

**AI- POWERED NUTRITION
ANALYZER FOR FITNESS
ENTHUSIASTS**



IBM NALAIYA THIRANPROJECT

REPORT SUBMITTED BY

TEAM ID : PNT2022TMID31020

SANKAR S	(620819104090)
SARAN D	(620819104092)
SELVARAJ S	(620819104098)
SIVANESH M	(620819104099)

INDUSTRY MENTOR

(SRI TULASI)

FACULTY MENTOR

(UMAMHESWARI R)



Table of Contents

1. INTRODUCTION	3
1.1 Project Overview	3
1.2 Purpose	3
2. LITERATURE SURVEY.....	4
2.1 Existing Problem.....	4
2.2 References	4
2.3 Problem Statement Definition	8
3. IDEATION & PROPOSED SOLUTION.....	9
3.1 Empathy map canvas.....	9
3.2 Ideation & Brainstorm.....	10
3.3 Proposed Solution	13
3.4 Problem Solution fit	14
4. REQUIREMENT ANALYSIS	15
4.1 Functional Requirement	15
4.2 Non-Functional Requirements.....	15
5. PROJECT DESIGN	16
5.1 Data Flow Diagrams	16
5.2 Solution & Technical Architecture	16
5.3 User Stories.....	17
6. PROJECT PLANNING & SCHEDULING	18
6.1 Sprint Planning & Estimation.....	18
6.2 Sprint Delivery Schedule.....	20
7. CODING AND SOLUTIONING	21
7.1 Feature 1	21
8. TESTING.....	23
8.1 Test Cases	23
8.2 User Acceptance Testing.....	30
9. RESULTS.....	31
9.1 Performance Metrics	31
10. ADVANTAGES AND DISADVANTAGES	33
11. CONCLUSION.....	34
12. FUTURE SCOPE	35
13. APPENDIX.....	36

1. INTRODUCTION

1.1 Project Overview

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.

1.2 Purpose

The main aim of the project is to building 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 to the trained model. The model analyses the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

2. LITERATURE SURVEY

2.1 Existing problem

Neutrino delivers nutrition-based data services and analytics to its users and wants to turn into a leading source of the nutrition-related platform. The platform employs NLP and mathematical models from the optimization theory as well as predictive analysis to enable individualized data compilation.

The application relies on Artificial Intelligence to produce custom data related to smart calorie counter powered by AI. Their artificial intelligence learns an individual's tastes, preferences, and body type. All of this is packaged in a comprehensive nutrition and activity tracker.

2.2 References

Title : Precision Nutrition Author : Daniel Kirk, Cagatay Catal, Bedir Tekinerdogan. Publications : Elsevier Year : April 2021.

Title : Promotion of healthy nutrition and physical lifestyle of teenagers Author : Maria Vanessa Villasans, Jvan Migual Pires, Francisco Florez, Revualta Publications : Journal of personalized medicine Year : March 2020.

Title : Fundamentals of food, nutrition and diet therapy Author : Sumati R. Mudambi, M. V. Rajagopal Publications : New Age International Year : July 2021.

Title : Artificial Intelligence in nutrition and science Author : Jaroslaw Sak, Magdalena Suchodalska Publications : Multidisciplinary Digital Publishing Institute Year : January 2021.

Title : Analysis of mobile applications reporting on nutritional recipes Author : Jose Huamani-Cahwana, Michael Cabonillas-Carbonell Publications : EDP science publications Year : 2019.

Title : ISSN Exercise and sport nutrition review Author : Richard B Kreider, Jose Antonio, Anton L. Almada Publications : Journal of the international society of sports nutrition Year : 15 May 2018.

Title : Consumer preference for nutrition and health claims Author : Marija Klopčič, Polona Sloan, Karman Erjavec Publications : University of Ljubljana Year : 1 December 2019

2.3 Problem Statement Definition

The main aim of the project is to building 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 to the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstroming

Sivanesh

Create meal plans for your specific digestive system	Best website for nutrition	Artificial intelligence made it possible to analyze personal health metrics
Calorie intake and shares food suggestions	Recipe building by performing depth market analysis	Food patterns and diet are the important factors to improve the lifestyle by preventing diseases
Registered dietitian nutritionists are reliable sources of nutrition information	Food recommendations for a healthy diet	Provide a real-time update about nutrition intake

Saran

Give the regular diet	Maintain the body metabolism	Set the alarm for regular food
Give the notification to take food	Recipe building by performing in-depth market analysis	Food quality analyzer
It helps to maintain a healthy body weight	Improve your well being	To detect the exact nutritional value of any given food item

Sankar

It ensures that the food has an optimal requirement of vitamins and nutrition	Improve your well being	Increase your energy level
It helps's to maintain a healthy body weight	Plan your meals and snacks to include nutrition dense food	To evaluate a food products varied physical qualities
Helps decrease a caregiver's risk	A healthy diet throughout life promotes healthy pregnancy outcomes	Support natural growth

Selvaraj

Nutrition industry overview	How nutrition AI determines people's health metrics	Personalized nutritional guide by Artificial Intelligence
Top machine learning and artificial intelligence-powered apps to maintain fitness	Food quality analyzer	Precision medicine and Artificial Intelligence
Instant multi food recognition	Food quality analyzer	Precision medicine and artificial intelligence

