

Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID26231

NIRANJANA SR	(211519205104)
KIRTHANA P	(211519205079)
HARSHA VARDINI G	(211519205060)
HARINI M	(211519205057)
NOWZIYA BANU Y	(211519205108)

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY



**PANIMALAR INSTITUTE OF TECHNOLOGY,
CHENNAI.**

1.INTRODUCTION

1.1 Overview In this project, two datasets named fruits and vegetables are collected. The collected datasets are trained and tested using a deep learning neural network called Convolutional Neural Networks (CNN). First, the fruit set is trained and then tested using CNN. It has 6 classes and all classes are trained and tested. Second, a vegetable dataset is trained and tested. The software used for training and testing datasets is Python. All Python code is first written in the Jupyter notebook that comes with Anaconda Python, and then the code is tested in the IBM cloud. Finally, a web framework is designed with the help of Flask and the Python library. 2 html files are created in the templates folder along with their associated files in the static folder. The python program 'app.py' used to link to these two web pages is written in Spyder-Anaconda python and tested.

1.2 Purpose This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.

2.LITERATURE SURVEY

2.1 Existing problem Indumathi proposed a method for leaf disease detection and suggest fertilizers to cure leaf diseases. But the method involves less number of train and test sets which results in poor accuracy. Pandi selvi proposed a simple prediction method for soil-based fertilizer recommendation system for predicted crop diseases. This method gives less accuracy and prediction. Shiva reddy proposed an IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

2.2 Proposed solution In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classification accuracies. By increasing

the more dense layers and by modifying hyperparameters such as number of epochs, batch size, the accuracy rate can be increased to 95% to 98%.

3.THEORITICAL ANALYSIS

3.1 Block diagram

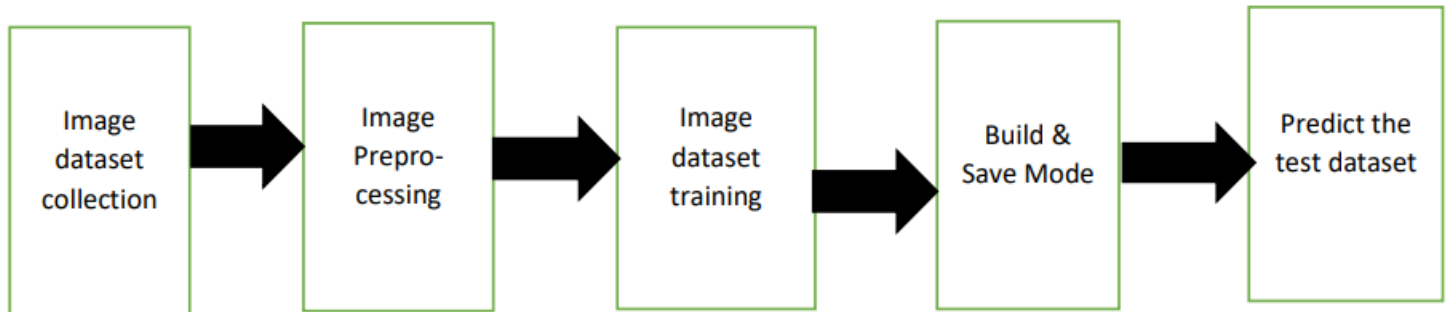


Figure.3.1. Block Diagram of the project

The block diagram of the entire project is shown in Fig.3.1. First step is the image dataset collection followed by image preprocessing. The third step is the training of image datasets with initializing different hyper parameters. Then build the model and save the model file with .h5 format. The final stage is the testing of existing or new datasets using the trained model.

3.2 Hardware/Software designing

The software used for training and testing the dataset is Python. The Jupyter notebook (Notebook of IBM cloud also) is used for python programming. The neural network used for training and testing the model is Convolutional Neural Network (CNN).

