

**Plant Health Monitoring System**

**INTERNSHIP REPORT**

**Quarter IV (Year 1)**

***Submitted by***

**V GAJA PRASANTH E0121049**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

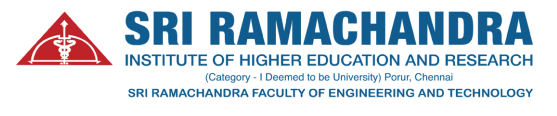
**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Machine Learning)**

**Sri Ramachandra Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**JULY, 2020**



**BONAFIDE CERTIFICATE**

Certified that this project report “**Plant Health Monitoring System**” is the bonafide work of **V Gaja Prasanth** Reg No. **E0121049** who carried out the internship work under my supervision

**Signature of Faculty Mentors Signature of Vice-Principal**

**Prof. Balaji Prasath Prof. M. Prema**

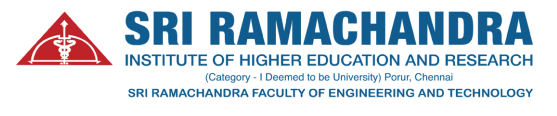
Professor, Vice-Principal,

Sri Ramachandra Engineering and Technology, Sri Ramachandra Engineering

Porur, Chennai-600116 and Technology

Porur, Chennai-600116

**Evaluation Date:**



**ACKNOWLEDGEMENT**

I express my sincere gratitude to our Chancellor, Vice-Chancellor and our sincere gratitude to our Provost **Dr. V. Raju** and our Vice-Principal **Prof. M. Prema** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor(s), **Prof.Balaji Prasath** Department of Computer Science and Engineering, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

**TABLE OF CONTENTS**

**Title Page**

1.Abstract

2.Introduction

3.Review of literature and products

4.Problem Statement

5.Workflow and architecture

5.1 Pipeline(flowchart)

5.2 Pre Processing

5.3 Faster response time to output

5.4 About dataset

5.5 Naming the output

6.Technologies used

6.1 Python

6.2 Matplotlib

6.3 Keras

6.4 VGG19 CNN

7.Implementation

8.Sample output

9.Future enhancements

10.Conclusion

11.Timeline and Worklog

12.References

**1.Abstract:**

India is a large country whose population is in the range of 1.35 to 1.38 billion people, to satisfy the needs of the fast growing country one of the major needs is food from agriculture but due to certain leaf/plant diseases crop production is being let down,

To tackle this problem farmers and agricultural scientists are being faced with the problem of detecting a disease with their naked eyes and intuitons, basic generational knowledge about how a disease symptoms look in a plant and then use the appropriate measures which in all in time consuming and not always 100% accurate.

To help with this we can use a plant health monitoring system which uses automated learning to scan the uploaded pictures of diseased plant and help detect the name of the disease with high accuracy,with the help of deep-learning.

This proposed model helps farmers to have an easier job and increase their yield with higher chances of saving crops from death.

**2.Introduction:**

* **Motivation**- To help increase yield of crops and increase detection of diseases which affect the yield of crops.
* **Approach to Problem**- Find easier and more accurate disease detection using VGG19,Deep-Learning.
* .**Application and Technology**- We can integrate this code into an app or website with user-friendly languages which will help farmers access and use it without much help from others.If we use a website we can host the webpage on a cloud server which can help others access it.

**3.Review of Literature and Products**

| **Year** | **Author** | **Product** | **Review** |
| --- | --- | --- | --- |
| March 2021 | Prabavathi S, Kanmani P | Plant Leaf Disease Detection and Classification using Optimized CNN Model | **Benefits-**  The accuracy of the model is at 93.18%  **Application-**  Leaf disease detection  **Limitation-**  No frontend/user interface UI  **Challenges-**  Using enhanced CNN |
| May 2021 | Mrs. M. Geetha Yadav ,  Rajasekhar Nennuri , Devarakonda Rajeshwari , Voggu Rishitha , Tandur Puneeth | Identification of Plant Leaf Diseases using Machine Learning Algorithms | **Benefits-**  Feature extraction using VGG19,Inception V3 model  **Application-**  Leaf disease detection  **Limitation-**  Frontend is not published in internet to be utilised by common people  **Challenges-**  DL is comparatively better than ML in this case |

