ST.JOSEPH COLLEGE OF ENGINEERING

(Affiliated to AICTE & ANNA UNIVERSITY)

Near Toll Plaza, Sriperumbudur. Chennai - 602117

PROJECT

AI-Powered Nutrition Analyser For Fitness Enthusiasts

TEAM ID: PNT2022TMID26694

Done By Team:

Lavanya Devi.M (212919205020)

Swarnalakshmi.R (212919205052)

Snekha.R (212919205047)

Saranya.M (212919205040)

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.

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 the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

1.2 Purpose

- Know fundamental concepts and techniques of Convolutional Neural Network.
- Gain a broad understanding of image data.
- Knowhow to pre-process/clean the data using different data pre-processing techniques.
- Know how to build a web application using the Flask framework.

2. LITERATURE SURVEY

2.1 Existing Problem

2.1.1 TITILE : Study for Food Recognition System Using Deep Learning

AUHTOR: Nareen O. M. Salim

The evaluation of weight reduction therapy has been shown in the literature to greatly benefit from accurate dietary assessment. The majority of contemporary nutritional evaluation techniques, however, rely on memory. Current computer-based food identification system development for accurate food evaluation is now possible via rich Cloud services and complete mobile devices. Addressing the issue of food detection and identification in images of various foods. The issue is worsened by the wide range of food products with low inter- and large intra-class differences and the scant information in a single image. By outlining the general application of numerous fusion-trained classifiers, it is possible to improve the identification and recognition of traits gleaned from different deep models. This essay investigated numerous methods for identifying foods. The Food identification is a challenging challenge since food products are presented; Sometimes, they are different within the same group. A sort of issue with categorizing fine-grained pictures as the identification of food pairwise local characteristics that take advantage of eight specific food ingredients' positional relationships. The proposed multi-food image recognition system that detects first food recognizes color, texture, gradient, and SIFT extracted by several detectors using multiple kernels learning regions. The food is divided into 300 blocks, and five classes are further classified, such as staple, main dish, side dish, fruit, and non-food from each block's extract color and DCT coefficients. Food identification and quantity estimation are part of the TADA dietary evaluation system.

2.1.2 TITILE : Efficient extraction of deep image features using convolution neural network (CNN) for applications in detecting and analysing complex food matrices

AUHTOR: Yao Liu , Hongbin Pu , Da-Wen Sun b

For the food sector, it is important to establish tools and processes for quickly and accurately identifying and analysing food quality and safety goods. traditional methods for machine learning Based on handcrafted traits, they typically perform poorly since they have a limited capacity to describe complex culinary properties. The convolutional neural network (CNN), which is the most widely used architecture of deep learning and has been increasingly used for the detection and analysis of complex food matrices, has recently emerged as an efficient and viable method for feature extraction. Goals and strategy: The current review introduces multi-feature aggregation techniques, 1-D, 2-D, and 3-D CNN models for feature extraction, and the structure of CNN. CNN's uses as a depth feature. With the improvement of the quality of life, people are increasingly conscious of high quality and safe food products in daily life, therefore the development of methods for reliably detecting and analysing food quality and safety is important for the industry. With the unique advantages of strong feature learning and good generalization ability, CNN is potential and attractive for effective and efficient analysis of complex food matrices. CNN can not only automatically locate important features, but can also obtain unparalleled performance under challenging conditions such as complex background, and different resolutions and orientations of the images. Despite the advantages of CNN in the provision of better performance, there still remain numerous challenges to its applications in the food domain.

2.1.3 TITILE : The Use of Different Image Recognition Techniques in Food Safety A Study

AUHTOR: Rijwan Khan, Santosh Kumar

In order to prevent foodborne illness and harm, food must be properly prepared, transported, and stored. Food products may encounter a variety of health risks from farm to factory and factory to fork. Food safety is therefore essential from a financial and moral standpoint. The consequences of not complying with food safety regulations are diverse.)e demand for precise, rapid, and neutral quality assessments of these qualities in food products is increasing as dietary requirements and high-quality standards are demanded more frequently. To accomplish these goals, computer vision offers an automated, non-destructive, and cost-effective method. Its usefulness for fruit and vegetable assessment and classification has been proven by a significant body of research. It highlights the key elements of image processing technology and provides an overview of the most recent developments in the food industry. Public health is consistently and significantly burdened by foodborne illnesses. After more than a century Large-scale changes in food production, distribution, and regulations were pushed and fed into macrosocial pressures like population growth, urbanisation, and globalisation . Compared to other economic sectors, the food industry and distribution network, in particular, have created huge amounts of data in recent years. To increase the safety of the food supply, several types of data were imaginatively examined at various points along the agricultural value chain . For instance, toxic contaminations on farmlands were forecasted in preharvest, field, and weather forecasts; in the retail setting, contactless audits and record-keeping were carried out for 1.4 million months; and observations of Hindawi Journal of Food Quality Volume

2.1.4 TITILE : Barriers to and Facilitators for Using Nutrition Apps: Systematic Review and Conceptual Framework

AUHTOR: Laura Maria König, Christiane Attig

Diet-related health risk factors and eating habits can both be modified with the help of nutrition apps. Although they might slow the rising rates of overweight and obesity, they haven't yet been widely adopted. Therefore, comprehensive understanding of The development of design recommendations targeted at promoting adoption and sustained use of nutrition apps requires an understanding of the factors encouraging and discouraging (long-term) app use. The literature on obstacles to and enablers of the use of nutrition apps across disciplines, as well as empirical qualitative and quantitative studies with current, former, and non-users of nutrition apps, has been synthesised in this systematic review. PsychINFO, PSYNDEX, PsycArticles, PubMed, Web of Science, and SPORTDiscus were among the six databases used in a comprehensive literature search that also included backward and forward citation searches. The anticipated data extraction procedure, the inclusion and exclusion criteria, and the search strategy were all registered in advance. All empirical qualitative and quantitative publications in German or English that focused on adolescents (aged 13–18) or adults who were either present, former, or non-users of nutrition apps were eligible for inclusion. Individual barriers and facilitators were extracted and put into categories based on a qualitative content analysis. Multiple factors influencing participation with mobile weight reduction and weight maintenance therapies have been discovered by two systematic evaluations. These elements include social support, customisation, ease of use, entertainment, and the availability of tools like self-monitoring, prompts, and feedback.

2.1.5 TITILE : Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning

AUHTOR: Dhanamjayulu C, Nizhal U N

The usable information on human faces can be used to determine an individual's age, gender, weight, etc. Body mass index (BMI) and weight are two of these biometrics that are reliable predictors of health. Based on recent health science studies, this work proposes a regression approach based on the 50-layer Residual network architecture to investigate ways to identify malnourished individuals and obese individuals by evaluating body weight and BMI from facial photos. Multi-task Cascaded Convolutional Neural Networks have been used for face detection. A method is developed to assess BMI, age, and gender using real-time photographs of human faces. Obesity and malnutrition are frequently identified with the aid of BMI. The estimation of height, weight, and BMI using automated methods was done in earlier publications. Today's social networks, like Facebook, Instagram, and Snapchat, contain a variety of functions, including the trading of images, looking for a job, dating, and blogging. More and more people around the world are capturing their lives with digital cameras and publishing the records as images or videos on social media networks. The proposed method is useful in establishing the relation between the characteristics of the human face and the body, such as body height and weight. The proposed method to identify Malnutrition and obese children from human faces. The proposed system does not require the full body real image of a person. Face detection is done with the Multi-task Cascaded Convolution Neural Networks on pictures with single/multiple faces. BMI, age, and gender are estimated from a person’s face using residual neural networks. The problems of BMI, age, and gender estimation are posed as three separate regression pattern classification problems. The dataset of facial images taken from the internet along with their metadata containing information like gender, age, and BMI.

