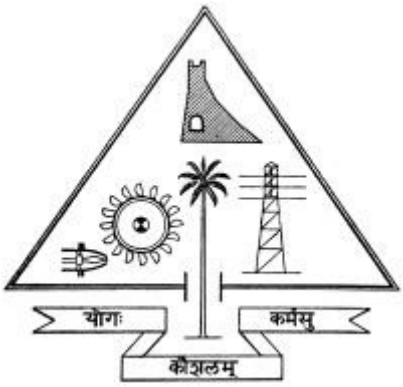
# PLANT LEAF DISEASE DETECTION USING RESNET-50

*Thesis submitted in partial fulfillment of the requirements for the award of the degree of* ***Master of Computer Applications*** *of the* ***APJ Abdul Kalam Technological University***

*submitted by*

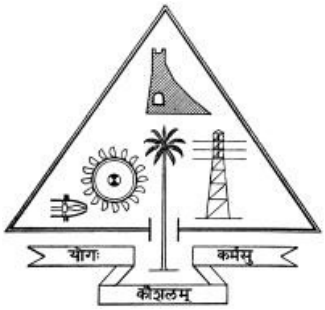
**AJAY JEEVAN JOSE**
**(TCR21MCA-2004)**




**DEPARTMENT OF COMPUTER APPLICATIONS**
**GOVERNMENT ENGINEERING COLLEGE**
**THRISSUR - 680009**

NOVEMBER 2022

**DEPARTMENT OF COMPUTER APPLICATIONS**
**GOVERNMENT ENGINEERING COLLEGE, THRISSUR**
THRISSUR, KERALA STATE, PIN 680009




# CERTIFICATE

*This is to certify that the main project titled* **"PLANT LEAF DISEASE DETECTION USING RESNET-50"** *is a bonafide work done by* **AJAY JEEVAN JOSE (TCR21MCA-)** *under my supervision and guidance, and is submitted in July 2022 in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications from APJ Abdul Kalam Technological University(KTU).*

Dr.Sminesh C N
**Project Coordinator and HOD**

Place : THRISSUR
Date : 21-10-2022

# DECLARATION

I hereby declare that the main project named, **PLANT LEAF DISEASE DETECTION USING RESNET-50**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published by another person nor material which has been accepted for the award of any other degree or course of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

**P**lace : THRISSUR

Date : 21-10-2022

Signature

**AJAY JEEVAN JOSE(TCR21MCA-2004)**

# ACKNOWLEDGEMENT

I would like to thank Department of Computer Application, GEC Thrissur, for giving me this opportunity to pursue this project and successfully complete it.

I am highly indebted to our project guide and head of the department **Dr. Sminesh C N** for his committed guidance, valuable suggestions, constructive criticisms and precious time that he invested throughout the work.

I express my heart-felt gratitude to **Prof. Soumia Chandran**, for her guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

I sincerely thank all other faculties of MCA department for guiding through the processes involved in the project.

# ABSTRACT

Early disease detection is crucial for improved crop yield and quality. Due to a decline in the quality of the agricultural produce, diseased plants can cause large financial losses for individual farmers.In a nation like India, where a substantial section of the population relies on agriculture for a living, it is essential to spot the disease at its earliest stages.A precise diagnosis of the plant disease might reduce losses.The objective of this research is to develop a model that can correctly forecast whether a leaf is disease-infected or not.The main objectives of this study include identifying plant disease and suggesting pesticides that can help to reduce the financial loss.

# CONTENTS

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

The productivity of the agriculture sector drives the Indian economy. Over 70% of rural homes depend on agriculture. Agriculture pays about 17% of the total GDP and employs over 60% of the population. Therefore the ability to identify plant diseases is crucial in the field of agriculture. In Indian agriculture, a variety of crops, including rice and wheat. Additionally, Indian farmers raise sugarcane, oil seeds, maize, and potatoes as well as non-food goods like cotton, rubber, coffee, and tea. These grains and vegetables come in contact with different diseases due to different climates and conditions in different places. As a result, cultivators in any country face severe losses because of these diseases. According to past studies, 42% of agricultural production is in loss, and that too only because of the increasing rate of loss due to plant leaf diseases. Hence, it is most important to identify plant disease in an accurate and timely way.