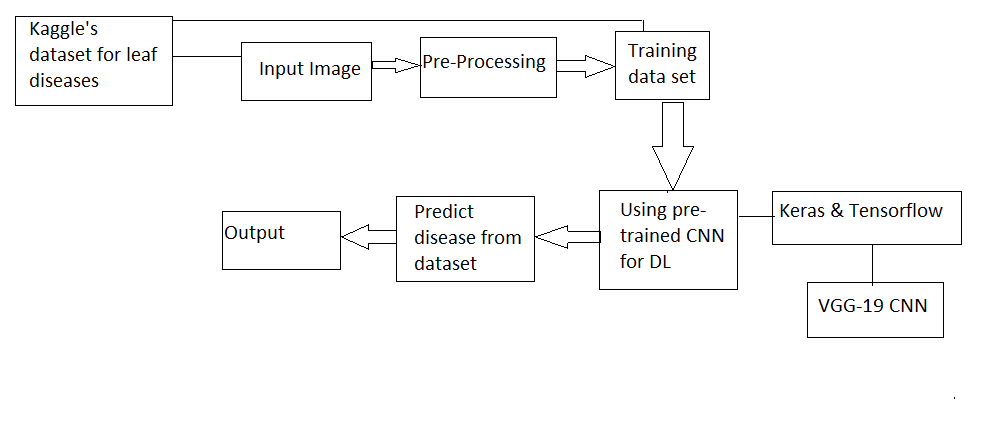
**4.Problem Statement**

Description: To analyse leaf disease picture dataset using Python and its modules with deep-learning algorithms to find the most accurate algorithm to help with disease detection.

| **Objective** | **Methodology** |
| --- | --- |
| 1. To collect the data samples | 1. Dataset - Kaggle |
| 2. To pre-process the dataset and prepare for deep-learning task | 2. Python |
| 3. To visualise the data | 3.Python |
| 4. To create DL model | 4. Python |

**5.Workflow and Architecture**

**5.1 Pipeline(Flowchart)**



**5.2 Pre-Processing**

****

We will use the keras module from python which will help in pre-processing the images we have in the dataset obtained from kaggle.

Pre-Processing helps in making it easier for the CNN to distinguish between healthy and diseased leaf by highlighting or making a difference in the diseased or symptom showing part.

For example,

The spots in below image in not easily recognized and maybe understood to be a healthy tomato leaf but infact the spots which are not prominent are visible just not to the naked eye under normal circumstances ,post pre-processing the spots are very prominent and easily visible which helps the DL program to easily find the difference.

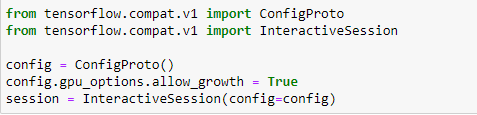


Fig. Leaf of Tomato with Late blight disease



Fig. Image of leaf after Pre Processing to show the prominent spots in different shades of green to differentiate the diseased part.

**5.3 Faster response time to output**



Trying to load the CNN in TensorFlow and fully utilise the power of GPUs. However, my GPUs only have 4GBs memory, which is quite small. So I need to use GPUs and CPUs at the same time.

Config to configure the session.

**5.4 About Dataset**

Plant village is the dataset I have chosen to use in my project as it has 20,638 images of healthy and diseased plants of 15 different classes namely:

'Pepper\_\_bell\_\_\_Bacterial\_spot', 'Pepper\_\_bell\_\_\_healthy', 'Potato\_\_\_Early\_blight', 'Potato\_\_\_Late\_blight', 'Potato\_\_\_healthy', 'Tomato\_Bacterial\_spot', 'Tomato\_Early\_blight', 'Tomato\_Late\_blight', 'Tomato\_Leaf\_Mold', 'Tomato\_Septoria\_leaf\_spot', 'Tomato\_Spider\_mites\_Two\_spotted\_spider\_mite', 'Tomato\_\_Target\_Spot', 'Tomato\_\_Tomato\_YellowLeaf\_\_Curl\_Virus', 'Tomato\_\_Tomato\_mosaic\_virus', 'Tomato\_healthy