2.2 REFERENCES

1. Nareen O. M. Salim, Study for Food Recognition System Using Deep Learning,2021
2. Yao Liu , Hongbin Pu, Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices, 2021
3. Rijwan Khan, Santosh Kumar, The Use of Different Image Recognition Techniques in Food Safety: A Study, 2021
4. Laura Maria König, Barriers to and Facilitators for Using Nutrition Apps: Systematic Review and Conceptual Framework, 2021
5. Dhanamjayulu C, Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning, 2021

2.3 PROBLEM STATEMENT DEFINITION

Food is a necessity for human life and has been addressed in numerous medical conventions. Modern dietary evaluation and nutrition analysis technologies give consumers more possibilities to explore nutrition patterns, comprehend their daily eating habits, and keep up a balanced diet.

The biggest challenge for fitness lovers is keeping track of their daily nutrition intake, which is crucial for staying in shape. But with today's busy world and the abundance of internet fitness resources, keeping track of your nutrition will become increasingly difficult and inaccurate. Fitness fanatics typically stick to their diet programmers, but they have trouble keeping track of the food's nutritional value.

Fruits are easily digestible since they are high in vitamins, fiber, and minerals, but eating too much of them can cause weight gain and even diabetes because fruit contains natural sugar.

Fitness aficionados eat a diet high in fruits, vegetables, foods high in protein, and low in carbohydrates. However, it is difficult to identify and keep track of the nutritional components of unknown foods, such as fiber, protein, and nutrition.

I am (USER)

User has to upload the food (fruits and vegetables) image to know the healthy content.

I am Trying To

Instead of waiting for a diet expert, users may acquire dietary specifics through this application.

But

This might be the result of a human error, such as a lack of quality control, poor customer service, or even a lack to provide healthy suggestions.

Because

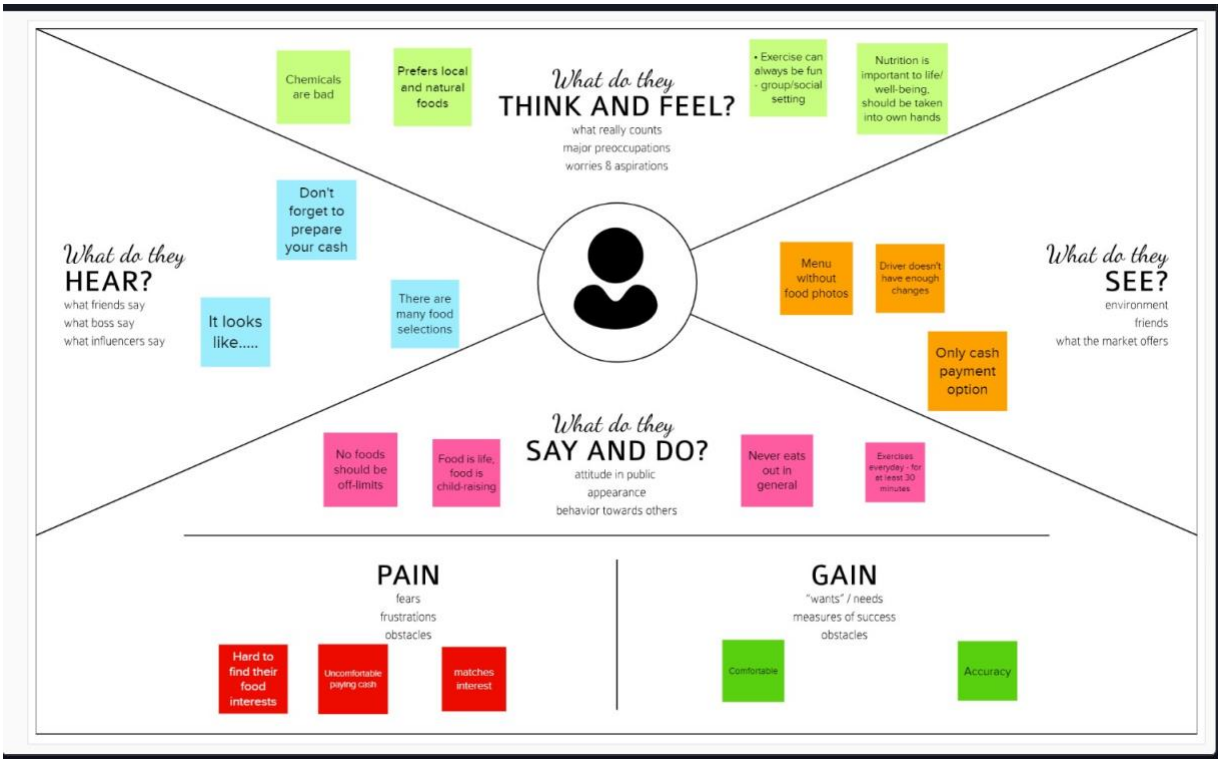
It is hard, and there is a delay to know about the food details and also awkward for providing our healthy facts.

Which makes me feel?

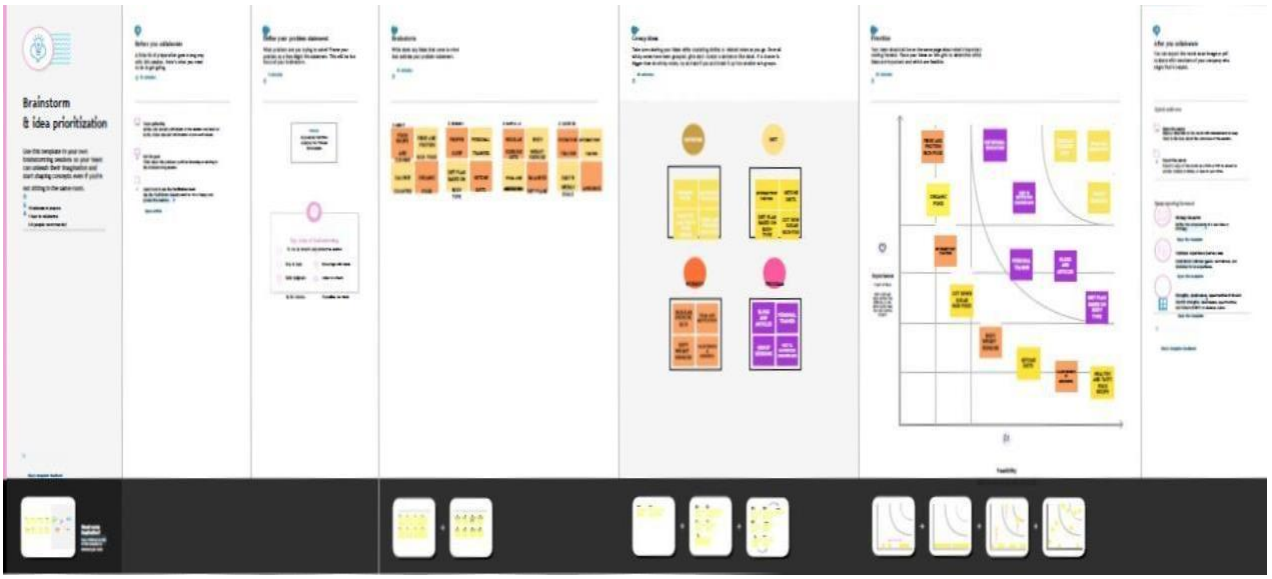
Deep learning algorithms may assist to address these challenges by automating nutrition content assessment. Finally, by analysing the nutritional components in the images, compute the calories, fat, carbs, and protein amounts to give a dietary evaluation report. The addition of more food kinds to the dataset will increase the system's efficiency and precision.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation &Brainstorming



