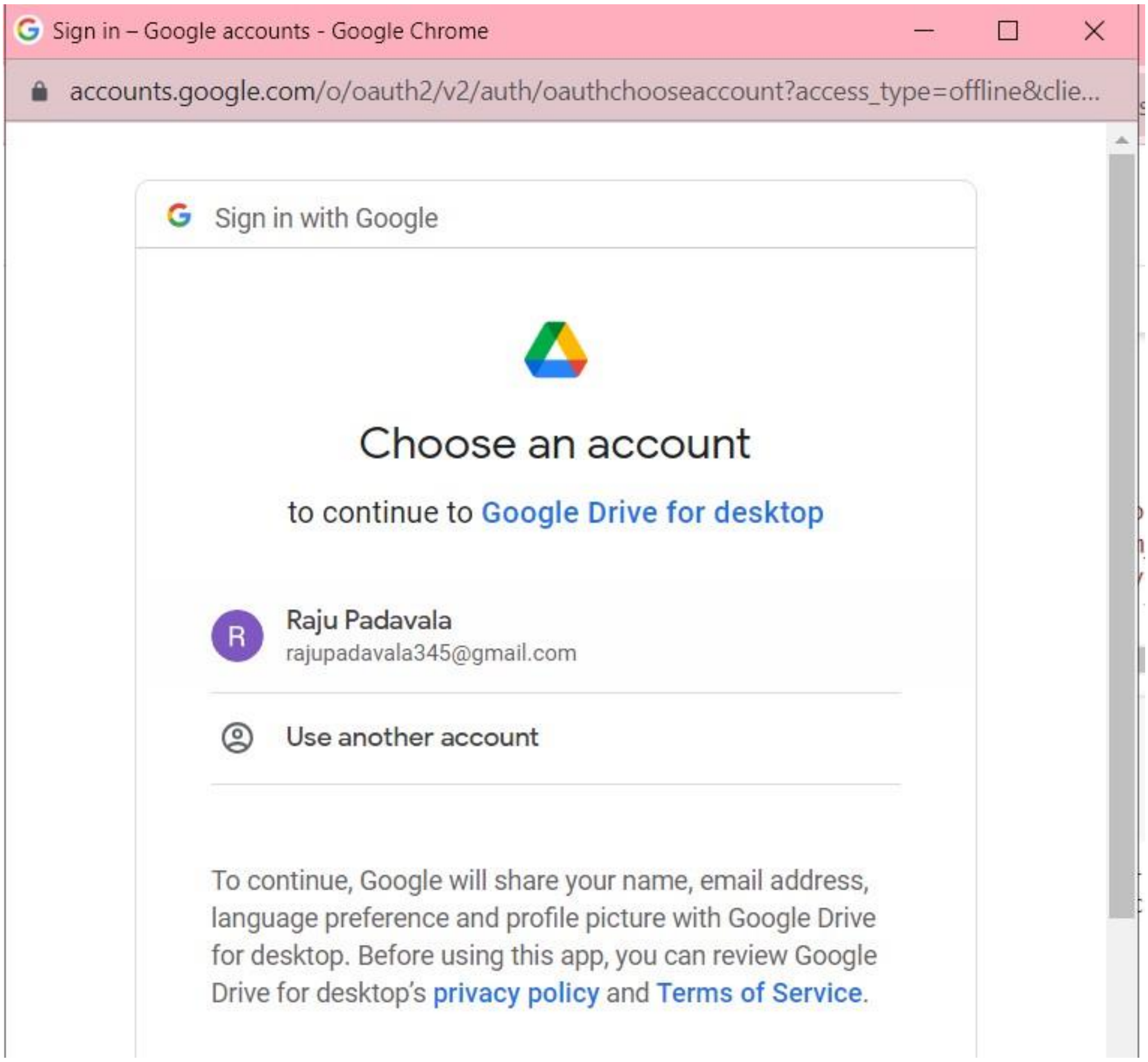


Figure 5 Prompt for Selecting Google Drive Account

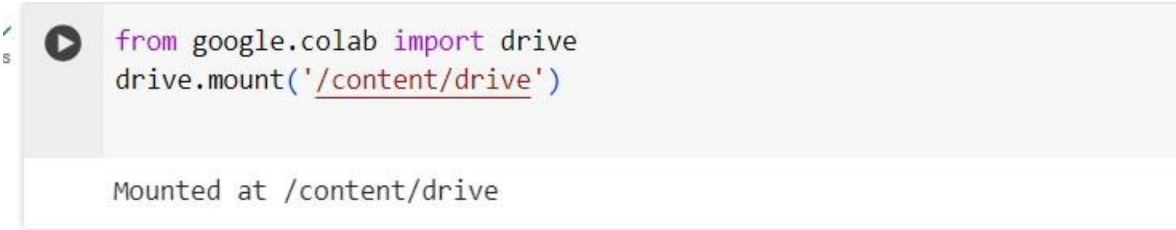


Figure 6 Drive Mounted

Step 2: Load Saved Model

2.1 Import all the needed libraries

```python
#Loading the saved_model
import tensorflow as tf
import time
import numpy as np
from PIL import Image
from google.colab.patches import cv2_imshow
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
```

Figure 7 Importing Required Libraries

2.2 Give the path to the saved model file where it was saved in Google Drive in detection.ipynd file;

2.3 Give the Label map file in detection.ipynd so that the object is correctly labelled

2.4 Give the image path in detection.ipynd to find the object in the picture

Figure 8 Reference for Providing paths

4.3 Outputs of the Faster RCNN and CenterNet Model

Figure 9 Sample output of CenterNet Model

Figure 10 Output from Faster RCNN Model

4.4 Loading YOLO V3 Model

4.4.1 Libraries Installation

1. Navigate to the content directory: "%cd /content".This will set our working directory to the content folder in Colab.

2. Clone the yolov3 repository (Ultralytics, n.d.) from GitHub: “! git clone https://github.com/ultralytics/yolov3.git".This will download the yolov3 code into a folder called "yolov3".

3. Move into the yolov3 directory:"%cd yolov3". Now we are inside the yolov3 folder that was just cloned.

4. Install the required Python packages:”! pip install -r requirements.txt". This will install all the necessary packages to run yolov3 based on the requirements file.