3.3 Proposed Solution

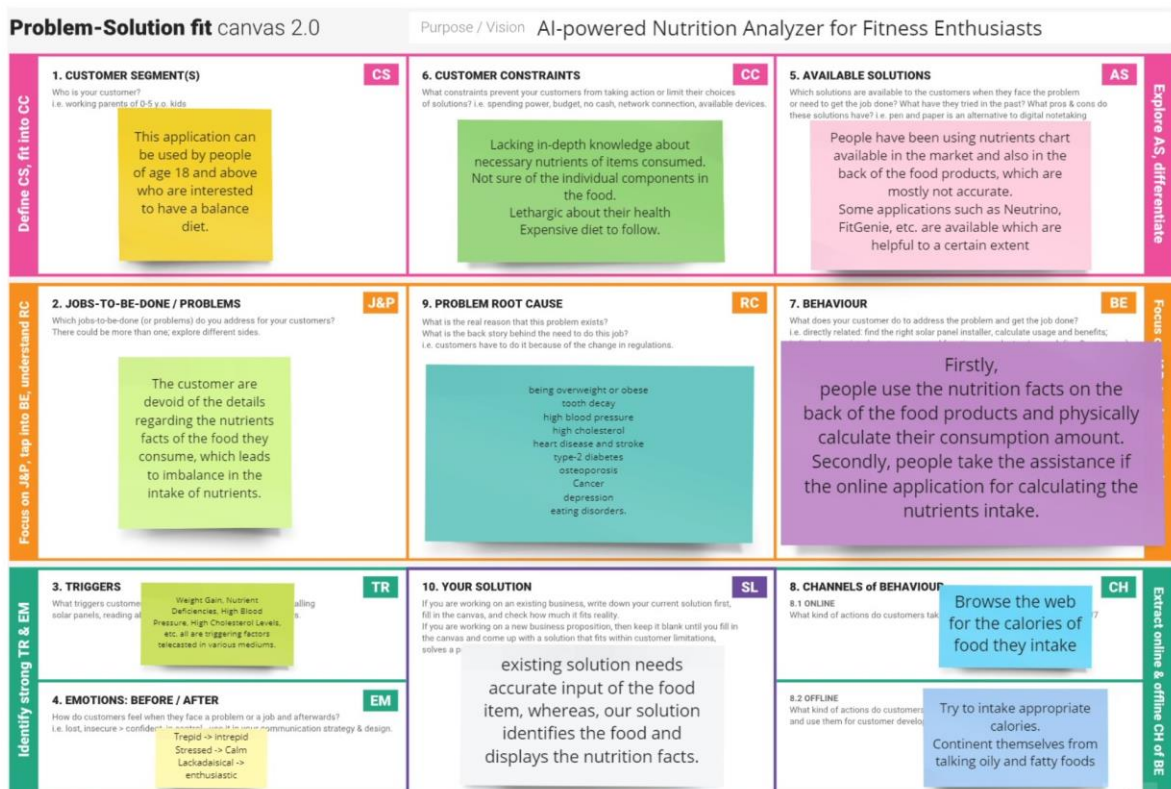
S.NO	PARAMETERS	SOLUTIONS
1.	Problem Statement	<ul style="list-style-type: none"> • Main objective is to detect the nutrition in a fruit from camera captured image. • The identification of nutrition and calories from a image is quite an interesting field. • Since nutrition monitoring plays an important role in leading healthy lifestyle, this product has the potential to become an essential in our day to day life.
2.	Idea / Solution description	<ul style="list-style-type: none"> • The solution is to develop AI-powered nutrition analyzer application. • By giving the image of the fruit as the input to the application, it will display the nutrition content in it. • By training the model with various inputs, image processing can be improved as well as the accuracy of the result.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> • Personalized nutrition for individuals. • Providing science based guidance for healthy living. • Balanced food diet and measured intake. • 24/7 support. • Serving size.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> • Economically stable product. • Change one's view towards health and fitness. • Quality of service. • High fiber food. • Accurate amount of nutrition.
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> • User friendly interface which improves the constant use of the product. • Hence, Economical growth improves. • Product will be delivered in pocket size which results in consuming low memory. • Nutrition and fitness related ads to earn profit
6.	Scalability of the Solution	<ul style="list-style-type: none"> • Offers ingredients substance detail in food • Suggest best health solution and meal plans for different criteria proposed by different individuals. • Virtualization of your long term plan to provide motivation to the customer.

3.4 ProblemSolution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.



4. REQUIREMENT ANALYSIS

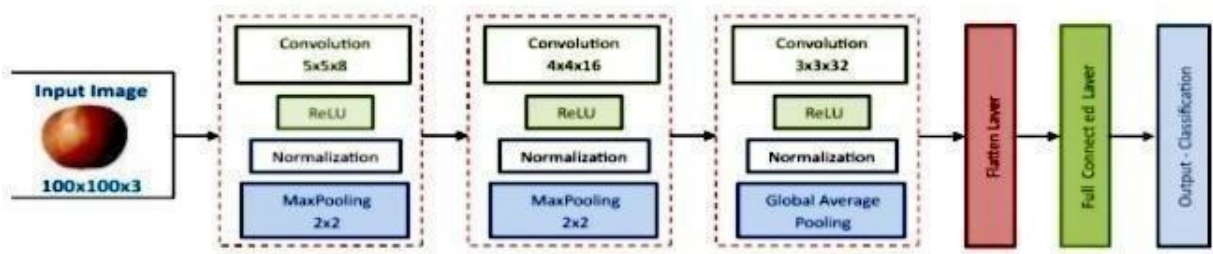
4.1 Functional requirement

- It will generate the diet plan as well as monitor the user's health to classify the category of the disease and to create the diet plan. It will also reduce the cost of consulting the person nutritionist.
- The task of food detection/classification is not easy as it seems. All possible options related to the given Image.
- Image classification, object detection, segmentation, face recognition.
- Classification of crystal structure using a convolutional neural network
- Nutrition is vital to the growth of the human body. Nutritional analysis guarantees that the meal meets the appropriate vitamin and mineral requirements, and the examination of nutrition in food aids in understanding the fat proportion, carbohydrate dilution, proteins, fiber, sugar, and so on. Another thing to keep in mind is not to exceed our daily calorie requirements

Computer-Assisted Nutritional Recognize Food Images – In order to solve this issue, a brand-new Convolutional Neural Network (CNN)- based food picture identification system was created, as described in this study. We utilized our suggested strategy on two sets of actual food picture data.

- Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fiber, Protein, Calories, etc.)
- The Ultimate Workout at Home Solution This fitness AI software is designed with personalized training regimens for each individual. It began as “gym only software,” but has now improved its system to satisfy “at home fitness” expectations.
- You take a picture, dial in data such as whether you are eating breakfast or lunch and add a quick text label, and the app estimates the calorie content.
- This software collaborated with IBM's natural language capability to provide 24-hour assistance and dietary recommendations.