This dataset takes into account leaves of 3 plants -

* Pepperbell(1+1)
* Potato(1+2)
* Tomato(1+9)

Given below is the link to the dataset:

<https://www.kaggle.com/datasets/emmarex/plantdisease>

**5.5 Naming the output**

* **Issue:**

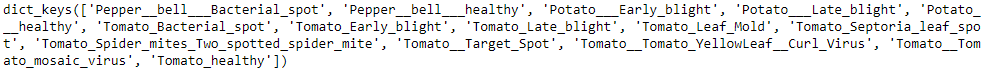
End output is displaced in terms of class indices value which is not easy for user to read unless he knows which indices key is for which indices value

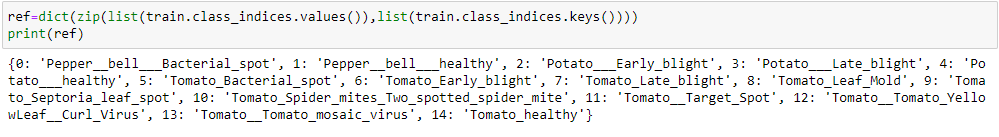


* **Fix to solve the issue:**

Taking both lists of either keys and values and ziping it to a dictionary with corresponding value,key.







* **Output Post-fix:**



**6.Technologies used**

**6.1 Python:**

Python is a high-level and one of the most popular programming languages, first designed by Guido van Rossum which was first released in 1991. It has a wide scale application and can be used for various tasks like data analysis, machine learning, building websites, automating tasks, etc.

Advantages-

* Simple and flexible to code with and easy to import necessary modules for implementation.

**6.2 Matplotlib:**

Matplotlib is an amazing visualisation library in Python for 2D plots of arrays.

Matplotlib is a multi-platform data visualisation library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in 2002.

Advantages-

* Visual access to huge amounts of data in easily digestible visuals like line, bar, scatter, histogram plots.

**6.3 Keras:**

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

Advantages-

* Keras helps reduce cognitive load and offers simple api.

**6.4 VGG19 CNN:**

VGG-19 is a convolutional neural network that is 19 layers deep. You can load a pre trained version of the network trained on more than a million images from the ImageNet database.

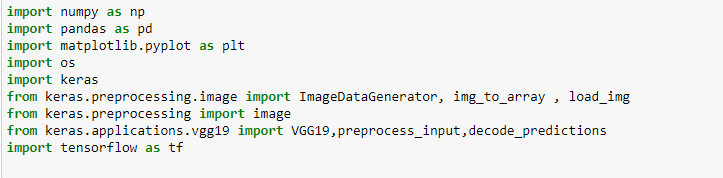
As a result, the network has learned rich feature representations for a wide range of images.

Advantages-

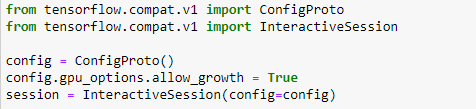
* faster training speed, fewer training samples per time, and higher accuracy.

**7.Implementation:**

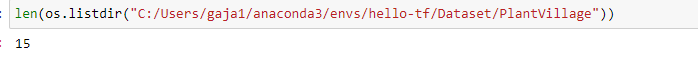
**7.1 Importing modules:**

****

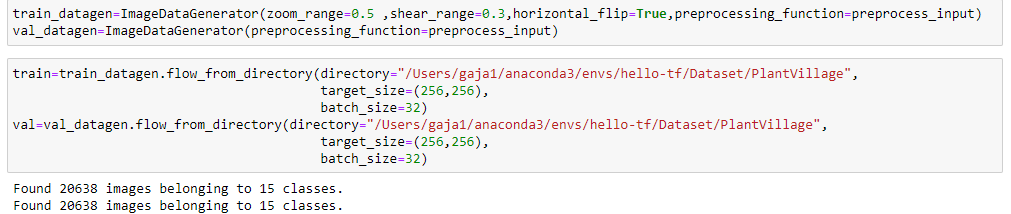
**7.2 Including cpu and gpu for runtime efficiency:**