In the modern century, the introduction of computer and enhancement in fields like machine learning is making difference in these problems. Digital image processing tools are employed by the used method to obtain the desired output. The subjective nature of the results makes it impossible for the human eye to precisely determine the extent of the disease. Usually, the extent of diseases in a production area is determined by observations made with the unaided eye. Image processing has significantly advanced the agricultural industry. Image processing techniques namely Image preprocessing, image augmentation, feature extraction, feature selection and classification are applied on leaf image dataset. Then it is designed a supervised machine learning which trains the dataset image and extracts the data

from it. This paper introduces a leaf diseases detection system using CNN architecture known as Residual Neural Network-50 (ResNet-50) ResNet-50 is a convolutional neural network that is 50 layers deep. It is a variant of the ResNet model which has 48 Convolution layers along with 1 Max Pool and 1 Average Pool layer. It has $3.8 \times 10^9$ Floating points operations.

Following the identification of the disease the system will run a query which turns out the desirable treatment for the disease and pesticides that can be used as remedy. Also, the system will provide the direction to the direction of experts.

# CHAPTER 2
# LITERATURE REVIEW

For the project study I referred to some of the latest conference papers which are related to our project on change detection and also observed some of the methods as well as procedures which are implemented and written those observations of each paper that will be useful in development of the project.

This section denominates the different approaches of previous studies which were used to identify and classify the various leaf diseases of plants. Zhou et al [1] proposed K-Means clustering algorithm and faster R-CNN Fusion algorithm to detect rice diseases as well as to address several complications such as blurred image edge, noise, large background interference and low detection accuracy using 3010 images which captured by camera. And also faster 2D-Otsu algorithm was used to classify the rice disease images for getting output. Sharma et al [2] introduced the image pre-processing along with k-means clustering, segmentation and four classifiers such as logistic regression, SVM, KNN, and CNN to detect and classify leaf diseases automatically wherein logistic regression performs quite well due to classes but highest accuracy was provided by CNN due to classification and detection of diseases using 20,000 images from GitHub and Kaggle. In [3], it was suggested an incorporated method to create heterogeneous data from Normalized Difference Vegetation Index for achieving a rich and comprehensive knowledge of wheat's health using IoT sensors, machine learning such as SVM and NB and drone technology.

Leaf disease detection is used for detection of disease in the leaves by

using ResNet-50 pre-trained models and also improves the accuracy of the model. Here the project use Paddya and Maize leaf for disease detection.

# CHAPTER 3
# ENVIRONMENTAL STUDY

## *3.1 System Configuration*

System configuration describe the hardware and software requirement of the system for development

### *3.1.1 Hardware Requirements*

- Memory : 4 GB RAM or above
- Processor : Intel Core i5 or above
- Hard Disk Space : 100 GB

### *3.1.2 Software Requirements*

- Operating system : Ubuntu 20.04 or above
- Front End : HTML
- Back End : Python, JavaScript
- IDE Used : VS Code, Jupyter Notebook, Collab

## *3.2 Software Specification*

### *3.2.1 Python*

Python is an interpreted, excessive-stage, widespread-purpose programming language. Created by means of guido van rossum and first launched in 1991, python's design philosophy emphasizes code readability with its

extraordinary use of significant white space. Its language constructs and item-oriented approach aims to help programmers write clean, logical code for small and huge-scale projects. Python is dynamically typed and rubbish-accrued. It supports multiple programming paradigms, together with procedural, item-oriented,and purposeful programming. Python is regularly defined as a "batteries included" language due to its comprehensive preferred library. Python is meant to be an effortlessly readable language. Its formatting is visually uncluttered, and it frequently uses English key phrases in which different languages use punctuation. In contrast to many other languages, it does now not use curly brackets limit blocks, and semicolons after statements are non-compulsory. It has fewer syntactic exceptions and unique instances than c or pascal. Python uses white space indentation, as opposed to curly brackets or keywords, to delimit blocks. An growth in indentation comes after positive statements; a decrease in indentation indicates the end of the present day block. Thus, the program's visual shape appropriately represents this system's semantic shape. This feature is likewise every now and then termed the off-side rule

### *3.2.2 VS Code*

Microsoft created the free open source text editor known as Visual Studio Code (often referred to as VS Code). Linux, macOS, and Windows all support VS Code. VS Code has recently become one of the most widely used development environment tools, despite the editor's relatively modest weight and robust capabilities. Numerous programming languages are supported by VS Code, including CSS, Go, and Dockerfile in addition to Java, C++, and Python. In addition, VS Code enables you to add new extensions like code linters, debuggers, and support for cloud and web development.

