

Configuration Manual

Plant Disease Detection using Deep Learning
MSc Research Project
MSc Data Analytics

Sayali Patil
Student ID: X20208162

School of Computing
National College of Ireland

Supervisor: Prashanth Nayak

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Sayali Patil
Student ID: x20208162
Programme: MSc in Data Analytics **Year:** 2021 – 2022
Module: Research Project
Lecturer: Prashanth Nayak
Submission Due Date: 15th August
Project Title: Plant Disease Detection using Deep Learning
Word Count: 730 **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: Sayali Patil

Date: 15th August

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

Sayali Patil
Student ID: x20208162

1 Introduction

This document details the procedures and steps that must be taken for this project to be successfully implemented. The hardware configuration used to run the code and the recommended setup are both specified. The processes involved in replicating the project's outcomes are detailed in the document, which may be utilized for future work.

2 System Specification

2.1 Hardware

The configuration of the system which was used for the implementation is stated below:

Processor – 11th Gen Intel, i5 @2.40Ghz

Storage – 256GB SSD + 1TB

Ram – 8GB

System type - 64-bit operating system, Windows 10 Home

Graphics card – Nvidia 2GB graphics card

2.2 Software

Jupyter Notebook is an open-source IDE that supports the Python programming language was used for the implementation. This notebook is accessible through the Anaconda application.

3 Download and Installation

3.1 Anaconda

Anaconda package was installed that supports Python and Jupyter Notebook needed for the implementation. Anaconda is an open-source package supporting thousands of open-source packages like Spyder, RStudio, and libraries. It can be easily downloaded from the website¹ as shown in figure 1. Anaconda is supported on Windows, macOS, and Linux.

¹ [Anaconda | The World's Most Popular Data Science Platform](#)

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.



Figure 1: Anaconda Download Page

Along with the installation of the anaconda package, the python version is also installed. After installation, the anaconda will display the screen shown in figure 2. Various IDE are available and Jupyter Notebook was used for this research.

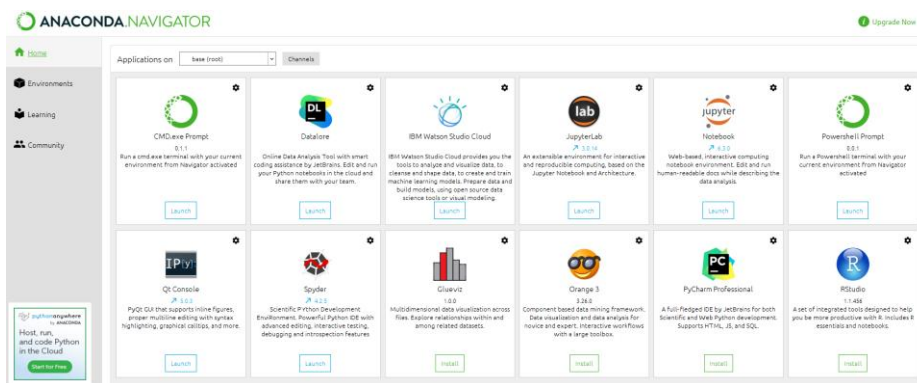


Figure 2: Anaconda Navigator Page

3.2 Python

Python version 3.8 was installed for implementing the script. Python is an open-source language and is used for pre-processing images, modeling algorithms, and visualization. The python version can be downloaded from the website ²by clicking on the download section as shown in Figure 3. After downloading the package, it can be installed by following the instructions given.



Figure 3: Python Download Page

3.3 Data Source

The dataset used in the project can be downloaded from google drive and stored locally on a computer. The dataset is accessible on this website ³by clicking on the LWDCD2020 icon, as

² [Welcome to Python.org](https://www.python.org/)

³ [Wheat Disease Detection | Lakshay goyal | Medium | Analytics Vidhya](#)

depicted in figure 4. The data is then automatically stored on google drive shown in figure 5. The dataset contains four folders containing images of wheat plant diseases.

Dataset

The dataset used is the *Large Wheat Disease Classification Dataset (LWDCD2020)*. It consists of around 4,500 images of three classes of wheat diseases and one normal class. The images have been curated for dimensional uniformity.

