AI-Powered Nutrition Analyser for Fitness Enthusiasts

1. INTRODUCTION

1.1 Overview: Artificial intelligence (AI) is a rapidly evolving area that offers unparalleled opportunities of progress and applications in many healthcare fields. In this review, we provide an overview of the main and latest applications of AI in nutrition research and identify gaps to address to potentialize this emerging field. AI algorithms may help better understand and predict the complex and non-linear interactions between nutrition-related data and health outcomes, particularly when large amounts of data need to be structured and integrated, such as in metabolomics. AI-based approaches, including image recognition, may also improve dietary assessment by maximizing efficiency and addressing systematic and random errors associated with self-reported measurements of dietary intakes.

1.2 Purpose: The main purpose of the project is to build a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).It classify an object with higher degree of accuracy by fine tuning the parameters of the network. The main motto is to reduce the training time and compute complexity of the network by adding a sub layer after each convolution layer.

2. LITERATURE SURVEY

2.1 Existing problem /approaches

A number of studies have been conducted on image categorization. Veggie-Vision was an initial attempt to develop a produce recognition system for use in supermarkets. The system could analyze color, texture and density, and thus was able to obtain more information. Density was calculated by dividing weight with the area of the fruit. The reported accuracy was approximately 95% when color and texture features were combined.

Faria et al. presented a framework for classifier fusion for the automatic recognition of produce in supermarkets. They combined low-cost classifiers trained for specific classes of interest to enhance the recognition rate. Chowdhury et al. recognized 10 different vegetables using color histogram and statistical texture features. They obtained a classification accuracy of up to 96.55% using neural network as a classifier.

Dubey proposed a framework for recognizing and classifying images of 15 different types produce. The approach involves segmenting an image to extract the region of interest, and then calculating the features from that segmented region, which is further used in training and classification by a multi-class support vector machine. Moreover, they proposed an improved sum and difference histogram (ISADH) texture feature for this kind of problem.

Fruit detection greatly affects the robot's harvesting efficiency because it is an unstructured environment with changing lighting conditions. Bulanon et al. enhanced the portion occupied by fruit in images using a red chromaticity coefficient and adopted a circle detection method for classifying individual fruits.

Jimenez et al. developed a method that can identify spherical fruits in the natural environment in which difficult situations are present: occlusions, shadows, bright areas, and overlapping fruits. Range and attenuation data are sensed by a laser range-finder sensor, and the 3-D position of the fruit with radius and reflectance are obtained after the recognition steps.

2.2 Proposed solution:
In this project, CNN algorithm helps to extract the important features from the images and train according to them.
These neural networks have proven to be successful in many different real-life case studies and applications like

1. Image classification, object detection, segmentation, face recognition.
2. Self driving cars that leverage CNN based vision systems.
3. Classification of crystal structure using a convolutional neural network.

This model involves 4 steps:

1. Convolution
2. Max pooling
3. Flattening
4. Full connection

And finally import the image using these packages to find the predictions:
from keras.models import load_model
from keras.preprocessing import image

The extension for saving the model is .h5 file.

1. **Data Collection**

I. Download the dataset
II. Load the dataset
III. In this project, we have used 5 datasets namely:

1. Apple
2. Banana
3. Pineapple
4. Orange

5. Watermelon

In training dataset ,we notice that 2626 images are belonging to 5 classes for training and 1055 images belong to 5 classes for testing purpose.

1. **Data Modelling**

Convolution Neural Network (CNN) algorithm is applied to this model .This helps us to extract the important features from the images and train them.

It has four steps namely:

1. Convolution
2. Max pooling
3. Flattening
4. Full connection

1.In convolution operation, the input image is convoid with feature detector or filter to get a feature map. By applying convolution operation, the size of image is reduced we may lose some information but feature detector or filter will helps us to extract main features from image and remove unwanted features.

2.Max pooling is a technique which helps us to avoid over-fitting of data and helps us to avoid special ingredients and distortions in data.

