Fertilizers Recommendation System For Disease Prediction

Team: SB

Name: Sharmila B S

Designation and Address: Assistant Professor, The National Institute of Engineering Mysuru.

1. Introduction:

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

2. Purpose:

To Detect and recognize the plant diseases and to recommend fertilizer, it is necessary to provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new fertilizers Recommendation System for crop disease prediction.

3. Methodology:

In this project we have implemented Artificial Intelligence based Fertilizers Recommendation System For Disease Prediction. In this work, we have implemented Framework using Convolution neutral network and implemented Web Page using Flask library in python. This method was trained using relu activation function with adam optimizer. The training was done for 50 epochs in fruit disease prediction and 50 epoch in vegetable disease prediction. Finally model is saves with “Fruit.h5” and Vegetable.h5” model names. The final saved model is called in webpage designed using Flask library. The below block diagram 1 gives overview of our project methodology.

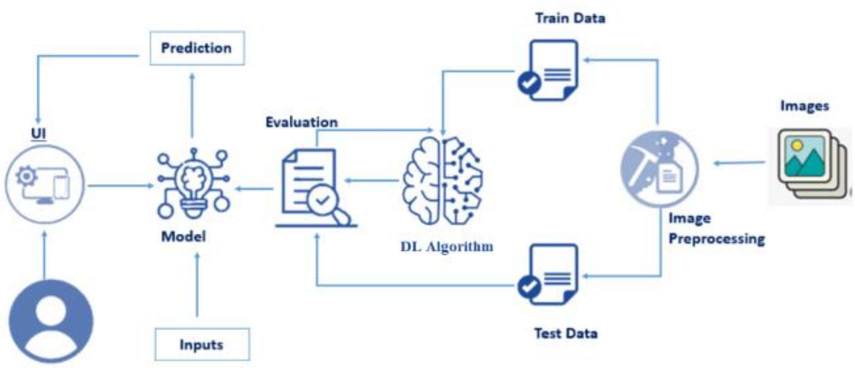


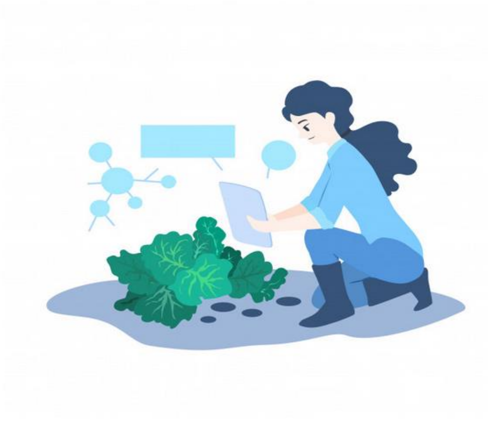
Figure 1: Methodology

4. Experimental Results:

The model trained with both Fruit leaf and vegetable leaf images on local Jupyter Notebook from Anaconda and IBM Watson Studio. The below table 1 shows the Accuracy obtained from both different Frameworks.

Frameworks	Fruit Leaf disease Accuracy	Vegetable Leaf Disease Accuracy
Jupyter Notebook	96.85	94.66
IBM Notebook	97.96	95.44

The below figure 2a show the home webpage and 2b shows the prediction results in webpage.

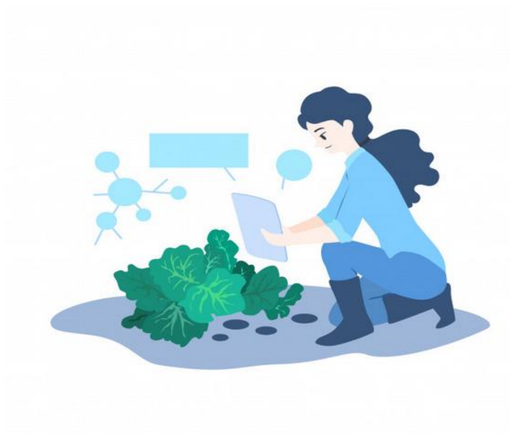


Drop in the image to get the prediction

Fruit

Choose...

Prediction: Yaayy!! Your apple plant is healthy. But, maintain the soil pH of 6.0 to 7.0 for healthy growth. Avoid planting apples in a low spot where cold air or frost can settle.



Drop in the image to get the prediction

Vegetable

Choose...

Prediction: Oopps!! Your pepper plant is infected by Bacterial Leaft Spot. The disease cycle can be stopped by using the Sango formula for disinfectants. Bleach treatment and hot water treatment is also helpful.

