

Project Development Phase

Sprint – 4

Date	16 November 2022
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiast
Team Id	PNT2022TMID21516

CREATE A HTML PAGE:

Home.html

Nutrion Image Analysis

HomeClassify

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food. It ensures compliance with trade and food laws.

Team details
Team Leader: Malathy G (917719IT053)
Team Member1: Prijitha S (917719IT069)
Team Member2: Srivishali K (917719IT104)
Team Member3: Sujatha K (917719IT106)

Image.html

Nutrion Image Analysis

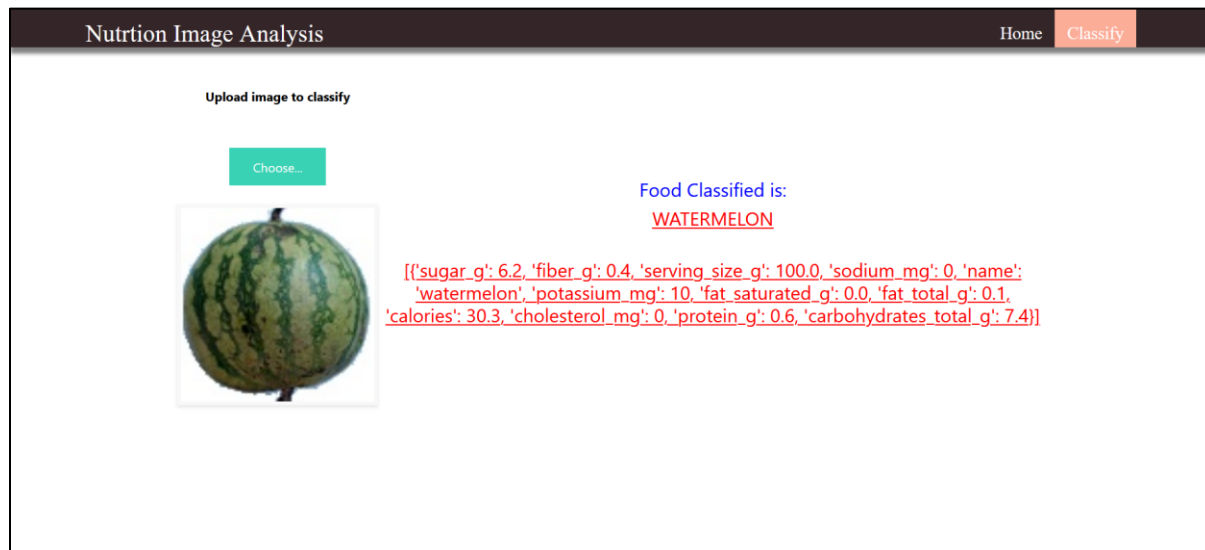
HomeClassify

Upload image to classify

Choose...

Classify

Imageprediction.html



BUILD PYTHON CODE:

Import libraries

```
from flask import Flask, render_template, request
# Flask-It is our framework which we are going to use to run/serve our application.
# request-for accessing file which was uploaded by the user on our application.
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model #to load our trained model
from tensorflow.keras.preprocessing import image
import requests
```

```
C:\Users\sujatha.k>pip install flask
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: flask in c:\users\suiatha.k\appdata\roaming\pyt
```

```
C:\Users\sujatha.k>pip install tensorflow
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: tensorflow in c:\users\sujatha.k\appdata\roaming
```

```
C:\Users\sujatha.k>pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\sujatha.k\appdata\roaming\python\python39\site-packages (1.23.4)
WARNING: You are using pip version 21.2.3; however, version 22.3.1 is available.
You should consider upgrading via the 'C:\Program Files\Python39\python.exe -m pip install --upgrade pip' command.
```

CREATING OUR FLASK APPLICATION AND LOADING OUR MODEL BY USING LOAD_MODEL METHOD:

```
app = Flask(__name__,template_folder="templates")
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")
```

ROUTING TO THE HTML PAGE:

Here, the declared constructor is used to route to the HTML page created earlier.

In the above example, the '/' URL is bound with the home.html function. Hence, when the home page of the webserver is opened in the browser, the HTML page is rendered.