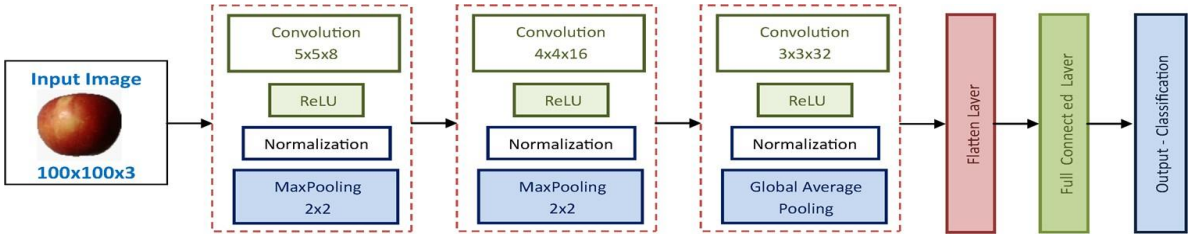
3.Flattening layer converts multi dimension pooled feature map to single dimension pooled feature map. Flattening layer is the input layer.

Start initializing the parameters and finally train the model with Keras fit() function. The model trains for 20 epochs.
Testing is similar to training, except that we don't need to compute gradients and training targets. Instead, we take the predictions from network output, and combine them to get the real detection output.
Fit generator is used to find the training and validation accuracy.
Image data generator is a class in keras.preprocessing package to apply some image processing to the images.

In this project the accuracy error is also good, so that the model is perfect.

3. THEORITICAL ANALYSIS

3.1 Block diagram Diagrammatic overview of the project:



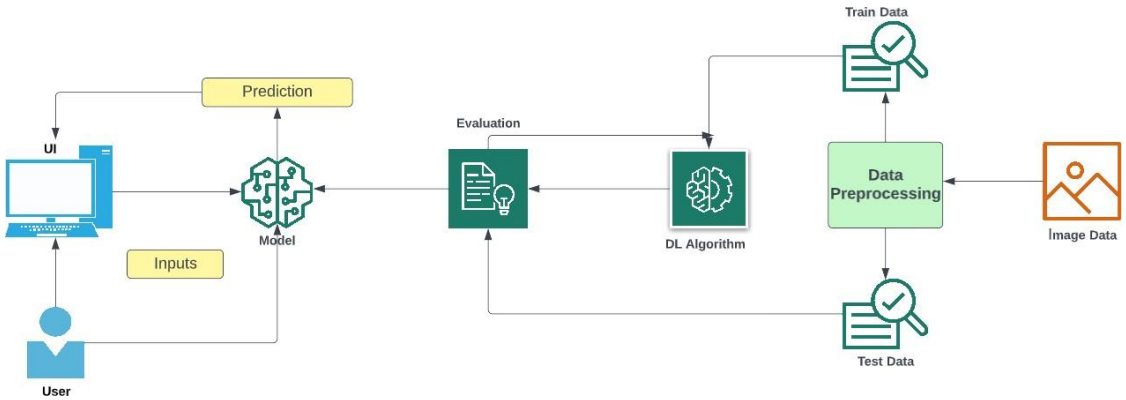
3.2 Hardware and software requirements of the project:

HARDWARE REQUIREMENTS	SOFTWARE REQUIREMENTS
• Laptop/PC with windows V.8 or above.	• Python Packages: numpy,pandas,sci-kit,tensorflow,keras.
• Backup Hard Drive/USB memory	• Jupyter Notebook/PyCharm
	• Anaconda Navigator
	• Flask

4. EXPERIMENTAL INVESTIGATIONS:

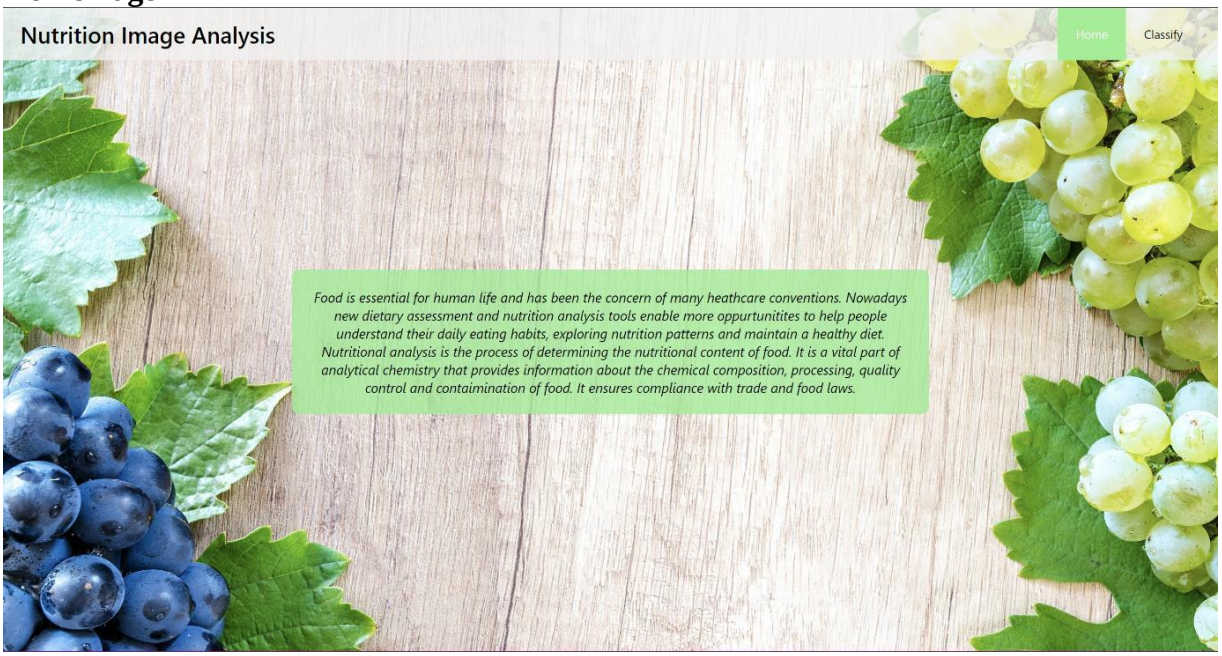
In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', 'pineapple' and 'watermelon', they are saved in the respective subdirectories with their respective names:85 % of the images from these are used for training, and 15 % are used for testing the model.
The network is trained for 20 epochs with a batch size of 32. The accuracy of the proposed model was 98.74 %.
The comparison of the proposed model with the conventional models shows that the results of this model are exceptionally good and promising to use in real-world applications. This sort of higher accuracy and precision will work to boost the machine's general efficiency in fruit recognition more appropriately.

5. FLOWCHART

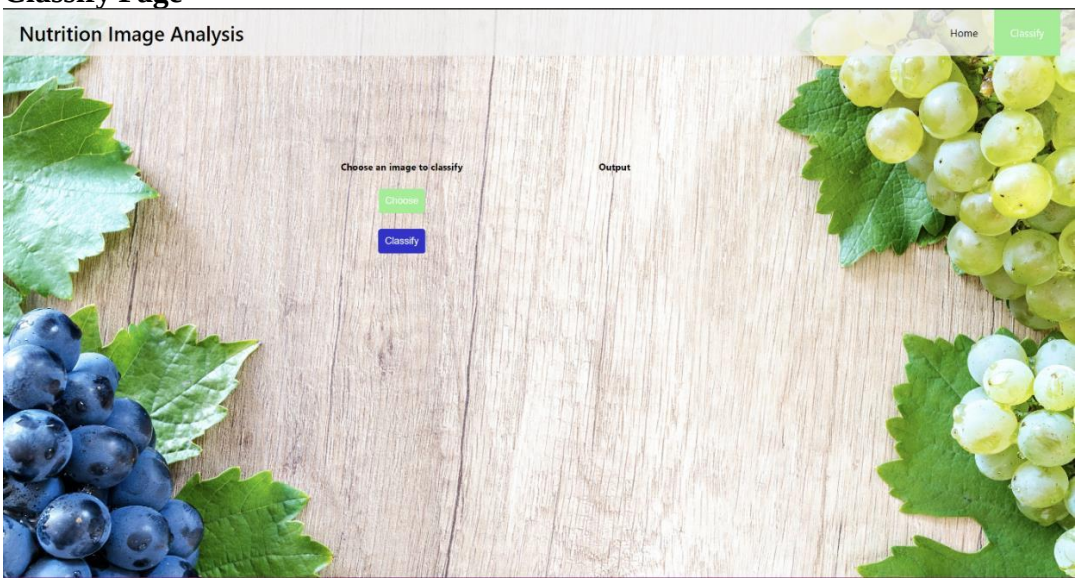


6. RESULT Final findings (Output) of the project along with screen-shots.

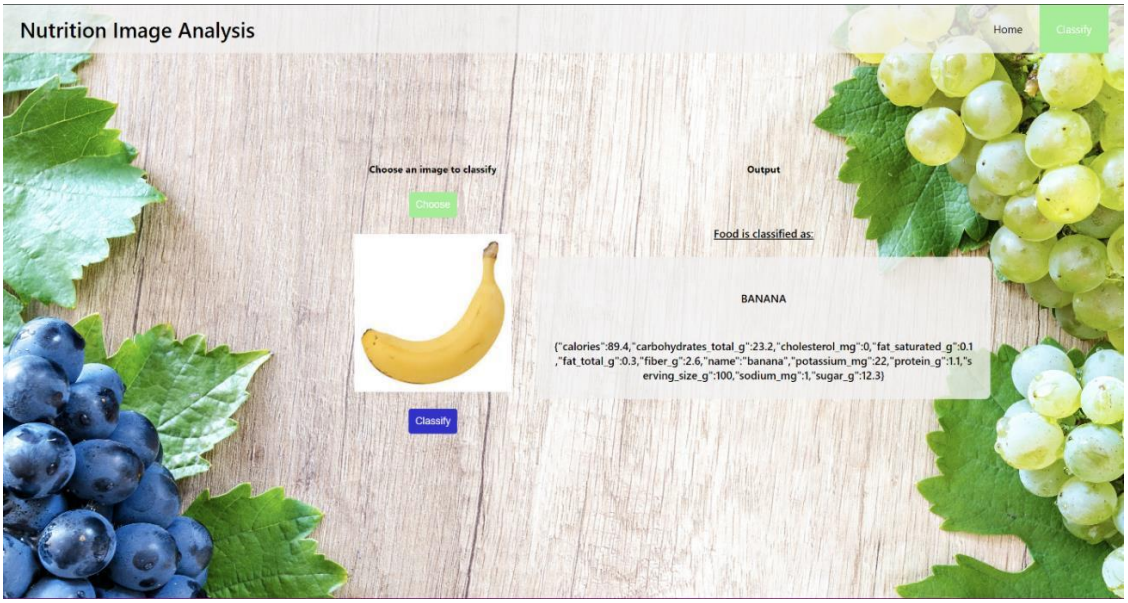
Home Page



Classify Page



Results



7. ADVANTAGES & DISADVANTAGES

Advantages:

1. Classification of fruits is a needful exercise to differentiate the particular variety of fruits of the same family. Most of the case, the variety of fruits of the same family differ in the sense of colour and size only.
2. The use of image processing for the grading of fruits involves categorization of fruits, with consideration of the severity of the disease, defects, and contamination on fruits. Grad-ing is an important step in the post-harvest process. Grading of fruits manually is a time taking and unreliable process. Therefore, it is needful to adapt the automated faster system in this regard.
3. Some of the other associated benefits include speed operation, production consistent, greater product stability and safety.

Disadvantages:

1. Most of the research conducted by taking the one-side view of fruits. In addition, by considering the one-side image of fruit, it is challenging to evaluate the quality fruits.
2. It does not provide stable recognition in adverse imaging condition.

8. APPLICATIONS:

Fruit classification is an important task in many industrial applications. A fruit classification system may be used to help a supermarket cashier identify the fruit species and prices. It may also be used to help people decide whether specific fruit species meet their dietary requirements. It may also help children and visually impaired people, and improve supermarket grocery self-checkouts.

Identified applications of fruit classification other than nutrition analyser are:

1. Food Industry- Quality Assessment
2. Agriculture-Robotic Harvesting
3. Retail-Supermarkets, Retail.

9. CONCLUSION:

During this project, we had the option to investigate some portion of the profound learning algorithms and find qualities and shortcomings. We picked up information on deep learning, and we got a product that can perceive fruits from pictures. A new method for classifying fruits using convolutional neural network algorithm is proposed. The above listed results were obtained using 7 test samples taken out from the actual number of 2626 and 1050 images used for training and testing.

The above algorithm was coded and tested using anaconda software. Different fruits varieties that had different backgrounds were taken for training and testing. The proposed algorithm gave 98% accuracy rate. This project explores a fruits classification based on CNN algorithm. The accuracy and loss curves were generated by using various combinations of hidden layers for five cases using fruits. CNN gave better performance to attain better fruit classification.

We trust that the outcomes and strategies introduced in this project can be additionally extended to a greater task. From our perspective, one of the principal goals is to improve the precision of the neural system. This includes further exploring different avenues regarding the structure of the system.

10. FUTURE SCOPE:

Hopefully, in the future, this project can be extended with a larger dataset having more categories of fruits & vegetables. We will also have the plan
to implement some other CNN based models to compare the accuracy on the same dataset, can also work on some more features for grading and classification, which can identify types of disease and/or texture structure of fruits. All these are future direction.

11. BIBILOGRAPHY :

1. https://www.youtube.com/embed/82GMw6espuQ

2. Mureșan, Horea & Oltean, Mihai. (2018). Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica. 10. 26-42. 10.2478/ausi-2018-0002.

3. https://github.com/garodisk/Fruits-classification-120-different-fruits-

4. https://www.electronicsforu.com/electronics-projects/electronics-design-guides/fruit-classification-quality-detection-using-deep-convolutional-neural-network

5. https://medium.com/ai-techsystems/fruits-classification-using-deep-learning-f8261b0ee0ca

6. Khatun, Mehenag & Nine, Julker & Ali, Md. Forhad & Sarker, Pritom & Turzo, Nakib. (2020). Fruits Classification using Convolutional Neural Network. 5. 1-6.

APPENDIX A.

In this section we present the source code and project structure used in this project.

1. **DATA COLLECTION, IMAGE PREPROCESSING AND MODEL BUILDING.**

```
 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./255)

# # Loading our data and performing data augmentation
#performing data augmentation to train the data
x_train=train_datagen.flow_from_directory(r'C:\Users\Asmi Bhardwaj\Downloads\AI-
Powered Nutrition Analyser for Fitness
Enthusiasts\Dataset\TRAIN_SET',target_size=(64,64),batch_size=5,color_mode='rgb',class_
mode='sparse')
#performing data augmentation to test the data
x_test=test_datagen.flow_from_directory(r'C:\Users\Asmi Bhardwaj\Downloads\AI-Powered
Nutrition Analyser for Fitness
Enthusiasts\Dataset\TEST_SET',target_size=(64,64),batch_size=5,color_mode='rgb',class_m
ode='sparse')


print(x_train.class_indices)
from collections import Counter as c
c(x_train.labels)
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.preprocessing.image import ImageDataGenerator
model=Sequential()

classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64,64,3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))

classifier.add(Conv2D(32, (3,3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))

classifier.add(Flatten())

classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5,activation='softmax'))

classifier.summary()
```

```
classifier.compile(optimizer='adam',loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
classifier.fit_generator(generator=x_train,steps_per_epoch =
len(x_train),epochs=20,validation_data=x_test,validation_steps = len(x_test))
classifier.save('nutrition.h5')


from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("nutrition.h5")
img = image.load_img(r"C:\Users\Asmi Bhardwaj\Downloads\AI-Powered Nutrition
Analyser for Fitness Enthusiasts\Dataset\TEST_SET\APPLES\5_100.jpg",
grayscale=False,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)
pred

index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']
result=str(index[pred[0]])
result
```

APPLICATION BUILDING:

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AI based analyzer for Fitness Enthusiasts </title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/styles.css')}}">
</head>
<body>
    <nav>
        <h2> <span>Nutrition</span> Image Analysis</h2>
        <ul>
            <li class="page-selected"><a href="/">Home</a></li>
            <li><a href="/image">Classify</a></li>
        </ul>
    </nav>
    <main id="home-main">
        <p>
            Food is essential for human life and has been the concern of many heathcare conventions. Nowadays new dietary assessment and
            nutrition analysis tools enable more oppurtunitites 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
            contaimination of food. It ensures compliance with trade and food laws.
        </p>
    </main>
</body>
</html>
```

image.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AI based analyzer for Fitness Enthusiasts </title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/styles.css')}}">
</head>
<body>
    <nav>
        <h2> <span>Nutrition</span> Image Analysis</h2>
        <ul>
            <li><a href="/">Home</a></li>
            <li class="page-selected"><a href="/image">Classify</a></li>
        </ul>
    </nav>
    <main id="classify-main">
        <div>
            <h3>Choose an image to classify</h3>
            <button class="primary-button">Choose</button>
            <img src="" alt="" id="image-viewer">
            <button class="secondary-button">Classify</button>
        </div>
        <div id="output">
             <h3>Output</h3>
             <p>Food is classified as:</p>
             <div id="output-wrapper">
                 <p id="output-result"></p>
                 <p id="output-api-result"></p>
             </div>
        </div>
    </main>
</body>
<script src="{{url_for('static', filename='js/app.js')}}"></script>
</html>
```

styles.css

```
}
nav > ul > li > a{
    color: black;
    transition: 0.25s;
}

/* Home Page */
#home-main{
    background: var(--primary-color-opacity);
    margin: 23rem auto;
    width: 70rem;
    padding: 2rem;
    font-size: 1.5rem;
    border-radius: 15px;
    text-align: center;
    font-style: italic;
}

/* Classify Page */
#classify-main{
    background: none;
    width: 50rem;
    margin: 13rem auto;
    display: grid;
    grid-template-columns: 1fr 2fr;
    grid-gap: 3rem;
}
#classify-main h3{
    margin-bottom: 2rem;
}
#classify-main .secondary-button{
    margin-top: 2rem;
}
#classify-main > div, #output{
    display: flex;
    flex-direction: column;
    align-items: center;
}
#output h3{
    margin-bottom: 4rem;
}
#output > p{
    text-decoration: underline;
    display: none;
}
#classify-main > div p{
    text-align: center;
    overflow-wrap: break-word;
    max-width: 50rem;
    font-size: 1.3rem;
    font-weight: 500;
    margin-top: 2rem;
}
#output-wrapper{
    background: rgba(243, 243, 243, 0.7);
    margin-top: 2rem;
    padding: 2rem;
    border-radius: 10px;
    display: none;
}
#output-wrapper > p:first-child{
    margin-top: none;
}
#output-wrapper > p:last-child{
    margin-top: 4rem;
```

app.js

```
const   chooseButton = document.querySelector('button.primary-button'),
        classifyButton = document.querySelector('button.secondary-button');
let     userFile;

// Event Listeners
chooseButton.addEventListener('click', (e)=>{
    // Creating an input element to select the file
    const input = document.createElement('input');
    input.setAttribute('type', 'file');
    input.setAttribute('accept', 'image/png, image/jpeg, image/jpg');
    input.setAttribute('name', 'file');
    input.click();
    input.onchange = function(){
        const imageViewer = document.querySelector('#image-viewer');

        // Displaying Image selected on the web page
        const reader = new FileReader();
        reader.onload = function(event){
            imageViewer.src = event.target.result;
            imageViewer.style.marginTop = '2rem';
            imageViewer.style.height = '300px';
            imageViewer.style.width = '300px';
        }
        reader.readAsDataURL(input.files[0]);
        userFile = input.files[0];
    }
})

classifyButton.addEventListener('click', (e)=> {
    const formData = new FormData();
    formData.append('file', userFile);
    fetch('/predict', {
        method: 'POST',
        body: formData
    })
    .then((response)=> response.json())
    .then((res)=> {
        const   result = document.querySelector('#output-result'),
                apiResult = document.querySelector('#output-api-result'),
                outputWrapper = document.querySelector('#output-wrapper'),
                p = document.querySelector('#output > p');


        console.log(res.apiResult[0])

        result.innerText = res.result;
        apiResult.innerHTML = `${JSON.stringify(res.apiResult[0])}`;
        p.style.display = 'block';
        outputWrapper.style.display = 'block';
    })
})
```

app.py

```python
from flask import Flask,render_template,request, jsonify

import os
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import requests

app = Flask(__name__,template_folder="../templates",
            static_folder='../static')

model = load_model('nutrition.hdf5.h5')
print("Loaded model from disk")

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

@app.route('/image')
def image1():
    return render_template("image.html")

@app.route('/imageprediction')
def imageprediction():
    return render_template("imageprediction.html")


@app.route('/predict',methods=['POST'])
def launch():
    if request.method=='POST':
        f=request.files['file']

        basepath=os.path.dirname('/')
        filepath=os.path.join(basepath, f.filename)
        f.save(filepath)

        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)
        print("prediction",pred)
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']

        result=str(index[pred[0]])
        apiResult=nutrition(result)

        final_result = {
                "result" : result,
                "apiResult" : apiResult
            }
        print(final_result)
        return final_result

def nutrition(index):

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

    querystring = {"query":index}

    headers = {  'X-RapidAPI-Host': 'calorieninjas.p.rapidapi.com',
    'X-RapidAPI-Key': '8c43e02098mshcb4fea7ab8fdea2p175878jsn0d0669a8826c'}

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