

# Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data

MSc Research Project  
Data Analytics

Saurabh Sharma  
Student ID: x19239301

School of Computing  
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	Saurabh Sharma
<b>Student ID:</b>	x19239301
<b>Programme:</b>	Data Analytics
<b>Year:</b>	2021
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Dr. Bharathi Chakravarthi
<b>Submission Due Date:</b>	31/01/2022
<b>Project Title:</b>	Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data
<b>Word Count:</b>	1017
<b>Page Count:</b>	8

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.

<b>Signature:</b>	Saurabh Sharma
<b>Date:</b>	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"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , 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.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data

Saurabh Sharma  
x19239301

## 1 Introduction

This document intends to provide all the hardware and software required to carry on this research successfully. All the steps involved in accomplishing this research are mentioned logically to be repeated easily by anyone. This research uses three models, namely, Res-Net50, a pre-trained MobileNetV2 model and a hybrid CNN-RF model, to classify plant disease into three classes: Healthy, Powdery and Rust. All three models are compared to see which model performs well with a limited plant disease dataset.

## 2 System Configuration

This section focuses on the software and hardware required to execute this research successfully.

### 2.1 Hardware Configuration

The research can be conducted on a personal laptop with configuration as mentioned in the Table 1:

Table 1: Hardware Configuration

Hardware	Configuration
System	Lenovo Yoga 710-14IKB
System Type	64 bit
RAM	8 GB
Graphics	None
SSD Memory	256 GB
Processor	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz

### 2.2 Software Configuration

All the requirements related to the software required for this research are mentioned in this section.

The details of the operating system required for the successful implementation of this research is mentioned in the Table 2:

Table 2: Operating System

Specification	Value
Edition	Windows 10 Home
Version	21H2
OS Build	19044.1348na

The dataset for this research contains high-quality images, and processing that data requires GPU; therefore, google colab becomes the ideal choice to execute python code. For this, you need to have a Gmail account and can be opened on any browser, and Google chrome explorer is used to access google colab in this research. Google colab makes it easy to write python code as it contains all the required libraries and provides suggestions while writing python code. The Figure 1 shows the version of chrome used in this research.

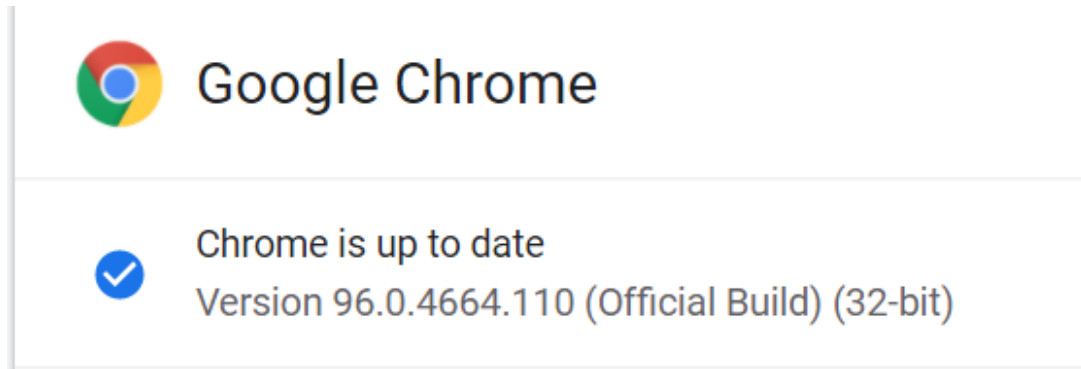


Figure 1: Browser Specification

### 2.3 Google Colab Configuration

The steps required to run a python code in Google Colab (Nelson and Hoover; 2020) is mentioned below:

- An active Gmail account is needed to access Google Colab.
- Visit the website [https://colab.research.google.com/?utm\\_source=scs-index](https://colab.research.google.com/?utm_source=scs-index).
- Login to Google Colab using your Gmail account as shown in Figure 2:
- Now open new notebook or upload an existing notebook to execute your python code.

