

Configuration Manual

MSc Research Project
Data Analytics

Iswarya Yogeashwaran
Student ID: X20155034

School of Computing
National College of Ireland

Supervisor: Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing



| | |
|-----------------------------|----------------------|
| Student Name: | Iswarya Yogeashwaran |
| Student ID: | X20155034 |
| Programme: | Data Analytics |
| Year: | 2021-2022 |
| Module: | MSc Research Project |
| Supervisor: | Giovani Estrada |
| Submission Due Date: | 31/01/2022 |
| Project Title: | Configuration Manual |
| Word Count: | 1132 |
| Page Count: | 7 |

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 2022 |

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

| | |
|--|--------------------------|
| Attach a completed copy of this sheet to each project (including multiple copies). | <input type="checkbox"/> |
| Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies). | <input type="checkbox"/> |
| 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. | <input type="checkbox"/> |

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

Iswarya Yogeashwaran
X20155034

1 Introduction

This configuration manual contains all the required documents and instructions for the Domestic Waste Segregation using Deep neural network approaches solution that was built as phase of the research study. All procedures for reproducing the outcomes are included in the report.

2 System Configuration

2.1 Hardware Requirements

The trained model solution is being executed in the Google Collaboratory environment, because the model requires more amount of CPU and GPU resources. The hosting machine's specifications are listed below:

The Google collaboratory environment gives you better Graphics and Memory, as well as 12 hours of constant GPU usage, which is plenty of to accomplish the entire training phase. The Colab comes with unlimited GPU performance, which speeds up overall completion time.

- Latest OS: Windows 10
- RAM: 8 gigabyte (GB) 64-bit
- Hard disk space: 256 GB SSD for 64-bit OS
- Graphics card: Nvidia GeForce GTX 1650
- Display: 800 x 600

2.2 Software Requirements

Hosted Google Colaboratory environment: The Google collaboratory environment is used to build the model. The project code is implemented in the google colab, which is written in Python 3.6.7. Custom CNN, MobileNet, Xception, and VGG16 are the deep learning models that were build by importing the necessary python library packages. These libraries were already installed in the google colab environment.

IDE: The recommended solution is carried out using the Visual Studio Code IDE. This IDE was selected because it offers free accessibility and free Microsoft Visual Studio Code experience that runs fully in the browser, allowing to explore source code repositories and make quick and safe code changes. Using this IDE. the generated h5 file from the Google Colab is integrated with the web application built with HTML, CSS, and JavaScript and powered by the Python Flask framework.

The python library packages involved in the model building process are

1. TensorFlow 2.x
2. Python Flask
3. Keras
4. Numpy
5. Matplotlib
6. pandas
7. tqdm
8. glob

The Softwares and their versions which were used in this project are,

- Google Colaboratory
- Visual Studio

Programming Languages: Python, HTML, CSS, Java Script is employed to implement the planned design. 70% of the project is implemented using python and python incorporated frameworks. The project code is applied in the local system and the dataset has loaded from google drive. The packages used in the project code are shown in the Figure 1.

| <i>S.no</i> | <i>Packages</i> |
|-------------|-------------------|
| <i>1</i> | <i>io</i> |
| <i>2</i> | <i>oprnpysl</i> |
| <i>3</i> | <i>numpy</i> |
| <i>4</i> | <i>pandas</i> |
| <i>5</i> | <i>matplotlib</i> |
| <i>6</i> | <i>seaborn</i> |
| <i>7</i> | <i>tensorflow</i> |
| <i>8</i> | <i>keras</i> |
| <i>9</i> | <i>Flask</i> |
| <i>10</i> | <i>werkzeug</i> |
| <i>11</i> | <i>gevent</i> |

Figure 1: Packages

3 Data Description

The Waste dataset is available at Kaggle Website, that contains 22500 images. This dataset is available for public access. It includes both Biodegradable and Non-Biodegradable images in two separate folders, which is in JPEG format. This dataset file is named as Waste_classification_datasets, which is present in the google drive that was mounted in the initial step. In the Figure 2, website for dataset is shown.