Whenever you enter the values from the HTML page the values can be retrieved using the POST Method.Here, "home.html" is rendered when the home button is clicked on the UI

```
@ app.route('/')# route to display the home page
def home():
    return render_template('home.html') #rendering the home page

@ app.route('/image1', methods=['GET', 'POST']) # routes to the index html
def image1():
    return render_template("image.html")
```

When "image is uploaded "on the UI, the launch function is executed

```
@ app.route('/predict' ,methods=['GET','POST']) # route to show the pr
def lanuch():
```

It will take the image request and we will be storing that image in our local system then we will convert the image into our required size and finally, we will be predicting the results with the help of our model which we trained and depending upon the class identified we will showcase the class name and its properties by rendering the respective html pages.

```

@app.route('/predict',methods=['GET','POST']) # route to show the pr
def lanuch():
    if request.method=='POST':
        f=request.files['file'] # requesting the file
        basepath=os.path.dirname('__file__') #storing the file directo
        filepath=os.path.join(basepath,"uploads",f.filename) #storing
        f.save(filepath) #saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and res
        x=image.img_to_array(img) #converting image to an array
        x=np.expand_dims(x,axis=0) #changing the dimensions of the ima

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred) #printing the prediction
        index=['APPLE','BANANA','ORANGE','PINEAPPLE','WATERMELON']

        result=str(index[pred[0]])
        print(result)
        x=result
        result=nutrition(result)
        print(result)

```

API Integration:

Here we will be using Rapid API. Using RapidAPI, developers can search and test the APIs, subscribe, and connect to the APIs — all with a single account, single API key and single SDK. Engineering teams also use RapidAPI to share internal APIs and microservice documentation.

API used:

The link above will allow us to test the food item and will result the nutrition content present in the food item.

NOTE: When we keep hitting the API the limit of it might expire. So, making a smart use of it will be an efficient way.

```

def nutrition(index):

    import requests

    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query":index}

    headers = {
        "X-RapidAPI-Key": "85887549f4msh51e7315b280a87ep1f43e0jsn585c940f2ea6",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
    }

    response = requests.request("GET", url, headers=headers, params=querystring)

    print(response.text)
    return response.json()['items']
if __name__ == "__main__":
    # running the app
    app.run(debug=False)

```

Finally, Run the application. This is used to run the application in a localhost. The local host runs on port number 5000.

RUN THE APPLICATION:

Open the anaconda prompt from the start menu.

Navigate to the folder where your app.py resides.

Now type the “python app.py” command.

It will show the local host where your app is running on <http://127.0.0.1:5000/>

Copy that localhost URL and open that URL in the browser. It does navigate to where you can view your web page.

Enter the values, click on the predict button and see the result/prediction on the web page.

```

C:\Users\sujatha.k>cd Downloads\Srivishali application Building
C:\Users\sujatha.k\Downloads\Srivishali application Building>python app.py

