

TEAM ID: PNT2022TMID18684

PROJECT NAME: AI-Powered Nutrition Analyzer for Fitness Enthusiasts

Project Report Format

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 build a model which is used for classifying the fruit depends on the different characteristics like color, 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, Fiber, Protein, Calories, etc.).

2. LITERATURE SURVEY

2.1 Existing problem

Nutrino 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

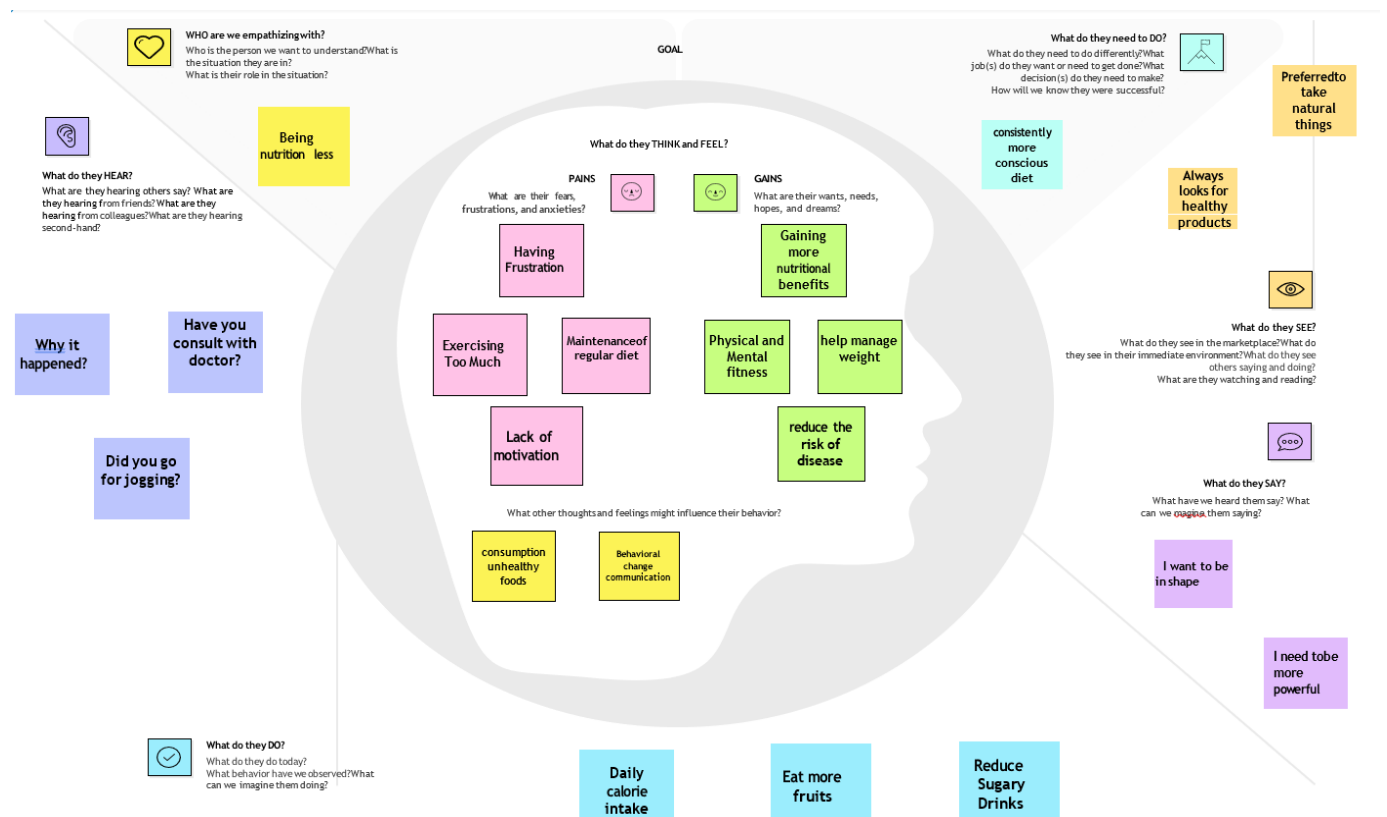
<https://www.nutrinohealth.com/>

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 color, shape, texture etc.. High-Calorie food intake can be harmful and result in obesity, which is a preventable medical condition that causes abnormal accumulation of fat in the body. It can result in numerous diseases such as obesity, diabetes, cholesterol, heart attacks, blood pressure, and other diet-related ailments. In order to deal with such problems, people are inclined towards making a difference in their diet plans by paying more attention to what type of food they are consuming. Diet management is a key concern among individuals belonging to different age groups. However, one major challenge in diet management is maintaining a balance between what one eats and how one monitors his/her food consumption. The immense increase in ailments such as high cholesterol, blood pressure, strokes, etc. demands nutritional and diet management for which people resort to expensive nutrition therapies. 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.).

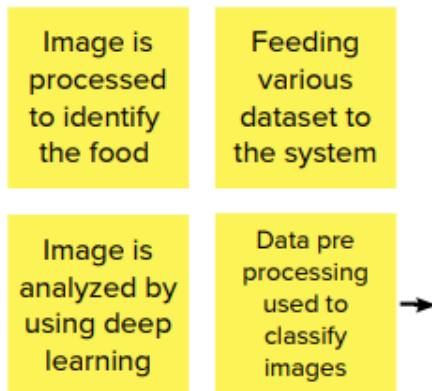
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