****

**7.3Reading number of classes in dataset folder:**

****

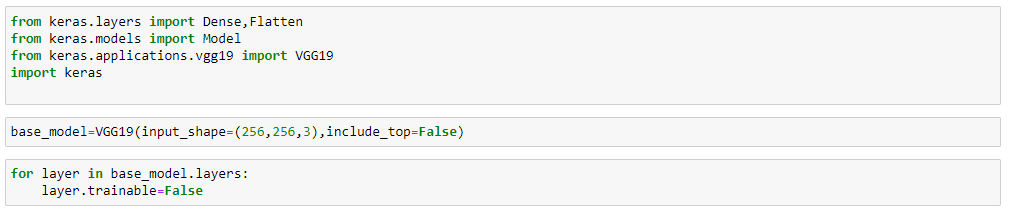
**7.4 Splitting Data:**

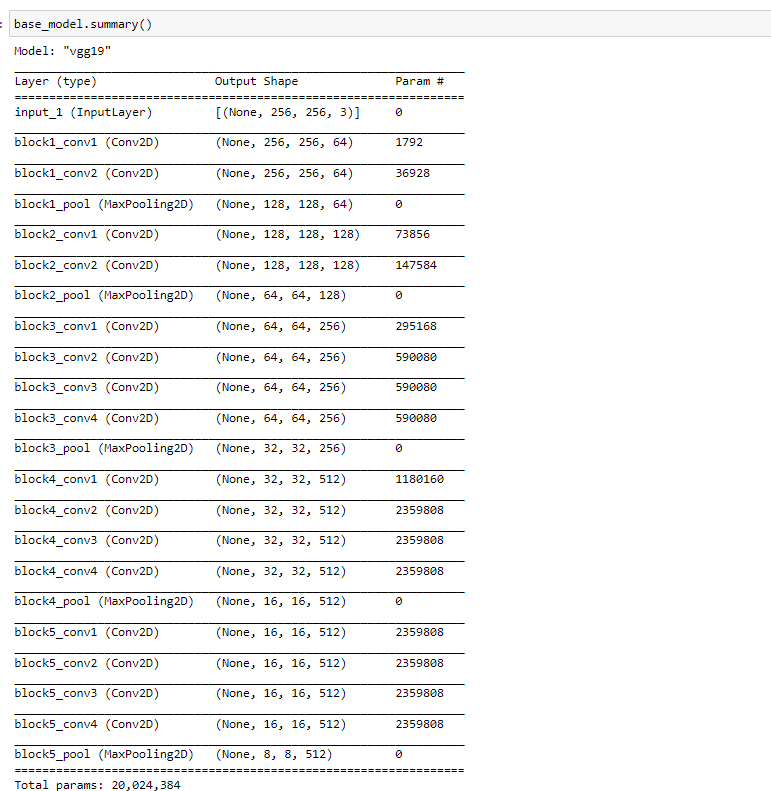
****

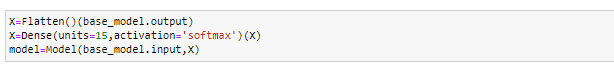
**7.5 Pre Processing:**

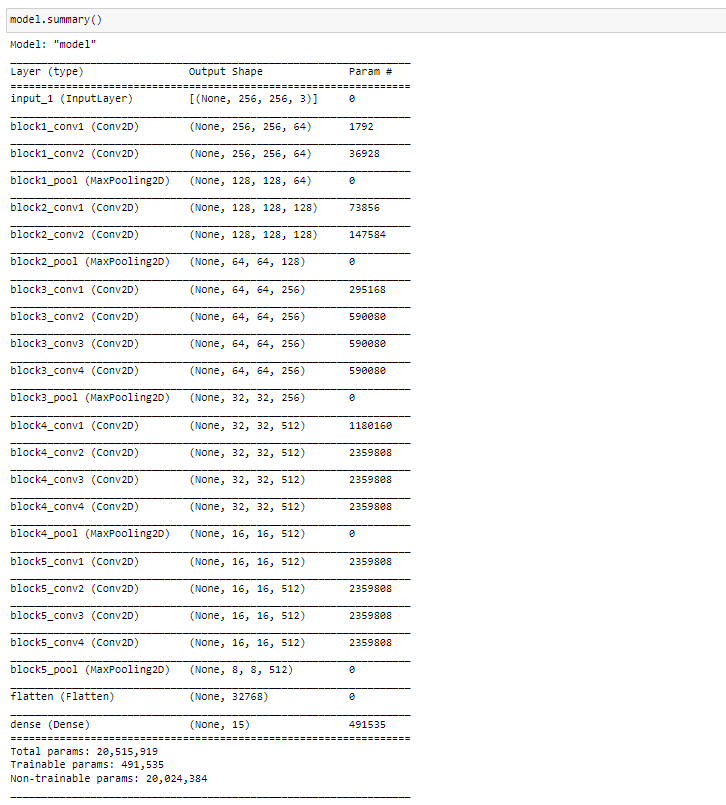
****

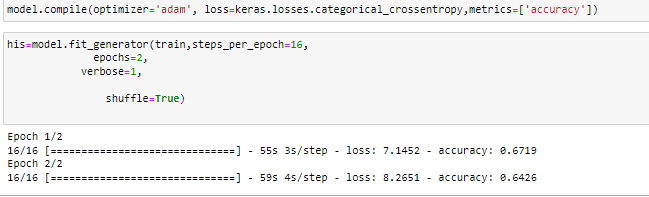
**7.6Training the data in the CNN**

****

****

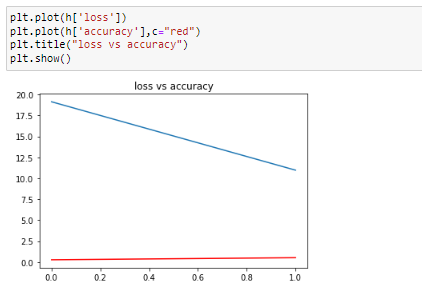






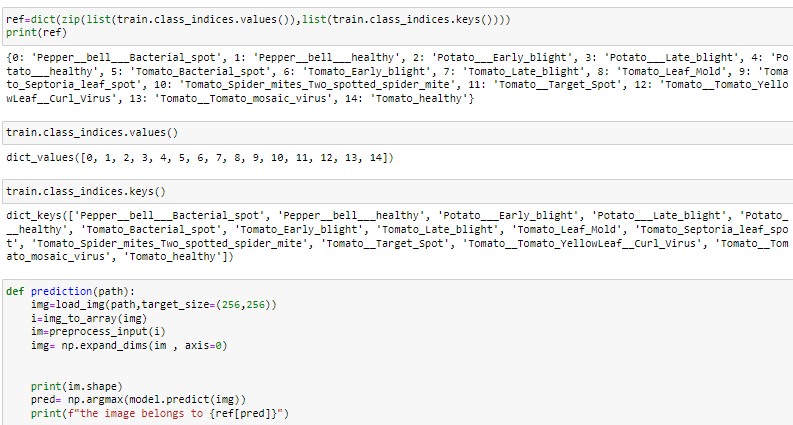
* **IMPORTANT NOTE** :epochs taken in this picture is 2 so accuracy is 67% but when epochs are taken at 25 or more, accuracy is close to around 80 to 90% or higher

**7.7 Information about the accuracy of model:**

****

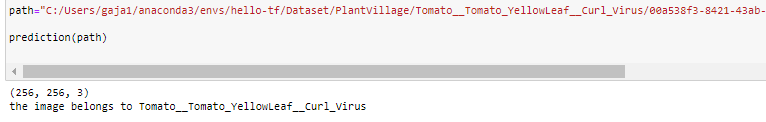
**Note:** Here the accuracy increase from 0.2 to 0.8 as epochs keep running at the same time the loss decrease from 20 to 11

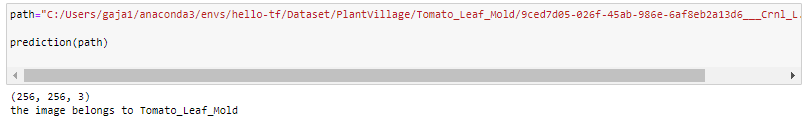
**7.7 Getting output for the input image with proper name:**