The dataset contains a total of 4 classes as listed below:

1. Leaf Rust
2. Crown and Root Rot
3. Healthy Wheat
4. Wheat Loose Smut

Figure 4: Dataset

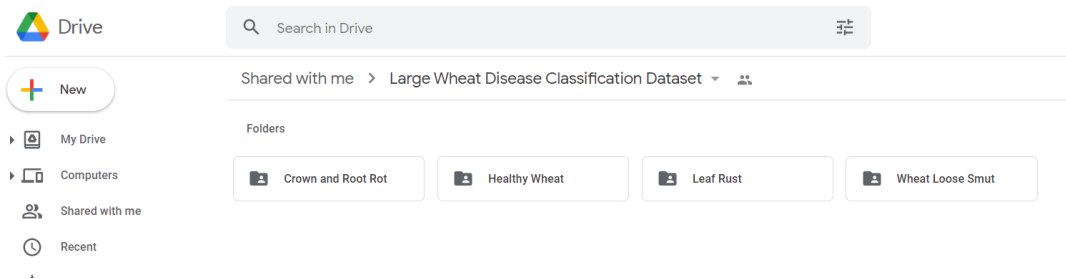


Figure 5: Images Stored on Google Drive

4 Project Implementation

After launching the Jupyter Notebook from Anaconda, a page will be displayed and then click on the “New” icon depicted in figure 6, and select Python 3, a page for the development of the project will load.

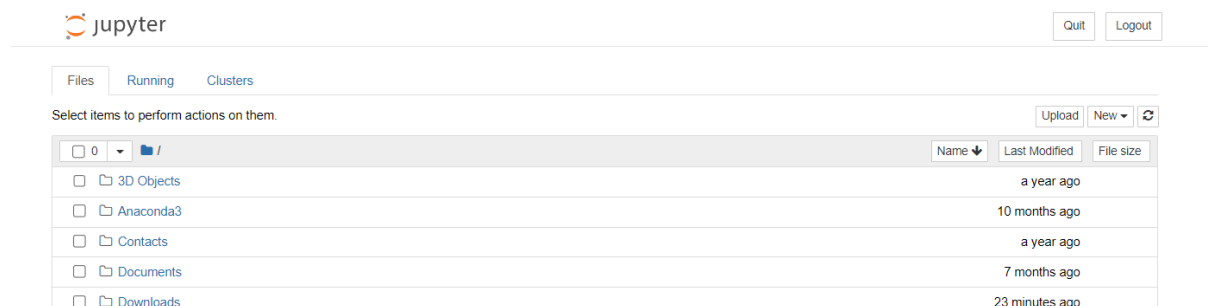


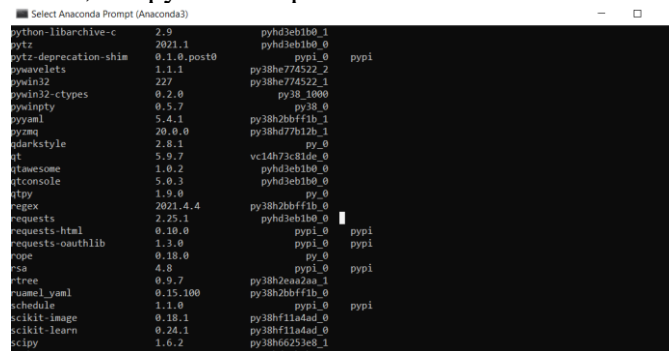
Figure 6: Jupyter Notebook

4.1 Installation of libraries

Before beginning the implementation of the project, the required libraries must be installed. The following libraries will be utilized for data manipulation, modeling, and visualization. The library versions used for this project are listed below.

- Tensorflow – 2.7.0
- Keras - 2.7.0
- scikit-learn - 0.24.1
- scikit-image – 0.18.1
- NumPy – 1.20.1
- matplotlib – 3.3.4
- pandas – 1.2.4
- OpenCV-python – 4.6.0
- imutils – 0.5.4
- seaborn – 0.11.1

These libraries can be installed on the Anaconda Prompt shown in figure 7 “pip” command like “pip install library name” For example: “pip install matplotlib”. After successful installation of the libraries, the python script can be executed.



```

Select Anaconda Prompt (Anaconda3)
python-libarchive-c 2.9 pyhd3eb1b0_1
pytz 2021.1 pyhd3eb1b0_0
pytz-deprecation-shim 0.1.0.post0 pypi
pynvml 1.1.1 py38he774522_2
pywin32 227 py38he774522_1
pywin32-ctypes 0.2.0 py38_1000
pywinpty 0.5.7 py38_0
pyyaml 5.4.1 py38h2bbff1b_1
pyzmq 20.0.0 py38hd77b12b_1
rdarkstyle 2.8.1 py_0
rt 5.9.7 vc14h73c81de_0
rtimezone 1.0.2 pyhd3eb1b0_0
rtconsole 5.0.3 pyhd3eb1b0_0
rtpy 1.9.0 py_0
regex 2021.4.4 py38h2bbff1b_0
requests 2.25.1 pyhd3eb1b0_0
requests-hlml 0.10.0 pypi_0
requests-oauthlib 1.3.0 pypi_0
rope 0.10.0 py_0
rsa 4.8 pypi_0
rtime 0.9.7 py38h2ba2aa1_1
ruamel_yaml 0.15.100 py38h2bbff1b_0
schedule 1.1.0 pypi_0
scikit-image 0.18.1 py38hf11a4ad_0
scikit-learn 0.24.1 py38hf11a4ad_0
scipy 1.6.2 py38h0252ed_1

