Fertilizers Recommendation System For Disease Prediction

1 INTRODUCTION :

1.1 Overview :

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.

1.2 Purpose :

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.

2 LITERATURE SURVEY:

2.1 Existing problem:

In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production. Due to climatic change, inconsistent rainfall patterns, pollution, and contamination in the water, farmers are experiencing issues with their irrigation systems. As we have seen in the past, an oversupply of water can sometimes cause agricultural damage.

Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an external display or an Android application. The application was created to measure the approximate values of temperature, humidity,and moisture sensors that were programmed into a microcontroller to manage the amount of water.

Pi Doctor: A Low Cost Aquaponics Plant Health Monitoring System Using Infragram Technology and Raspberry Pi present a low-cost aquaponics plant health monitoring system based on a modified web camera that gathers both infrared and visible data in a single image in this research. The NDVI is calculated from the NGB picture (nearinfrared, green, and blue) to quantify the health and state of the vegetation in the system. Exposing the system to diverse conditions, such as moderate, sunny, and dark, allows for the investigation of other factors in the aquaponics system, such as pH and nitrogen usage by the plants. The study also determines the ideal pH for the water in the fish tank and the soil in the aquaponics system's grow bed.

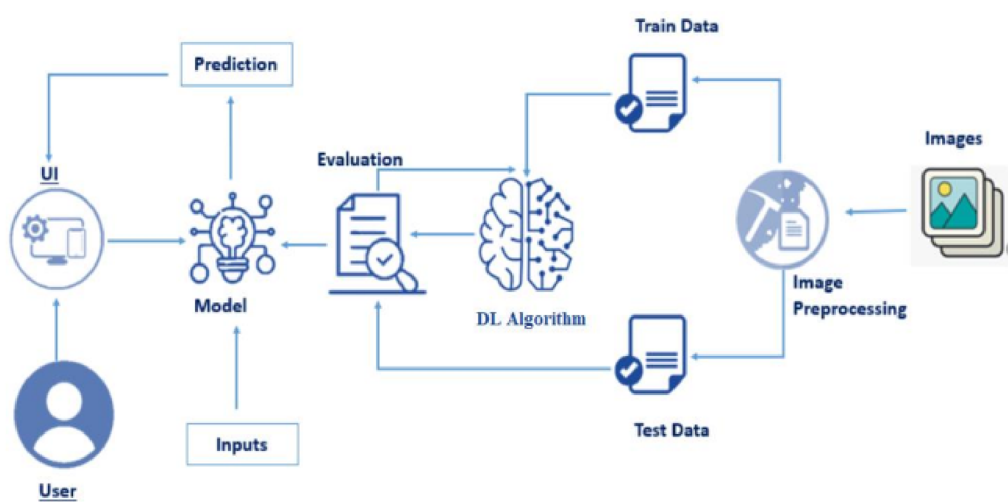
2.2 Proposed solution:

A web Application is built where:

1. **Farmers interact with the portal build**
2. **Interacts with the user interface to upload images of diseased leaf**
3. **Our model built analyses the Disease and suggests the farmer with fertilizers are to be used**

3 THEORITICAL ANALYSIS :

3.1 Block diagram :



3.2 Hardware / Software designing:

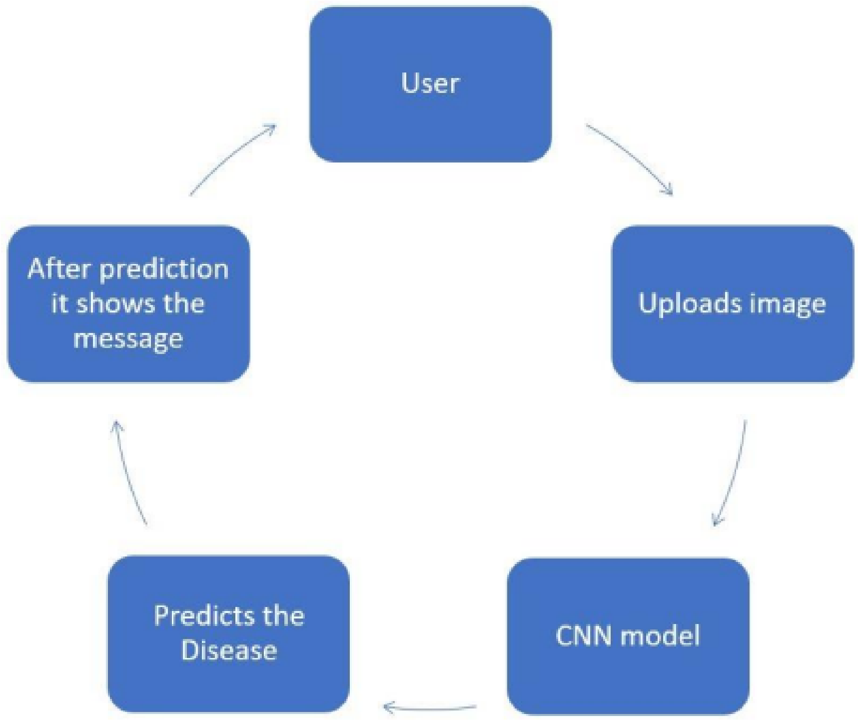
software :
Anaconda navigator
Vs code / py charm
Jupiter notebook
IBM watson studio

By using the above listed softwares we build this application to take the input (image) from the farmer and detects whether the pant is infected or not.
Here we use Deep learning techniques and give the out put to the user (Farmer).