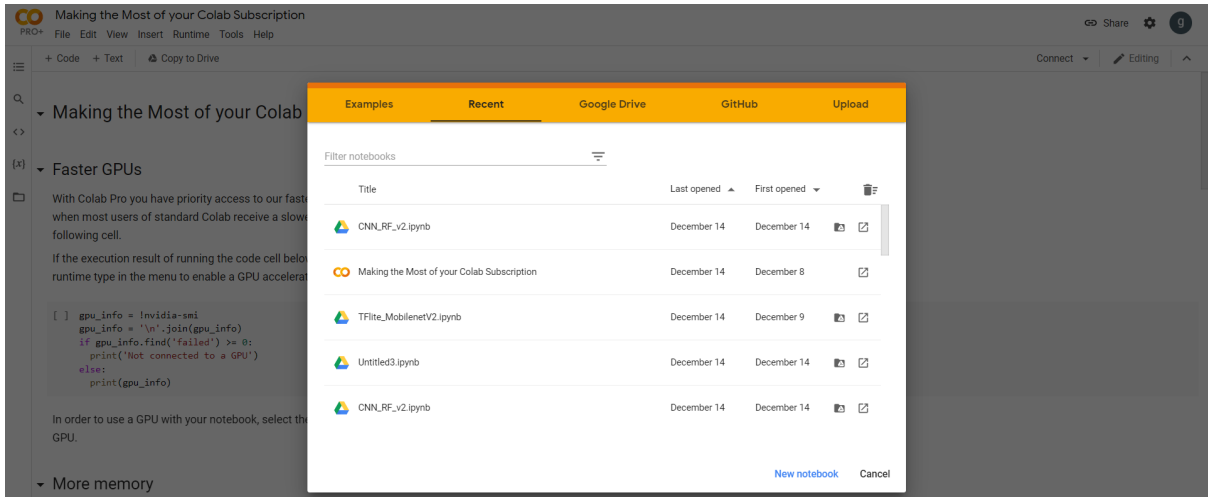


Figure 2: Google Colab

## 2.4 Google Colab Cloud Configuration

The limitation of Google Colab is that it gives GPU only for a small amount of time. Hence, code might stop by providing a pop-up that GPU allowed time is finished. To overcome this issue, a Google Cloud Platform is used along with Google Colab. The steps involved in doing this is mentioned below:

- First you would need a active google account.
- Visit the link <https://cloud.google.com/> and sign in with an Gmail accout as shown in Figure3.
- Click on console.
- Look for "Colab" in search box.
- As shown in Figure 4 click on Colab and then click on launch.
- Set up the configuration needed for the project and then open Colab. The design specification required for this project is shown in the Table 3:

Table 3: Instance Configuration

Instance Setting	Configuration
Machine Type	n1-highmem-2
GPU	Nvidia Tesla T4
SSD Memory	200 GB
CPU	Intel Haswell

- 
- Once the instance is created. Click on the arrow as shown in the Figure 5 and then click on the link to open the Google Colab instance as shown in Figure 6

- Final step is to connect to GCE VM machine and provide the required details as shown in Figure 7.