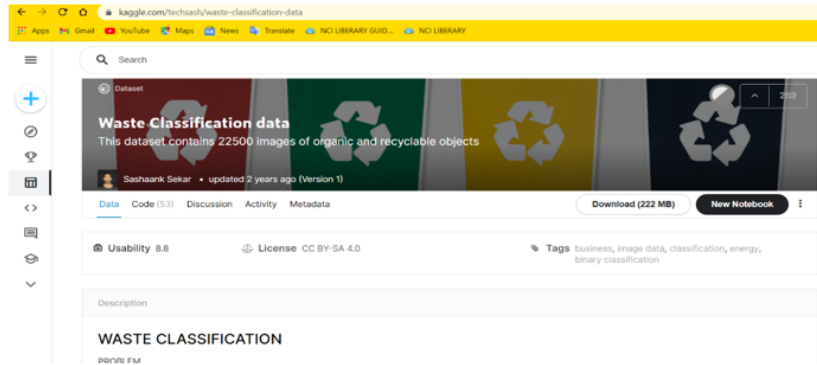


Figure 2: Waste Dataset in Kaggle

4 Environment Setup

To conduct this study, the Google Colab environment must be established. The program and data files must be in location before the model can be properly implemented and tested. Thoroughly explained with the requirements outlined here.

Step 1: The version of tensorflow is checked before importing the dataset from the google drive, which can be seen in the figure 3. Since the data cannot be accessed from google colab to the local system that is collected from the Kaggle. All the data downloaded from the Kaggle is uploaded to the google drive.

```

1 import tensorflow
2 print(tensorflow.__version__)

2.7.0

```

Figure 3: Tensor Flow version

Step 2: Logged into Google account and the executed the two lines of code and acquired the permission code. Then, enter is hit after pasting the access code. If everything works smoothly, the result will be shown as "Mounted at /content/drive" as a response, which is shown in the figure 4. After mounting, disk folder is checked whether the drive is mounted properly using 'ls command'.

```

WAST_CLASSIFICATION.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text

[2] 1 from google.colab import drive
     2 drive.mount('/content/drive')

Mounted at /content/drive

```

Figure 4: Mounted drive

Step 3: In-order to import large amount of dataset to google Colab and execute

certain task quickly such as Data Pre-processing, Data augmentation, and Auto-tuning and to build Custom CNN and Transfer Learning models such as VGG16, MobileNet, Xception. The computing engine of the google colab is changed to GPU as shown in the below figure 5.

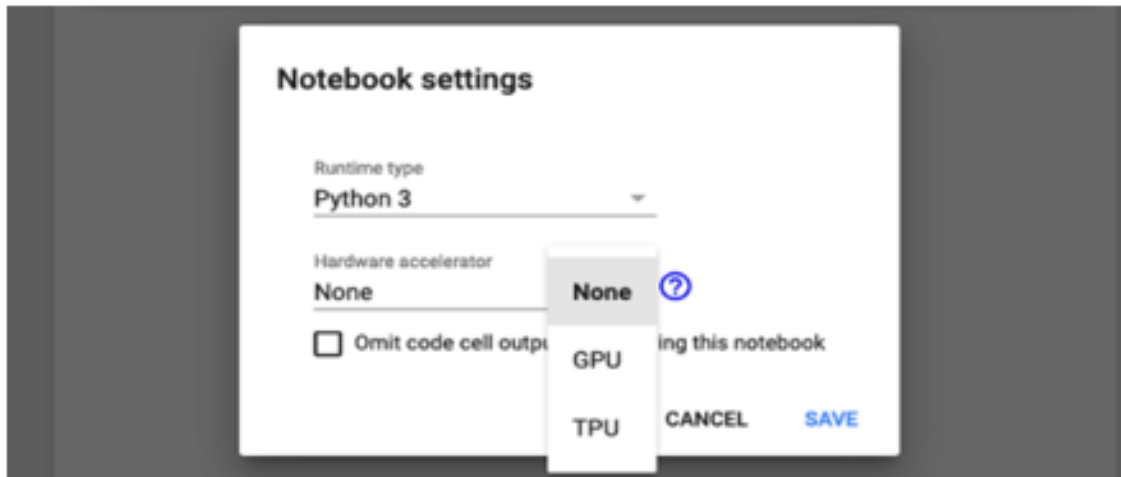


Figure 5: GPU setting

Step 4: All the essential packages are imported in the google colab which is shown in the figure ?? to execute tasks such as date pre-processing, data augmentation, Data normalization and finally building the architecture of the model and training it with the processes dataset in the figure 6

