

# **FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION**

## **ABSTRACT:**

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

## **PROJECT OBJECTIVES:**

- Pre-process the images.
- Applying the CNN algorithm to the dataset.
- How deep neural networks detect the disease.
- You will be able to know how to find the accuracy of the model.
- You will be able to build web applications using the Flask framework.

## **USE CASES:**

- Detection of plant disease through some automatic technique is beneficial as it reduces a large work of monitoring in big farms of crops.
- Plant disease identification by visual way is more laborious task and at the same time, less accurate and can be done only in limited areas.
- Whereas if automatic detection technique is used it will take less efforts, less time and become more accurate.
- At very early stage itself it detects the symptoms of diseases i.e. when they appear on plant leaves.

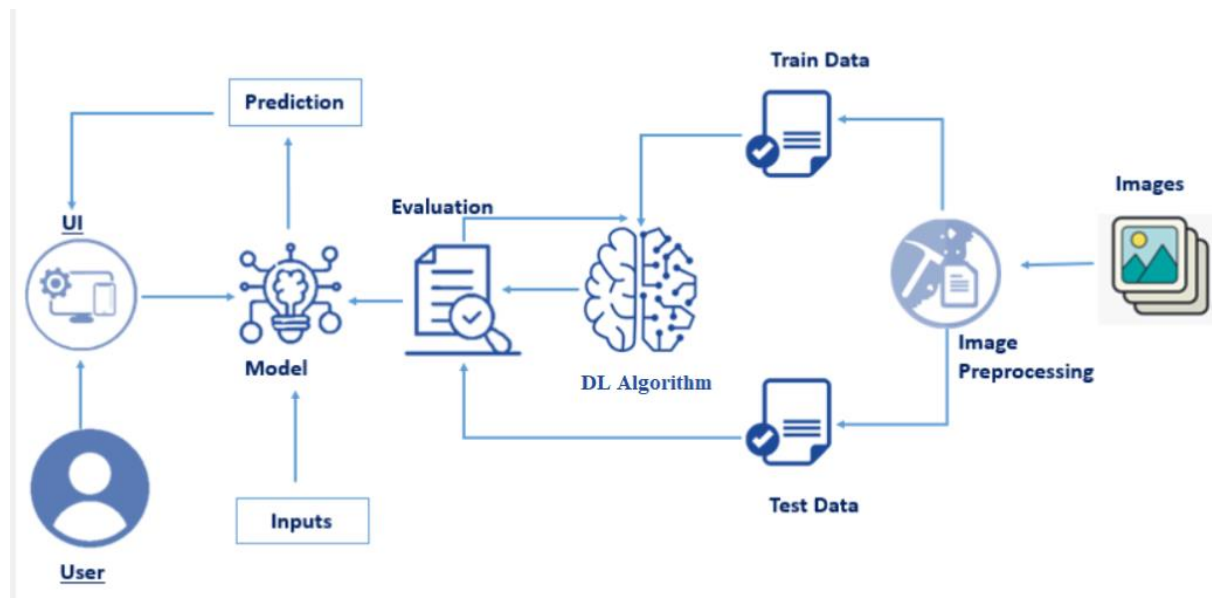
## **PROJECT FLOW:**

A web Application will be built in that

- Farmers can interact with the portal
- Farmers can upload images of diseased leaf
- Model analyses the Disease and suggests the farmer with fertilizers are to be used

To accomplish the above task following tasks should be performed

- Download the dataset and classify into train and test sets.
- Add the neural network layers.
- Load the trained images and fit the model.
- Test the model.
- Save the model and its dependencies.
- Build a Web application using a flask that integrates with the model built.



## **SOFTWARE REQUIREMENTS:**

- Anaconda – Jupyter Notebook

## **LIBRARIES REQUIRED:**

- Tensorflow
- Keras

## **FRAMEWORK REQUIRED:**

- Flask

## **STEPS INVOLVED:**

### 1.Data collection :

- Collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.
- For this project dataset is provided by the IBM.

### 2.Image Pre-processing:

- import ImageDataGenerator Library.
- Configure ImageDataGenerator Class.
- Applying ImageDataGenerator functionality to the trainset and test set.

### 3.Model building :

- Import the model building Libraries.
- Initializing the model.
- Adding CNN Layers.
- Adding Hidden Layer.
- Adding Output Layer.
- Configure the Learning Process.
- Training and testing the model.
- Saving the model.

### 4.Creating user interface:

After building the model, integrate the model into web application using flask framework.

## **By:**

1905096 - Murugalakshmi M

1905098 - Neharika A R

1905118 - Tharunika PA

2005207 - Harshita K