### *3.2.3 Jupyter Notebook*

An open-source web tool called the Jupyter Notebook enables you to create and share documents with real-time code, equations, visualisations, and

narrative text. It can be used for a variety of tasks, such as machine learning, statistical modelling, data visualisation, and data cleaning and transformation. A web-based interactive computing environment called Jupyter Notebook is used to create Jupyter notebook documents. Depending on the context, the term "notebook" can refer to a wide range of objects, principally the Jupyter web application, Jupyter Python web server, or Jupyter document format.

### *3.2.4 Google Colab*

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

### *3.2.5 HTML*

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

### *3.2.6 JavaScript*

JavaScript often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[10] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.

# CHAPTER 4
# SYSTEM ANALYSIS

The practise of conducting a systems analysis on a business scenario in order to create a system solution to a problem or come up with improvements to that situation is known as systems improvement. Before the creation of any system can start, a project proposal is created by the intended users and/or systems analysts and submitted to the proper managerial structure within the company.

The path toward examining a business situation with the aim of improving it through improved strategies and processes is part of a good system inquiry. System analysis is done to investigate a system or its components in order to pinpoint its goals. It is a technique for solving problems that makes the system better and makes sure that all of its parts function effectively to serve their intended purposes.

## *4.1 Requirements Analysis*

Requirements analysis, often known as requirements engineering, is the process of determining what customers want from a new or modified product. These highlights, also known as requirements, must be quantitative, meaningful, and listed. Such requirements are commonly referred to be helpful judgments in programming design. In this project the requirements are:

- Image of disease affected leaf
- Analysis of the disease classification
- Pesticide Recomondation

### *4.2 Existing System*

There are many model that are avilable for detecting plant leaf disease. In those system they use CNN, K-means algorithm which are widely used in image classification problems.

### *4.3 Limitations of existing system*

- Low accuracy
- Using lesser amount of dataset
- Nothing for recommending the remedial

### *4.4 Proposed System*

In the proposed system, it will analyse the disease affected plant leaf and provide an overall view of the disease. Also the system will provide the pesticides which are recommended to reduce further growth of disease and protect the field.

### *4.5 Advantages of proposed System*

- High accuracy
- User friendly
- Similar to YouTube, viewers can search for videos
- Recommendation for the identified disease

### *4.6 Feasibility Study*

A significant level container form of the entire system analysis and design process is a feasibility study. The problem definition is first grouped for the examination. Determine whether the task is feasible before attempting it. A credible model of the framework is constructed by the expert once

an acknowledgement issue specification has been developed. The search for options is carefully examined. The feasibility study is divided into 3 sections.

### *4.6.1 Technical Feasibility*

Technical feasibility is an investigation of function performance and requirements that may influence the capacity to accomplish an adequate framework. It is every now and again the most troublesome area to evaluate at this phase of framework advancement process. During specialized examination, the investigator assesses the specialized benefits of the system idea, while simultaneously gathering extra data about performance,reliability, viability and reducibility, the principle specialized issues generally raised during attainability phase of examination incorporate.

In this project, rather than using convolutional image processing technique which are less accurate ResNet-50 an exponentially growing image processing technique is used. Thus the system will yield a good performance. Current technical resources are sufficient for the plant leaf disease detection system. By considering these facts this project is technically feasible.

### *4.6.2 Operational Feasibility*

The project is profitable only if it deduces the correct disease affected by the leaf from the given image. In short, this crash test check whether the system will work if it is build and installed. Here are some points to evaluate the project's performance:

- System should analyze all feasible solution for finding the disease.
- System should provide accurate analysis report about the disease.
- System should provide correct disease preventation methodology.

### *4.6.3 Economical Feasibility*

Cost-benefit-analysis is among the most significant data contained in a feasibility study, which is an appraisal of economic justification or a PC based framework project cost-benefit-analysis investigation portrays costs for projects advancement and loads them against tangible and intangible of system. To create system actually for the utilization it need money related advantages or on the other hand surpasses the expenses or made equivalent. In this project the softwares used to develop the system were open source. Therefore the project is economically feasible.

# CHAPTER 5
# SYSTEM DESIGN

## *5.1 Application Architecture*

Fig. 5.1 is the architecture of the proposed system.