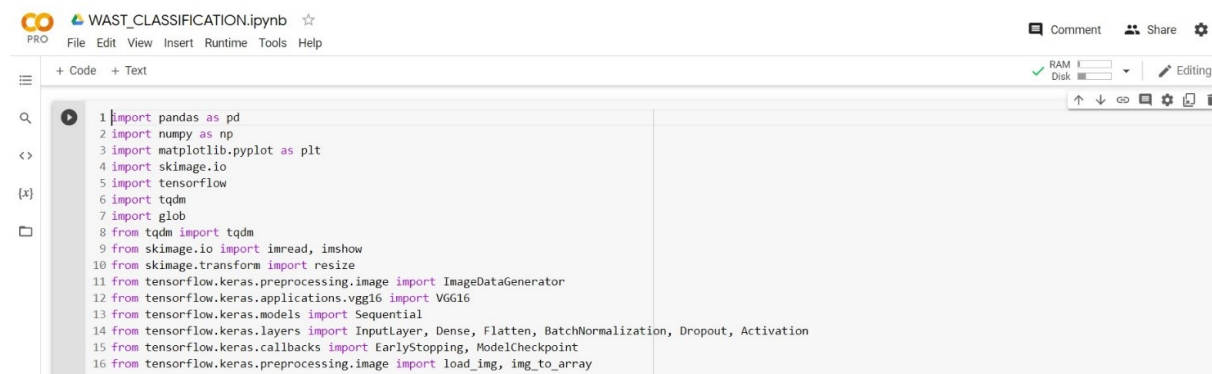


Figure 6: Essential packages

Step5: To train the transfer learning models the weights of each models are downloaded from the imagenet. After downloading the weights for all the transfer learning models the model is rained with the waste classification data with the final dense layer of 2 unit since it has two classes. The weights that is downloaded from the imagenet is shown in the figure 7.

Step6: The trained model is exported and saved in separate path file to integrate that path with front-end of the webpage using Flask. The model path created by saving the model is shown in the figure 10.

```
[ ] 1 preprocess_input = tf.keras.applications.vgg16.preprocess_input
      2
      3 rescale = tf.keras.layers.Rescaling(1./127.5, offset=-1)

[ ] 1 IMG_SHAPE = IMG_SIZE + (3,)

[ ] 1 base_model = VGG16(input_shape=(224,224,3),
      2 |           |           |           |           |           |           |
      3 |           |           |           |           |           |           |
      4 |           |           |           |           |           |           |
      5 |           |           |           |           |           |           |
      6 |           |           |           |           |           |           |
      7 |           |           |           |           |           |           |
      8 |           |           |           |           |           |           |
      9 |           |           |           |           |           |           |
     10 |           |           |           |           |           |           |
     11 |           |           |           |           |           |           |
     12 |           |           |           |           |           |           |
     13 |           |           |           |           |           |           |
     14 |           |           |           |           |           |           |
     15 |           |           |           |           |           |           |
     16 |           |           |           |           |           |           |
     17 |           |           |           |           |           |           |
     18 |           |           |           |           |           |           |
     19 |           |           |           |           |           |           |
     20 |           |           |           |           |           |           |
     21 |           |           |           |           |           |           |
     22 |           |           |           |           |           |           |
     23 |           |           |           |           |           |           |
     24 |           |           |           |           |           |           |
     25 |           |           |           |           |           |           |
     26 |           |           |           |           |           |           |
     27 |           |           |           |           |           |           |
     28 |           |           |           |           |           |           |
     29 |           |           |           |           |           |           |
     30 |           |           |           |           |           |           |
     31 |           |           |           |           |           |           |
     32 |           |           |           |           |           |           |
     33 |           |           |           |           |           |           |
     34 |           |           |           |           |           |           |
     35 |           |           |           |           |           |           |
     36 |           |           |           |           |           |           |
     37 |           |           |           |           |           |           |
     38 |           |           |           |           |           |           |
     39 |           |           |           |           |           |           |
     40 |           |           |           |           |           |           |
     41 |           |           |           |           |           |           |
     42 |           |           |           |           |           |           |
     43 |           |           |           |           |           |           |
     44 |           |           |           |           |           |           |
     45 |           |           |           |           |           |           |
     46 |           |           |           |           |           |           |
     47 |           |           |           |           |           |           |
     48 |           |           |           |           |           |           |
     49 |           |           |           |           |           |           |
     50 |           |           |           |           |           |           |
     51 |           |           |           |           |           |           |
     52 |           |           |           |           |           |           |
     53 |           |           |           |           |           |           |
     54 |           |           |           |           |           |           |
     55 |           |           |           |           |           |           |
     56 |           |           |           |           |           |           |
     57 |           |           |           |           |           |           |
     58 |           |           |           |           |           |           |
     59 |           |           |           |           |           |           |
     60 |           |           |           |           |           |           |
     61 |           |           |           |           |           |           |
     62 |           |           |           |           |           |           |
     63 |           |           |           |           |           |           |
     64 |           |           |           |           |           |           |
     65 |           |           |           |           |           |           |
     66 |           |           |           |           |           |           |
     67 |           |           |           |           |           |           |
     68 |           |           |           |           |           |           |
     69 |           |           |           |           |           |           |
     70 |           |           |           |           |           |           |
     71 |           |           |           |           |           |           |
     72 |           |           |           |           |           |           |
     73 |           |           |           |           |           |           |
     74 |           |           |           |           |           |           |
     75 |           |           |           |           |           |           |
     76 |           |           |           |           |           |           |
     77 |           |           |           |           |           |           |
     78 |           |           |           |           |           |           |
     79 |           |           |           |           |           |           |
     80 |           |           |           |           |           |           |
     81 |           |           |           |           |           |           |
     82 |           |           |           |           |           |           |
     83 |           |           |           |           |           |           |
     84 |           |           |           |           |           |           |
     85 |           |           |           |           |           |           |
     86 |           |           |           |           |           |           |
     87 |           |           |           |           |           |           |
     88 |           |           |           |           |           |           |
     89 |           |           |           |           |           |           |
     90 |           |           |           |           |           |           |
     91 |           |           |           |           |           |           |
     92 |           |           |           |           |           |           |
     93 |           |           |           |           |           |           |
     94 |           |           |           |           |           |           |
     95 |           |           |           |           |           |           |
     96 |           |           |           |           |           |           |
     97 |           |           |           |           |           |           |
     98 |           |           |           |           |           |           |
     99 |           |           |           |           |           |           |
    100 |           |           |           |           |           |           |

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [=====] - 0s 0us/step
58900480/58889256 [=====] - 0s 0us/step
```

