```
# -*- coding: utf-8 -*-

"""Copy of Test the Veg model.ipynb

Automatically generated by Colaboratory.

Original file is located at

    https://colab.research.google.com/drive/1RHpmLZRIo1sq5mAhS8EUL_PAcVbNWolZ

"""


!unzip '/content/drive/MyDrive/ibm dataset/Fertilizers_Recommendation_ System_For_Disease_ Prediction.zip'


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)


x_train=train_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch_size=2,class_mode='categorical')

x_test=test_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_size=2,class_mode='categorical')


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


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)


x_train=train_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch_size=16,class_mode='categorical')
```

```
x_test=test_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_size=16,class_mode='categorical')


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=300,kernel_initializer='uniform',activation='relu'))

model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)

model.save('fruit.h5')

model.summary()

from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as nps

model=load_model('fruit.h5')

img=image.load_img('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/011d02f3-5c3c-4484-a384-b1a0a0dbdec1___RS_HL 7544.JPG',grayscale=False,target_size=(128,128))

img
```

```
x=image.img_to_array(img)
x=nps.expand_dims(x,axis=0)

pred=(model.predict(x) > 0.5).astype("int32")

pred

import requests
from tensorflow.keras.preprocessing import image

from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request ,  render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app= Flask(__name__)
model = load_model("fruit.h5")
@app.route('/')
def home():
  return render_template('home.html')

@app.route('/prediction')
def prediction():
  return render_template('predict.html')

@app.route('/predict',methods=['POST'])
```

```
def predict():
  if request.method=='POST':
    f= request.files['images']
    basepath=os.path.dirname(__file__)
    file_path==os.path.join(
       basepath, 'uploads',secure_filename(f.filename))
    f.save(file_path)
    img=image.load_img(file_path, target_size=(128,128))
    x=image.img_to_array(img)
    x=np.expand_dims(x, axis=0)
    plant=request.form['plant']
    print(plant)
    if(plant=="fruit"):
      preds=model.predict_classess(x)
      print(preds)
      df=pd.read_excel('precautions-veg.xlsx')
      print (df.iloc[preds[0]]['cautions'])
    else:
      pred=model1.predict_classes(x)
      df=pd.read_excel('precautions-fruits.xlsx')
      print(df.iloc[preds[0]]['caution'])
      return df.iloc[preds[0]]['caution']


if __name__=="__main__":
  app.run(debug=False)
```