Conclusion: In this project, we developed Fertilizers Recommendation System for Disease Prediction using Artificial Intelligence. We also learnt and implemented Webpage using python and Flask. In future we will implement the same for different diseas prediction dataset and we will try to increase accuracy.

Acknowledgment: I would like to thank Hemanth Kumar, Project mentors, SmartIntenz, and IBM Watson Studio to give opportunity to learn on this platform.

Appendix:

Source Code Screen Shot:

In [1]:
```
import numpy as np
import pickle
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
#from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator

from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import tensorflow
```

In [2]:
```
import random
random.seed(123)
```

In [3]:
```
TRAINING_DIR='E:/IDS/Intenrship_IBM/Fertilizers/Dataset/fruit-dataset/fruit-dataset/train/'
TEST_DIR = 'E:/IDS/Intenrship_IBM/Fertilizers/Dataset/fruit-dataset/fruit-dataset/test/'
```

```python
x_train =train_datagen.flow_from_directory(TRAINING_DIR,
                                           target_size=(128,128),
                                           batch_size=32,
                                           class_mode='categorical')
x_test =train_datagen.flow_from_directory(TEST_DIR,
                                           target_size=(128,128),
                                           batch_size=32,
                                           class_mode='categorical')
```

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

```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
```

In [7]:

```python
from keras import backend as K
K.clear_session()
model = Sequential()
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))
model.add(MaxPooling2D(2, 2))

model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))
model.add(MaxPooling2D(2, 2))

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

In [8]:

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

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
flatten (Flatten)	(None, 28800)	0
dense (Dense)	(None, 40)	1152040
dense_1 (Dense)	(None, 20)	820
dense_2 (Dense)	(None, 6)	126

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

```
CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=30,validation_data=x_test, validation_steps = 27).history
```

```
<ipython-input-10-08b1e90848e3>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Pl
ease use `Model.fit`, which supports generators.
  CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=30,validation_data=x_test, validation_steps = 27).history
```

```
Epoch 1/30
89/89 [==============================] - 65s 637ms/step - loss: 1.0910 - accuracy: 0.5815 - val_loss: 0.5214 - val_accuracy: 0.
8241
Epoch 2/30
89/89 [==============================] - 46s 515ms/step - loss: 0.4352 - accuracy: 0.8569 - val_loss: 0.3922 - val_accuracy: 0.
8715
Epoch 3/30
89/89 [==============================] - 55s 622ms/step - loss: 0.3306 - accuracy: 0.8856 - val_loss: 0.3572 - val_accuracy: 0.
8866
Epoch 4/30
89/89 [==============================] - 48s 541ms/step - loss: 0.2869 - accuracy: 0.9010 - val_loss: 0.2542 - val_accuracy: 0.
9167
Epoch 5/30
89/89 [==============================] - 57s 646ms/step - loss: 0.2266 - accuracy: 0.9196 - val_loss: 0.2449 - val_accuracy: 0.
9155
Epoch 6/30
89/89 [==============================] - 49s 554ms/step - loss: 0.2137 - accuracy: 0.9324 - val_loss: 0.2182 - val_accuracy: 0.
9259
Epoch 7/30
89/89 [==============================] - 50s 567ms/step - loss: 0.2182 - accuracy: 0.9270 - val_loss: 0.1795 - val_accuracy: 0.
9271
```

Model Save

```
In [11]: model.save("fruit.h5")
```

Prediction:

```
In [47]: model = load_model("fruit.h5")
```

```
In [33]: img = image.load_img('E://IDS//Intenrship_IBM//Fertilizers//Dataset//fruit-dataset//fruit-dataset//test//Apple___healthy//00fca0c
```

```
In [48]: x = image.img_to_array(img)
         x = np.expand_dims(x, axis = 0)
```

```
In [49]: Pred = model.predict(x)
```

```
In [50]: print("The Predicted Value of Healthy Apple is:", np.argmax(Pred))

         The Predicted Value of Healthy Apple is: 1
```

```
In [56]: img = image.load_img('E://IDS//Intenrship_IBM//Fertilizers//Dataset//fruit-dataset//fruit-dataset//test//Peach___Bacterial_spot//
         x = image.img_to_array(img)
         x = np.expand_dims(x, axis = 0)
         Pred = model.predict(x)
         print("The Predicted Value of Peach___Bacterial_spot is:", np.argmax(Pred))

         The Predicted Value of Peach___Bacterial_spot is: 4
```