Figure 7: Weights from Imagenet

```
import keras
import keras.utils
from keras import utils as np_utils
vgg_model.save("/content/drive/MyDrive/waste_classification_models_history/VGG16_model_dl.h5")
VGG_model_dl = keras.models.load_model("/content/drive/MyDrive/waste_classification_models_history/VGG16_model_dl.h5") #look for local saved file
```

Figure 8: Model Path

Step 7 The exported model is downloaded from the drive where it is generated in the given path. The Model.h5 file is then integrated with the app.py file. The app.py file integrates all the static HTML, CSS and Java Script files. If the app.py file is excited a local server is hosted were the web application can be accessed using the browser which is illustrated in the below figure 10.

```
PS C:\Users\Asus TUF\OneDrive\Desktop\mk_Dataset> & "C:/Users/Asus TUF/anaconda3/python.exe" "c:/Users/Asus TUF/OneDrive/Desktop/mk_Dataset/app.py"
2021-12-16 02:06:22.084455: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network
Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
2021-12-16 02:06:36.780322: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network
Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 141-027-580
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 9: Web application

The file structure of the web application is shown below in the figure 10. In the below file model_dl.h5 file is generated in the google drive where the google colab account is logged in. The downloaded h5 file is integrated with the app.py file where all the other files are integrated together. The app.py file should be executed for hosting the web application is shown in the figure 10

Step 8 After executing the code the web application shows up in the default browser

- The image has to be uploaded in the web application
- After uploading the waste image, a predict button shows up in the front end of the web application.
- Then if the predict button is hit, the image uploaded gets loaded to the deep learning model and the output response from the deep learning model after classification is shown in the front end of the web application.

```
#OPT = tensorflow.keras.optimizers.Adam(lr=0.0001)
base_learning_rate = 0.0001

vgg_model.compile(loss='binary_crossentropy',
                  metrics=['accuracy', 'Precision', 'Recall'],
                  optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate))