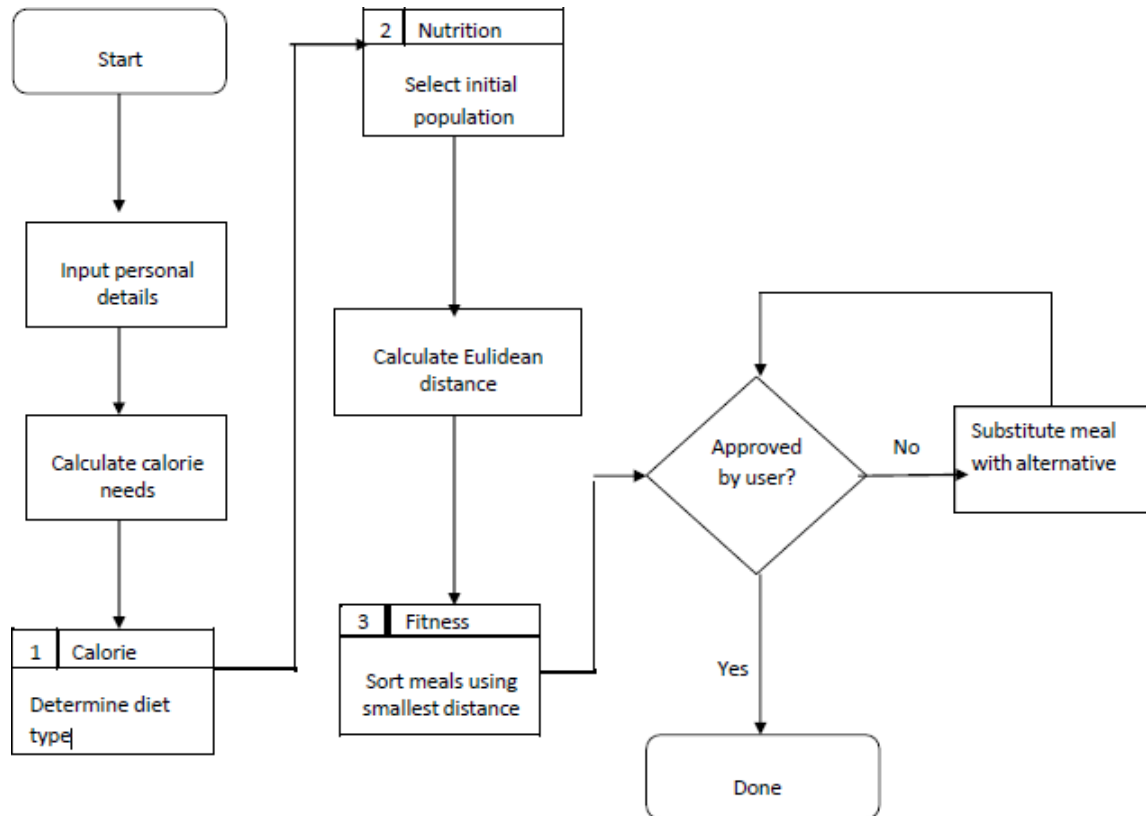
4.2 Non Technical Requirement



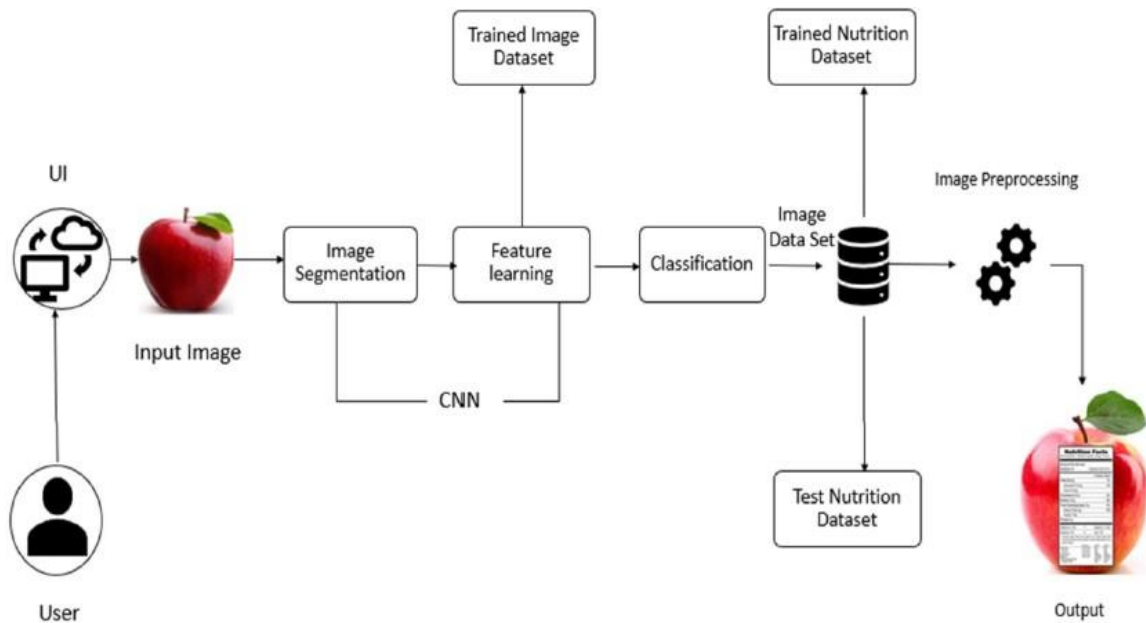
- The comparison of the proposed model with the conventional models shows that the results of this model are exceptionally good and promising to use in real-world applications.
- This sort of higher accuracy and precision will work to boost the machine's general efficiency in fruit recognition more appropriately.
- A generic model for the dietary protein requirement (as with any nutrient) defines the requirement in terms of the needs of the organism,
- i.e. metabolic demands, and the dietary amount which will satisfy those needs, i.e. efficiency of utilization, thus: dietary requirement = metabolic demand/efficiency of utilization.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



S.No	Component	Description	Technology
1.	App	User interacts with application for the prediction of Nutrition	Python, Java, HTML, SQLite, Android studio
2.	Database	Data Type, Configurations and data will be stored	MySQL, JS
3.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
4.	File Storage	File storage requirements	Cloud -- > drive
5.	Machine Learning Model	Purpose of Machine Learning Model	ANN, CNN, RNN
6.	Notification	Notification will be sent from the server	SendGrid

5.3 User Stories

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks used	SendGrid, Python, JQuery
2.	Security Implementations	Request authentication using encryption	Encryptions, SSL certs
3.	Scalable Architecture	The scalability of architecture consists of 3 tiers	Web Server – HTML, CSS ,Javascript Application Server –Python Flask Database Server – IBM Cloud IBM Cloud hosting
4.	Availability	Availability is increased by loads balancers in cloud VPS	
5.	Performance	The application is expected to handle up to 4000 predications per second	IBM Load Balance

6. Project Planning & Scheduling

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a analyzer , I can register for the application by entering my email, password, and confirming my password	?	High	SIVANESH M
Sprint 1		USN-2	As a analyzer, I can register for the application through Gmail	1	Medium	SANKAR S
Sprint 1	User Confirmation	USN-3	As a analyzer, I will receive confirmation email once I have registered for the application	1	High	SARAN D
Sprint1	Login	USN-4	As a analyzer, I can log into the application by entering email & password	2	High	SELVARAJ S

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Total Story Points	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	24 Oct 2022	27 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	28 Oct 2022	01 Nov 2022	20	04 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	09 Nov 2022	20	11 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	18 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Team ID : PNT2022TMD31020

Data Collection

```
In [4]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
In [5]: cd /content/drive/MyDrive/Dataset/Fruit dataset
```

/content/drive/MyDrive/Dataset/Fruit dataset

```
In [6]: ls
```

TEST_SET/ TEST_SET-20221031T142419Z-001.zip TRAIN_SET/ TRAIN_SET.zip

```
In [7]: !unzip TEST_SET-20221031T142419Z-001.zip
```

Archive: TEST_SET-20221031T142419Z-001.zip
replace TEST_SET/WATERMELON/r_71_100.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:

```
In [ ]: !unzip TRAIN_SET.zip
```

Archive: TRAIN_SET.zip
creating: TRAIN_SET/APPLES/

DATA PREPROCESSING

```
In [8]: # Importing the ImageDataGenerator Library
from keras.preprocessing.image import ImageDataGenerator
```

```
In [9]: # Configure ImageDatagenerator class
train_datagen = ImageDataGenerator(rescale = 1./255, horizontal_flip = True, shear_range = 0.2, zoom_range = 0.2) #for train data
```

```
In [10]: test_datagen = ImageDataGenerator(rescale = 1./255) # for test data
```

```
In [11]: # Applying ImageDataGenerator functionality to Testset
x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/Dataset/Fruit dataset/TRAIN_SET", target_size =(64,64), class_mode = "sparse", batch_size = 32)

Found 2626 images belonging to 5 classes.
```

```
In [12]: # Applying ImageDataGenerator functionality to Trainset
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/Dataset/Fruit dataset/TEST_SET", target_size = (64,64), class_mode = "sparse", batch_size = 32)

Found 1055 images belonging to 5 classes.
```

```
In [13]: print(x_train.class_indices)

{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

