

```

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
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")
@app.route('/')# route to display the home page
def home():
    return render_template('home.html')
@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("image.html")
@app.route('/predict',methods=['GET','POST'])# route to show the predictions in a web UI
def launches():
    if request.methods=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file
        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to an array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image
        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred)#printing the prediction
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']
        result=str(index[pred[0]])
        x=result
        print(x)
        result=nutrition(result)
        print(result)
        return render_template("0.html",showcase=(result))
import http.client
conn = http.client.HTTPSConnection("calorieninjas.p.rapidapi.com")
headers = {
    'X-RapidAPI-Key': "e5805fbf62mshf8d7308c0600c2dp197087jsn93407e3cce35",

```

```

    'X-RapidAPI-Host': "calorieninjas.p.rapidapi.com"
}
conn.request("GET", "/v1/nutrition?query=Pineapple", headers=headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
import requests
url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"
querystring = {"query": "Pineapple"}
headers = {
    "X-RapidAPI-Key": "e5805fbf62mshf8d7308c0600c2dp197087jsn93407e3cce35",
    "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
}
response = requests.request("GET", url, headers=headers, params=querystring)
print(response.text)
if __name__ == "__main__":
    # running the app
    app.run(debug=False)

```

## 7.1.Feature 2

### home.html

```

<!DOCTYPE
html>

<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Home</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">
</script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<style>
body
{
background-image: url("https://img.freepik.com/free-photo/top-view-healthy-balanced-vegetarian-food_1150
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;

```

```
background-size: 100% 100% ;
}
.bar
{
margin: 0px;
padding:5px;
background-color: #c0df84;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
text-align:center;
width: 400px;
}
h3
{
margin: 0px;
padding:5px;
background-color:#c0df84;
width: 400px;
color:#000000;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:15px;
}
a
{
color:#c0df84;
float:center;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
.div1{
background-color: lightgrey;
width: 500px;
border: 10px solid peach;
```

```
padding: 20px;
margin: 20px;
height: 500px;
}
```

```
.header {position: relative;
        top:0;
        margin:0px;
        z-index: 1;
        left: 0px;
        right: 0px;
        position: fixed;
        background-color: #8B008B ;
        color: white;
        box-shadow: 0px 8px 4px grey;
        overflow: hidden;
        padding-left:20px;
        font-family: 'Josefin Sans'
        font-size: 2px;
        width: 100%;
        height:8%;
        text-align: center;
    }
    .topnav {
        overflow: hidden;
        background-color: #FCAD98;
    }

    .topnav-right a {
        float: left;
        color: black;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 10px;
    }

    .topnav-right a:hover {
        background-color: #FF69B4;
        color: black;
    }
```

```
.topnav-right a.active {
    background-color: #DA70D6;
    color: black;
}
```

```
.topnav-right {
    float: right;
    padding-right: 100px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<!--Brian Tracy-->
```

```
<div class="header">
```

```
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black; padding-top:1%;
padding-left:5%;">Nutrition Image Analysis</div>
```

```
<div class="topnav-right" style="padding-top:0.5%;">
```

```
<a class="active" href="{{ url_for('home')}}">Home</a>
```

```
<a href="{{ url_for('image1')}}">Classify</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<h1>
```

```
<center>
```

<h3>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.</h3>

```
</center>
```

```
</h1>
```

```
</body>
```

```
</html>
```

## image.html

```
<div style="float:left">
```

```
<br>
```

```
<br>
```

```
<h5><font color="black" size="3" font-family="sans-serif">
```

```
<b>Upload image to classify</b></font></h5><br><br>
```

```
<div>
```

```
<form id="upload-file" method="post" enctype="multipart/form-data">
```

```
<label for="imageUpload" class="upload-label">
```

```
    Choose...
```

```
</label>
```

```
<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
```

```
</form>
```

```
<center> <div class="image-section" style="display:none;">
```

```
<div class="img-preview">
```

```
<div id="imagePreview">
```

```
</div></center>
```

```
</div>
```

```
<center><div>
```

```
<button type="button" class="btn btn-primary btn-lg "
id="btn-predict">Classify</button>
```

```
</center></div>
```

```
</div>
```

```
<div class="loader" style="display:none;margin-left: 450px;"></div>
```

```
<h3 id="result">
```

```
<span><p style="padding-top: 25px;"><h4>Food Classified is : <h4><b><u>{{showcase}}</u>{{showcase}}
</h3>
```

```
</div>
```

```
</div>
```

# ImagePrediction.html

!DOCTYPE html>

```
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Predict</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">
</script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>
<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<style>
body
{
    background-image: url("https://i.pinimg.com/originals/be/21/1a
/be211ad5043a8d05757a3538bdd8f450.jpg");
    background-size: cover;
}
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:15px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
```

```
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
.div1{
background-color: lightgrey;
width: 500px;
border: 10px solid peach;
padding: 20px;
margin: 20px;
height: 500px;
}
```

```
.header { position: relative;
top:0;
margin:0px;
z-index: 1;
left: 0px;
right: 0px;
position: fixed;
background-color: #8B008B ;
color: white;
box-shadow: 0px 8px 4px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav {
overflow: hidden;
background-color: #FCAD98;
}
```

```
.topnav-right a {
float: left;
color: black;
text-align: center;
padding: 14px 16px;
text-decoration: none;
```



```

    font-size: 18px;
}

.topnav-right a:hover {
    background-color: #FF69B4;
    color: black;
}

.topnav-right a.active {
    background-color: #DA70D6;
    color: black;
}

.topnav-right {
    float: right;
    padding-right: 100px;
}
</style>
</head>
<body>
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black; padding-top:1%;padding-left:5%;">Nut
    <div class="topnav-right"style="padding-top:0.5%;">

        <a href="{{ url_for('home')}}">Home</a>
        <a class="active" href="{{ url_for('image1')}}">Classify</a>
    </div>
</div>
<br>

</div>
<div class="container">
<center>
<div id="content" style="margin-top:2em"></div></center>
</div>
</body>

<footer>
<script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>
</html>

```

o.html

```
<html lang="en" dir="ltr">
```

```
<head>
```

```
<style>
```

```
</style>
```

```
<meta charset="utf-8">
```

```
<title>Nutrition Image Analysis</title>
```

```
<link rel="shortcut icon" href="{{ url_for('static',  
filename='diabetes-favicon.ico') }}">
```

```
<link rel="stylesheet" type="text/css" href="{{ url_for('static',  
filename='style.css') }}">
```

```
<script src="https://kit.fontawesome.com/5f3f547070.js"  
crossorigin="anonymous"></script>
```

```
<link href="https://fonts.googleapis.com/css2?family=  
Pacifico&display=swap" rel="stylesheet">
```

```
</head>
```

```
<!-- Result -->
```

```
<div class="results">
```

```
<p style="padding-top: 150px; color:blue;"><h4  
style="color:blue;">Food Classified is: <h4><b><h4  
style="color:red;"><u>{{showcase1}}<h4><br><h4  
style="color:red;"><u>{{showcase}}<h4></p>
```

```
</div></div>
```

```
</body>
```

```
</html>
```

## 8. TESTING

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('train.h5')
model=load_model('dataset.h5')
model=load_model('nutrition.h5')
img=image.load_img(r"/content/drive/MyDrive
/CNN/Dataset/TEST_SET/PINEAPPLE/125_100.jpg")
img
```



```
img=image.load_img(r"/content/drive/MyDrive
/CNN/Dataset/TEST_SET/PINEAPPLE/125_100.jpg",
target_size=(64,64))
img
```



```
x=image.img_to_array(img)
x
array([[[[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]]],

       ...,
       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]]],

       ...,
       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]]],

       ...,
       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]]]])
```

```

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],
...,

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]]], dtype=float32)
x=np.expand_dims(x,axis=0)
[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

[[255., 255., 255.],

```

```

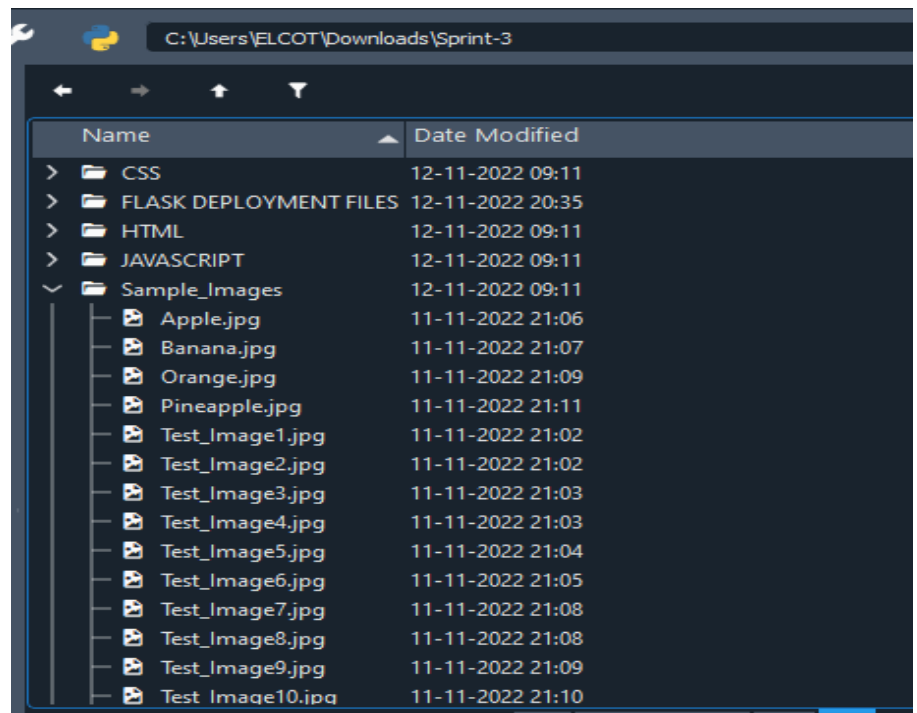
[255., 255., 255.],
[255., 255., 255.],
...
[255., 255., 255.],
[255., 255., 255.],
[255., 255., 255.]],

[[255., 255., 255.],
[255., 255., 255.],
[255., 255., 255.],
...
[255., 255., 255.],
[255., 255., 255.],
[255., 255., 255.]]], dtype=float32)
pred = model.predict
pred
array
([[0.25227112, 0.17414774, 0.15219809, 0.20493415, 0.21644896],
 [0.26760292, 0.1759095 , 0.15206912, 0.19424875, 0.21016978],
 [0.26474723, 0.165203 , 0.14452063, 0.20434381, 0.2211853 ],
 ...,
 [0.24550524, 0.1721549 , 0.16282505, 0.21065485, 0.20885986],
 [0.25395462, 0.1735253 , 0.16055605, 0.20655352, 0.20541045],
 [0.24495909, 0.15889102, 0.16927534, 0.20705006, 0.21982446]],
 dtype=float32
<bound method Model.predict of <keras.engine.
sequential.Sequential object at 0x7f94abfd7c10>>
predict_x=model.predict(x_test)

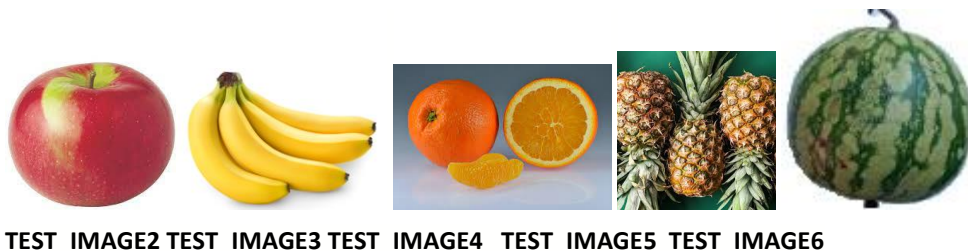
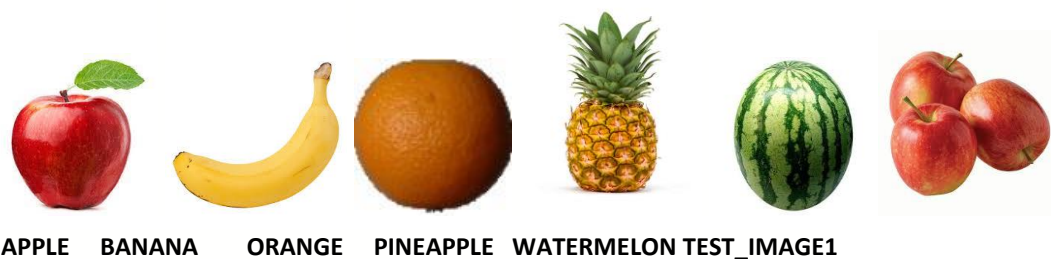
classes_x=np.argmax(predict_x,axis=1)
classes_x
array([0, 0, 0, ..., 0, 0, 0])
x_test.class_indices
index=['APPLE','BANANA','ORANGE','WATERMELON','PINEAPPLE']
result=str(index[classes_x[0]])
result
'Pineapple'

```

## TEST CASES



## 8.2 USER ACCEPTANCE TESTING



## PERFORMANCE TESTING:

Epoch 1/10  
110/110 [=====] - 27s 242ms/step - loss: 0.4205 - accuracy:  
0.8861 - val\_loss: 48.9065 - val\_accuracy: 0.1488

Epoch 2/10  
110/110 [=====] - 27s 245ms/step - loss: 0.0082 - accuracy:  
0.9989 - val\_loss: 62.1670 - val\_accuracy: 0.1280  
Epoch 3/10  
110/110 [=====] - 28s 255ms/step - loss: 0.0014 - accuracy:  
1.0000 - val\_loss: 66.6759 - val\_accuracy: 0.1488  
Epoch 4/10  
110/110 [=====] - 27s 242ms/step - loss: 3.3364e-04 - accuracy:  
1.0000 - val\_loss: 70.6794 - val\_accuracy: 0.1488  
Epoch 5/10  
110/110 [=====] - 27s 248ms/step - loss: 1.9990e-04 - accuracy:  
1.0000 - val\_loss: 74.1865 - val\_accuracy: 0.1488  
Epoch 6/10  
110/110 [=====] - 26s 236ms/step - loss: 4.5090e-04 - accuracy:  
1.0000 - val\_loss: 75.5190 - val\_accuracy: 0.1308  
Epoch 7/10  
110/110 [=====] - 27s 248ms/step - loss: 1.0600e-04 - accuracy:  
1.0000 - val\_loss: 78.4789 - val\_accuracy: 0.1488  
Epoch 8/10  
110/110 [=====] - 26s 237ms/step - loss: 7.9529e-05 - accuracy:  
1.0000 - val\_loss: 80.7918 - val\_accuracy: 0.1403  
Epoch 9/10  
110/110 [=====] - 26s 236ms/step - loss: 9.2201e-05 - accuracy:  
1.0000 - val\_loss: 80.3610 - val\_accuracy: 0.1431  
Epoch 10/10  
110/110 [=====] - 29s 266ms/step - loss: 9.1324e-05 - accuracy:  
1.0000 - val\_loss: 83.0943 - val\_accuracy: 0.1393  
<keras.callbacks.History at 0x7fbc5cb4b10>