3.3 Proposed Solution

To recognise and find food items from the given photographs, develop an automated nutrition analysis system for the proposed system. By locating promising locations and classifying them with deep neural networks, you may create a three-step process only for detecting various cuisines in photographs. From the provided photos, the automatic algorithm initially generates a large number of suggestion regions. Then, it aggregates each region of ideas by placing them on feature maps, categorising them into different food groups, and identifying their locations in the original photos. Finally, by analysing the nutritional elements in the photographs, determine the quantities of calories, fat, carbs, and protein to generate a dietary evaluation report. The system's effectiveness and accuracy will also be increased by expanding the dataset to cover a larger variety of food kinds.

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Due to the uncountable food items as well as the inefficiency of details, usually hard to recognize which food is healthy
2.	Idea / Solution description	Digitalized process might be useful to users and fitness people analyze the nutrition for healthy and diet plan

3.	Novelty / Uniqueness	Provide sustenance such as protein, fat, carbs, vitamin, calories etc.,
4.	Social Impact / Customer Satisfaction	Accurate findings increase people's satisfaction and health.
5.	Business Model (Revenue Model)	-
6.	Scalability of the Solution	Capable of detecting and providing food contents

3.4 Problem Solution Fit

1.CUSTOMER SEGMENTS User has to upload the food (fruits and vegetables) image to know the nutrition data	**5. AVAILABLE SOLUTIONS** Helps to know the facts of food habits and health	**8.CHANNELS OF BEHAVIOUR** Users should be able to interact with the recommended system and obtain information both online and offline.
2.JOBS TO BE DONE / PROBLEM Ineffectual to get the details systematically	**6.CUSTOMER CONSTRAINTS** Takes more time to get the information	**9.PROBLEM ROOT CAUSE** There isn't a systematic approach to gather dietary information rapidly. One must wait hours to visit a diet specialist.
3.TRIGGERS Help to fitness people to analyze and to know the food calories and so on	**7.BEHAVIOUR** The digitalized nutrition assistant makes it simpler for people to obtain information.	**10 YOUR SOLUTION** Analyze the nutritional elements in the images, and then compute the calories, fat, carbs, and protein levels to give a dietary evaluation report. The system's effectiveness and accuracy will also be increased by expanding the dataset to cover a larger variety of food kinds.
4. EMOTIONS: BEFORE / AFTER Before, waiting for a diet expert took a lot of time. After, getting aware of health foods just image.		

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements

Upload Image

In this module, upload the nutrition datasets in the form of CSV file format. In addition, the data is saved in a database for future use. Fruits and vegetables calorie, protein, fat, carbohydrate, vitamin, and cholesterol values are included in the dataset. These values are taken from the Kaggle website and saved as integer values.

Filtering Noise

Filter techniques are used to remove noise in images in order to evaluate nutrients based on the fruits or vegetables. The filter's objective is to remove noise from photos. It is supported by a statistical methodology. The usual frequency response of a filter is built. Filtering is a nonlinear image processing technique used to minimise "salt and pepper" noise. When edge preservation and noise reduction are concerns, a median filter is superior to convolution.

Classification

The food image uploaded from the user end will be compared with the food items in the system database for the features obtained in the feature extraction step. The specific food item will be recognised when the perfect match is obtained based on the attributes matched. The name of the detected food item and the nutrition details will be displayed over the food.

Nutrition Detection

The request for an insurance claim can be viewed and approved by the insurance company. Once the damaged image has been uploaded and the degree of the damage has been determined, the user may receive insurance only if the firm accepts the damaged image and the condition is greater than 80%.

Non - Functional Requirements

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

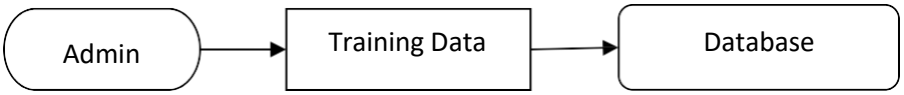
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

5. PROJECT DESIGN

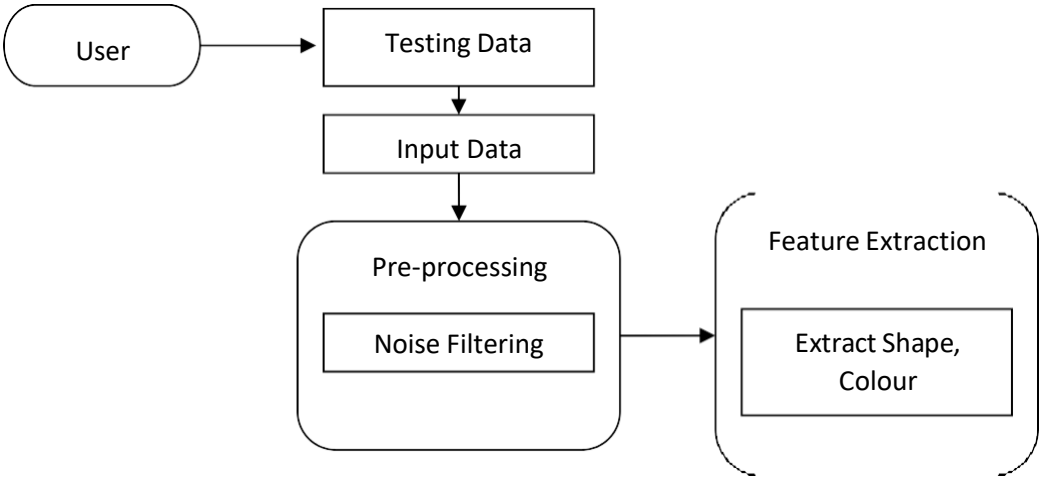
5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information and where data is stored.