5. Return back to the content directory:"%cd /content/yolov3”. So now we're back in the content folder in the yolov3 folder and ready to use the yolov3 code.

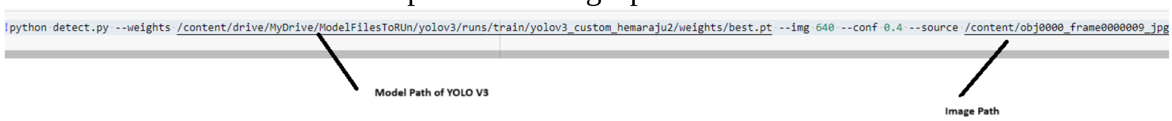
```
%cd /content
!git clone https://github.com/ultralytics/yolov3.git
%cd yolov3
!pip install -r requirements.txt
%cd /content/yolov3
```

```
/content
Cloning into 'yolov3'...
remote: Enumerating objects: 11001, done.
remote: Counting objects: 100% (923/923), done.
remote: Compressing objects: 100% (445/445), done.
remote: Total 11001 (delta 631), reused 730 (delta 477), pack-reused 10078
Receiving objects: 100% (11001/11001), 9.88 MiB | 5.83 MiB/s, done.
Resolving deltas: 100% (7430/7430), done.
/content/yolov3
Collecting gitpython>=3.1.30 (from -r requirements.txt (line 5))
  Downloading GitPython-3.1.40-py3-none-any.whl (190 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 190.6/190.6 kB 4.4 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=3.3 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 6)) (3
Requirement already satisfied: numpy>=1.22.2 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 7)) (1.2
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 8
Collecting Pillow>=10.0.1 (from -r requirements.txt (line 9))
```

Figure 11 Libraries Installation for YOLOV3

4.4.2 Running YOLO-V3

Step 1: Provide the saved model path and image path after 4.4.1 and run the inference:



4.4.3 Output of the YOLO V3

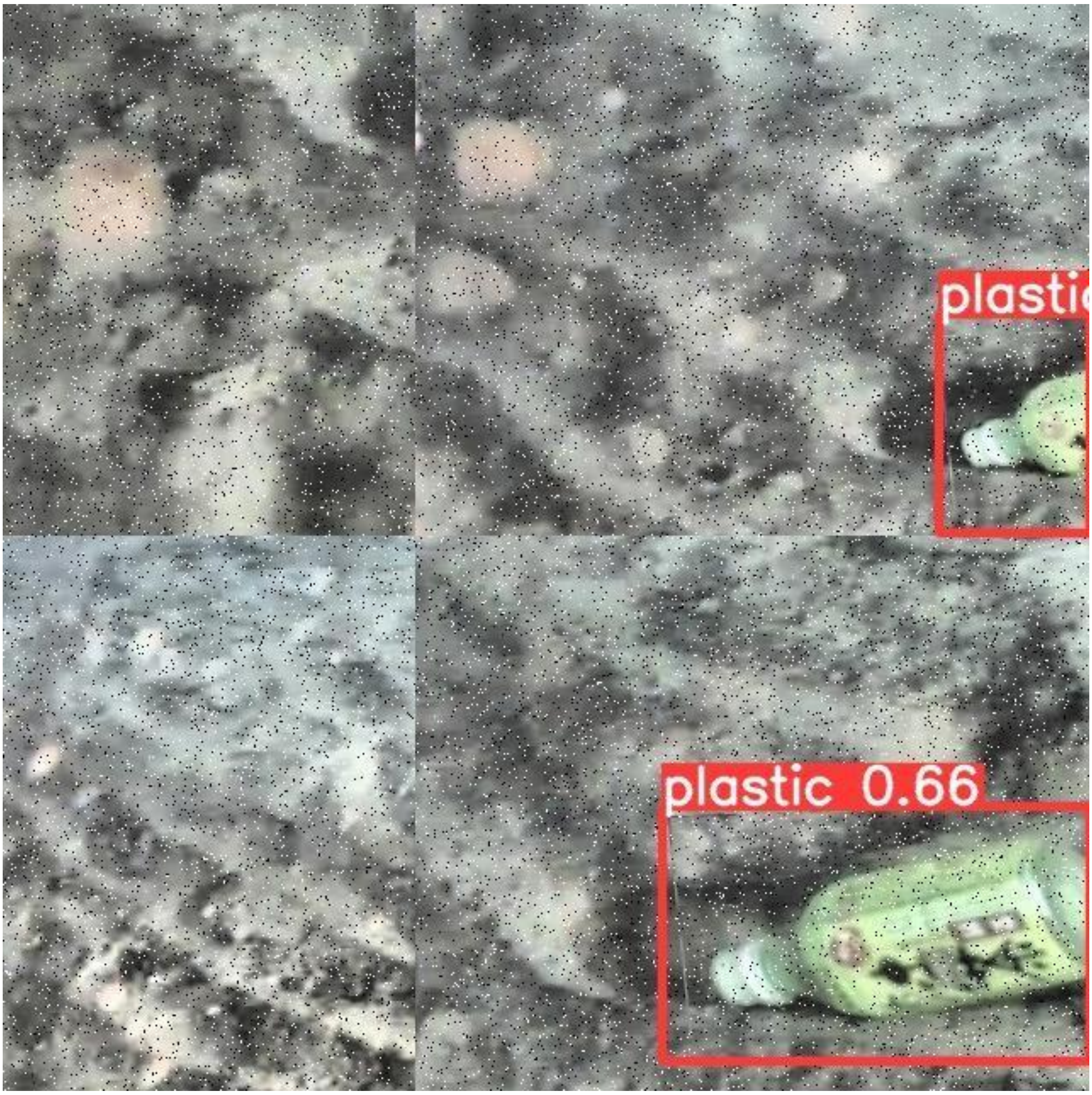


Figure 12 YOLO V3 Sample Output

References

Google *Colaboratory*. Available from:

https://colab.research.google.com/?utm_source=scs-index [Accessed 13/12/23].

Tensorflow *models/research/object_detection/g3doc/tf2_detection_zoo.md at master tensor flow/models*. [Online] GitHub. Available from: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md [Accessed 13/12/23].

Ultralytics *YOLOv3* [Online] Ultralytics YOLOv3 Docs. Available from: https://docs.ultralytics.com/models/yolov3/ [Accessed 13/12/23].