```
In [14]: print(x_test.class_indices)

{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

```
In [15]: from collections import Counter as c
c(x_train.labels)
```

```
Out[15]: Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})
```

MODEL BUILDING

Importing model building libraries

```
In [16]: from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

```
In [17]: model = Sequential()
```

Initializing the model

```
In [18]: classifier = Sequential ()
```

Adding CNN layers

```
In [19]: # Adding Convolutional Layer
        classifier.add(Convolution2D(32,(3,3),activation = "relu", input_shape = (64,64,3)))
```

```
In [21]: classifier.add(MaxPooling2D(pool_size=(2,2))) #adding pooling layer
```

```
In [22]: classifier.add(Convolution2D(32,(3,3),activation = "relu"))
```

```
In [23]: classifier.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [24]: classifier.add(Flatten()) #adding flatten layer
```

Adding Dense layers

```
In [25]: classifier.add(Dense(units = 128,activation='relu')) #adding dense layer
        #classifier.add(Dense(300,activation='relu')) #adding dense layer 2, we can add any no.of hidden layers, here it is 2
        classifier.add(Dense(units = 5, activation = 'softmax')) #adding output layer to the model
```

```
In [26]: classifier.summary()
```

```
In [24]: classifier.add(Flatten()) #adding flatten layer
```

Adding Dense layers

```
In [25]: classifier.add(Dense(units = 128,activation='relu')) #adding dense layer
        #classifier.add(Dense(300,activation='relu')) #adding dense layer 2, we can add any no.of hidden layers, here it is 2
        classifier.add(Dense(units = 5, activation = 'softmax')) #adding output layer to the model
```

```
In [26]: classifier.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 32)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 128)	147584
dense_1 (Dense)	(None, 5)	645

=====
Total params: 158,373
Trainable params: 158,373
Non-trainable params: 0
=====

Configuring the learning process

```
In [27]: classifier.compile(loss = "sparse_categorical_crossentropy", metrics = ["accuracy"], optimizer = 'adam') #compiling the model
```

Training the model

```
In [28]: classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_train),epochs=20, validation_data=x_test,validation_steps = len(x_test))
```

Epoch 1/20
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
 """Entry point for launching an IPython kernel.

526/526 [=====] - 307s 585ms/step - loss: 0.1565 - accuracy: 0.9429 - val_loss: 0.0374 - val_accuracy: 0.9839
Epoch 2/20
526/526 [=====] - 23s 44ms/step - loss: 9.1779e-04 - accuracy: 1.0000 - val_loss: 0.0445 - val_accuracy: 0.9801
Epoch 3/20
526/526 [=====] - 25s 48ms/step - loss: 5.5660e-04 - accuracy: 1.0000 - val_loss: 0.0572 - val_accuracy: 0.9791
Epoch 4/20
526/526 [=====] - 25s 48ms/step - loss: 1.4845e-04 - accuracy: 1.0000 - val_loss: 0.0647 - val_accuracy: 0.9791
Epoch 5/20
526/526 [=====] - 23s 45ms/step - loss: 9.2684e-05 - accuracy: 1.0000 - val_loss: 0.0549 - val_accuracy: 0.9791
Epoch 6/20
526/526 [=====] - 23s 44ms/step - loss: 0.0809 - accuracy: 0.9802 - val_loss: 0.0348 - val_accuracy: 0.9810
Epoch 7/20
526/526 [=====] - 23s 44ms/step - loss: 3.0526e-04 - accuracy: 1.0000 - val_loss: 0.0261 - val_accuracy: 0.9839
Epoch 8/20
526/526 [=====] - 23s 44ms/step - loss: 1.3504e-04 - accuracy: 1.0000 - val_loss: 0.0107 - val_accuracy: 0.9981
Epoch 9/20
526/526 [=====] - 23s 43ms/step - loss: 6.7562e-05 - accuracy: 1.0000 - val_loss: 0.0216 - val_accuracy: 0.9858
Epoch 10/20
526/526 [=====] - 24s 47ms/step - loss: 1.8923e-04 - accuracy: 1.0000 - val_loss: 5.8745e-04 - val_accuracy: 1.0000
Epoch 11/20
526/526 [=====] - 23s 44ms/step - loss: 3.0109e-05 - accuracy: 1.0000 - val_loss: 0.0256 - val_accuracy: 0.9848
Epoch 12/20
526/526 [=====] - 23s 43ms/step - loss: 8.8911e-06 - accuracy: 1.0000 - val_loss: 0.0265 - val_accuracy: 0.9839
Epoch 13/20
526/526 [=====] - 23s 44ms/step - loss: 9.1767e-06 - accuracy: 1.0000 - val_loss: 0.0137 - val_accuracy: 0.9905
Epoch 14/20
526/526 [=====] - 23s 44ms/step - loss: 8.1939e-06 - accuracy: 1.0000 - val_loss: 0.0325 - val_accuracy: 0.9829
Epoch 15/20
526/526 [=====] - 25s 47ms/step - loss: 4.7228e-06 - accuracy: 1.0000 - val_loss: 0.0126 - val_accuracy: 0.9924
Epoch 16/20
526/526 [=====] - 23s 43ms/step - loss: 2.6643e-06 - accuracy: 1.0000 - val_loss: 0.0306 - val_accuracy: 0.9829
Epoch 17/20

Testing the model

```
In [36]: from tensorflow.keras.models import load_model  
from tensorflow.keras.preprocessing import image  
import numpy as np  
model = load_model("Nutrition Analyzer.h5") #Loading the model
```

```
In [37]: from tensorflow.keras.utils import img_to_array  
img = image.load_img(r'/content/drive/MyDrive/Dataset/Fruit dataset/TEST_SET/APPLES/151_100.jpg', grayscale = False, target_size =(64,64)) #Loading the image  
x = img_to_array(img) #image to array  
x = np.expand_dims(x,axis = 0)  
predict_x=model.predict(x)  
classes_x=np.argmax(predict_x,axis=-1)  
classes_x
```

1/1 [=====] - 0s 108ms/step

```
Out[37]: array([0])
```

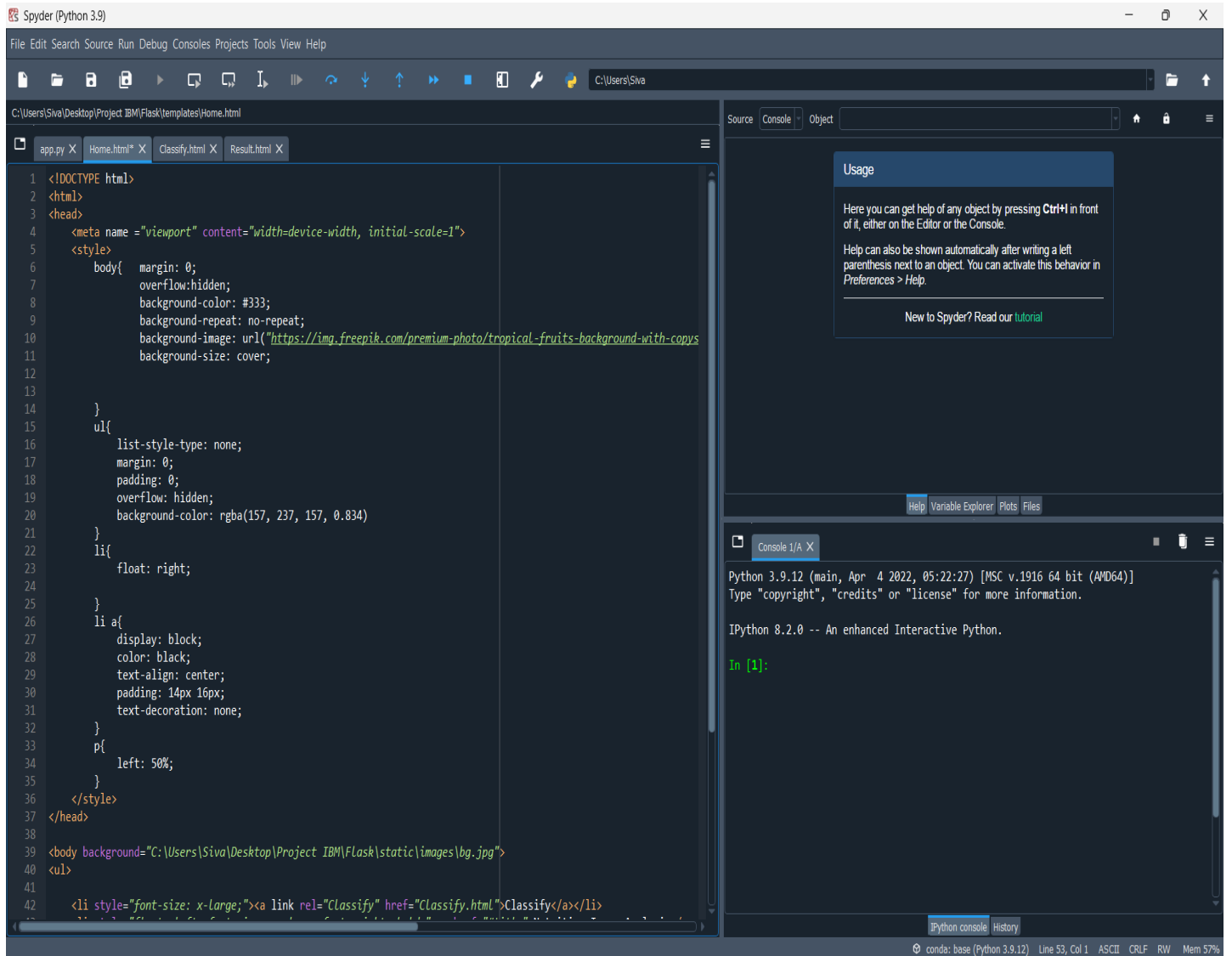
```
In [38]: img #display the loaded image
```

Out[38]: 

```
In [39]: index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']  
result=str(index[classes_x[0]])  
result
```

Out[39]: 'APPLES'

Feature 2



File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Siva

C:\Users\Siva\Desktop\Project IBM\Flask\templates\Classify.html

app.py X Home.html X Classify.html X Result.html X

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89

```
});  
</script>  
<script>  
function preview() {  
    frame.src=URL.createObjectURL(event.target.files[0]);  
}  
  
$(document).ready(function() {  
    $('#clear_button').on('click', function() {  
        $('#image').val('');  
        $('#frame').attr('src','');  
    });  
});  
</script>  
<body>  
<body background = "C:\Users\Siva\Desktop\Project IBM\Flask\bg.jpg ">  
<ul>  
    <li style="font-size: x-large;"><a link rel="Home" href="Home.html">Home</a></li>  
    <li style="float: left; font-size: xx-large;font-weight: bold;"><a href="#title">Nutrition Image Analysis</a></li>  
</ul>  
<section id="content">  
    <div class="left">  
        <form action="/predict" method="POST" enctype="multipart/form-data">  
            <label style="font-size: x-large;font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-s">  
                <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>  
                <img id="frame" src="" width="250px" height="250px"/>  
                <div class="buttons_div">  
                    <button type="submit" class="btn btn-dark" id="predict_button" style="width: fit-content;font-size: La">  
                    <button type="button" class="btn btn-dark" id="clear_button" style="font-size: Large;font-family: 'Fra">  
                </div>  
            </form>  
        </div>  
    </section>  
</body>  
</html>
```

Source Console Object

Usage
Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in [Preferences > Help](#).
New to Spyder? Read our [tutorial](#)

Help Variable Explorer Plots Files

Console 1/A X
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:

IPython console History
conda: base (Python 3.9.12) Line 1, Col 1 ASCII CRLF RW Mem 57%

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Siva\Desktop\Project IBM\Flask\templates\Result.html

app.py X Home.html X Classify.html X Result.html X

```
1 <html lang="en" dir="ltr">
2 <head>
3 <style>
4   body{
5     background-image: url('https://img.freepik.com/premium-photo/tropical-fruits-background-with-copyspace-left
6     background-size: cover;
7     background-repeat: no-repeat;
8   }
9
10 </style>
11 <meta charset="utf-8">
12 <title>Nutrition Image Analysis</title>
13 <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
14 <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
15 <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
16 <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
17 </head>
18
19 <!-- Result -->
20 <div class="results">
21   <p style="padding-top: 150px; color:blue;text-align: center;">
22     <h1 style="color:blue;font-size:xx-large;text-align: center;font-family: 'Segoe UI', Tahoma, Ger
23     <b><h2 style="color:red;text-align: center;font-size: xx-large;font-family: 'Franklin Gothic
24     <br><h3 style="color:black;text-align: center;width: 58%;margin-left: 20%;font-size: Lar
25   </div>
26   <br>
27   <br>
28
29 </div>
30 </body>
31 </html>
```

Source Console Object

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Help Variable Explorer Plots Files

Console 1/A X

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:

Python console History

conda: base (Python 3.9.12) Line 31, Col 8 ASCII CRLF RW Mem 57%

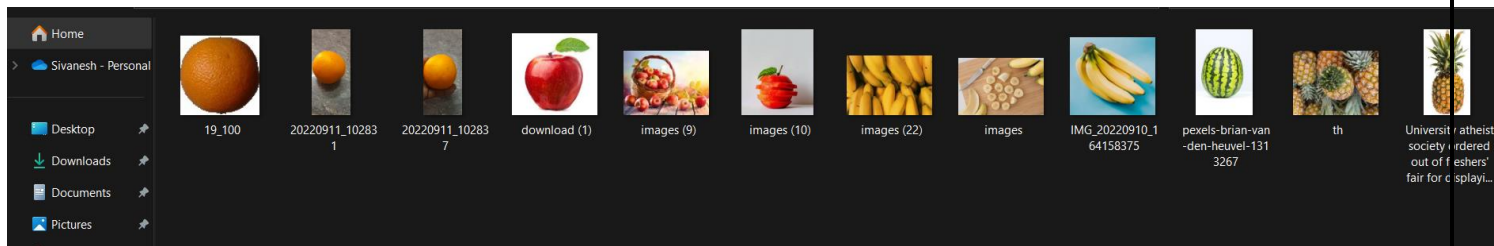
8. TESTING

8.1 Test Cases

Steps To Execute	Test Data	Expected Result	Actual Result	Status	
1.Enter URL and click go 2.Verify home page displayed or not	http://127.0.0.1:5000/	Home page and elemets should display	Working as expected	Pass	
1.Enter URL and click go 2.Click on dropdown button 3.click to URL open the home page	http://127.0.0.1:5000/	Application should show below UI elements: click to URL open the home page	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	img.jpg	It should predict the fruit	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	1.jpg	it should predict the fruit in the image	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	2.jpg	it should predict the fruit as Apple	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	3.jpg	it should predict the fruit as Banana	Working as expected	Pass	

Steps To Execute	Test Data	Expected Result	Actual Result	Status	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	4.jpg	it should predict the fruit as Orange	orking as expect	Pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file	5.jpg	it should predict the fruit as Watermelon	orking as expect	Pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file 4.upload to the image nutrition data	1.jpg data	it should predict the fruit as Pineapple	orking as expect	Pass	
1.Enter URL(http://127.0.0.1:5000/)	2.jpg data				

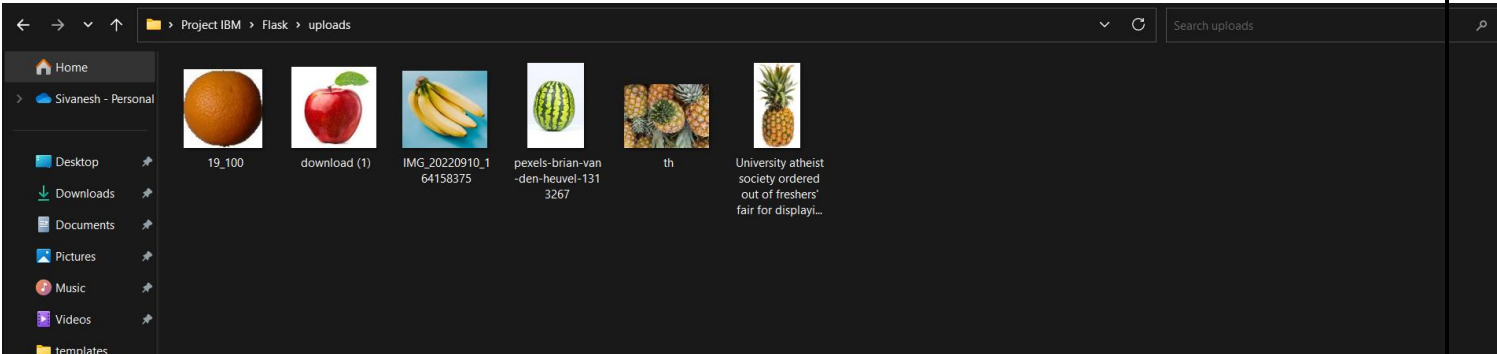
Steps To Execute	Test Data	Expected Result	Actual Result	Status	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file 4.upload to the image nutrition data	3.jpg data	it should predict the fruit as Banana	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file 4.upload to the image nutrition data	4.jpg data	it should predict the fruit as Orange	Working as expected	pass	
1.Enter URL(http://127.0.0.1:5000/) and click go 2.Click on choose file button 3.upload a image file 4.upload to the image nutrition data	5.jpg data	it should predict fruit as Orange	Working as expected	pass	
Successful view to nutrition information		it should show the nutrition content of the fruit.		Pass	



8.2 User Acceptance Testing

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	11	2	3	4	20
Duplicate	1	0	1	0	2
External	2	3	1	1	7
Fixed	10	2	3	17	32
Not Reproduced	0	1	1	0	2
Skipped	1	0	1	1	3
Won'tFix	1	4	1	3	9
Totals	26	12	11	26	75

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	55	0	0	55
Security	2	0	0	2



9. RESULTS

9.1 Performance Metrics

```
2022-11-19 09:05:46.357961: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-AHNEOAFD
2022-11-19 09:05:46.358089: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-AHNEOAFD
2022-11-19 09:05:46.360295: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

```
In [2]: runfile('C:/Users/Siva/Desktop/Project IBM/Flask/app.py', wdir='C:/Users/Siva/Desktop/Project IBM/Flask')
```

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/Nov/2022 09:06:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 09:06:14] "GET /favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 09:06:17] "GET /Classify.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 09:06:17] "GET /favicon.ico HTTP/1.1" 200 -
1/1 [=====] - 1s 703ms/step
127.0.0.1 - - [19/Nov/2022 09:06:40] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 09:06:40] "GET /static/style.css HTTP/1.1" 200 -
Apple
127.0.0.1 - - [19/Nov/2022 09:06:40] "GET /static/diabetes-favicon.ico HTTP/1.1" 200 -
```

Python console History

LSP Python: ready conda: base (Python 3.9.12) Line 26, Col 1 ASCII CRLF RW Mem 72%

Output

IBM 127.0.0.1:5000

← → ↻ 127.0.0.1:5000

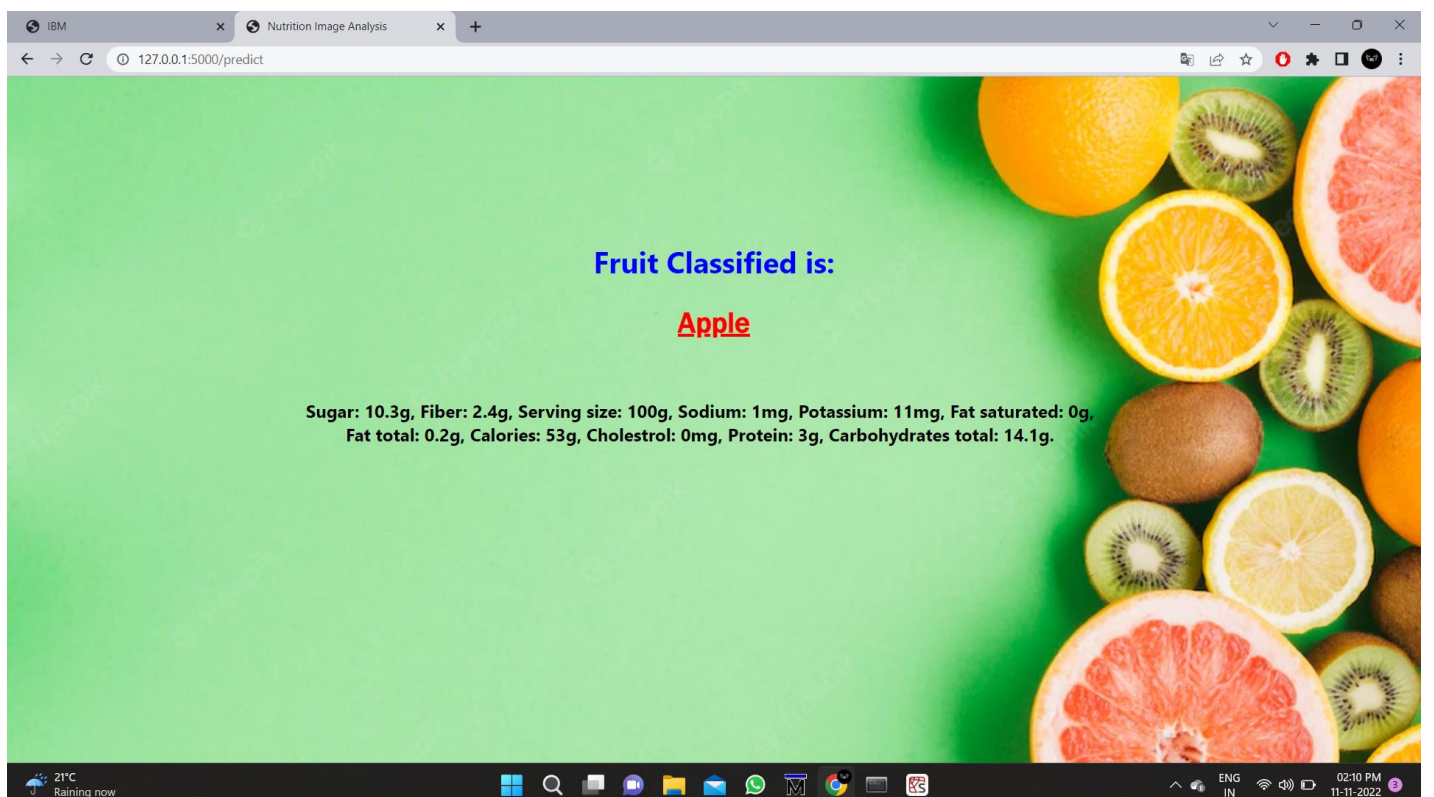
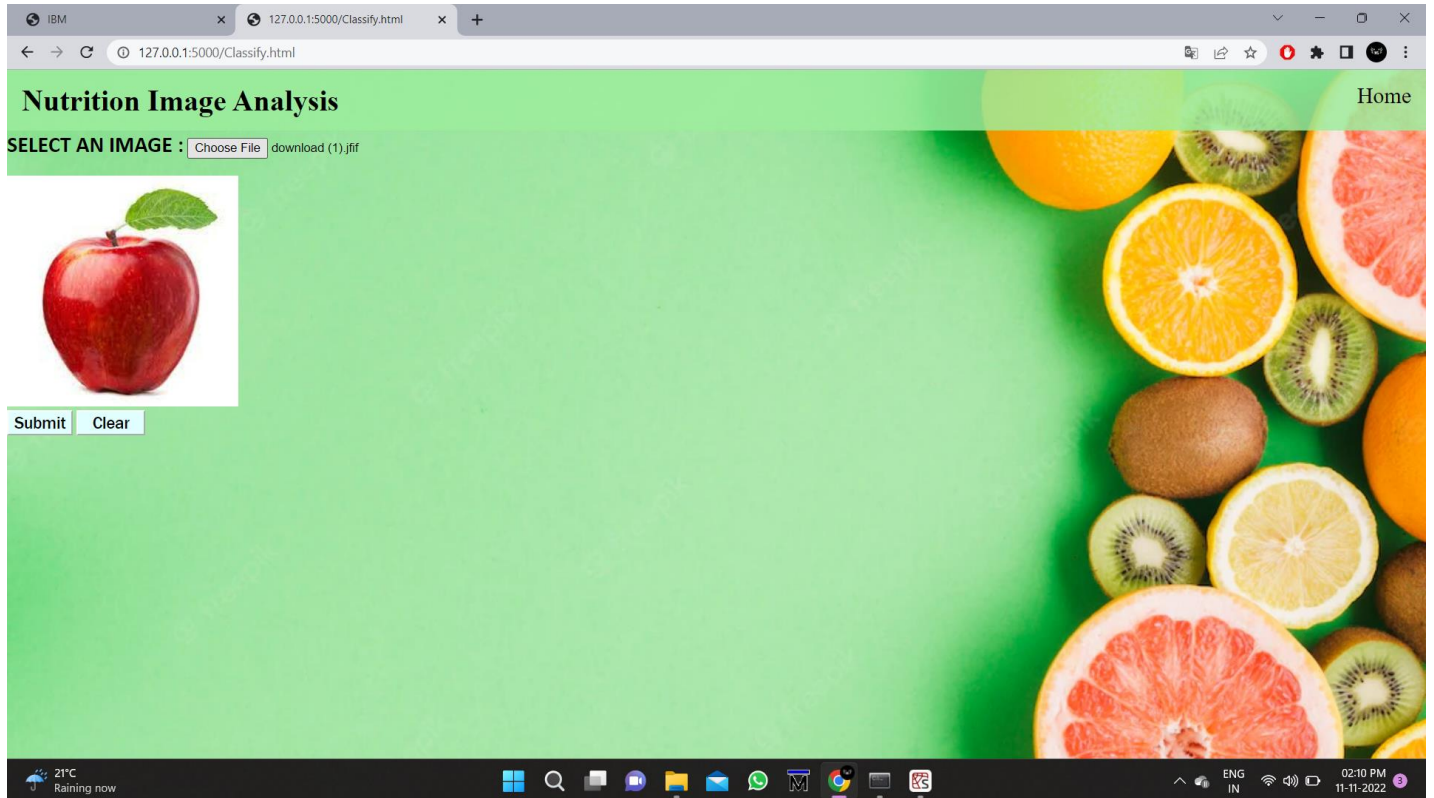
Nutrition Image Analysis

Classify

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.

21°C Raining now

ENG IN 02:08 PM 11-11-2022



10. Advantages & Disadvantages

Advantages :

- Easy to use.
- Gives information about the nutrition content of the fruits
- Attractive user interface.
- Quick and reliable output.
- Exact prediction.
- User friendly

Disadvantage :

- Requires internet connection.
- Need input data to be in the image format.

11. CONCLUSION

By the end of this project we will

- know fundamental concepts and techniques of Convolutional Neural Network.
- gain a broad understanding of image data
- know how to build a web application using the Flask framework.
- know how to pre-process data and
- know how to clean the data using different data preprocessing techniques.

12. FUTURE SCOPE

- AI is revolutionizing the health industry.
- It is majorly used in improving marketing and sales decisions, AI is now also being used to reshape individual habits.
- In future we don't want to go to gym and do any diets. By using this nutrition fitness analyzer we can maintain our diet plans without any help from others and we can lead a happy and healthy life with good wealth.
- AI can easily track health behaviors and repetitive exercise patterns and use the data to guide you towards your fitness journey and diet plans .

13. Appendix

Source Code:

Home.html

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body{
      margin: 0;
      overflow: hidden;
      background-color: #333;
      background-repeat: no-repeat;
      background-image: url("https://img.freepik.com/premium-photo/tropical-fruits-
background-with-copyspace-left_23-2147822402.jpg?w=1060");
      background-size: cover;

    }
    ul{
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: rgba(157, 237, 157, 0.834)
    }
    li{
      float: right;

    }
    li a{
      display: block;
      color: black;
```

```

        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }
    p{
        left: 50%;
    }
</style>
</head>

<body background="C:\Users\Siva\Desktop\Project IBM\Flask\static\images\bg.jpg">
<ul>

    <li style="font-size: x-large;"><a href="Classify.html">Classify</a></li>
    <li style="float: left; font-size: xx-large;font-weight: bold;"><a href="#title">Nutrition Image
        Analysis</a></li>
</ul>
<p style="text-align: center; margin: 20ch;font-size: x-large; background-color:
    lightcyan;border-radius: 0.5cm;width: 60%;margin-left: 20%;font-style: oblique;font-weight:
    500; color: black;">
<br>Food is essential for human life and has been the concern of many healthcareconventions.
<br>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. <br>It is a vital part of analytical chemistry that provides information about
the chemical composition, processing, quality control and contamination of food.<br>

<br></p>
</body>
</html>

```

Classify.html

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body{ margin: 0;
          overflow:hidden;
          background-color: #333;
          background-image: url("https://img.freepik.com/premium-photo/tropical-fruits-background-with-
copyspace-left_23-2147822402.jpg?w=1060");
          background-size: cover;

          }

    ul{
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: rgba(157, 237, 157, 0.834)
    }
    li{
      float: right;

    }
    li a{
      display: block;
      color: black;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
    }
    p{
      left: 50%;
    }

  </style>
</head>
<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
  }

  $(document).ready(function() {
    $('#clear_button').on('click', function() {
      $('#image').val("");
      $('#frame').attr('src','');
    });
  });
},
```

```
</script>
```

```
<script>
function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}
```

```
$(document).ready(function() {
    $('#clear_button').on('click', function() {
        $('#image').val("");
        $('#frame').attr('src','');
    });
});
```

```
</script>
```

```
<body>
```

```
<body background = "C:\Users\Siva\Desktop\Project IBM\Flask\bg.jpg ">
```

```
<ul>
```

```
<li style="font-size: x-large;"><a href="Home.html">Home</a></li>
```

```
<li style="float: left; font-size: xx-large;font-weight: bold;"><a href="#title">Nutrition Image Analysis</a></li>
</ul>
```

```
<section id="content">
```

```
<div class="left">
```

```
<form action="/predict" method="POST" enctype="multipart/form-data">
```

```
<label style="font-size: x-large;font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;font-weight: bold;">SELECT AN IMAGE :</label>
```

```
<input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
```

```
<img id="frame" src="" width="250px" height="250px"/>
```

```
<div class="buttons_div">
```

```
<button type="submit" class="btn btn-dark" id="predict_button" style="width: fit-content;font-size:
large;font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;background-color: lightcyan;border-
color: white;" >Submit</button>
```

```
<button type="button" class="btn btn-dark" id="clear_button" style="font-size: large;font-family: 'Franklin
Gothic Medium', 'Arial Narrow', Arial, sans-serif;background-color: lightcyan;border-color: white;">&nbsp;Clear
&nbsp;</button>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</section>
```

```
</body>
```

```
</html>
```

Result.html

```
<html lang="en" dir="ltr">
<head>
<style>
    body{
        background-image: url('https://img.freepik.com/premium-photo/tropical-fruits-background-with-
        copyspace-left_23-2147822402.jpg?w=1060');
        background-size: cover;
        background-repeat: no-repeat;
    }

</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;text-align: center;">
            <h1 style="color:blue;font-size:xx-large;text-align: center;font-family: 'Segoe
            UI', Tahoma, Geneva, Verdana, sans-serif;">Fruit Classified is: </h1>
            <b><h2 style="color:red;text-align: center;font-size: xx-large;font-
            family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;"><u>{ { showcase1 } }</u></b></h2>
            <br><h3 style="color:black;text-align: center;width:
            58%;margin-left: 20%;font-size: larger;font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-
            serif;">{ { showcase } }</h3></p>
        </div>
        <br>
        <br>

    </div>
</body>
</html>
```

App.py

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request
app = Flask(__name__) #initializing the flask object in the variable name as app
@app.errorhandler(404)
def not_found(e):
# defining function
    return render_template("Classify.html")
model = load_model("Nutrition Analyzer.h5")
@app.route('/',methods=['GET']) #routing the html
def home():
    return render_template('Home.html') #displayed on the html page
@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(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)
        index = ["Apple","Banana","Orange","Pineapple","Watermelon"]
        result = str(index[pred[0]])
        print(result)
        x=result
        if(result == "Apple"):
            content = "Sugar: 10.3g,\nFiber: 2.4g, Serving size: 100g, Sodium: 1mg, Potassium: 11mg, Fat saturated: 0g, Fat total: 0.2g, Calories: 53g, Cholestrol: 0mg, Protein: 3g, Carbohydrates total: 14.1g."
        elif(result == "Banana"):
            content = "Sugar: 12.3g,\nFiber: 2.6g, Serving size: 100g, Sodium: 1mg, Potassium: 22mg, Fat saturated: 0.1g, Fat total: 0.3g, Calories: 89.4g, Cholestrol: 0mg, Protein: 1.1g, Carbohydrates total: 23.2g."
        elif(result == "Orange"):
            content = "Sugar: 16.8g,\nFiber: 4.3g, Serving size: 100g, Sodium: 1mg, Potassium: 22mg, Fat saturated: 0g, Fat total: 0.2g, Calories: 84g, Cholestrol: 0mg, Protein: 1.7g, Carbohydrates total: 21.2g."
        elif(result == "Pineapple"):
            content = "Sugar: 9.9g,\nFiber: 21.4g, Serving size: 100g, Sodium: 0mg, Potassium: 8mg, Fat saturated: 0g, Fat total: 0.1g, Calories: 50.8g, Cholestrol: 0mg, Protein: 10.5g, Carbohydrates total: 13g."
        elif(result == "Watermelon"):
            content = "Sugar: 16.3g,\nFiber: 2.3g, Serving size: 100g, Sodium: 1mg, Potassium: 22mg, Fat saturated: 0g, Fat total: 0.3g, Calories: 83g, Cholestrol: 0mg, Protein: 0.9g, Carbohydrates total: 21.6g."
        else:
            content = ""
        return render_template("Result.html",showcase=(content),showcase1=(x))
if __name__=="__main__":
    app.run(debug = False) #running the app
```

GitHub:

<https://github.com/IBM-EPBL/IBM-Project-46269-1660744141>

Demo Link:

<https://drive.google.com/drive/u/0/folders/1dkW4SiQg8xLJYLN5r0PPXbIvT3TqKcHO>

<https://github.com/IBM-EPBL/IBM-Project-46269-1660744141/blob/main/Final%20deliverables/Demo/Project%20Demo.mp4>