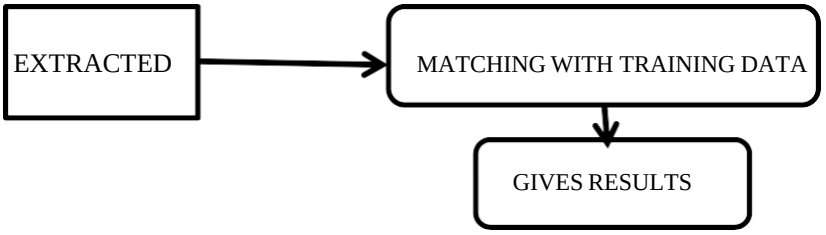
LEVEL 1



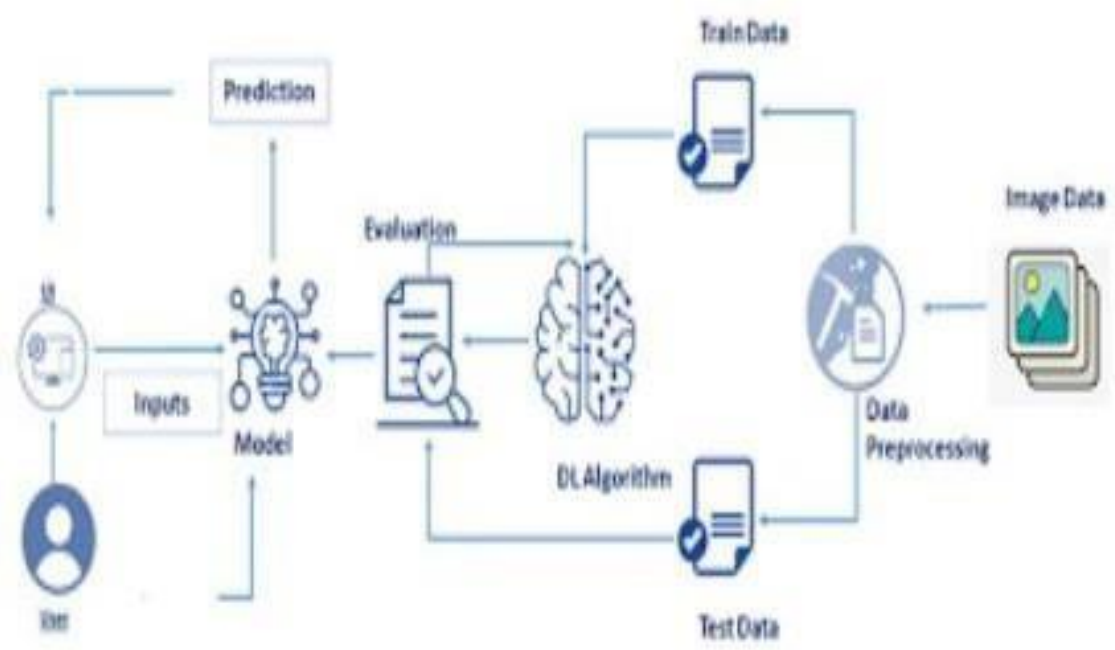
LEVEL 2



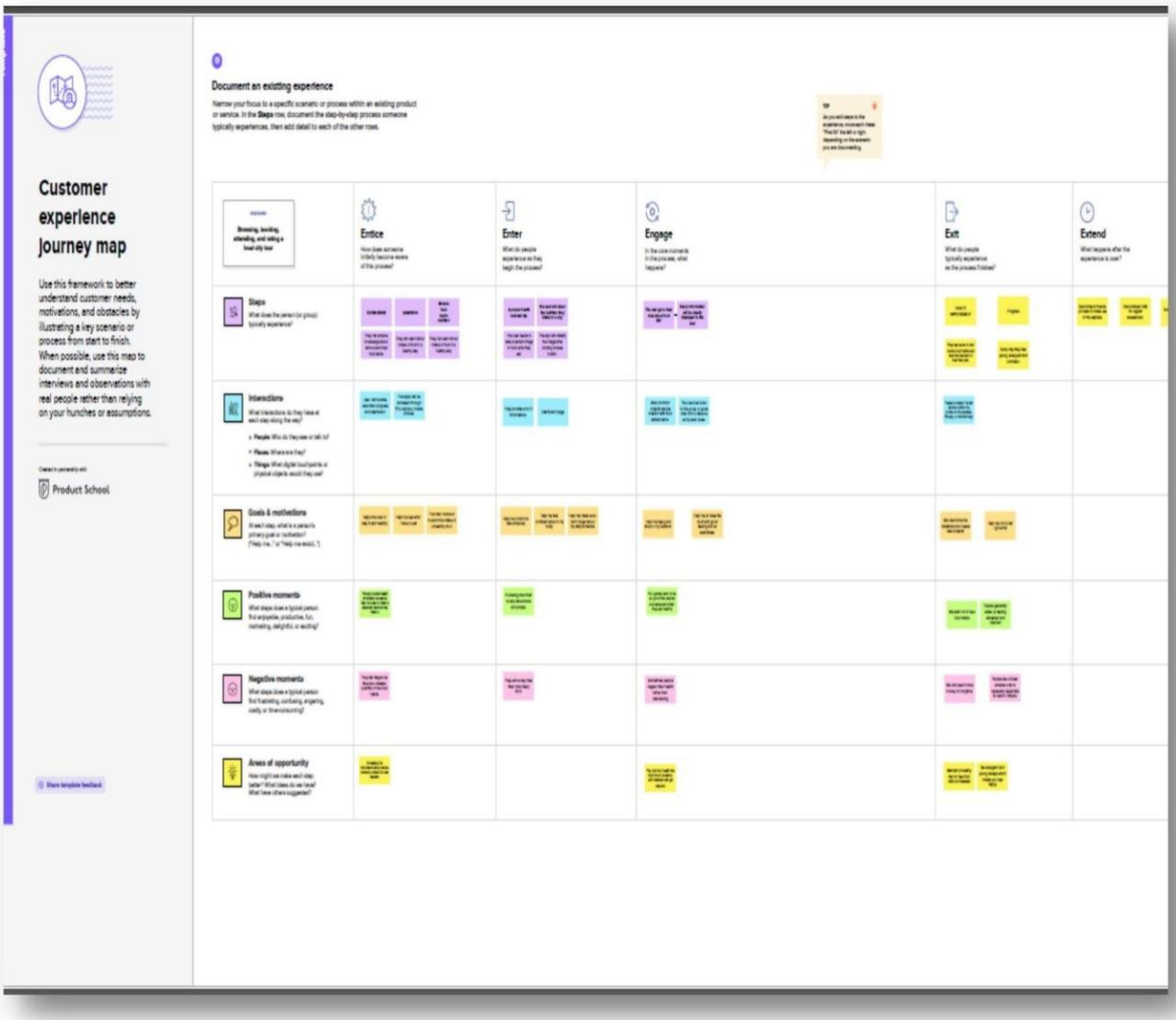
LEVEL 3



5.2 Solution & Technical Architecture



5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

PRODUCT BACKLOG, SPRINT SCHEDULE, AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Dataset - Collecting images of food itemsapples , banana, orange, pineapple, watermelon for analysis	5	High	Sneka R
Sprint-1	Image Preprocessing	USN-2	Image data augmentation - Increasing theamount of data by generating new data points from existing data	4	Medium	Saranya M
Sprint-1		USN-3	Image Data Generator Class - Used for getting the input of the original data	4	Medium	Lavanya Devi
Sprint-1		USN-4	Applying image data generator functionality to train set and test set	4	Medium	Saranya M
Sprint-2	Modeling Phase	USN-5	Defining the model architecture - Buildingthe model using deep learning approach and adding CNN layers	4	High	Sneka R
Sprint-2		USN -6	Training , saving, testing and Initializing the model	5	High	Sneka R
Sprint-2		USN- 7	Adding Dense layer of model	4	High	Sneka R

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Development phase	USN- 8	Project development-delivery of sprint1	3	Medium	Saranya
Sprint-2	Development phase	USN- 9	Project development-delivery of sprint2	2	Low	Lavanya Devi
Sprint-2	Development phase	USN-10	Project development-delivery of sprint 3	2	Low	Lavanya Devi M
Sprint-3	Development phase	USN-11	Project development-delivery of sprint4	2	Low	Swarnalakshmi
Sprint-4	Model on IBM	USN-12	Cloud deployment – Deployment of application by using Register for IBM cloud	4	High	Sneka R
Sprint-4	Testing Phase	USN-13	Functional testing – Checking usability and accessibility	**3**	Medium	Saranya M
Sprint-4	Testing Phase	USN-14	Non Functional testing – Checking scalability and performance of the application	**3**	Medium	Saranya M

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	08	5 Days	29 Oct 2022	02 Nov 2022	20	3 Nov 2022
Sprint-2	15	5 Days	03 Oct 2022	07 Nov 2022	20	8 Nov 2022
Sprint-3	15	5 Days	08 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	25	5 Days	13 Nov 2022	17 Nov 2022	20	16 Nov 2022

Velocity:

Average Velocity= 12/4 =3

7. CODING & SOLUTIONING

7.1 Features 1

```
from flask import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
import warnings
import  datetime
import cv2
app = Flask(_name_)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
return render_template('index.html')
@app.route("/Test")
def Test():
return render_template('NewUser.html')
@app.route("/testimage", methods=['GET', 'POST'])
def testimage():
if request.method == 'POST':
file = request.files['fileupload']
file.save('static/Out/Test.jpg')
img = cv2.imread('static/Out/Test.jpg')
if img is None:
print('no data')
img1 = cv2.imread('static/Out/Test.jpg')
print(img.shape)
```

```
    img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5)))
original = img.copy()
neworiginal = img.copy()
cv2.imshow('original', img1)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
img1S = cv2.resize(img1, (960, 540))
cv2.imshow('Original image', img1S)
grayS = cv2.resize(gray, (960, 540))
cv2.imshow('Gray image', grayS)
gry = 'static/Out/gry.jpg'
cv2.imwrite(gry, grayS)
from PIL import ImageOps,Image
im = Image.open(file)
im_invert = ImageOps.invert(im)
inv = 'static/Out/inv.jpg'
im_invert.save(inv, quality=95)
dst = cv2.fastNlMeansDenoisingColored(img1, None, 10, 10, 7, 21)
cv2.imshow("Nosie Removal", dst)
noi = 'static/Out/noi.jpg'
cv2.imwrite(noi, dst)
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
classifierLoad = tf.keras.models.load_model('model.h5')
import numpy as np
from keras.preprocessing import image
```

```
test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
img1 = cv2.imread('static/Out/Test.jpg')
# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)
print(result)
out = ''
fer = ''
if result[0][0] == 1:
out = "APPLES"
fer = '52 calories/1per' \
'Potassium 107mg' \
'Sodium 1mg' \
'Sugar 10g '
elif result[0][1] == 1:
out = "BANANA"
fer = '100 Calories/1per'
elif result[0][2] == 1:
out = "ORANGE"
fer = '50 calories/1per'
elif result[0][3] == 1:
out = "PINEAPPLE"
fer = '60 calories/1per'
elif result[0][4] == 1:
out = "WATERMELON"
fer = '400 calories/1per ' \
```

```
'99% of Water'
org = 'static/Out/Test.jpg'
return render_template('NewUser.html',fer=fer,result=out,org=org)
if__name__== '_main_':
    app.run(debug=True, use_reloader=True
```

7.2 FEATURE 2

```
# Part 1 - Building the CNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.models import model_from_json
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
batch_size = 32

from tensorflow.keras.preprocessing.image import ImageDataGenerator
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)
# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
'Data', # This is the source directory for training images
target_size=(200, 200), # All images will be resized to 200 x 200
batch_size=batch_size,
# Specify the classes explicitly
classes = ['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON'],
# Since we use categorical_crossentropy loss, we need categorical labels
class_mode='categorical')
```

```
import tensorflow as tf
model = tf.keras.models.Sequential([
# Note the input shape is the desired size of the image 200x 200 with 3 bytes color
# The first convolution
tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200, 3)),
tf.keras.layers.MaxPooling2D(2, 2),
# The second convolution
tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# The third  convolution
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# The fourth convolution
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# The fifth convolution
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
# Flatten the results to feed into a dense layer
tf.keras.layers.Flatten(),
# 128 neuron in the fully-connected layer
tf.keras.layers.Dense(128, activation='relu'),
# 5 output neurons for 5 classes with the softmax activation
tf.keras.layers.Dense(5, activation='softmax')
])
```

```
model.summary()
from tensorflow.keras.optimizers import RMSprop
early = tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=5)
model.compile(loss='categorical_crossentropy',
optimizer=RMSprop(lr=0.001),
metrics=['accuracy'])
total_sample=train_generator.n
n_epochs = 10
history = model.fit_generator(
train_generator,
steps_per_epoch=int(total_sample/batch_size),
epochs=n_epochs,
verbose=1)
model.save('model.h5')
acc = history.history['accuracy']
loss = history.history['loss']
epochs = range(1, len(acc) + 1)
# Train and validation accuracy
plt.plot(epochs, acc, 'b', label=' accurarcy')
plt.title('  accurarcy')
plt.legend()
plt.figure()
# Train and validation loss
plt.plot(epochs, loss, 'b', label=' loss')
plt.title('  loss')
plt.legend() plt.show()
```

8. TESTING

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behaviour of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

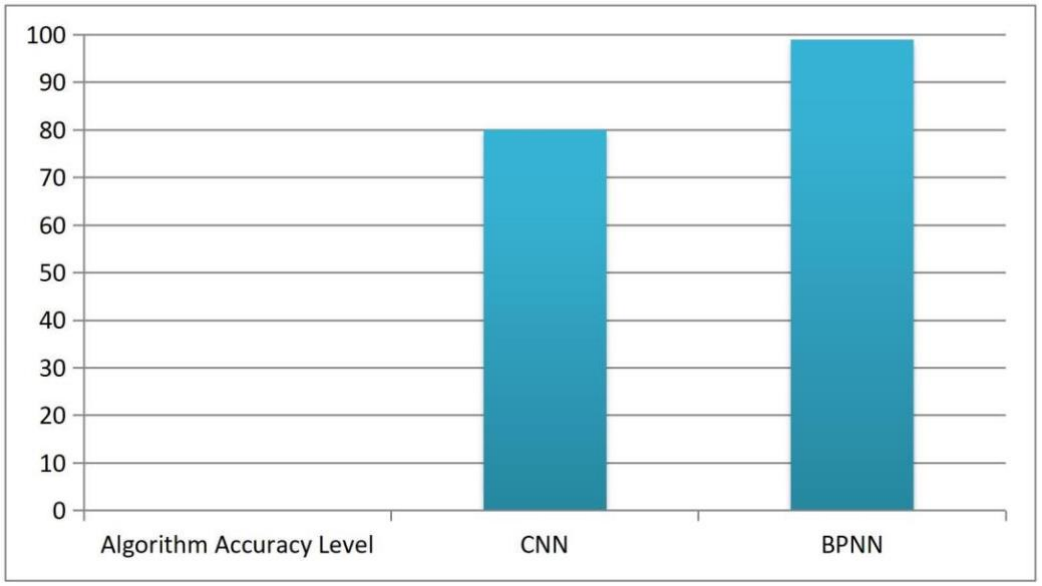
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload input image (fruits and vegetables)	Predicting calorie, fat, carbs and food content of given image	Details are stored in a database.

8.2 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULTS

9.1 PERFORMANCE METRICS



10. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Provide the nutrition content of Multifoods
- Helps for fitness people to maintain and know the proteins and calories of the food
- Gives accurate results in real-time application

DISADVANTAGE

- Hard to know the details of nutrition and calories of food
- Doesn’t ask to provide the users health condition
- Required more time to know the Multifoods

11. CONCLUSION

The approach for an automated food nutrition detection system that can determine the amount of nutrients in food is proposed in this project work. The machine has so far been able to place the meal into one of the many categories listed in the dataset. The well-known food dataset was used for the categorization. The classification of the food photos into their appropriate classifications using a deep learning approach. By reducing noise from the dataset, the classification process may be made better. The same research may be done with a larger dataset, more classes, and more photos in each class since a larger dataset increases accuracy by teaching the algorithm additional features and lowers the loss rate. The model's weights may be saved and used to create designs for food categorization, calorie extraction, and picture classification.

12. FUTURE SCOPE

The food photographs in this research study are categorised into the appropriate groups using a deep learning approach. In terms of future improvement, the classification task may be made better by reducing noise from the dataset. The same research may be done with a larger dataset, more classes, and more photos in each class since a larger dataset increases accuracy by teaching the algorithm additional features and lowers the loss rate. The model's weights may be saved and utilised to create a web or mobile application that classifies images and also extracts the calories from the food that has been identified.

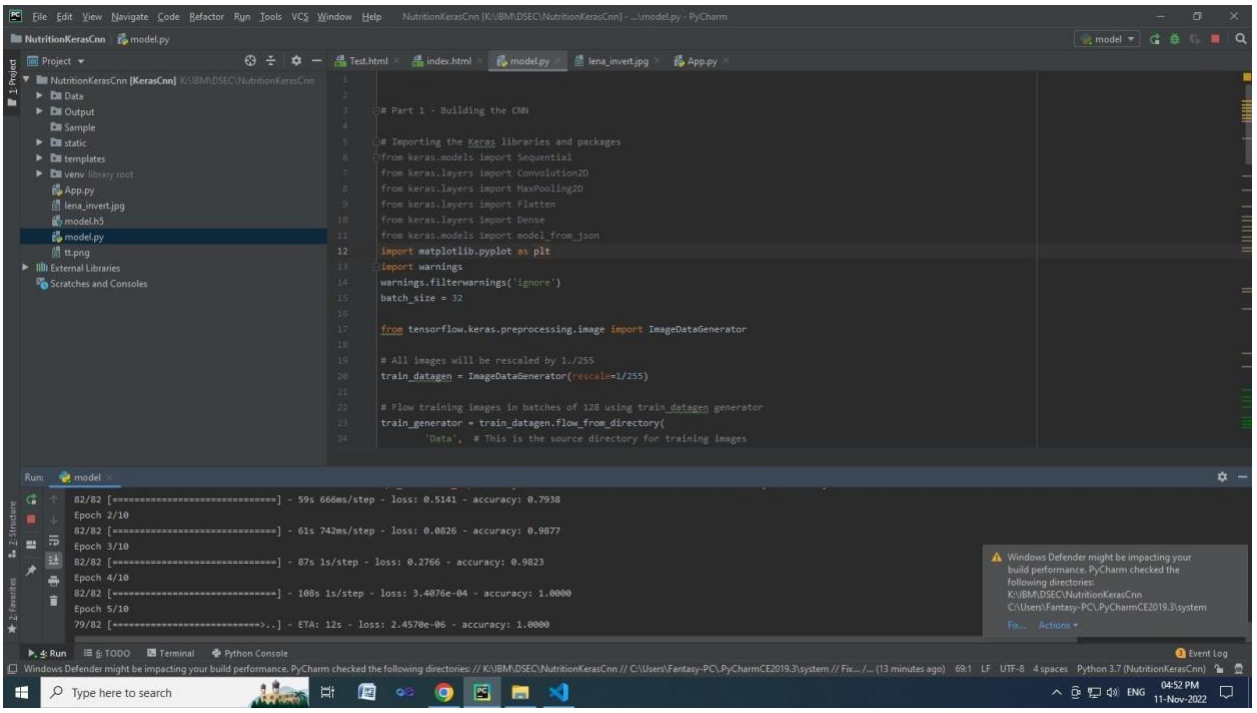
1. APPENDIX

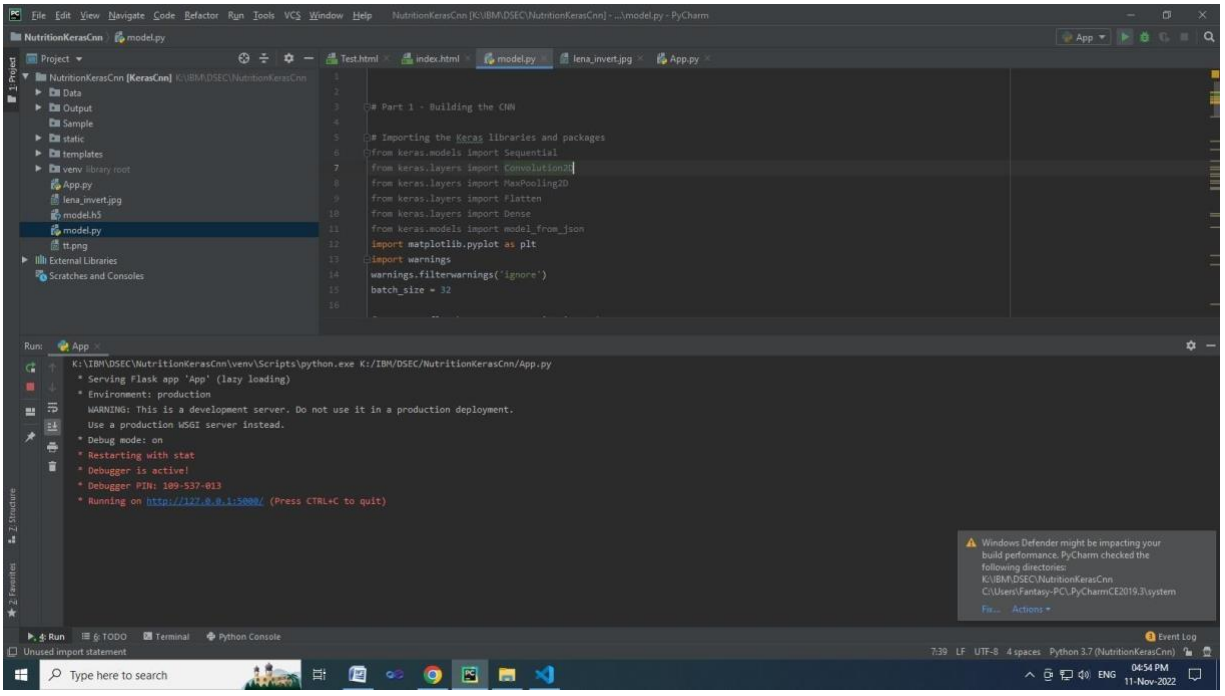
SOURCE CODE

```
from flask import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
import warnings
import  datetime
import cv2
app = Flask(_name_)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
return render_template('index.html')
@app.route("/Test")
def Test():
return render_template('NewUser.html')
@app.route("/testimage", methods=['GET', 'POST'])
def testimage():
if request.method == 'POST':
file = request.files['fileupload']
file.save('static/Out/Test.jpg')

img = cv2.imread('static/Out/Test.jpg')
if img is None:
print('no data')

img1 = cv2.imread('static/Out/Test.jpg')
print(img.shape)
```

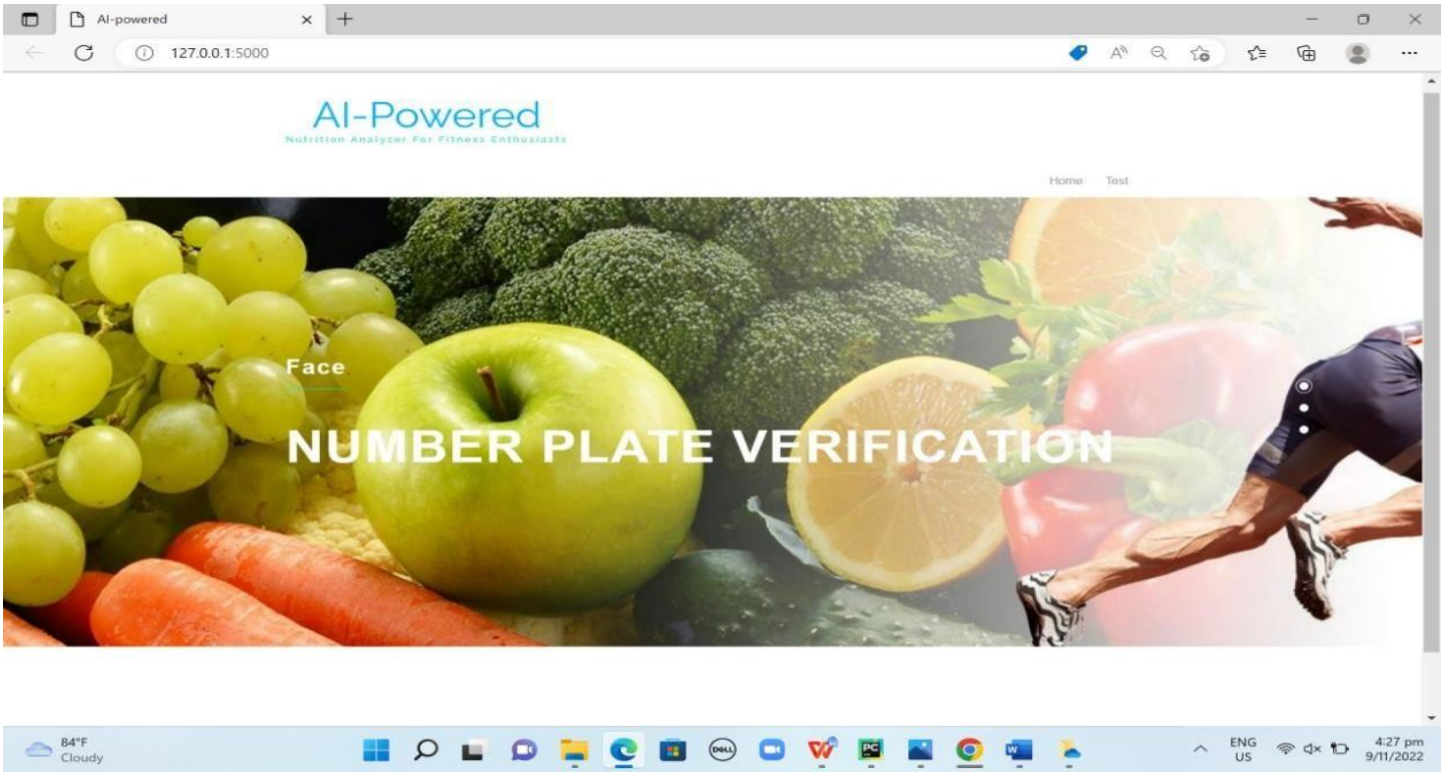
```
img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5)))
original = img.copy()
neworiginal = img.copy()
cv2.imshow('original', img1)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
img1S = cv2.resize(img1, (960, 540))
cv2.imshow('Original image', img1S)
grayS = cv2.resize(gray, (960, 540))
cv2.imshow('Gray image', grayS)
gry = 'static/Out/gry.jpg'
cv2.imwrite(gry, grayS)
from PIL import ImageOps,Image
im = Image.open(file)
im_invert = ImageOps.invert(im)
inv = 'static/Out/inv.jpg'
im_invert.save(inv, quality=95)
dst = cv2.fastNlMeansDenoisingColored(img1, None, 10, 10, 7, 21)
cv2.imshow("Nosie Removal", dst)
noi = 'static/Out/noi.jpg'
cv2.imwrite(noi, dst)
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
classifierLoad = tf.keras.models.load_model('model.h5')
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
img1 = cv2.imread('static/Out/Test.jpg')
# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)
print(result)
out = ''
fer = ''
```

```
 if result[0][0] == 1:
 out = "APPLES"
 fer = '52 calories/1per'
 elif result[0][1] == 1:
 out = "BANANA"
 fer = '100 calories/1per'
 elif result[0][2] == 1:
 out = "ORANGE"
 fer = '50 calories/1per'
 elif result[0][3] == 1:
 out = "PINEAPPLE"
 fer = '60 calories/1per'
 elif result[0][4] == 1:
 out = "WATERMELON"
 fer = '400 calories/1per'
 org = 'static/Out/Test.jpg'
 return render_template('NewUser.html',fer=fer,result=out,org=org)
 if__name__== '_main_':
app.run(debug=True, use_reloader=True)
```

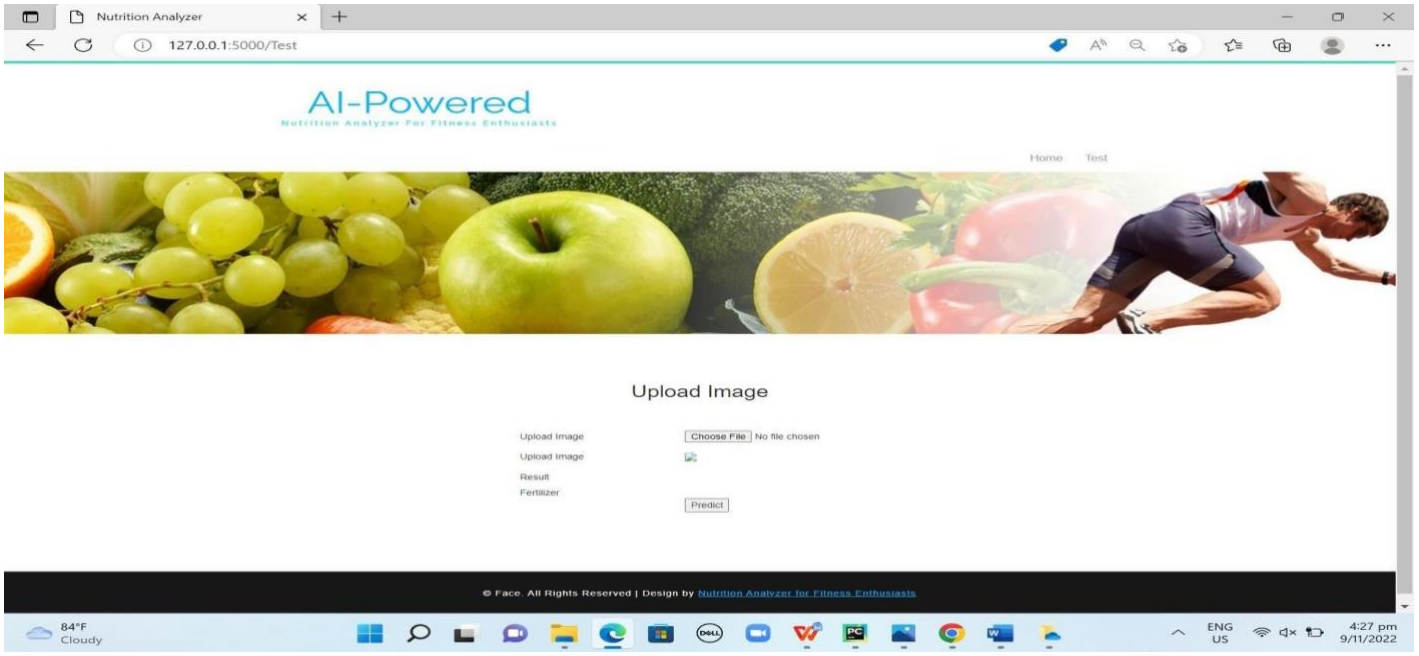

File Edit View Navigate Code Refactor Run Tools VCS Window Help
NutritionKerasCnn [K:\IBM\DSEC\NutritionKerasCnn] - ...\model.py - PyCharm
NutritionKerasCnn model.py
Project
NutritionKerasCnn [KerasCnn] K:\IBM\DSEC\NutritionKerasCnn
Data
Output
Sample
static
templates
venv library root
App.py
lena_invert.jpg
model.h5
model.py
tt.png
External Libraries
Scratches and Consoles
Test.html
index.html
model.py
lena_invert.jpg
App.py
# Part 1 - Building the CNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.models import model_from_json
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
batch_size = 32
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)
# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
'Data', # This is the source directory for training images
Run: model
82/82 [==============================] - 59s 666ms/step - loss: 0.5141 - accuracy: 0.7938
Epoch 2/10
82/82 [==============================] - 61s 742ms/step - loss: 0.0826 - accuracy: 0.9877
Epoch 3/10
82/82 [==============================] - 87s 1s/step - loss: 0.2766 - accuracy: 0.9823
Epoch 4/10
82/82 [==============================] - 108s 1s/step - loss: 3.4076e-04 - accuracy: 1.0000
Epoch 5/10
79/82 [==========================>..] - ETA: 12s - loss: 2.4570e-06 - accuracy: 1.0000
Windows Defender might be impacting your build performance. PyCharm checked the following directories:
K:\IBM\DSEC\NutritionKerasCnn
C:\Users\Fantasy-PC\.PyCharmCE2019.3\system
Fix... Actions
Run TODO Terminal Python Console
Event Log
Windows Defender might be impacting your build performance. PyCharm checked the following directories: // K:\IBM\DSEC\NutritionKerasCnn // C:\Users\Fantasy-PC\.PyCharmCE2019.3\system // Fix... / ... (13 minutes ago)
69:1 LF UTF-8 4 spaces Python 3.7 (NutritionKerasCnn)
Type here to search
ENG
04:52 PM
11-Nov-2022


File Edit View Navigate Code Refactor Run Tools VCS Window Help
NutritionKerasCnn [K:\IBM\DSEC\NutritionKerasCnn] - ...\model.py - PyCharm
NutritionKerasCnn model.py
App
Project
NutritionKerasCnn [KerasCnn] K:\IBM\DSEC\NutritionKerasCnn
Data
Output
Sample
static
templates
venv library root
App.py
lena_invert.jpg
model.h5
model.py
tt.png
External Libraries
Scratches and Consoles
Test.html
index.html
model.py
lena_invert.jpg
App.py
# Part 1 - Building the CNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.models import model_from_json
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
batch_size = 32
Run: App
K:\IBM\DSEC\NutritionKerasCnn\venv\Scripts\python.exe K:/IBM/DSEC/NutritionKerasCnn/App.py
* 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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 109-537-013
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Windows Defender might be impacting your build performance. PyCharm checked the following directories:
K:\IBM\DSEC\NutritionKerasCnn
C:\Users\Fantasy-PC\.PyCharmCE2019.3\system
Fix... Actions
Run TODO Terminal Python Console
Event Log
Unused import statement
7:39 LF UTF-8 4 spaces Python 3.7 (NutritionKerasCnn)
Type here to search
ENG
04:54 PM
11-Nov-2022

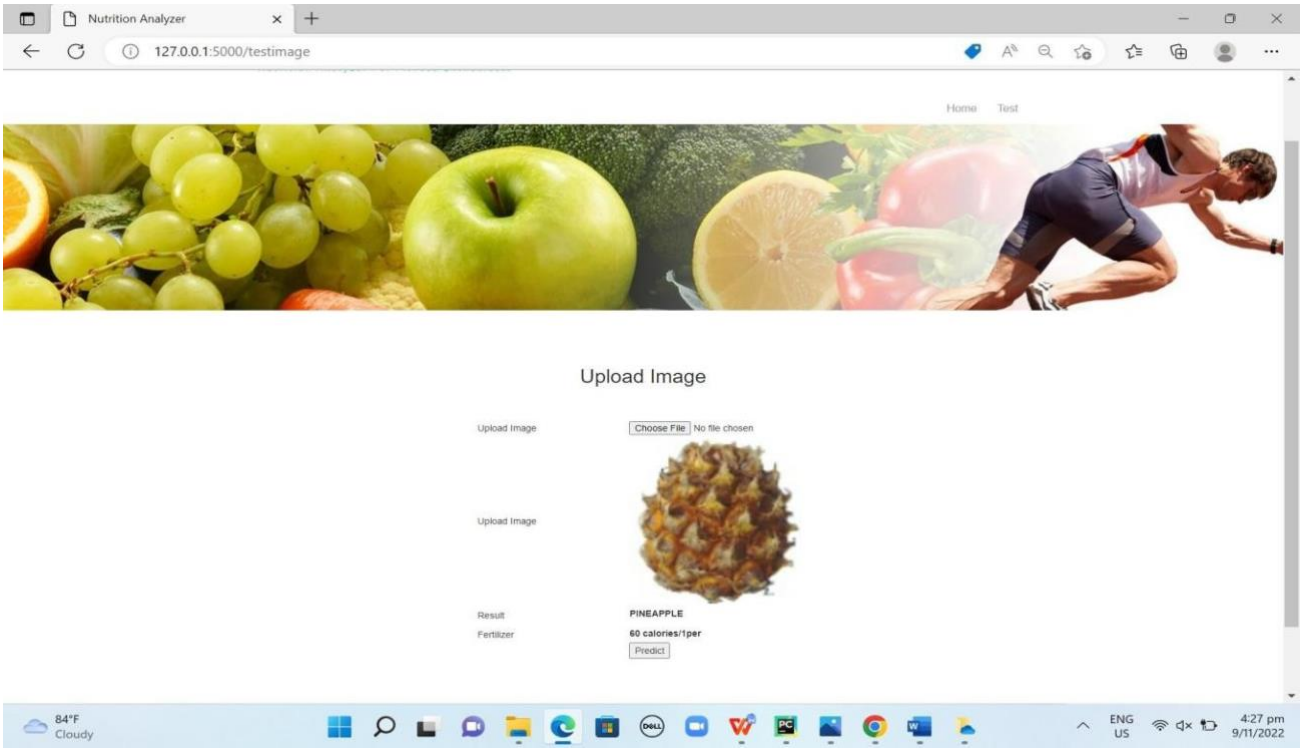
HOME PAGE



TEST PAGE



PREDICT PAGE



GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-24157-1659938736