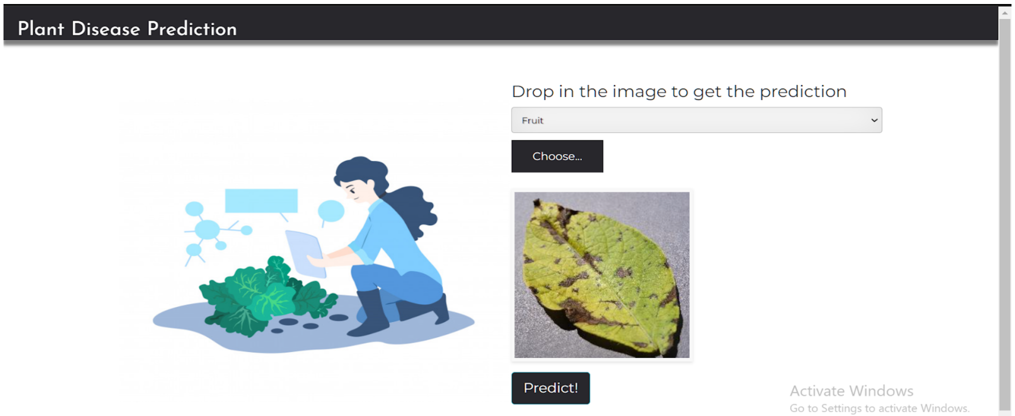
4 EXPERIMENTAL INVESTIGATIONS :

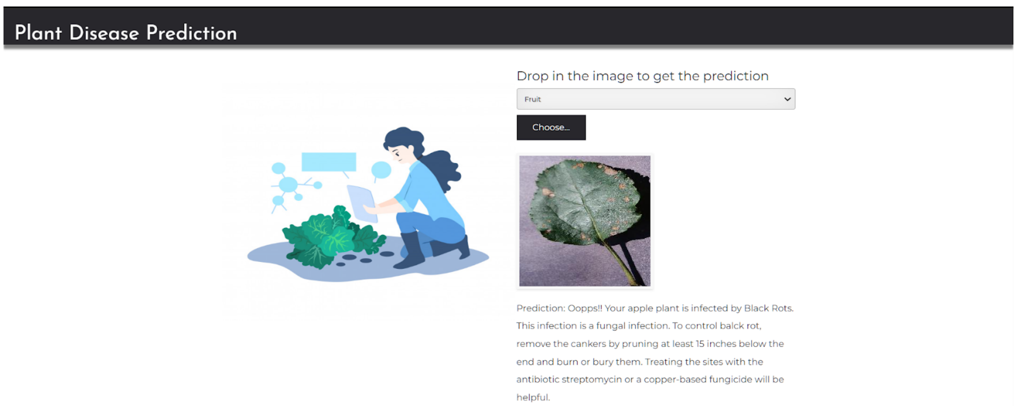
Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential 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. Analysis or the investigation made while working on the solution.

5 FLOWCHART :



6 RESULT:





7 ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

1. The proposed model could predict the disease just from the image of a particular plant
2. Easy to use UI
3. Model has some good accuracy in detecting the plant just by taking the input(leaf).

DISADVANTAGES:

1. Prediction is limited to few plants as we havent trained all the plants

8 APPLICATIONS :

This webapplication can be used by farmers or users to check whether their plant is infected or not and can also show the remedy so that the user can take necessary precautions.

These kind of web applications can be used in the agricultural sector as well as for small house hold plants as well.

9 CONCLUSION :

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.
Usage of such applications could help the farmers to necessary precautions so that they dont face any loss as such.

10 FUTURE SCOPE :

As of now we have just build the web application which apparently takes the input as an image and then predict the out in the near future we can develop an application which computer vision and AI techniques to predict the infection once you keep the camera near the plant or leaf this could make our project even more usable.