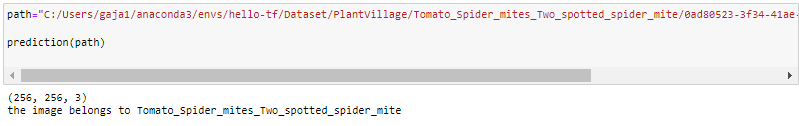
****

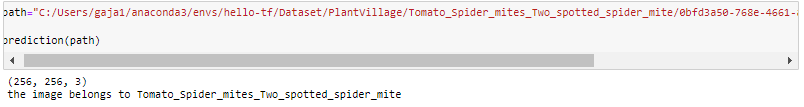
**8. Sample Output:**

* The path where the image is stored is the correct answer to what disease the plant leaf’s image has
* When the code gives output as curl\_virus we can check the credentiality of the output by seeing the path of image from above the prediction/output

****

****

****

****

**9.Future enhancements :**

This project would be better if there was an api integrating the code to webpage which is hosted in a cloud server to easily allow the functioning and access to almost everyone with an internet.

It would have also been better to see the plots between VGG16 and VGG19 running the same code to show the clear difference between the accuracy of either CNN’s ,As i have researched thus far i have found VGG19 to be having 0.9% higher accuracy.

Including the disease datasets of different plants is obviously one of my biggest concerns as i was limited by the kaggle dataset which had only 3 plants in its dataset, In the future i might do data augmentation between different datasets with different plants to help the program/model easily recognise the differences between different plant diseases.

Along with the webpage hosted in a cloud server, an application would also prove to be a portable and helpful alternative which should be looked into.

**10.Conclusion:**

This Model helps easily detect diseases in plant leaves without much loss and higher accuracy than other ML & DL programs thanks to the implementation of VGG19 which has increased the accuracy by 2 to 8% compared to traditional ML disease detection program

The utilisation of both gpu and cpu helps easily improve the response time for the code to run faster.

**11.Timeline and Worklog**

| **DAY** | **DATE** | **WORK** |
| --- | --- | --- |
| Day1 | 21-06-2022 | Researching about phms |
| Day2 | 22-06-2022 | Reading papers on leaf disease detection |
| Day3 | 23-06-2022 | Obtaining Datasets which are optimum for my model |
| Day4 | 24-06-2022 | Implementing different Modules in python |
| Day5 | 24-06-2022 | Using VGG19 |
| Day6 | 25-06-2022 | Preparing workflow |
| Day7 | 26-06-2022 | Preparing for 1st Review |
| Day8 | 27-06-2022 | Preparing ppt for 1st Review |
| Day9 | 28-06-2022 | Preparing ppt for 1st Review |
| Day10 | 29-06-2022 | 1st Review |
| Day11 | 30-06-2022 | Researching different preprocessing methods |
| Day12 | 1-07-2022 | Test Train Data Split |
| Day13 | 2-07-2022 | Using different preprocessing methods |
| Day14 | 3-07-2022 | Using different preprocessing methods |
| Day15 | 4-07-2022 | Preparing for 2nd review |
| Day16 | 5-07-2022 | Preparing ppt for 2nd Review |
| Day18 | 7-07-2022 | Preparing ppt for 2nd Review |
| Day20 | 9-07-2022 | Reviewing objectives for 2nd Review |
| Day22 | 11-07-2022 | 2nd Review |
| Day24 | 13-07-2022 | Final touches to code for naming output |
| Day26 | 15-07-2022 | Final touches to code |
| Day28 | 17-07-2022 | Preparing ppt for 3rd Review |
| Day30 | 22-06-2022 | 3rd Review |

**12.References:**

* ***PLANT DISEASE DETECTION USING VGG AND DJANGO*** *by* Jyothirmai Sai Sri Gelli, Lakshmi Akhila Madduri, Roshan Tanveer, Udaya Bhanu, G Krishna Kishore.
* **Dataset**- Plant Village Dataset from kaggle
* **Leaf Disease Identification Using Model Hybrid Based on Convolutional Neural Networks and K-Means Algorithms** by Joel Bejar Mallma; Ciro Rodriguez; Yuri Pomachagua; Carlos Navarro
* **Identification of Plant Leaf Diseases using Machine Learning Algorithms** by Mrs. M. Geetha Yadav , Rajasekhar Nennuri , Devarakonda Rajeshwari , Voggu Rishitha , Tandur Puneeth