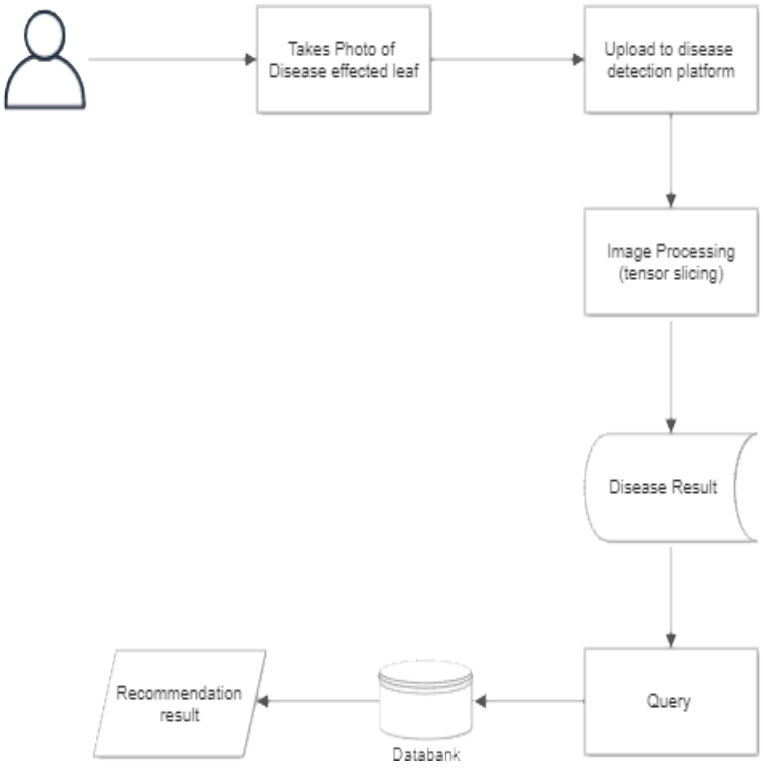



Fig. 5.1: Architecture Diagram

The user will take photo from the field which is affected by leaf disease. Then they upload the photo to disease detection platform where the photo is preprocessed and analysed for result. The picture undergo tensor slicing and analysed using pre trained ResNet-50 Model. The system will find the affected disease and then lead to a query search where it will find the remedy to the disease.

### *5.2 Process Flow Diagram*

The figure 5.2 is the process flow diagram of the system. It shows the overall system.

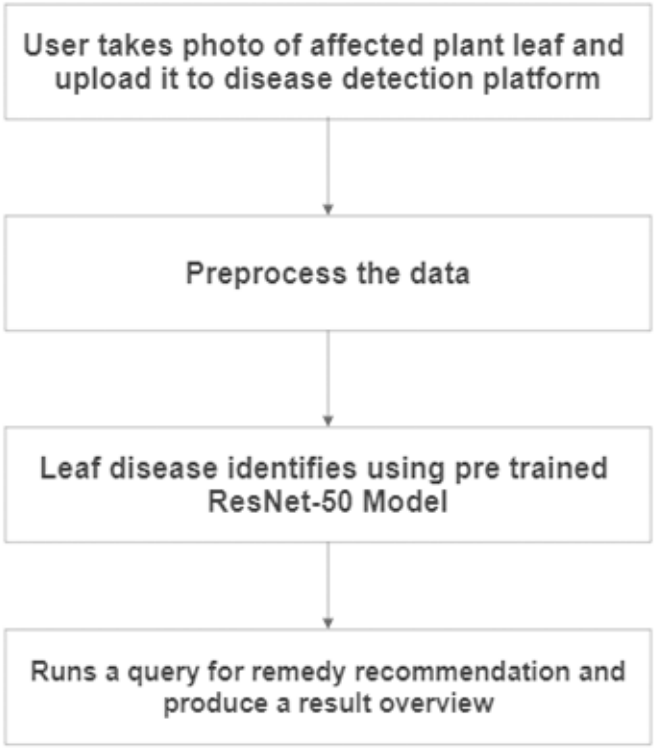



Fig. 5.2: Process Flow Diagram

### *5.3 List of Modules*

Plant Leaf Disease Detection Using ResNet-50 consists of mainly 4 modules. They are:

- Dataset Preparation

- Data Pre-processing
- Training and testing the Model
- Deploy the system

### *5.3.1 Dataset Preparation*

This module consists of dataset with images of Paddy and Maize leaf which are affected by various diseases. The datset is collected from Kaggle. The dataset of Paddy is by HUY MINH DO [4] and Maize from SMARANJIT GHOSE [5]. Each datset consist of different range of images. For equal use of all disease here the project use 523 image of each set.

### *5.3.2 Data Pre-processing*

The selected dataset is pre-processed using tensorflow slicing. Here system resize the original leaf image dataset using flipping, cropping and rotation techniques as well as to convert the leaf images into RGB using color transformation technique. It is done to maintain the balanced quality and size of images in the healthy and unhealthy leaf datasets.

