Fertilizer Recommendation Model for Fruits and Vegetables

Dr Maheswari S
Vellore Institute of Technology
Chennai Campus

Project Objectives

- Preprocess the images.
- Applying the CNN algorithm to the dataset.
- How deep neural networks detect the disease.
- You will be able to know how to find the accuracy of the model.
- You will be able to build web applications using the Flask framework.

Data set used for this project
Fruits

Vegetables

```
In [55]: x_train.class_indices

Out[55]: {'Pepper,_bell___Bacterial_spot': 0,
          'Pepper,_bell___healthy': 1,
          'Potato___Early_blight': 2,
          'Potato___Late_blight': 3,
          'Potato___healthy': 4,
          'Tomato___Bacterial_spot': 5,
          'Tomato___Late_blight': 6,
          'Tomato___Leaf_Mold': 7,
          'Tomato___Septoria_leaf_spot': 8}
```

Preprocess the images.
Under Image Augmentation, the processes are train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False).
Image resized to 64*64*3 as input size for the image processing. The same is used in

the API application.

Deep Neural Network Prediction

Applied CNN algorithm to predict the diseases of the fruits and vegetable plants for the given dataset. Used Convolution2D for feature selection, relu and softmax as activation function. Also used two hidden layers of 300 and 150 respectively for capturing high level details of the images. Finally six and nine output layers are used for fruits and vegetables respectively. With prediction function the model able to predict with more than 90 percent accuracy.

```
Epoch 9/10
225/225 [==============================] - 38s 169ms/step - loss: 0.0777 - accuracy: 0.9736 - val_loss: 0.1289 - val_accuracy:
0.9573
Epoch 10/10
225/225 [==============================] - 37s 164ms/step - loss: 0.0838 - accuracy: 0.9710 - val_loss: 0.1417 - val_accuracy:
0.9508
```

Out click to expand output; double click to hide output 18ae56bf7c0>

```
In [37]: img
```

Out[37]:

```
In [38]: img=image.load_img(r"C:\Users\Admin\Desktop\fig1.jpg",target_size=(64,64))

In [39]: img
```

Out[39]:

```
In [40]: x=image.img_to_array(img)
```

Fertilizer Recommendation Model for Fruits and Vegetables

```
In [49]: index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp

In [61]: index[y[0]]

Out[61]: 'Corn_(maize)___healthy'

In [65]: img=image.load_img(r"C:\Users\Admin\Desktop\fig4.jpg",target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         y=np.argmax(model.predict(x),axis=1)
         index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp
         index[y[0]]

         1/1 [==============================] - 0s 23ms/step

Out[65]: 'Corn_(maize)___healthy'

In [57]: img=image.load_img(r"C:\Users\Admin\Desktop\fig2.jpg",target_size=(64,64))
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         y=np.argmax(model.predict(x),axis=1)
         index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp
         index[y[0]]
```

Program for Application Development (Flask)

