Train and save the Model

Date	11 November 2022
Team ID	PNT2022TMID18696
Project Name	Fertilizers Recommendation System For Disease Prediction

```
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 [2]:

```
x_train=train_datagen.flow_from_directory(r'C:\Users\akash\project\Dataset
Plant Disease\Veg-dataset\Veg-
dataset\train_set',target_size=(128,128),batch_size=2,class_mode='categori
cal')
x_test=test_datagen.flow_from_directory(r'C:\Users\akash\project\Dataset
Plant Disease\Veg-dataset\Veg-
dataset\test_set',target_size=(128,128),batch_size=2,class_mode='categoric
al')
```

```
Found 11386 images belonging to 9 classes.
Found 3416 images belonging to 9 classes.
```

In [3]:

```
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 [4]:

```
model=Sequential()
```

In [5]:

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

In [6]:

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

In [7]:

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

In [8]:

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

In [9]:

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

In [10]:

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

In [11]:

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

In [12]:

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

In [13]:

```
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,vali
dation_steps=27)
```

```
Epoch 1/20
89/89 [==============================] - 95s 1s/step - loss: 2.1765 - accu
racy: 0.1404 - val_loss: 107.0669 - val_accuracy: 0.2407
Epoch 2/20
89/89 [==============================] - 61s 679ms/step - loss: 2.1010 - a
ccuracy: 0.2303 - val_loss: 73.7251 - val_accuracy: 0.0741
Epoch 3/20
89/89 [==============================] - 67s 755ms/step - loss: 2.1514 - a
ccuracy: 0.1348 - val_loss: 56.0996 - val_accuracy: 0.1111
Epoch 4/20
89/89 [==============================] - 64s 717ms/step - loss: 2.0868 - a
ccuracy: 0.1573 - val_loss: 23.7097 - val_accuracy: 0.3148
Epoch 5/20
89/89 [==============================] - 60s 671ms/step - loss: 2.0239 - a
ccuracy: 0.3090 - val_loss: 99.1493 - val_accuracy: 0.2222
Epoch 6/20
89/89 [==============================] - 72s 807ms/step - loss: 1.9236 - a
ccuracy: 0.2753 - val_loss: 172.7210 - val_accuracy: 0.1296
Epoch 7/20
89/89 [==============================] - 86s 971ms/step - loss: 1.9143 - a
ccuracy: 0.2753 - val_loss: 107.2718 - val_accuracy: 0.2778
Epoch 8/20
89/89 [==============================] - 58s 646ms/step - loss: 1.7796 - a
ccuracy: 0.3034 - val_loss: 64.2221 - val_accuracy: 0.3148
Epoch 9/20
89/89 [==============================] - 60s 676ms/step - loss: 1.7756 - a
ccuracy: 0.3427 - val_loss: 182.9076 - val_accuracy: 0.3519
Epoch 10/20
89/89 [==============================] - 59s 665ms/step - loss: 1.8444 - a
ccuracy: 0.2978 - val_loss: 138.7072 - val_accuracy: 0.2407
Epoch 11/20
89/89 [==============================] - 53s 598ms/step - loss: 1.7811 - a
ccuracy: 0.2640 - val_loss: 111.3470 - val_accuracy: 0.3333
Epoch 12/20
89/89 [==============================] - 57s 637ms/step - loss: 1.8700 - a
ccuracy: 0.2809 - val_loss: 104.8549 - val_accuracy: 0.2778
Epoch 13/20
89/89 [==============================] - 53s 593ms/step - loss: 1.8179 - a
ccuracy: 0.3371 - val_loss: 88.9790 - val_accuracy: 0.3519
Epoch 14/20
89/89 [==============================] - 53s 590ms/step - loss: 1.7108 - a
ccuracy: 0.2921 - val_loss: 79.7810 - val_accuracy: 0.4074
Epoch 15/20
89/89 [==============================] - 57s 644ms/step - loss: 1.8212 - a
ccuracy: 0.2416 - val_loss: 187.6725 - val_accuracy: 0.2222
Epoch 16/20
89/89 [==============================] - 53s 589ms/step - loss: 1.7251 - a
ccuracy: 0.3483 - val_loss: 148.7835 - val_accuracy: 0.4259
Epoch 17/20
```

```
89/89 [==============================] - 56s 624ms/step - loss: 1.6795 - a
ccuracy: 0.3146 - val_loss: 109.7393 - val_accuracy: 0.2593
Epoch 18/20
89/89 [==============================] - 54s 608ms/step - loss: 1.8275 - a
ccuracy: 0.3202 - val_loss: 90.4495 - val_accuracy: 0.3148
Epoch 19/20
89/89 [==============================] - 53s 591ms/step - loss: 1.7737 - a
ccuracy: 0.3427 - val_loss: 141.6376 - val_accuracy: 0.2593
Epoch 20/20
89/89 [==============================] - 53s 592ms/step - loss: 1.7060 - a
ccuracy: 0.4213 - val_loss: 96.3340 - val_accuracy: 0.3889
```

Out[13]:

In [14]:

```
model.save(r'C:\Users\akash\project\flask\uploads\vegetable.h5')
```

In [15]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 300)               38102700

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 75)                11325

 dense_3 (Dense)             (None, 9)                 684

=================================================================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________
```