```

Then it will run on localhost:5000

```

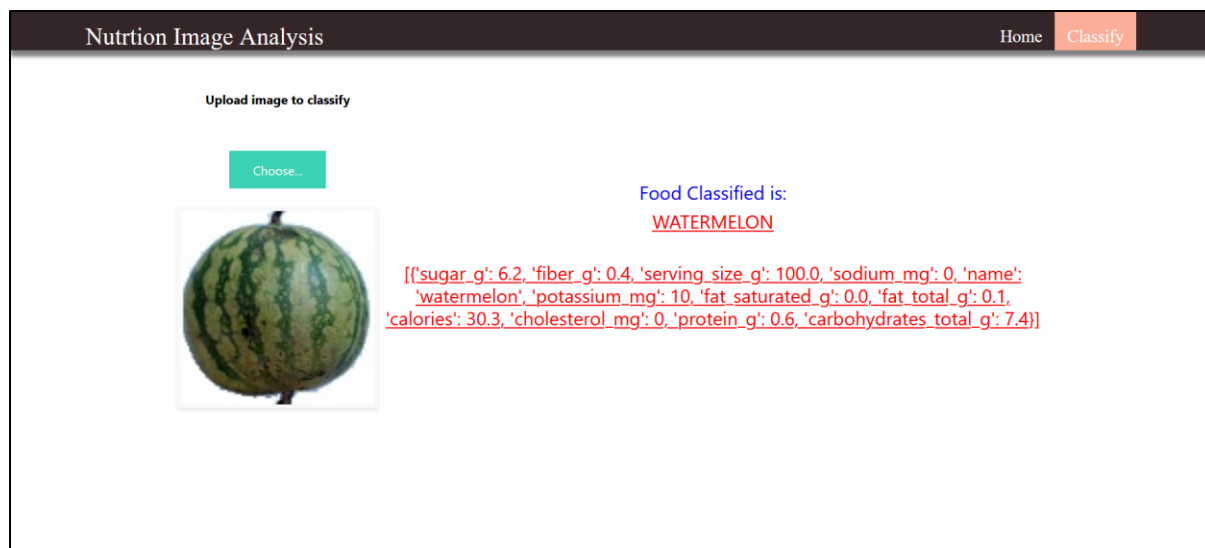
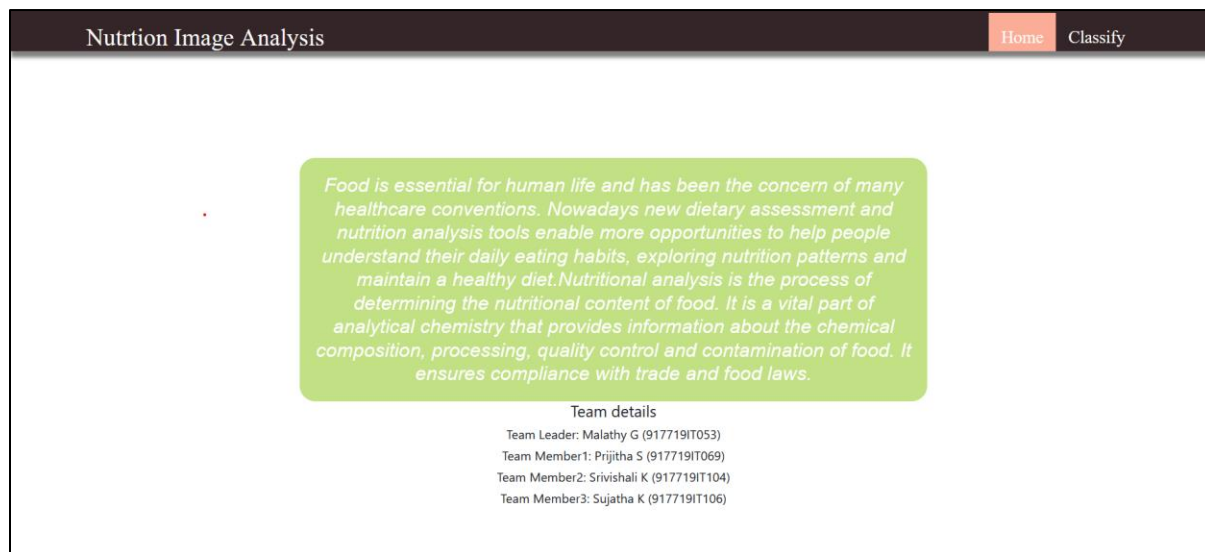
2022-11-18 17:39:19.926104: I tensorflow/core/platform/cp
h oneAPI Deep Neural Network Library (oneDNN) to use the
AVX AVX2
To enable them in other operations, rebuild TensorFlow wi
Loaded model from disk
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a
* Running on http://127.0.0.1:5000

```

Navigate to the localhost (<http://127.0.0.1:5000/>) where you can view your web page.

Click on classify button to see the results.

Output screenshots:




Nutrtn Image Analysis

HomeClassify

Upload image to classify

Choose...



Food Classified is:

PINEAPPLE


[[{"sugar_g": 9.9, "fiber_g": 1.4, "serving_size_g": 100.0, "sodium_mg": 0, "name": "pineapple", "potassium_mg": 8, "fat_saturated_g": 0.0, "fat_total_g": 0.1, "calories": 50.8, "cholesterol_mg": 0, "protein_g": 0.5, "carbohydrates_total_g": 13.0}]]

Nutrtn Image Analysis

HomeClassify

Upload image to classify

Choose...



Food Classified is:

BANANA


[[{"sugar_g": 12.3, "fiber_g": 2.6, "serving_size_g": 100.0, "sodium_mg": 1, "name": "banana", "potassium_mg": 22, "fat_saturated_g": 0.1, "fat_total_g": 0.3, "calories": 89.4, "cholesterol_mg": 0, "protein_g": 1.1, "carbohydrates_total_g": 23.2}]]

Nutrtn Image Analysis

HomeClassify

Upload image to classify

Choose...



Food Classified is:

APPLE

[[{"sugar_g": 10.3, "fiber_g": 2.4, "serving_size_g": 100.0, "sodium_mg": 1, "name": "apple", "potassium_mg": 11, "fat_saturated_g": 0.0, "fat_total_g": 0.2, "calories": 53.0, "cholesterol_mg": 0, "protein_g": 0.3, "carbohydrates_total_g": 14.1}]]

Fitness tip:

The user is aware of the nutritional value of the food they consume, according to our project study. Users are given access to a nutrition chart based on the food's nutritional composition. Thus, the user will be in good physical and mental health.

Feedback:

Feedback and reviews are collected from the users so the future users will refer it and make use of our product.