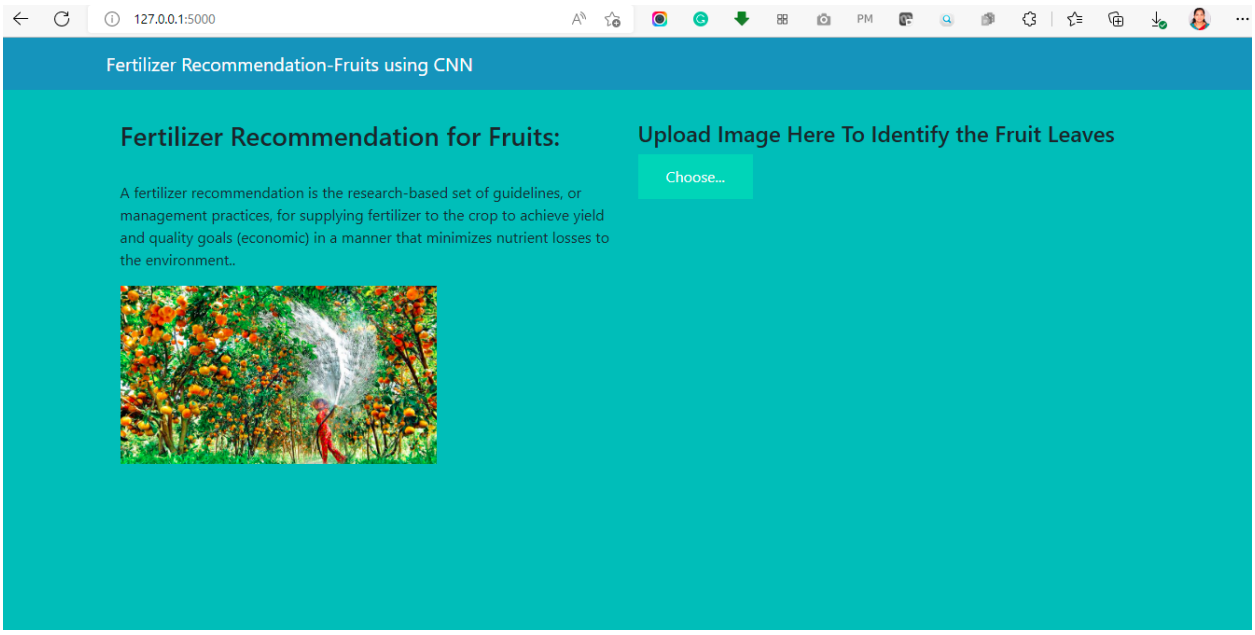
```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model

import numpy as np
import pandas as pd
import os
import tensorflow as tf

from flask import Flask, request, render_template, redirect, url_for
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)

model = load_model("Fertilizer_Fruits.h5")


#render home page
@app.route('/')
def index():
    return render_template('index.html')


#render predict page
```