```

Figure 7: Anaconda Prompt

4.2 Model Development

The data is read from the local machine along with the label and the image data is pre-processed by changing colour, resizing, and converting into an array. The following pre-processing is done using the script shown in figure 8.

```

label = "D:/Research project/Implement/lb.pickle"
LABELS = set(["Crown and Root Rot", "Healthy Wheat", "Leaf Rust", "Wheat Loose Smut"])
imagePaths = list(paths.list_images('D:/Research project/Implement/datasets/train'))
data = []
labels = []
# Loop over the image paths
for imagePath in imagePaths:
    label = imagePath.split(os.path.sep)[-2]
    if label not in LABELS:
        continue

    image = cv2.imread(imagePath)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.resize(image, (224, 224))

    data.append(image)
    labels.append(label)

```

Figure 8: Data Pre-processing

The image data is then augmented with parameters and values depicted in Figure 9. The data augmentation is performed on the training dataset.

```
# initialize the training data augmentation object
trainAug = ImageDataGenerator(
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.20,
    horizontal_flip=True,
    fill_mode="nearest")
```

Figure 9: Data Augmentation

After successfully writing the script, the code can be executed in the jupyter notebook itself by clicking on the “Run” icon. The results or errors that occurred will be displayed at the bottom of each section of code as shown in figure 10

```
plt.figure(figsize=(10,7))
cm = pd.DataFrame(confusionMatrix, index = ['Crown and Root Rot', 'Healthy Wheat', 'Leaf Rust', 'Whe
sns.heatmap(cm, annot=True, fmt = "d", cmap="Blues")

plt.title('Confusion Matrix')
plt.ylabel('Actual Values')
plt.xlabel('Predicted Values')
plt.show()
```

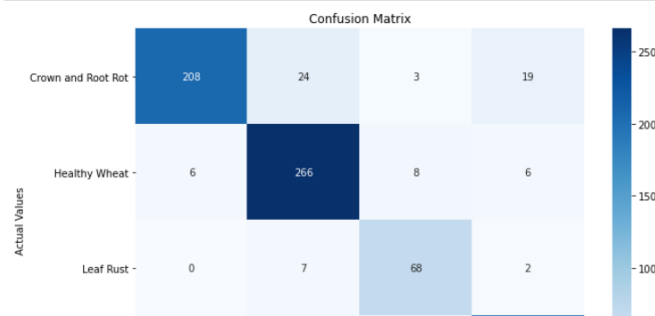


Figure 10: Display of Results

After the successful implementation of the code, the models can be evaluated based on the achieved results and used to identify wheat plant disease. The results of the model will be displayed in the format shown in figure 11.

	precision	recall	f1-score	support
Crown and Root Rot	0.95	0.82	0.88	254
Healthy Wheat	0.87	0.93	0.90	286
Leaf Rust	0.84	0.88	0.86	77
Wheat Loose Smut	0.89	0.93	0.91	231
accuracy			0.89	848
macro avg	0.89	0.89	0.89	848
weighted avg	0.90	0.89	0.89	848

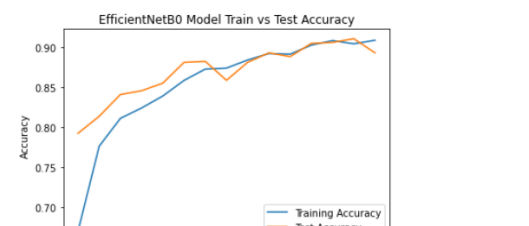


Figure 11: Results of the model

The final results of the identified disease for a particular image are displayed as shown in figure 12. The entire script along with the dataset is submitted. The script can be downloaded as ipynb file and can be directly used for the execution.



Figure 12: Disease Identification