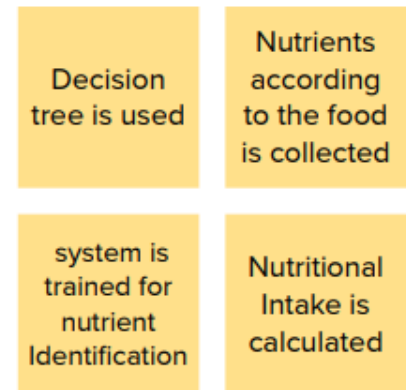


3.2 Ideation & Brainstorming

ABIRAMI M



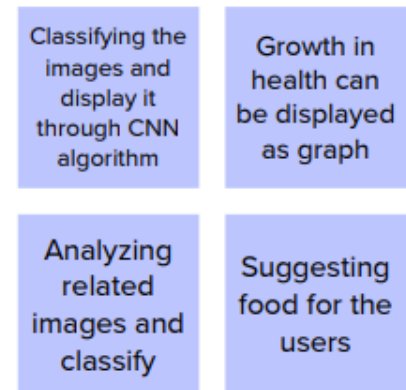
DEEPIKA T



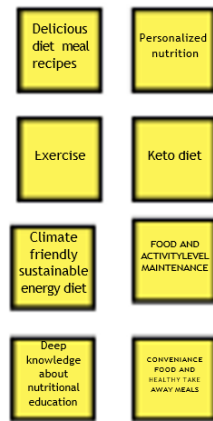
HARIPRIYAA I



JANSI RANI T



NUTRITIONAL



WORKOUT



PROGRAMS



3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The main aim 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 to the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).
2.	Idea / Solution description	The user can capture images and can get the details about the food and nutrients that are present in the food and suggestions related to the health and fitness of the user.
3.	Novelty / Uniqueness	This system analyses the images that are given as input and It displays the nutrients that are available in the fruit.
4.	Social Impact / Customer Satisfaction	Customer is satisfied when the nutrients are displayed by the system. And the suggestions are provided.
5.	Business Model (Revenue Model)	Subscription for premium users is being implemented for revenue purposes.
6.	Scalability of the Solution	Recommendation system are also included for scalability purposes.

3.4 Problem Solution 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.

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none"> ❖ People who want to maintain themselves fit and healthy. ❖ Mostly fitness 	6. CUSTOMER CONSTRAINTS CC <ul style="list-style-type: none"> ❖ May be unclear images that make difficult for us to compute the accurate output . ❖ Lack of awareness about the app among the users. 	5. AVAILABLE SOLUTIONS AS <ul style="list-style-type: none"> ❖ Try to have healthy solutions for all the problems. ❖ Act according to the suggestions provided by the app. 	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE /PROBLEMS J&P <ul style="list-style-type: none"> ❖ Calculating accurate results according to the given input. 	9. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none"> ❖ Unavailability of authentic source ❖ Different shapes and 	7. BEHAVIOUR BE <ul style="list-style-type: none"> ❖ The customer will have to upload the image input ❖ The application will display the Nutritional content based on the shape and texture of the 	Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR <ul style="list-style-type: none"> ❖ User urge to be healthy and fit ❖ Through social media and advertisement 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> ❖ We track health care plan for Individuals ❖ And provide health care suggestions 	8. CHANNELS of BEHAVIOUR CH <p>ONLINE</p> <ul style="list-style-type: none"> ❖ Through Social media and through advertisements <p>OFFLINE</p> <ul style="list-style-type: none"> ❖ Through other users
4. EMOTIONS: BEFORE / AFTER EM <p>Before: Lack of accuracy After: Customer is now self reliant Availability of authentic source to provide information</p>		

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

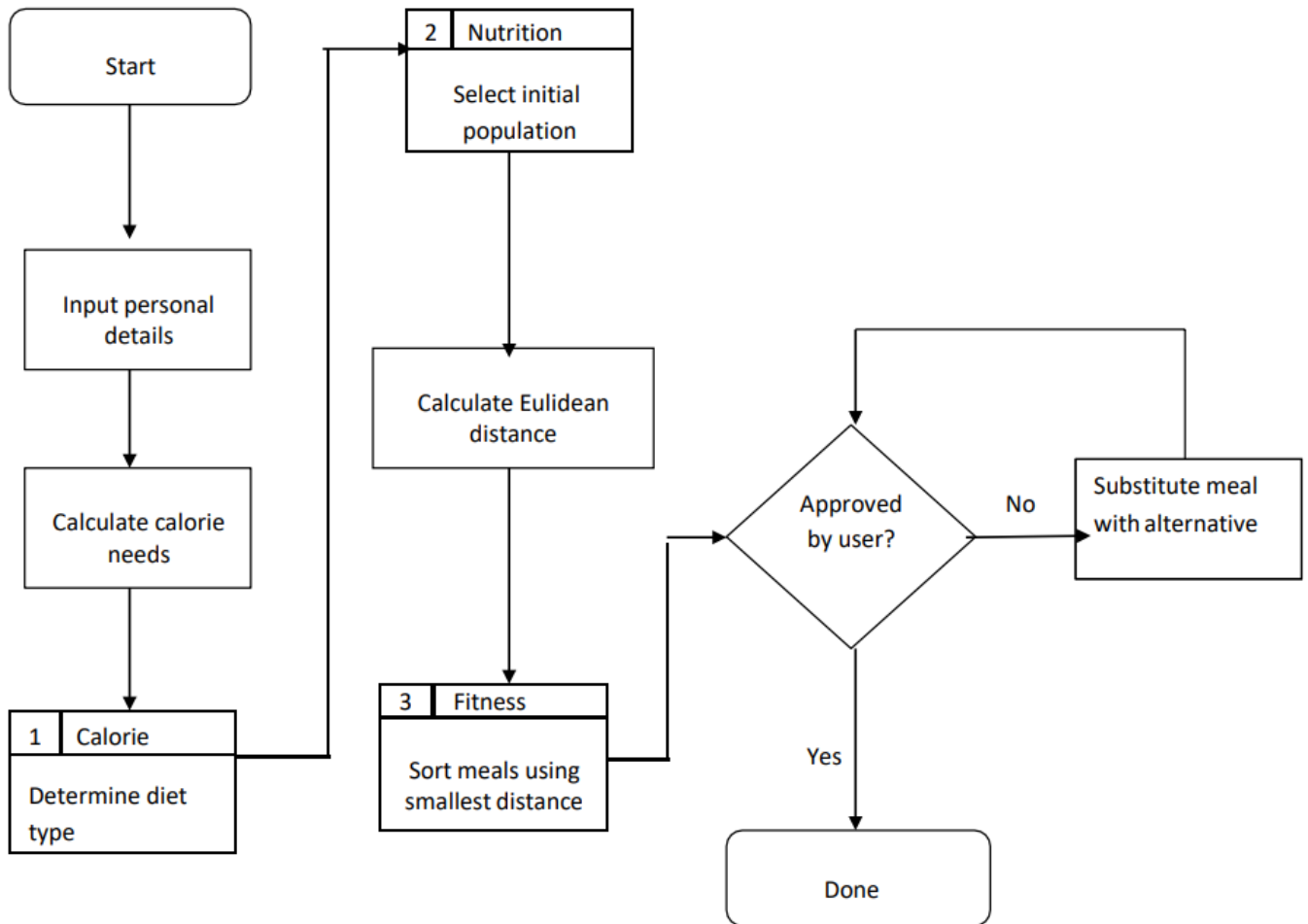
FR NO.	FUNCTIONAL REQUIREMENT(EPIC)	SUB REQUIREMENT(SUB-TASK)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Google Login through Email
FR-4	Choose package	Selection of desired package
FR-5	Generate the daily plan	Daily plans will be generated by dietician.
FR-6	Manage progress report	Gathering information from database and generating report
FR-7	Query	The user can ask for changes in plan

4.2 Non-Functional requirements

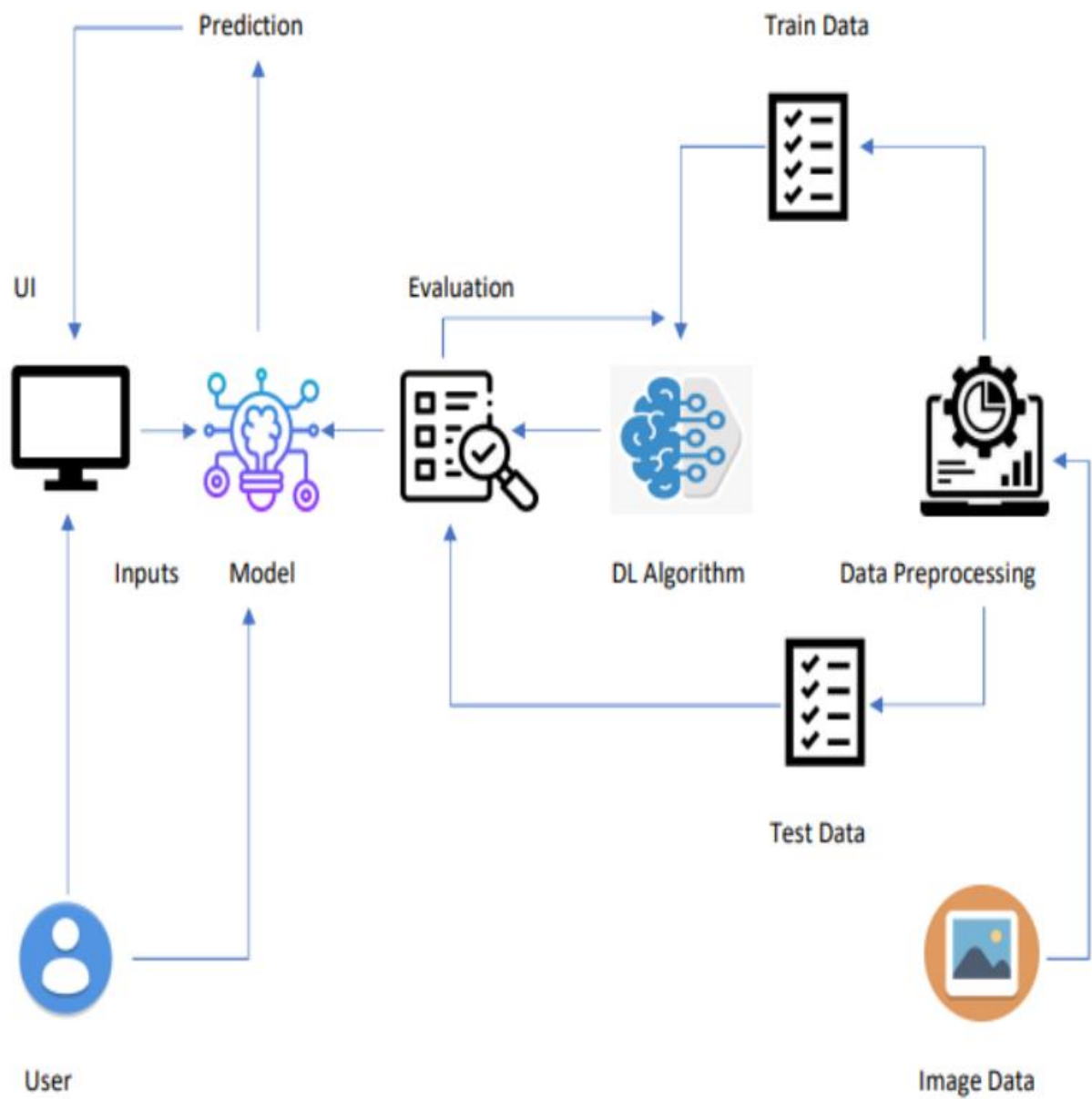
FR NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	Usability	Easy to use with interactive User Interface
NFR-2	Security	User can access only their personal information and not that of other users.
NFR-3	Reliability	The average time of failure shall be 7 days.
NFR-4	Performance	The results has to be shown within 10 sec
NFR-5	Availability	The dietician shall be available to users 24 hours a day, 7 days a week.
NFR-6	Scalability	Supports various food items

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

Application Characteristics:

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
4.	Availability	Availability is increased by loads balancers in cloud VPS	IBM Cloud hosting
5.	Performance	The application is expected to handle up to 4000 predications per second	IBM Load Balance

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can Access my Dashboard		Medium	Sprint - 1
Customer (Webuser)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	Sprint -1
Customer Care Executive	Solution	USN-5	Responding to each email you receive can make a lasting impression on customers.	Offer a solution for how your company can improve the customer experience	High	Sprint-1
Administrator	Manage	USN-5	Do-it yourself service for delivery Everything	Set of predefined requirements that must be met to mark a user story complete	High	Sprint-1]

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection	USN-1	Download Food Nutrition Dataset	2	Medium
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low
Sprint-1		USN-3	Handling Missing Data	3	Medium
Sprint-1		USN-4	Feature Scaling	3	Low
Sprint-1		USN-5	Data Visualization	3	Medium
Sprint-1		USN-6	Splitting Data into Train and Test	4	High
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High
Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium
Sprint-2		USN-9	Initializing The Model	1	Medium

Sprint-2		USN-10	Adding LSTM Layers	2	High
Sprint-2		USN-11	Adding Output Layers	3	Medium
Sprint-2		USN-12	Configure The Learning Process	4	High
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-2		USN-13	Train The Model	2	Medium
Sprint-2		USN-14	Model Evaluation	1	Medium
Sprint-2		USN-15	Save The Model	2	Medium
Sprint-2		USN-16	Test The Model	3	High
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium
Sprint-3		USN-18	Build Python Code	4	High
Sprint-3		USN-19	Run The App in Local Browser	4	Medium
Sprint-3		USN-20	Showcasing Prediction On UI	4	High
Sprint-4	Train The Model On IBM	USN-21	Register For IBM Cloud	4	Medium
Sprint-4		USN-22	Train The ML Model On IBM	8	High
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	03 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17 Nov 2022

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

3.1 Feature 1

Data Collection:

Dataset:

<https://drive.google.com/drive/folders/1yNVuLA2hxIstOcDV58enyD74Y9drEs6Y?usp=sharing>

Image Processing:

```
In [7]: import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense Layer is the regular deeply connected neural network Layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional Layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator

In [ ]: #setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)

In [ ]: #performing data agumentation to train data
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')

Found 1055 images belonging to 5 classes.
Found 2626 images belonging to 5 classes.
```

Model Building:

Importing Neccessary Libraries

```
In [1]: import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense Layer is the regular deeply connected neural network Layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional Layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Image Data Agumentation

```
In [2]: #setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Loading our data and performing data agumentation

```
In [4]: #performing data agumentation to train data
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.

```
In [5]: print(x_train.class_indices)#checking the number of classes

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

```
In [6]: print(x_test.class_indices)#checking the number of classes

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

```
In [7]: from collections import Counter as c
c(x_train .labels)
```

Out[7]: Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})

Creating the model

```
In [8]: # Initializing the CNN
classifier = Sequential()

# First convolution Layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution Layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the Layers
classifier.add(Flatten())

# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2
```

```
In [9]: classifier.summary()#summary of our model
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 128)	802944
dense_1 (Dense)	(None, 5)	645
=====		
Total params: 813,733		
Trainable params: 813,733		
Non-trainable params: 0		

Compiling the model

```
In [10]: # Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Fitting the model

```
In [11]: classifier.fit_generator(
    generator=x_train, steps_per_epoch = len(x_train),
    epochs=10, validation_data=x_test, validation_steps = len(x_test)) # No of images in test set
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

This is separate from the ipykernel package so we can avoid doing imports until

```
Epoch 1/10
526/526 [=====] - 756s 1s/step - loss: 0.1651 - accuracy: 0.9459 - val_loss: 0.0253 - val_accuracy: 1.0000
Epoch 2/10
526/526 [=====] - 34s 64ms/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 0.1257 - val_accuracy: 0.9526
Epoch 3/10
526/526 [=====] - 31s 60ms/step - loss: 1.7961e-04 - accuracy: 1.0000 - val_loss: 0.0836 - val_accuracy: 0.9555
Epoch 4/10
526/526 [=====] - 34s 64ms/step - loss: 8.7034e-05 - accuracy: 1.0000 - val_loss: 0.0989 - val_accuracy: 0.9555
Epoch 5/10
526/526 [=====] - 32s 60ms/step - loss: 1.3759e-04 - accuracy: 1.0000 - val_loss: 0.1146 - val_accuracy: 0.9706
Epoch 6/10
526/526 [=====] - 34s 65ms/step - loss: 3.1752e-05 - accuracy: 1.0000 - val_loss: 0.0752 - val_accuracy: 0.9564
Epoch 7/10
526/526 [=====] - 32s 61ms/step - loss: 1.9019e-05 - accuracy: 1.0000 - val_loss: 0.0980 - val_accuracy: 0.9564
Epoch 8/10
526/526 [=====] - 34s 64ms/step - loss: 1.4092e-05 - accuracy: 1.0000 - val_loss: 0.1118 - val_accuracy: 0.9555
Epoch 9/10
526/526 [=====] - 32s 60ms/step - loss: 8.7305e-06 - accuracy: 1.0000 - val_loss: 0.1017 - val_accuracy: 0.9611
Epoch 10/10
526/526 [=====] - 32s 62ms/step - loss: 1.1566e-05 - accuracy: 1.0000 - val_loss: 0.1225 - val_accuracy: 0.9564
```

Out[11]:

```
In [35]: from google.colab import files
files.download("nutrition.h5")
```

Nutrition Image Analysis using CNN

Predicting our results

```
In [36]: from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
In [25]: img = image.load_img("/content/drive/MyDrive/Colab Notebooks/TRAIN_SET/BANANA/0_100.jpg",target_size= (64,64))#Loading of the image
img
```

Out[25]:



```
In [26]: x=image.img_to_array(img)#conversion image into array
```

In [27]:

x

```
Out[27]: 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.]]], dtype=float32)
```

```
In [28]: x.ndim
```

Out[28]: 3

```
In [29]: x=np.expand_dims(x,axis=0) #expand the dimension
```

```
In [30]: x.ndim
```

Out[30]: 4

In [31]: `pred = classifier.predict(x)`

1/1 [=====] - 0s 24ms/step

In [32]: `pred`


Out[32]: `array([[0., 1., 0., 0., 0.]], dtype=float32)`




In [33]: `labels=['APPLE', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']`
`labels[np.argmax(pred)]`








Out[33]: 'BANANA'




3.2 Feature 2



0.html:

 IBM-EPBL / IBM-Project-26935-1660040759 Public





 Notifications  Fork 0  Star 2

 Code  Issues  Pull requests  Actions  Projects  Security  Insights

 main  IBM-Project-26935-1660040759 / Project Development Phase / Sprint 3 / Fruit_analyzer / templates / 


 JANSIRANI-T Add files via upload 96a6409 yesterday  History



..

 0.html	Add files via upload	yesterday
 home.html	Add files via upload	yesterday
 image.html	Add files via upload	yesterday
 imageprediction.html	Add files via upload	yesterday

home.html:

 IBM-EPBL / IBM-Project-26935-1660040759 Public

 Notifications  Fork 0  Star 2


 Code  Issues  Pull requests  Actions  Projects  Security  Insights

 main  IBM-Project-26935-1660040759 / Project Development Phase / Sprint 3 / Fruit_analyzer / templates / 0.html  Go to file 

 JANSIRANI-T Add files via upload Latest commit 96a6409 yesterday  History


 1 contributor




25 lines (20 sloc) | 880 Bytes








 Raw  Blame   


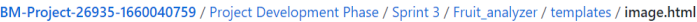


```
1
2 <html lang="en" dir="ltr">
3 <head>
4 <style>
5
6 </style>
7
8     <meta charset="utf-8">
9     <title style="color:whitesmoke">Nutrition Analyzer for Fitness Enthusiasts</title>
10    <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
11    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
12    <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
13    <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
14
15    <!-- Result -->
16    <div class="results">
```



Image.html:


 IBM-EPBL / IBM-Project-26935-1660040759 Public

 Notifications  Fork 0  Star 2






 Code  Issues  Pull requests  Actions  Projects  Security  Insights

 main  IBM-Project-26935-1660040759 / Project Development Phase / Sprint 3 / Fruit_analyzer / templates / image.html  Go to file 

 JANSIRANI-T Add files via upload Latest commit 96a6409 yesterday  History

 1 contributor

40 lines (28 sloc) | 1.09 KB

 Raw  Blame   

```
1
2
3
4 {% extends "imageprediction.html" %} {% block content %}
5 <div style="float:left">
6 <br>
7 <br>
8 <h5><font color="black" size="3" font-family="sans-serif"><b>Upload image to classify</b></font></h5><br><br>
9
10 <div>
11     <form id="upload-file" method="post" enctype="multipart/form-data">
12     <label for="imageUpload" class="upload-label">
13         Choose...
14     </label>
15     <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
16 </form>
```

Imageprediction.html

IBM-EPBL / IBM-Project-26935-1660040759 Public

Notifications

Fork 0

Star 2

<> Code Issues Pull requests Actions Projects Security Insights

main IBM-Project-26935-1660040759 / Project Development Phase / Sprint 3 / Fruit_analyzer / templates / imageprediction.html

Go to file

...

JANSIRANI-T Add files via upload

Latest commit 96a6409 yesterday History

1 contributor

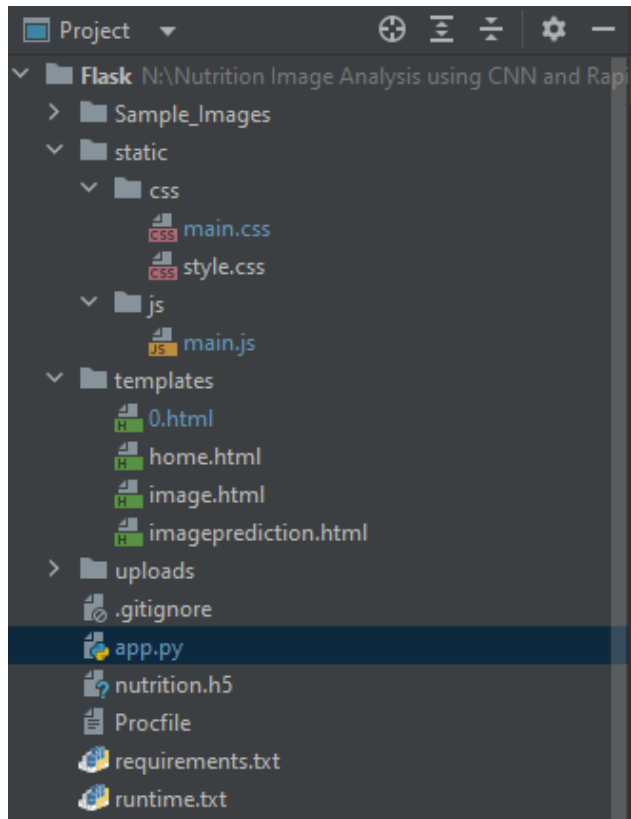
130 lines (116 sloc) 2.82 KB

Raw Blame

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Predict</title>
8   <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
10  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
11  <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
12  <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
13 <style>
14   body
15   {
16     background-image: url("https://i.pinimg.com/originals/be/21/1a/be211ad5043a8d05757a3538bdd8f450.jpg");
```

4. TESTING

4.1 Test Cases

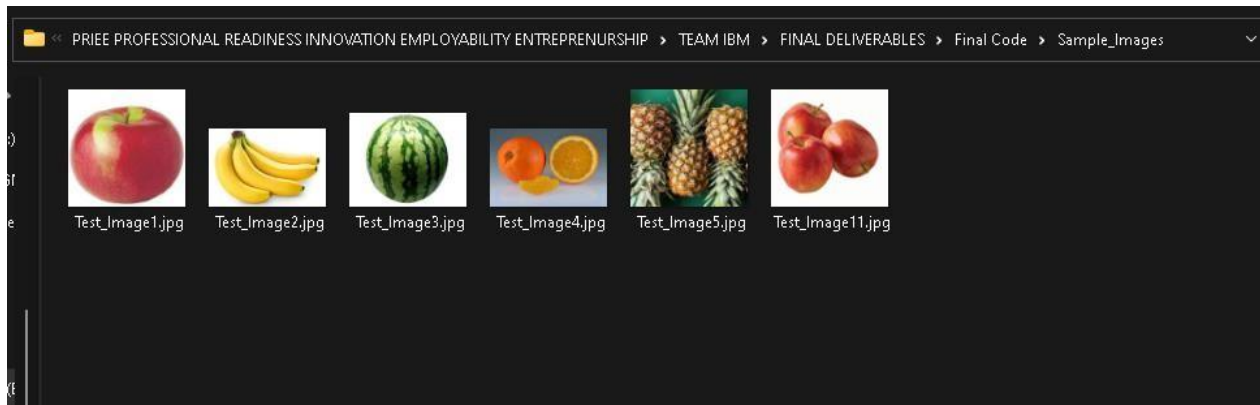




Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation (Y/N)	BUG ID	Executed By
Home Page	UI	Home Page	Verify the UI elements	1. Enter the local host url and click go. 2. Verify home page with below ui element	localhost/webapp/image.html	Application should show below UI elements: 1. Home button	Working as expected	PASS	Successful	Y		JANSI RANI & ABIRAMI
Classify Page Tc 001	Functional	PredictPage	Verify user is able to upload image	1. upload the image . 2. Click analyze button	Upload image	User should upload the image	Working as expected	PASS	Successful	Y		HARIPRIYA & JANSI

Classify Page Tc 002	Functional	PredictPage	Verify user is able to upload image	1. upload the image . 2. Click analyze button	Upload image	User should upload the image	Working as expected	PASS	Successful	Y		DEEPIKA & JANSI
Classify Page Tc 003	Functional	PredictPage	Verify user is able to upload image	1. upload the image . 2. Click analyze button	Upload image	User should upload the image	Working as expected	PASS	Successful	Y		JANSI & HARIPRIYA A
Classify Page Tc 004	Functional	PredictPage	Verify user is able to upload image	1. upload the image . 2. Click analyze button	Upload image	User should upload the image	Working as expected	PASS	Successful	Y		ABIRAMI & DEEPIKA

4.2 User Acceptance Testing



Purpose of User Acceptance Testing

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Severity 5	Subtotal
By Design	2	2	1	1	1	7
Duplicate	1	0	1	0	0	2
External	2	0	0	2	0	4

Fixed	3	2	1	1	0	7
Not Reproduced	0	0	1	1	0	2
Skipped	0	0	0	0	0	0
Won't Fix	0	0	0	0	0	0
Totals	8	4	4	5	1	22

Test Case Analysis

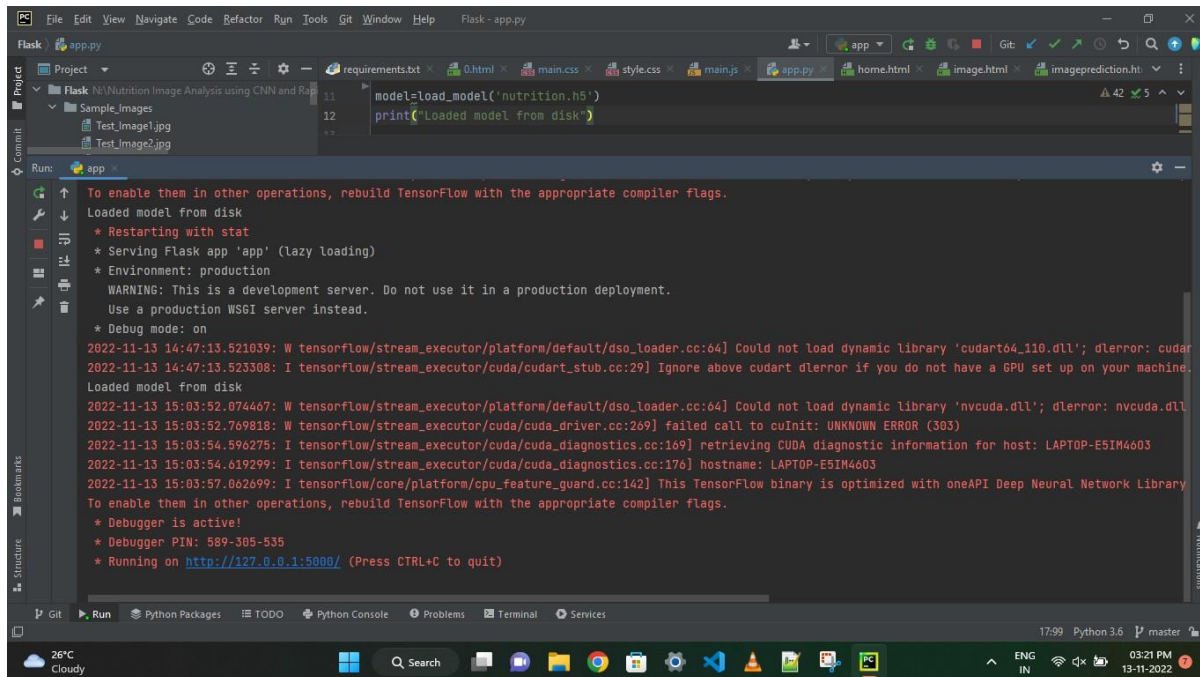
shows the number of test cases that have passed, failed, and untested

Section	Total cases	Not Tested	Fail	Pass
Home page	6	0	0	6
Image Page	5	0	0	5

Prediction Page	3	0	0	3
Report Page	3	0	0	3

5. RESULTS

5.1 Performance Metrics



```

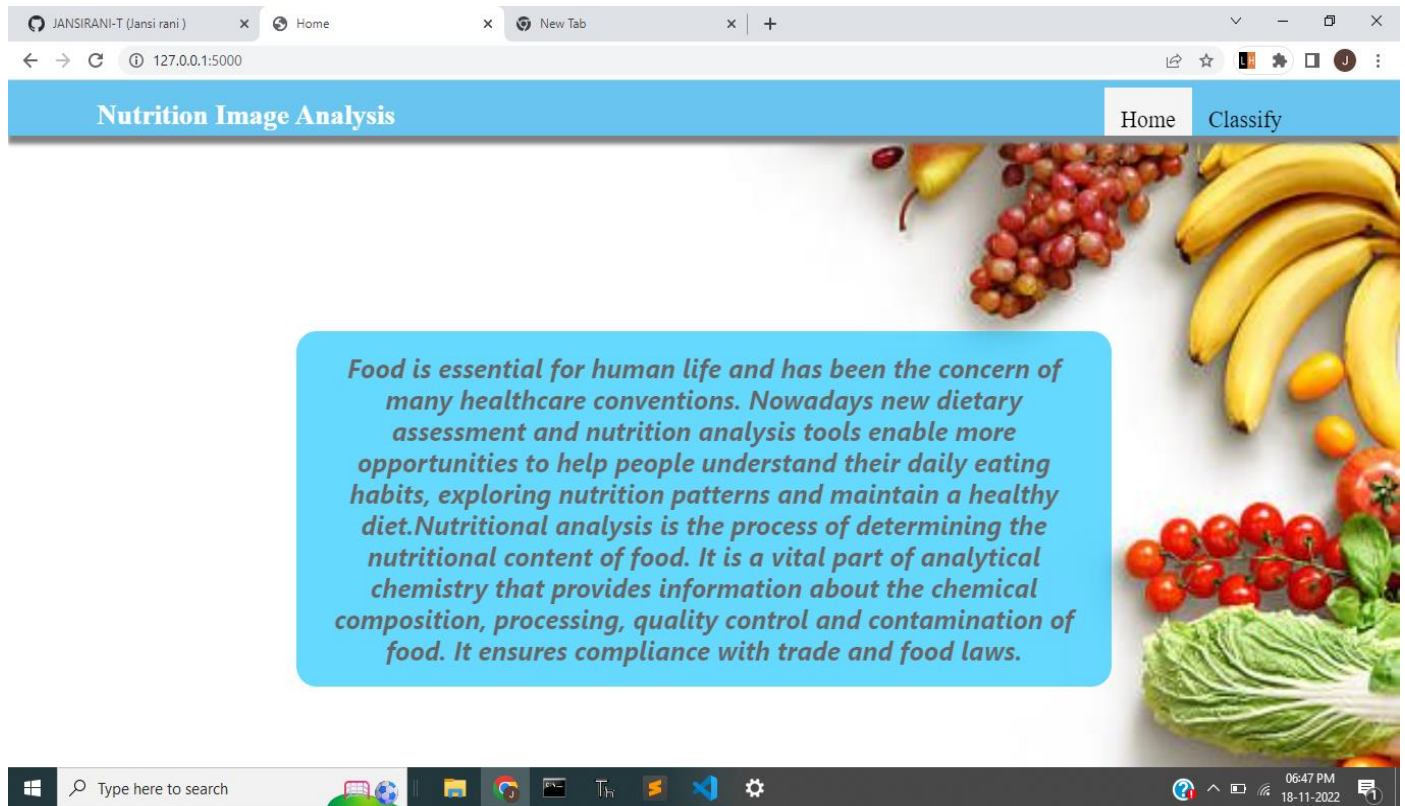
File Edit View Navigate Code Refactor Run Tools Git Window Help Flask - app.py
Project
  Flask Nutrition Image Analysis using CNN and RNN
  Sample_Images
    Test_Image1.jpg
    Test_Image2.jpg
  requirements.txt
  0.html
  main.css
  style.css
  main.js
  app.py
  home.html
  image.html
  imageprediction.ht

Run: app
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
* Restarting with stat
* 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: on
2022-11-13 14:47:13.521039: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dLError: cudart
2022-11-13 14:47:13.523308: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Loaded model from disk
2022-11-13 15:03:52.074467: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dLError: nvcuda.dll
2022-11-13 15:03:52.769818: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-13 15:03:54.596275: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-E5IM4603
2022-11-13 15:03:54.619299: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-E5IM4603
2022-11-13 15:03:57.062699: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 589-305-535
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

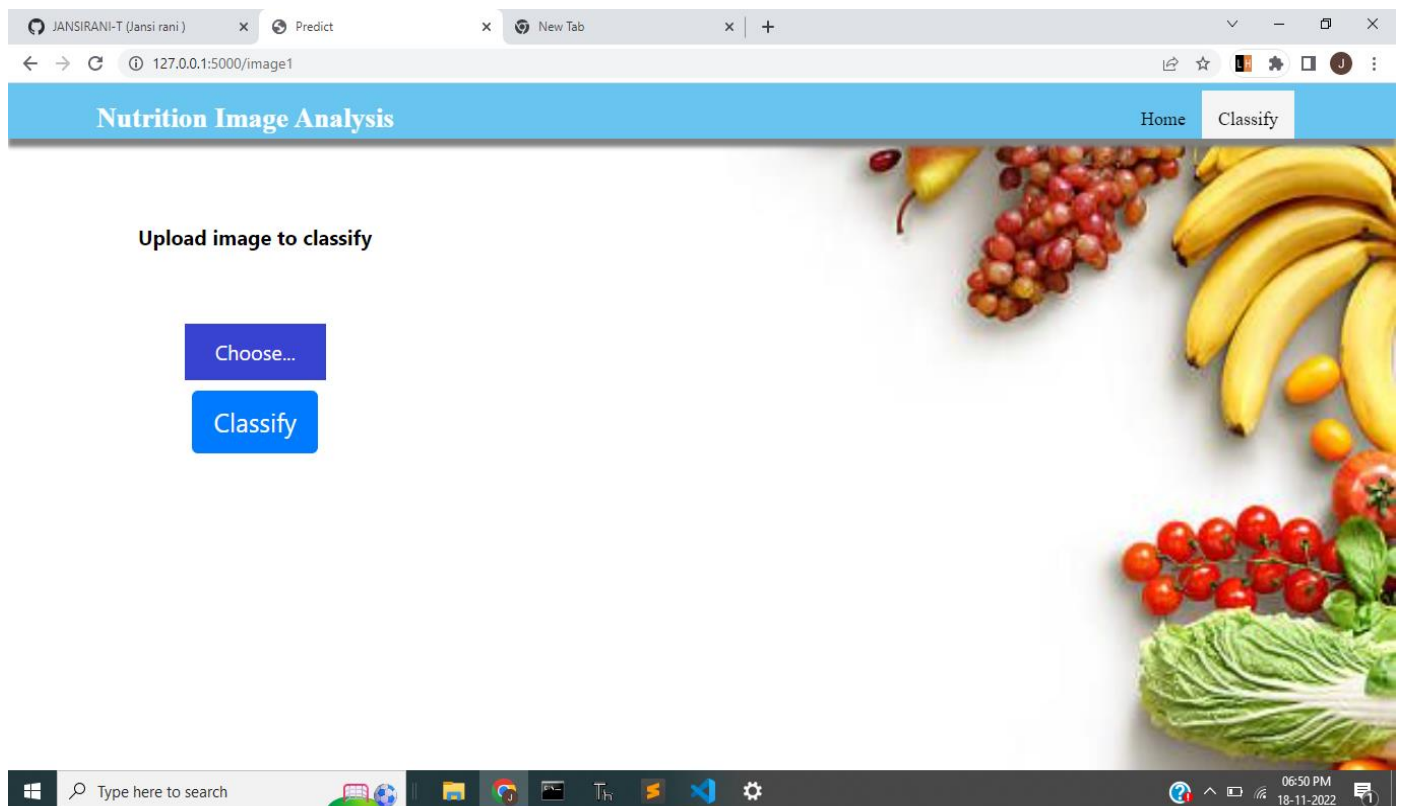
```

5.2 Output:

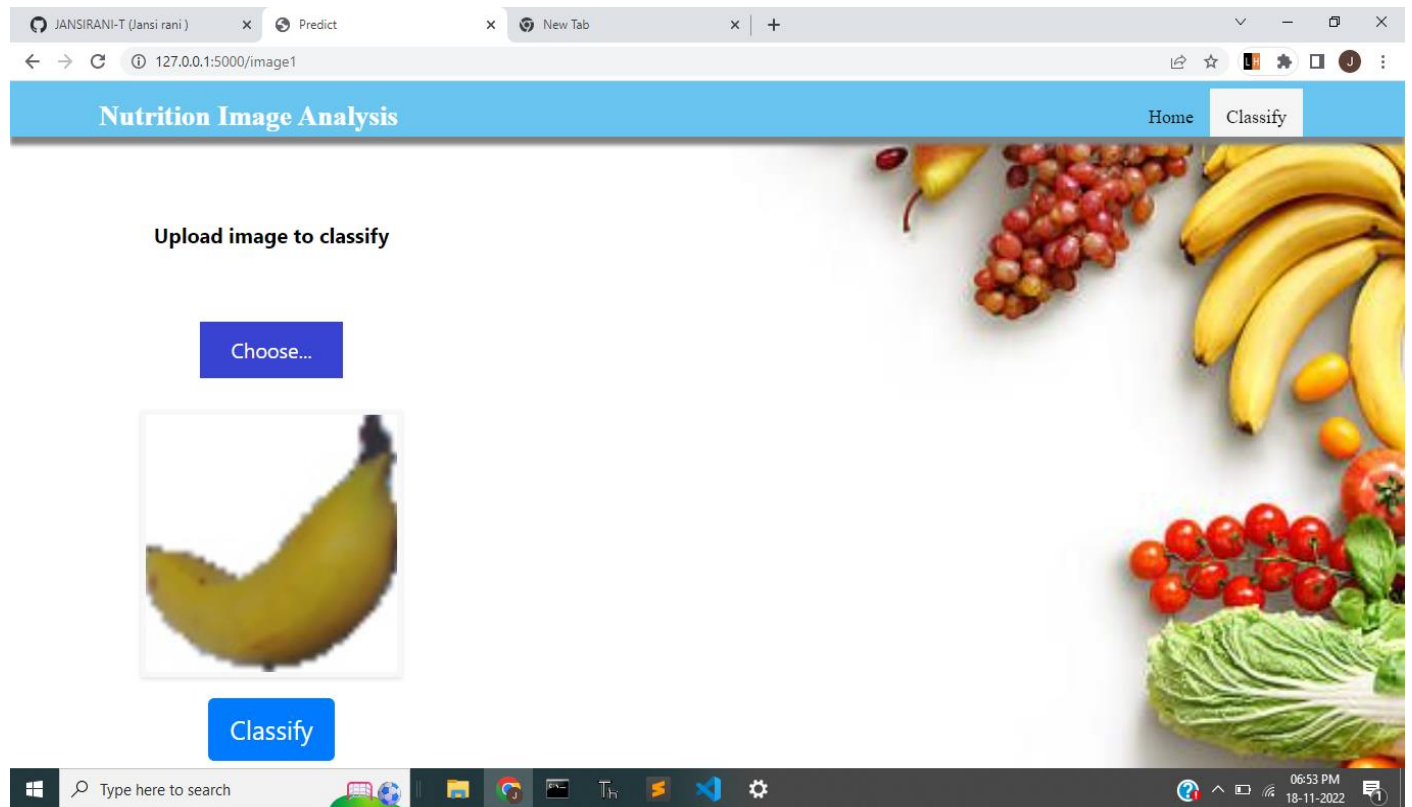
Home page:



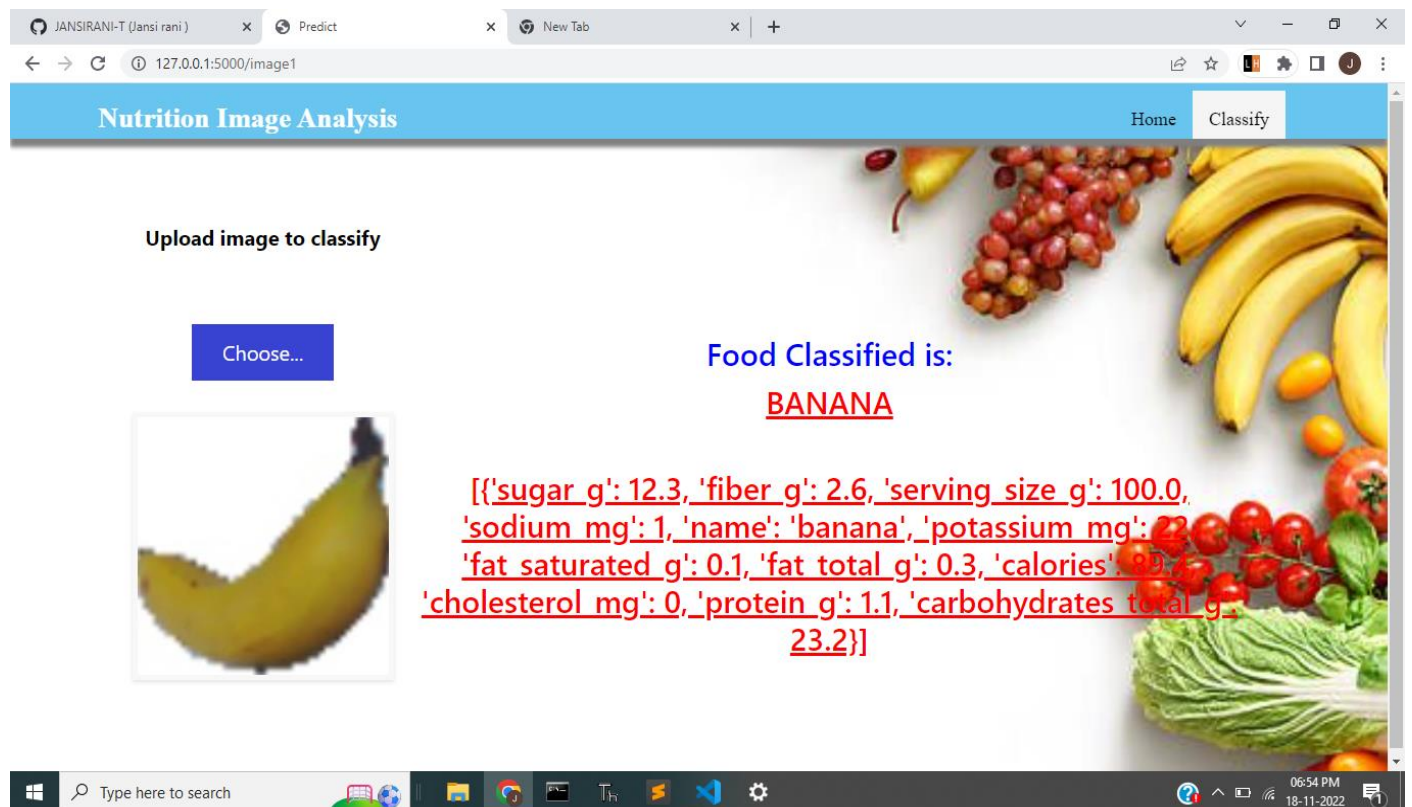
Classify:



Upload:



Output Screen:



6. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- The new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits
- It helps in exploring the nutrition patterns in their daily routines and this is very useful for people to maintain a healthy diet balances.
- The nutritional analysis is used to determine the nutritional content of food.
- This application eliminates the travelling cost in visiting a dietician.
- The usage of this application greatly reduces the time required to get the best diet plan

DISADVANTAGES

- The android mobile user will not be able to insert or view details if the server goes down.
- Thus, there is disadvantage of single point failure.

7. 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.

8. 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.

9. APPENDIX

GitHub - <https://github.com/IBM-EPBL/IBM-Project-26935-1660040759>

Demo link -

<https://drive.google.com/file/d/15iiwPocORFO2Tg2wrLH7u3P26JN4i7Xg/view?usp=drivesdk>