[ ] model_history=vgg_model.fit(train_generator,
                               validation_data=test_generator,
                               epochs = 6)
```

Figure 10: Integration

- This shows the real time performance of the web application.
- The below figure 11 shows the front end interface of the web application with the predicted image results.

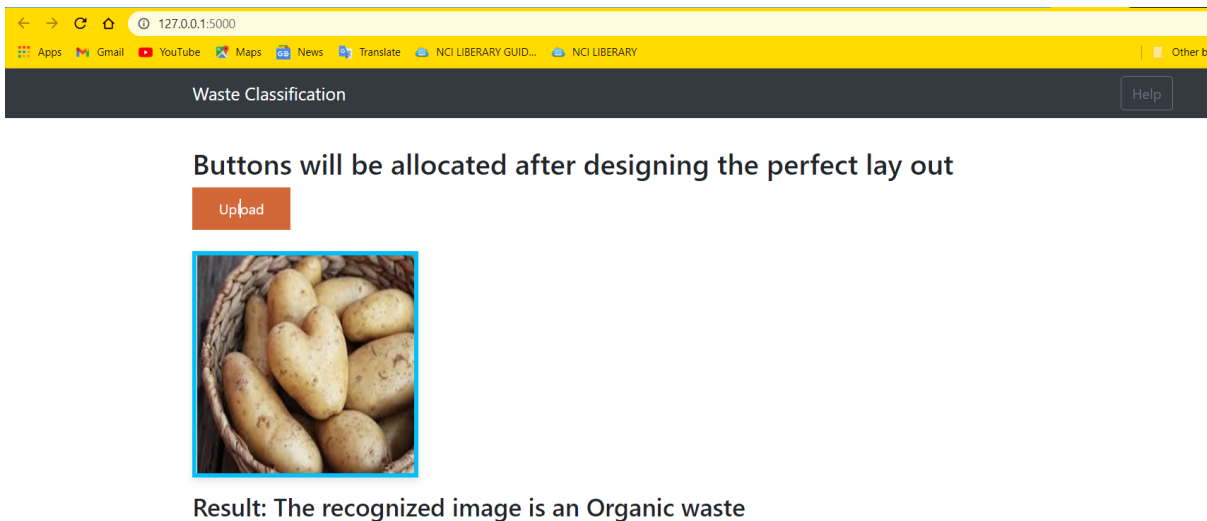


Figure 11: Web application

5 Training And Evaluating the model

After processing the image dataset and splitting it into two sections such as testing and training set. The datasets are loaded into the models architecture for training the model with waste classification dataset. The metrics that are assigned to evaluate the deep learning model are accuracy, precision and recall. The below figure 12 illustrates the above implementation. The below code is executed for training the model with the image dataset.


```
#OPT = tensorflow.keras.optimizers.Adam(lr=0.0001)
base_learning_rate = 0.0001

vgg_model.compile(loss='binary_crossentropy',
                  metrics=['accuracy', 'Precision', 'Recall'],
                  optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate))

[ ] model_history=vgg_model.fit(train_generator,
                               validation_data=test_generator,
                               epochs = 6)
```

Figure 12: Implementation

6 Making Predictions

After integration with the web application. The model is tested with the real time images by uploading it in the web application which is shown in the figure 13 . In this way the prototype is built and the model is evaluated in real time. The model is selected by comparing the processing time of all the models after integrating it with the web application.

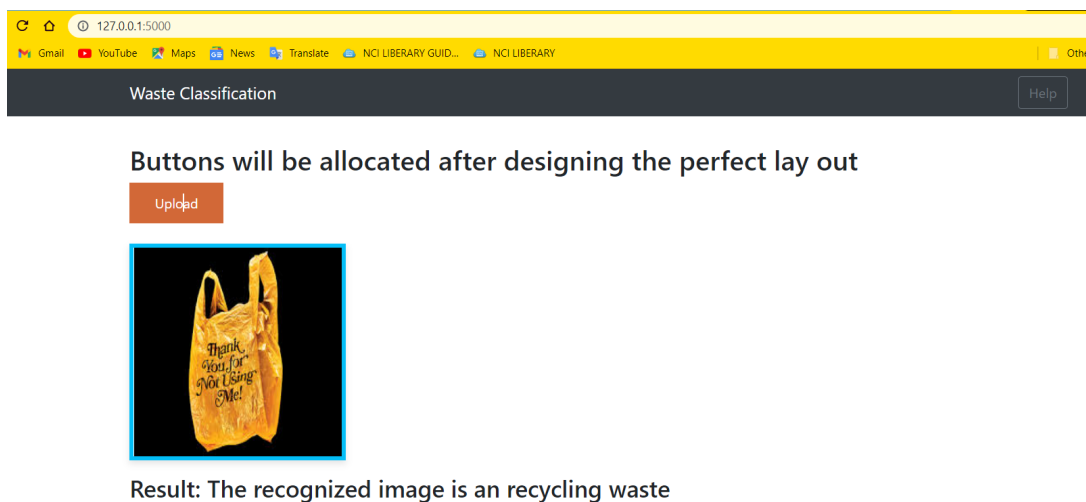


Figure 13: Implementation