

MODEL BUILDING FOR FRUIT DISEASE PREDICTION

Team ID	PNT2022TMID51209
Project Name	Project - Fertilizers Recommendation System For Disease Prediction

```
jupyter Model Building for Fruit disease prediction Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [23]: #Image Augmentation

In [24]: 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)

In [25]: x_train=train_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\fruit-dataset\fruit-dataset\train',target_size=(180,180))
x_test=test_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\fruit-dataset\fruit-dataset\test',target_size=(180,180))

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

In [26]: #Import required Libraries

In [27]: 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 [28]: #Initializing Sequential model

In [29]: model=Sequential()
```

```
jupyter Model Building for Fruit disease prediction Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [30]: #Adding Layers

In [31]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))# convolution Layer

In [32]: model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling Layer

In [33]: model.add(Flatten())# Flatten Layer

In [34]: model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))# Hidden Layer 1
model.add(Dense(units=70,kernel_initializer='random_uniform',activation='relu'))# Hidden Layer 2
model.add(Dense(units=6,kernel_initializer='random_uniform',activation='softmax')) # Output Layer

In [35]: # Compiling the model

In [36]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

In [37]: #Train the model

In [38]: model.fit(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=52)

Epoch 1/3
168/168 [=====] - 51s 266ms/step - loss: 1.5775 - accuracy: 0.3601 - val_loss: 133.8710 - val_accuracy: 0.4038
Epoch 2/3
168/168 [=====] - 57s 342ms/step - loss: 1.1549 - accuracy: 0.5625 - val_loss: 93.9611 - val_accuracy: 0.6442
Epoch 3/3
168/168 [=====] - 42s 249ms/step - loss: 0.9021 - accuracy: 0.6280 - val_loss: 48.2599 - val_accuracy: 0.7308
```



Out[38]: <keras.callbacks.History at 0x1d85a6c3790>

In [39]: `#Save the model`

In [40]: `model.save(r'C:\Users\Elcot\project\flask\uploads\fruit.h5')`

In [41]: `model.summary()`

```
Model: "sequential_2"
-----
Layer (type)                 Output Shape              Param #
-----
conv2d_5 (Conv2D)            (None, 126, 126, 32)      896
max_pooling2d_2 (MaxPooling (None, 63, 63, 32)        0
2D)
flatten_2 (Flatten)          (None, 127008)            0
dense_6 (Dense)               (None, 40)                5080360
dense_7 (Dense)               (None, 70)                2870
dense_8 (Dense)               (None, 6)                 426
-----
Total params: 5,084,552
Trainable params: 5,084,552
Non-trainable params: 0
-----
```