The CNN has following layers:

- Convolutional layer (32x32 kernal (3x3))

- Max-pool layer (kernel(2x2))-
- Flatten layer
- Dense layer (different layers with different size)
- Drop out layer (optional)
- Final output dense layer(size 6x1 for fruit dataset and 9x1 for Vegetable dataset)

In the preprocessing step, images are normalized to 1 and then resized to 128x128. The images are arranged in different batch sizes. Then train set and test set are formed from the collected datasets. In order to do the above steps in Python, the following Python libraries must be imported before starting the process:

- NumPy
- TensorFlow
- Keras
- Matplotlib (optional for data visualization)

The following activation functions used in the CNN training:

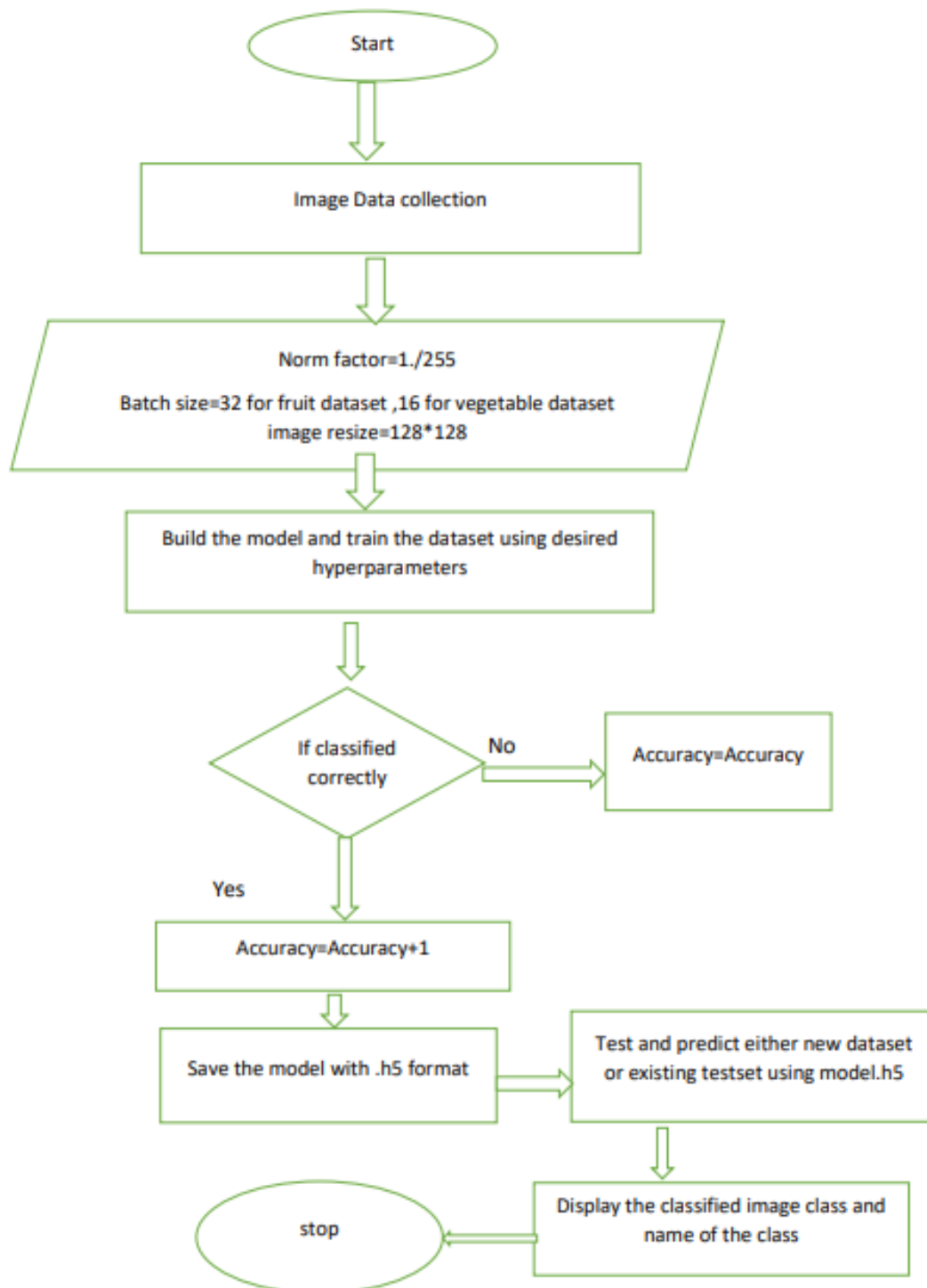
- RELU at the end of convolution layer and Max Pool layer
- SoftMax at the end of output dense layer
- For testing the dataset argmax is used, its an optional