### *5.3.3 Training and Testing the Data*

In this module, the ResNet-50 pre-trained network models are performed to automatically classify the leaf images of paddy and maize plants according to their disease classes. At first, ResNet-50 architectures are executed on 4184 different leaf images where 3200 is taken for training and rest 984 for testing. The system classifies the paddy leaf as,

- Brown Spot
- Leaf Blast images
- Hispa (beettle disease)
- Healthy

and maize images as,

- Early blight
- Common rust
- Grey leaf spot
- Healthy images

#### *5.3.4 Deployment of the system*

In this module, user interface was created using HTML and JavaScript. In the user interface there is place for uploading the image for user. There is also a button to submit. An interface is provided for giving back the result to the user.

# CHAPTER 6
# SYSTEM IMPLEMENTATION

## *6.1 Dataset Collection & Preparation*

Dataset collection and preparation is a key role in making this project live. From over 7000 image datset, 4184 images are selected for tarining and testing. This images are labelled and taken for next step. The images are prepared to 224x224 desire input size. Some of the sample images after labelling and augmentation are shown below,


Fig. 6.1: Dataset

### *6.2 Loading Data set*

The image dataset collected from Kaggle is loaded to the model after labelling and augmenting. The image is aumented using tensorflow pre-processing

### *6.3 Tensorflow Pre-processing*

Here using tensorflow the dataset image is turns into numbers, normalize pixel values, resize shape of the image

```
# read, turn image into number, normalize, resize
def preprocess_image(image_path, labels=None):
    # read image
    image = tf.io.read_file(image_path)
    # turn jpeg into numbers
    image = tf.image.decode_jpeg(image, channels=3)
    # scaling / normalize (0,255) becomes (0,1)
    image = tf.image.convert_image_dtype(image, dtype=tf.float32)
    # resize to (224,224)
    image = tf.image.resize(image, size=[IMAGE_SIZE, IMAGE_SIZE])
    # return
    return image, labels
```

Fig. 6.2: Fitting the model

### *6.4 Model Creation*

Modeling in machine learning is an iterative phase where a data scientist continually train and test machine learning models to discover the best one for the given task. The ResNet-50 is used here creating model.

```
[ ]  model = tf.keras.Sequential([
                                  # transfer learning model
                                  hub.KerasLayer("https://tfhub.dev/tensorflow/resnet_50/feature_vector/1"),

                                  # output layer
                                  tf.keras.layers.Dense(units=num_unique_label, activation='softmax')
     ])
```

Fig. 6.3: Fitting the model

### *6.5 ResNet-50*

ResNet-50 is a convolutional neural network that is 50 layers deep. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

When working with deep convolutional neural networks to solve a problem related to computer vision, machine learning experts engage in stacking more layers. These additional layers help solve complex problems more efficiently as the different layers could be trained for varying tasks to get highly accurate results. While the number of stacked layers can enrich the features of the model, a deeper network can show the issue of degradation. In other words, as the number of layers of the neural network increases, the accuracy levels may get saturated and slowly degrade after a point. As a result, the performance of the model deteriorates both on the training and testing data. This degradation is not a result of overfitting. Instead, it may result from the initialization of the network, optimization function, or, more importantly, the problem of vanishing or exploding gradients.