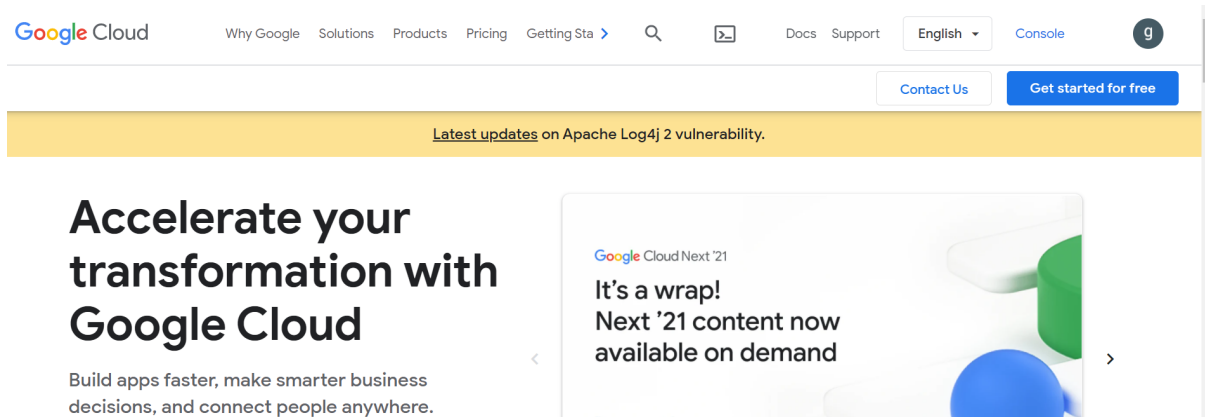


Figure 3: Google Cloud Platform

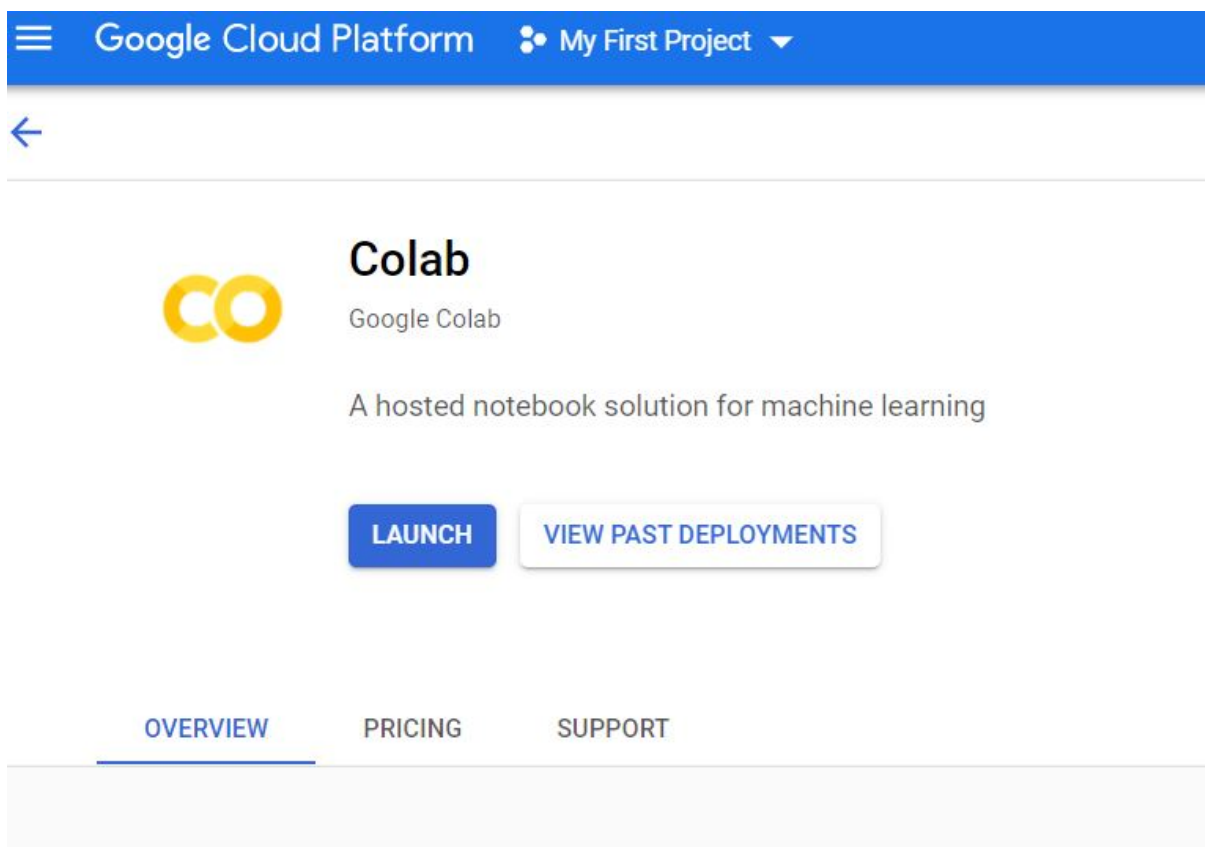


Figure 4: Google Cloud-Colab Launch

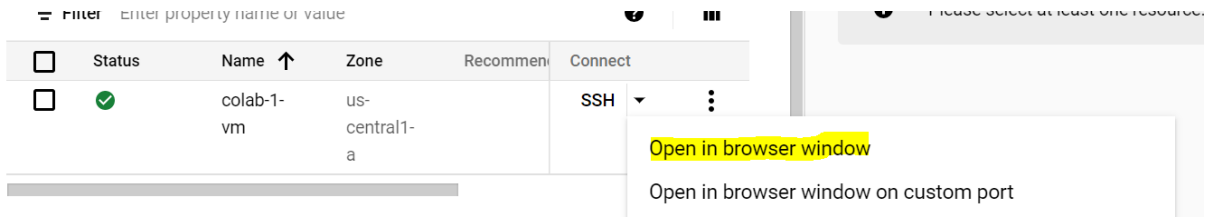


Figure 5: Google Cloud Machine

```

Connected, host fingerprint: ssh-rsa 0 24:20:A2:B5:E2:FC:AE:76:2C:A2:A1:BA:F7:1A
:BE:BD:17:C7:E2:8B:81:C9:60:67:B1:7A:8B:17:96:40:A0:5C

To use this VM in the Colab UI, navigate to the URL:

https://colab.research.google.com/#gceVm=basic-buttruss-334422/us-central1-a/c
olab-1-vm
saurabhire@colab-1-vm ~ $

```

Figure 6: Google Colab Link

## Connect to a custom GCE VM

Learn more about how to start a GCE VM for Colab via GCP Marketplace by checking out [these instructions](#).

Project\*  
basic-buttruss-334422

---

Zone\*  
us-central1-a

---

Instance\*  
colab-1-vm

---

[Copy auto-connect link](#)

**Cancel**   **Connect**

Figure 7: Google Colab VM

## 3 Project Implementation

### 3.1 Data Collection

The plant disease dataset is publicly available and can be downloaded from kaggle<sup>1</sup>. The dataset is recently uploaded in the Kaggle and is a novel dataset for the research. Datasd contains high quality images, and the size of the dataset is around 1.1 GB as shown in Figure 8 below.

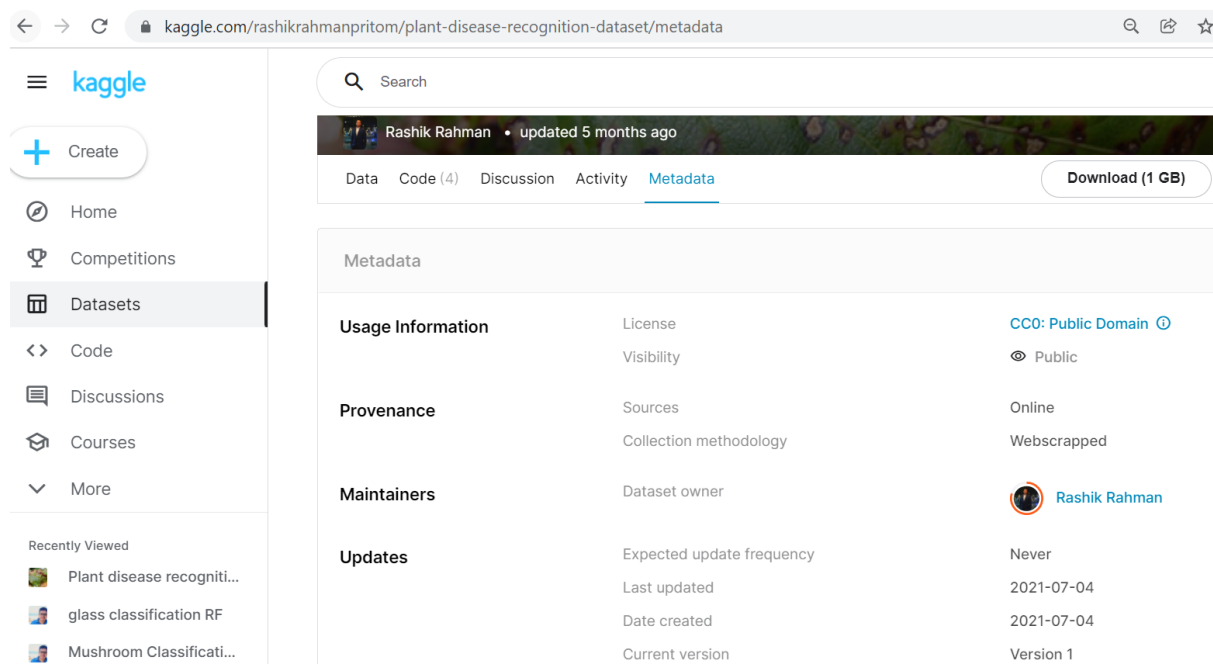


Figure 8: Kaggle Data Source

### 3.2 Data Preparation

The dataset contains three folders train, test and validation. Each folder contains three subfolders; namely, Healthy, Powdery and Rust. Below Table 4 shows count of each class in train, test and validation set:

Table 4: Data Description

<b>Dataset</b>	<b>Healthy</b>	<b>Powdery</b>	<b>Rust</b>
Train	458	430	434
Test	50	50	50
Validation	20	20	20

### 3.3 Data Preprocessing and Data Augmentation

Data preprocessing and data argumentation is the basis for creating any deep learning model. The focus of this research is to apply both this methodology to build an effective

<sup>1</sup><https://www.kaggle.com/rashikrahmanpritom/plant-disease-recognition-dataset>



model as the dataset for this research is small. Using the above method helps minimise over-fitting and increase the amount of data fed to the model for training. The below Figure 9 shows the preprocessing and augmentation techniques applied in this research. It can be seen that daugmentation techniques like height shift, flipping, and rotation have been applied in the training dataset.

```
3 # val_samples = 60
4 epochs = 100
5 batch_size = 128
6 class_idx = 3
7
8 #Data Augmentation # Research on other parameters
9 train_datagen = ImageDataGenerator(rescale = 1./255.,
10                                   rotation_range = 90,
11                                   width_shift_range = 0.2,
12                                   height_shift_range = 0.2,
13                                   shear_range = 0.1,
14                                   zoom_range = 0.1,
15                                   brightness_range=(0.4, 1.0),
16                                   horizontal_flip = True,
17                                   channel_shift_range=80.0,
18                                   vertical_flip = False)
19
20 test_datagen = ImageDataGenerator(rescale = 1. / 255)
21
22
```

Figure 9: Data Augmentation and Preprocessing

### 3.4 Model Building

This research uses python as the scripting language to build deep learning models. There are specific essential libraries or packages which needs to present for the successful execution of all the models. The critical packages required for this research are:

- NumPy: It is required to convert data into arrays and assign labels to each class of disease present in the dataset.
- Keras: Keras is used to import packages to build deep learning models like MobileNetV2, ResNet50, and the CNN-RF model.
- TensorFlow: It is used to build large-scale neural networks and can be used in areas such as classification, prediction etc.
- TensorFlow-hub: It contains a set of pre-trained machine learning models ready for fine-tuning and training new data. MobileNetV2 mode is creatrd using this library.

### 3.5 Model Evaluation and Visualizations

Once all the models are trained. The next step is to evaluate each model using a specific evaluation metric. For comparing each model, it is essential to have some plot, confusion matrix, and a heat map. Hence, to do this, certain packages need to be installed in python as mentioned below:

- sklearn: To plot confusion\_matrix, or to know the value of metrics like recall, precision classifier\_report is required. All this can be done using this library.

- matplotlib: To better understand the model, it is required to plot a graph comparing the values of loss accuracy, recall with train and test data. This is done using this library.
- seaborn: This package is needed to see how accurately the model predicts based on a heatmap.

## References

Nelson, M. J. and Hoover, A. K. (2020). Notes on using google colaboratory in ai education, *Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education*, pp. 533–534.