4.EXPERIMENTAL INVESTIGATIONS

Analysis made while working on the solution The batch sizes are varied and tested. For different batch sizes, the CNN gives different accuracies. The batch size

determines the number of iterations per epoch. Another important hyper parameter is the number of epochs. This determines accuracy and it has high influence on accuracy compared to other hyper parameters. The accuracy can be varied from 80% to 90% in vegetable dataset and 95% to 98% in the case of fruit dataset by increasing the number of epochs. The size of test dataset and train dataset also has very high influence on accuracies. The accuracy can be increased by using more number of images in train dataset. The computational time for model building is increased when the size of the train dataset increased and also number of epochs increased. The batch size of train dataset and test dataset also play a vital role in computational time. The Neural Network complexity is increased when more number of convolutional layers increased. If the number of layers increased, better accuracy result will obtain. At the same increasing the number of layers in CNN leads to more training time and also requires more time to build a model. The model .h5 size depends on the size of train datasets. But the memory requirement depends on the size of train dataset and CNN architecture complexity.

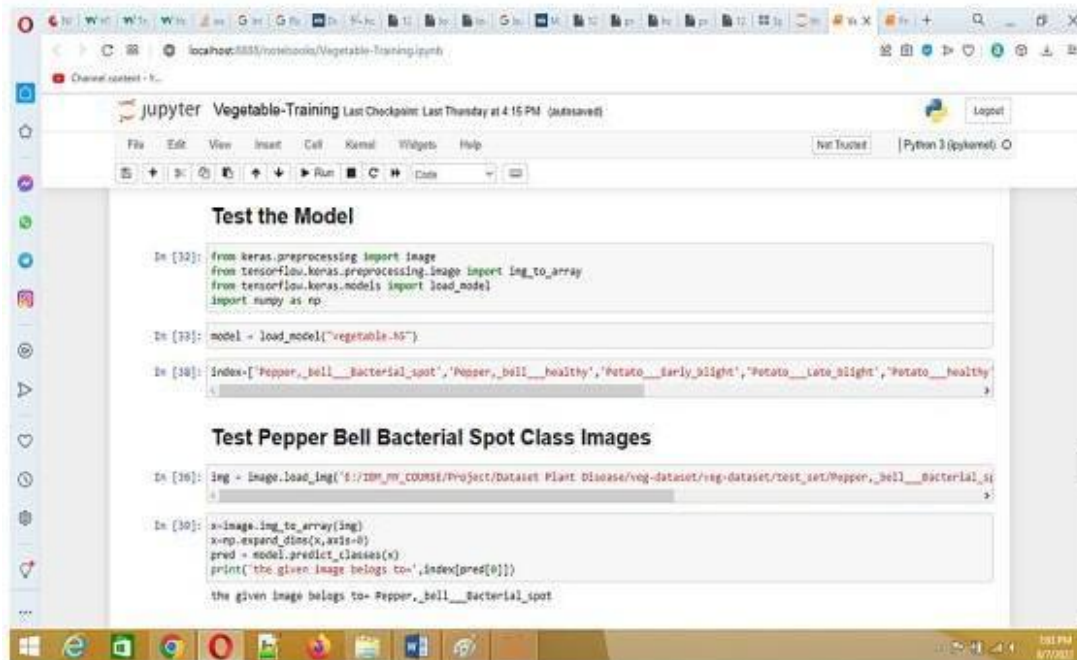
5.FLOWCHARTS



6.RESULTS

Final findings(output) of the project given below in the form of screenshot:
Training and Testing of Fruit dataset

Figure.6.3. Train the Vegetable dataset



```
Test the Model

In [32]: from keras.preprocessing import image
         from tensorflow.keras.preprocessing.image import img_to_array
         from tensorflow.keras.models import load_model
         import numpy as np

In [33]: model = load_model("vegetable.h5")

In [38]: index=['Pepper_bell__Bacterial_spot', 'Pepper_bell__healthy', 'Potato__Early_blight', 'Potato__Late_blight', 'Potato__healthy']

Test Pepper Bell Bacterial Spot Class Images

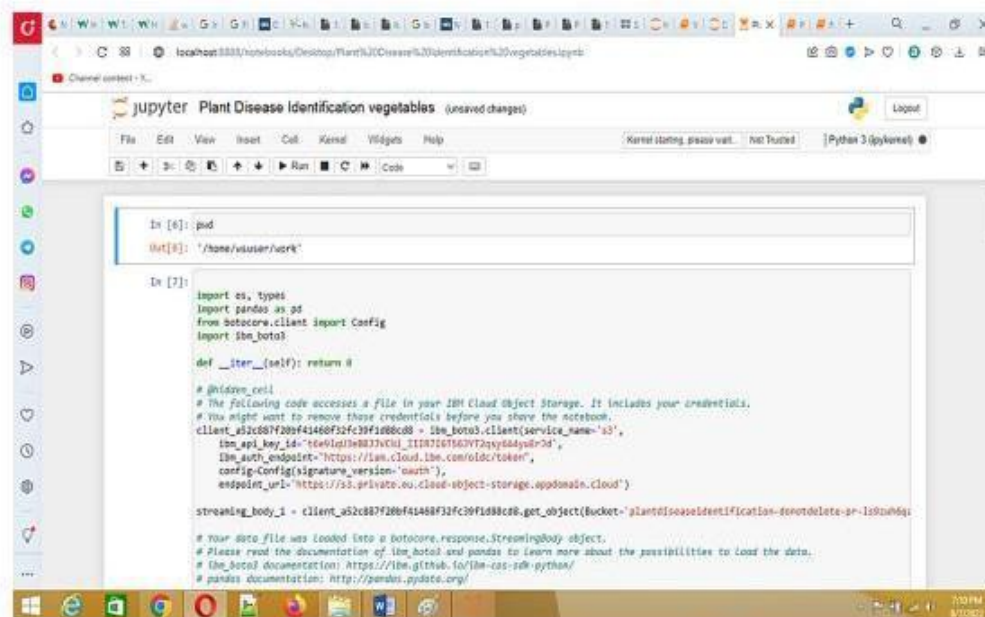
In [36]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Pepper_bell__bacterial_spot.jpg')

In [39]: x = image.img_to_array(img)
         x = np.expand_dims(x, axis=0)
         pred = model.predict_classes(x)
         print('the given image belongs to-', index[pred[0]])

the given image belongs to- Pepper_bell__Bacterial_spot
```

Figure.6.4. Test the Vegetable dataset

Figure.6.5. Training Fruit Dataset in IBM Cloud



```
jupyter Plant Disease Identification vegetables (unsaved changes)

In [6]: pwd
Out[6]: '/home/user/work'

In [7]: import os, types
         import pandas as pd
         from boto3.client import Config
         import boto3

         def __iter__(self): return 0

         # @i3dev_cell
         # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
         # You might want to remove these credentials before you share the notebook.
         client = boto3.client(service_name='s3',
                               iam_key_id='tc6f1d1b88373VU1_11R7G75G7V72qy6d4p8r3d',
                               iam_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
                               config=Config(signature_version='auth'),
                               endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

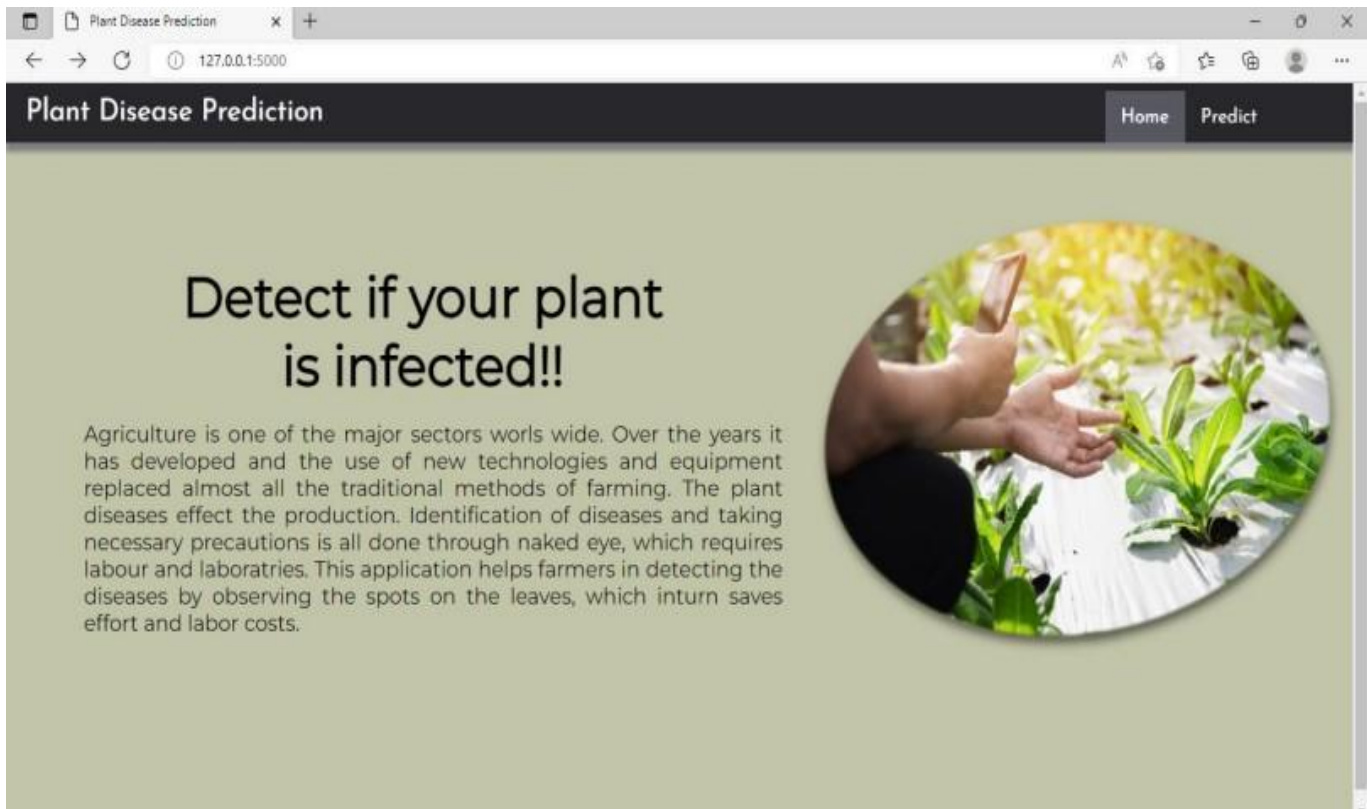
         streaming_body_1 = client.get_object(Bucket='plantdiseaseidentification-donutdelete-pr-129uh6q',
                                               Key='data/vegetable_dataset.zip')

         # Your data file was loaded into a boto3.response.StreamingBody object.
         # Please read the documentation of lib_boto3 and pandas to learn more about the possibilities to load the data.
         # lib_boto3 documentation: https://github.com/ibm-cas-sdx/python/
         # pandas documentation: http://pandas.pydata.org/
```

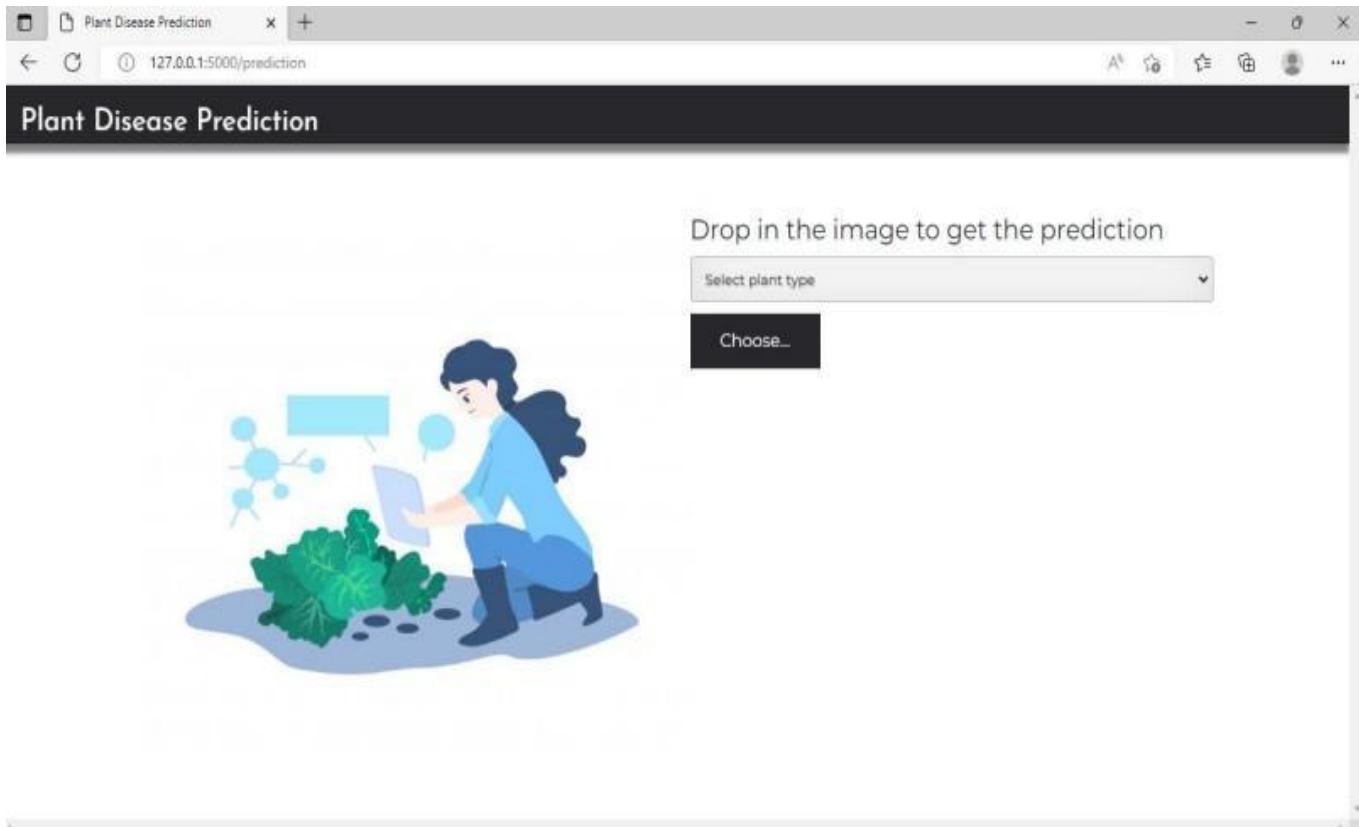
Figure.6.6. Training Vegetable Dataset in IBM Cloud