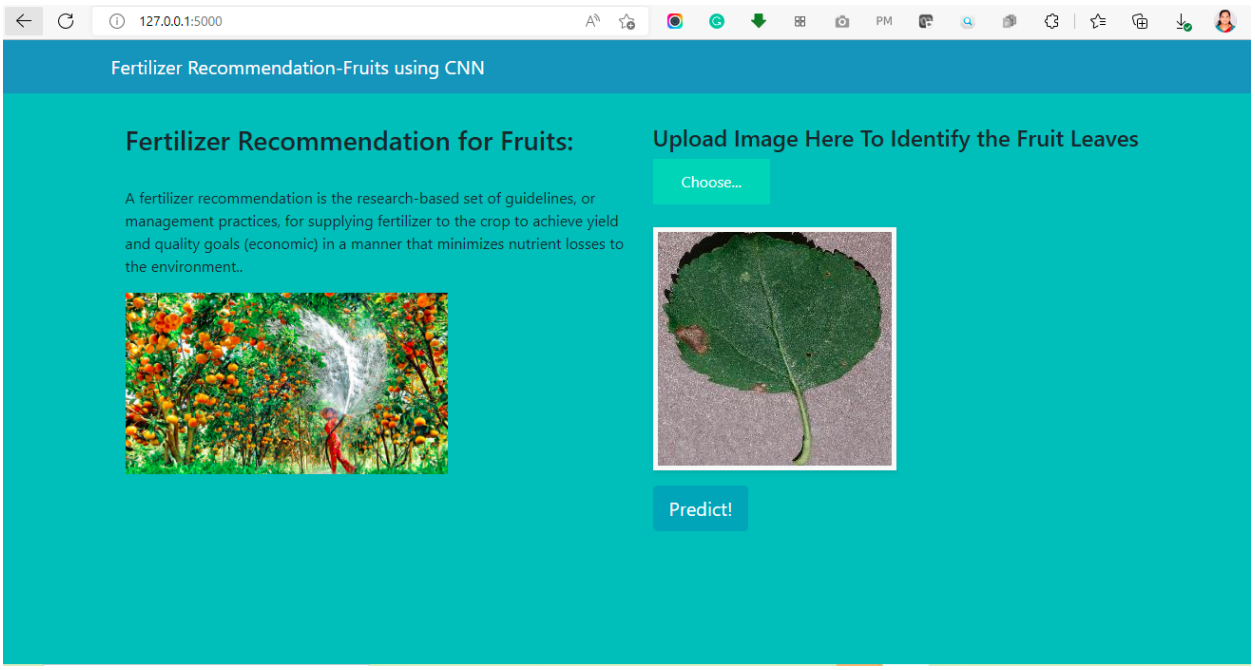
Fertilizer Recommendation Model for Fruits and Vegetables

```
@app.route('/predict',methods=['GET','POST'])
def upload():
    if request.method == 'POST':
        f=request.files['image']
        basepath = os.path.dirname(__file__)
        filepath = os.path.join(basepath,'uploads',f.filename)
        f.save

        img = image.load_img(filepath,target_size = (64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x),axis=1)
        index=['Pepper,_bell___Bacterial_spot',
               'Pepper,_bell___healthy',
               'Potato___Early_blight',
               'Potato___Late_blight',
               'Potato___healthy',
               'Tomato___Bacterial_spot',
               'Tomato___Late_blight',
               'Tomato___Leaf_Mold',
               'Tomato___Septoria_leaf_spot']

        text = 'The classied fruit is:'+str(index[pred[0]])
    return text


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

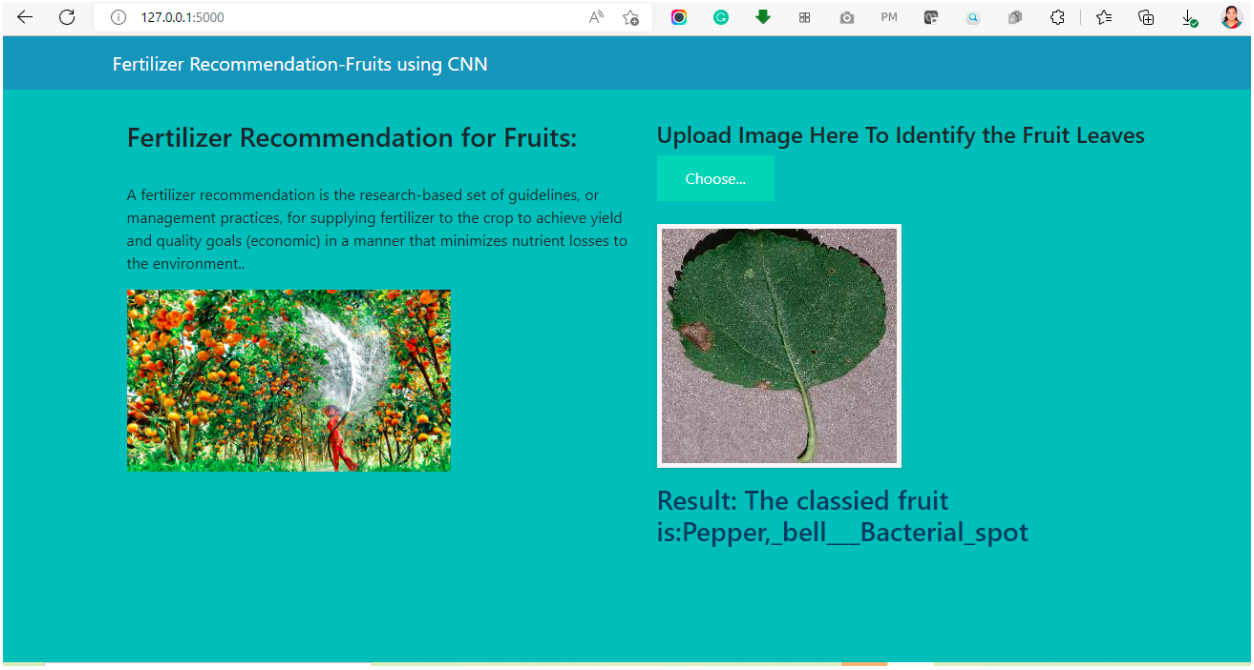
Output

Fertilizer Recommendation Model for Fruits and Vegetables





Fertilizer Recommendation Model for Fruits and Vegetables



Conclusion

Thus the project has been completed by fulfilling the given objectives with more than 90 percent accuracy.