11 BIBILOGRAPHY :

- http://www.ijstr.org/final-print/nov2019/Fertilizers-Recommendation-System-For-Disease-Prediction-In-Tree-Leave.pdf

- https://www.semanticscholar.org/paper/Fertilizers-Recommendation-System-For-Disease-In-Neela-Nithya/495379d3ef2b461fabd2de8d0605c164cb1e396f

- http://www.ijetajournal.org/volume-8/issue-2/IJETA-V8I2P1.pdf

- https://ieeexplore.ieee.org/document/8878781

- https://www.sciencedirect.com/science/article/pii/S0168169921004245
- https://www.irjet.net/archives/V7/i10/IRJET-V7I1004.pdf

APPENDIX :

A. Source Code:

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen =ImageDataGenerator(rescale = 1)
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_96505fb499a44047b92e8974d557e092 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='i9yXmsVr9-SXp9rzTMTR7sZliB-mD6rNljW0U8fwZ4bF',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')



# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
streaming_body_5 = client_96505fb499a44047b92e8974d557e092.get_object(Bucket='fertilizersrecommendationsystemfo-donotdelete-pr-hc
```

```
In [ ]: from io import BytesIO
        import zipfile
        unzip = zipfile.ZipFile(BytesIO(streaming_body_5.read()),'r')
        file_paths = unzip.namelist()
        for path in file_paths:
            unzip.extract(path)
```

```
In [ ]: pwd
```

```
In [ ]: import os
        filenames=os.listdir('/home/wsuser/work/dataset/Veg-dataset/Veg-dataset/train_set')
```

```
In [ ]: x_train = train_datagen.flow_from_directory('/home/wsuser/work/dataset/Veg-dataset/Veg-dataset/train_set',
                                                     target_size = (128,128),
                                                     batch_size = 16,
                                                     class_mode = 'categorical')
```

```
In [ ]: x_test = test_datagen.flow_from_directory('/home/wsuser/work/dataset/Veg-dataset/Veg-dataset/test_set',
                                                   target_size = (128,128),
                                                   batch_size = 16,
                                                   class_mode = 'categorical')
```

```
In [ ]: print(x_train.class_indices)
```

```python
from keras.models import Sequential
from tensorflow.keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

```python
from tensorflow.keras.models import Sequential
model = Sequential()
```

```python
model.add(Convolution2D(32,(3,3), input_shape = (128,128,3), activation = 'relu'))
```

```python
model.add(MaxPooling2D(pool_size = (2,2)))
```

```python
model.add(Flatten())
```

```python
model.add(Dense(units = 300, activation ='relu'))
```

```python
model.add(Dense(units = 150, activation ='relu'))
```

```python
model.add(Dense(units = 75, activation ='relu'))
```

```python
model.add(Dense(units= 9,activation = 'softmax'))
```

```python
model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])
```

```python
model.fit_generator(x_train,steps_per_epoch = 89,
                    epochs = 5,
                    validation_data = x_test,
                    validation_steps = 27)
```

```python
model.save('vegetable.h5')
```

```python
import keras

from keras.utils import generic_utils
from tensorflow.keras import models
from keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array



import numpy as np
import tensorflow as tf
import cv2
```

```python
model = load_model("vegetable.h5")
```

```python
model.summary()
```

```python
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

```python
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
test_datagen =ImageDataGenerator(rescale = 1)
```

```python

import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_96505fb499a44047b92e8974d557e092 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='OUsZVUuvjK9rSkM6oD6iGXFQUuHPUHiZ7sZUvXv4c-UP',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

streaming_body_1 = client_96505fb499a44047b92e8974d557e092.get_object(Bucket='fruit-donotdelete-pr-04om4ggimzzq8c', Key='dataset.
```

```python
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```python
pwd
```

```
'/home/wsuser/work'
```

```python
import os
filenames=os.listdir('/home/wsuser/work/dataset/fruit-dataset/fruit-dataset/train')
```

```python
x_train = train_datagen.flow_from_directory('/home/wsuser/work/dataset/fruit-dataset/fruit-dataset/train',
                                            target_size = (128,128),batch_size = 32, class_mode = 'categorical')
x_test =  test_datagen.flow_from_directory('/home/wsuser/work/dataset/fruit-dataset/fruit-dataset/test',
                                           target_size = (128,128),batch_size = 32, class_mode = 'categorical')
```

```
Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.
```

```
print(x_train.class_indices)
```

```
{'Apple___Black_rot': 0, 'Apple___healthy': 1, 'Corn_(maize)___Northern_Leaf_Blight': 2, 'Corn_(maize)___healthy': 3, 'Peach___
Bacterial_spot': 4, 'Peach___healthy': 5}
```

```
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))
```

```
model.add(MaxPooling2D(pool_size = (2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(units = 40 ,activation = 'relu'))
model.add(Dense(units = 20 ,activation = 'relu'))
model.add(Dense(units = 6,activation = 'softmax'))
```

```
model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])
```

```
model.fit_generator(x_train, steps_per_epoch = 70,epochs = 3,validation_data = x_test,validation_steps = 52)
```

```
model.save("fruit.h5")
```

```
model.summary()
```