Output:

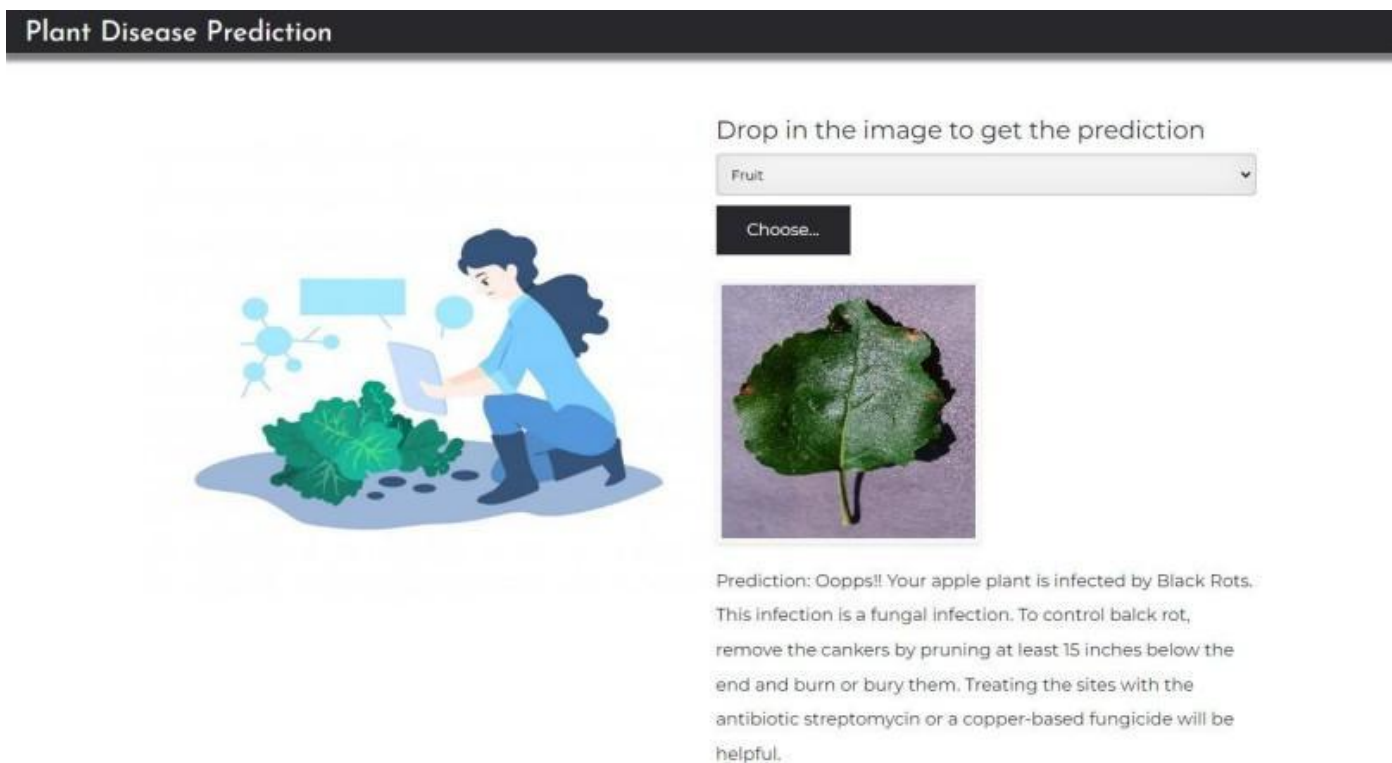
Home page:

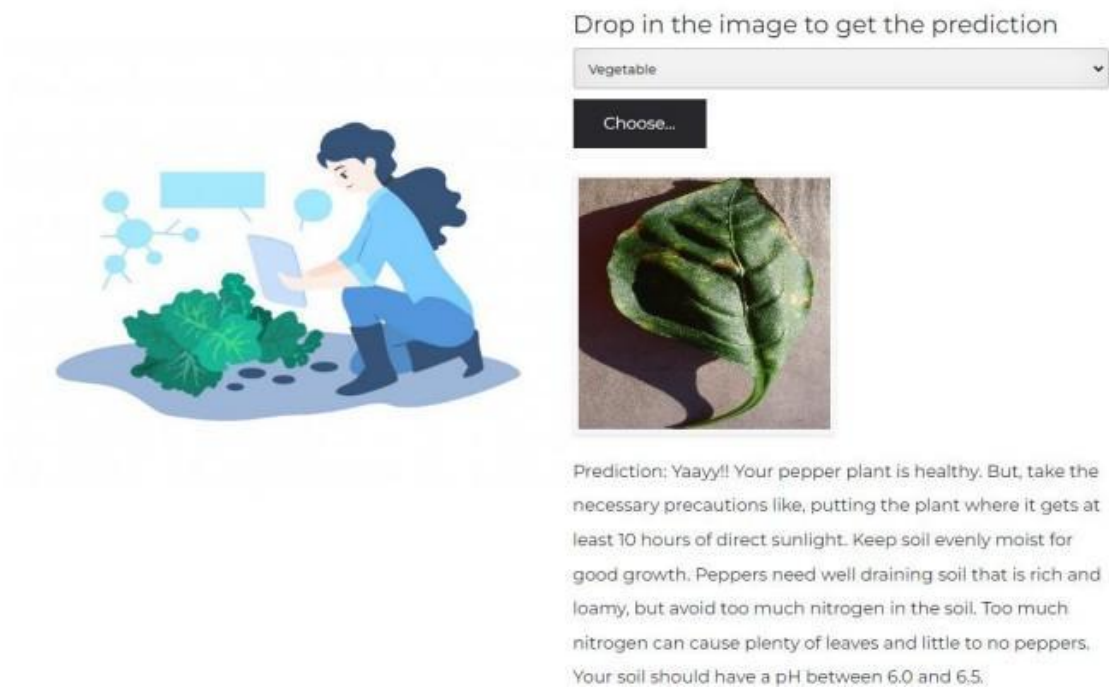


➤ Prediction Page:



➤ Result Page:





7.ADVANTAGES & DISADVANTAGES

List of advantages

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself.

List of disadvantages

- For training and testing, the proposed model requires very high computational time.
- The neural network architecture used in this project work has high complexity

8.APPLICATIONS

1. The trained network model used to classify the image patterns with high accuracy.
2. The proposed model not only used for plant disease classification but also for other image pattern classification such as animal classification.
3. This project work application involves not only image classification but also for pattern recognition.

9.CONCLUSIONS

The model proposed here involves image classification of fruit datasets and vegetable datasets. The following points are observed during model testing and training:

- The accuracy of classification increased by increasing the number of epochs.
- For different batch sizes, different classification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets

10. FUTURE SCOPE

The proposed model in this project work can be extended to image recognition. The entire model can be converted to application software using python to exe software. The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

11.BIBILOGRAPHY

[1]. R Indumathi Leaf Disease Detection and Fertilizer Suggestion", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 29-30 March 2019, DOI: 10.1109/ICSCAN.2019.8878781.

[2]. P. Pandi Selvi, P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System", International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2, Mar-Apr 2021.

[3]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395- 005.