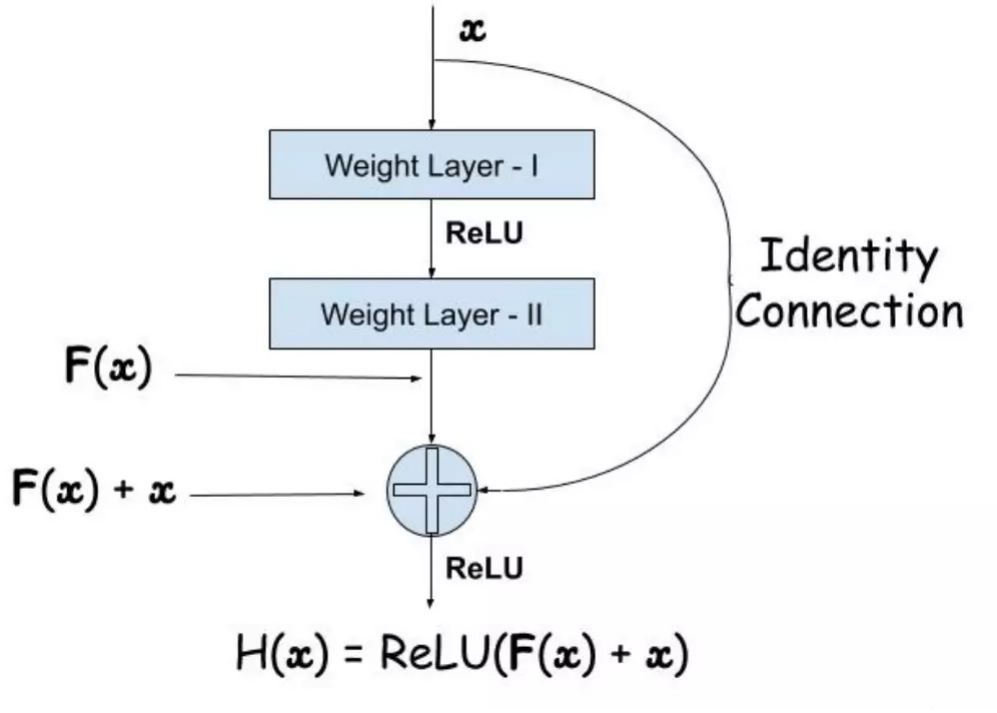



Fig. 6.4: A Residual Block of Deep Residual Network

ResNet was created with the aim of tackling this exact problem. Deep residual nets make use of residual blocks to improve the accuracy of the models. The concept of "skip connections," which lies at the core of the residual blocks, is the strength of this type of neural network.

### *6.5.1 Skip Connections in ResNet-50*

These skip connections work in two ways. Firstly, they alleviate the issue of vanishing gradient by setting up an alternate shortcut for the gradient to pass through. In addition, they enable the model to learn an identity function. This ensures that the higher layers of the model do not perform any worse than the lower layers. In short, the residual blocks make it considerably easier for the layers to learn identity functions. As a result, ResNet improves the efficiency of deep neural networks with more neural layers while minimizing the percentage of errors. In other words, the skip connections add the outputs from previous layers to the outputs of stacked layers, making it possible to train much deeper networks than previously

possible.

### *6.5.2 Keras and ResNet-50*

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. Here in the project, Keras is used in creating the model,creating early stopping and saving the model.

```
import keras
from keras.models import load_model

model.save('resnet50_project.h5')
!mkdir -p saved_model
model.save('saved_model/my_model')
```

Fig. 6.5: Saving model using keras

## *6.6 Early Stopping Callback*

A callback is an object that can perform actions at various stages of training (e.g. at the start or end of an epoch, before or after a single batch, etc).Early Stopping monitors the performance of the model for every epoch on a held-out validation set during the training, and terminate the training conditional on the validation performance. In this project early stopping callback is used for preventing overfitting.

```
# EARLYSTOPPING CALLBACK
# monitor the val loss (prevent overfitting)

early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3)
```

Fig. 6.6: Saving model using keras

### *6.7 Testing*

Software Testing is the process of evaluation of the application to that of the user preference or requirements, and evaluating whether the requirements are met. It is performed along with the stages of development, as each individual changes introduced are tested multiple times to make sure the desired output is generated. The software is tested by searching different Malayalam Youtube videos to check whether the system gives accurate results. The model also tested by giving custom English-Malayalam code-mix comments.

#### *6.7.1 Unit Testing*

The process of testing each and every individual units of module is called Unit testing. It is used to make sure that the application runs error free and effortlessly after actual deployment. The Youtube comments were scraped using google-api-client tool. It's tested whether it scrape all the comments or not. In the analysis part, it's tested whether the model provides accurate results or not.

# BIBLIOGRAPHY

[1] G. Zhou, W. Zhang, A. Chen, M. He, and X.Ma, *"Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion,"*
IEEE Access, vol. 7, pp. 143190-143206, Sep. 2019.

[2] P. Sharma ; P. Hans ; S. C. Gupta, *"Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques,"*
in Int. Conf. on Cloud Com., Data Sc. Eng., Noida, INDIA, 2020

[3] U. Shafi, R. Mumtaz, N. Iqbal, S. M. H. Zaidi, S. A. R. Zaidi, I. Hussain,Z. Mahmood, *"A multi-modal approach for crop health mapping using low altitude remote sensing, Internet of Things (IoT) and machine learning,"*
IEEE Access, vol. 8, pp. 112708-112724, Jun. 2020

[4] Rice Diseases Image Dataset by HUY MINH DO *https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset*

[5] Corn or Maize Leaf Disease Dataset by SMARANJIT GHOSE *https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset*