AI-Powered Nutrition Analyzer For Fitness Enthusiasts

PROJECT REPORT

TEAM ID – PNT2022TMID26372

Submitted by

KEERTHANA VS(211719104062)

MAYA PADHY (211719104075)

MADHUBALA R (211719104071)

RASIKA M (211719104102)

ROSZHAN RAJ MS(211719104108)

in partial fulfillment for the award of the degree of,

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

CONTENTS

1. Introduction	**1.1 Technical Architecture**
	1.2 Project objective
	1.3 Project structure
2.Ideation phase	**2.1 Brainstorm**
	2.2 Prioritize
	2.3 Empathy Map
	2.4 Literature survey
	2.5 Problem statement
3.Project Design phase-1	**3.1 Proposed solution**
	3.2 Problem solution Fit
	3.3 Solution Architecture
4.Project Design phase-2	**4.1 Customer journey**
	4.2 Functional Requirement
	4.3 Data Flow Diagrams
	4.4 Technology Architecture
5.Project Planning Phase	**5.1 Prepare Milestone & Activity list**
	5.2 Sprint Delivery Plan
6. Project Development Phase	**6.1 Project Development - Delivery Of Sprint-1**
	6.1.1 Data collection
	6.1.2 Data processing
	6.2 Project Development -

Delivery Of Sprint-2

6.3 Project Development -

Delivery Of Sprint-3

6.4 Project Development -

Delivery Of Sprint-4

6.4.1 IBM cloud

6.4.2 Train model on IBM

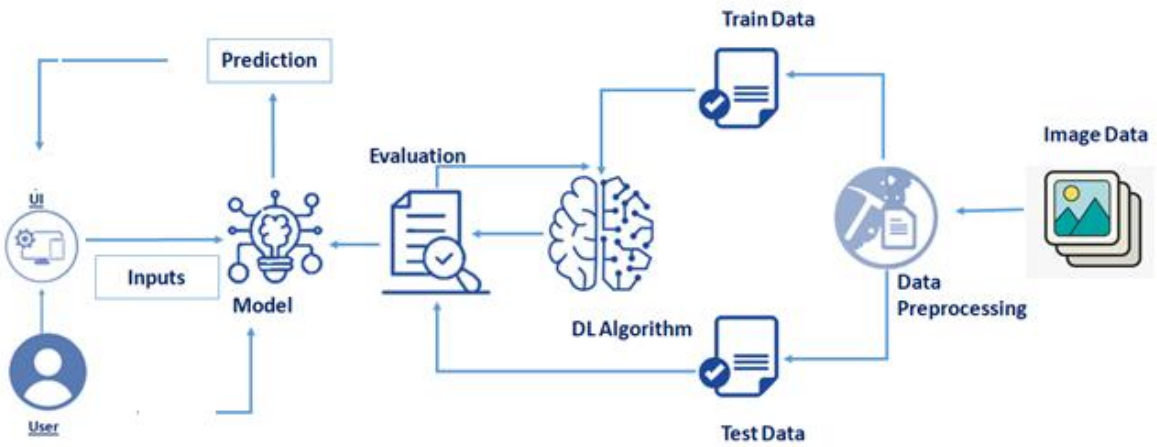
7. Conclusion

INTRODUCTION:

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.).

1.1 Technical Architecture:



1.2 Project Objectives

By the end of this project you will:

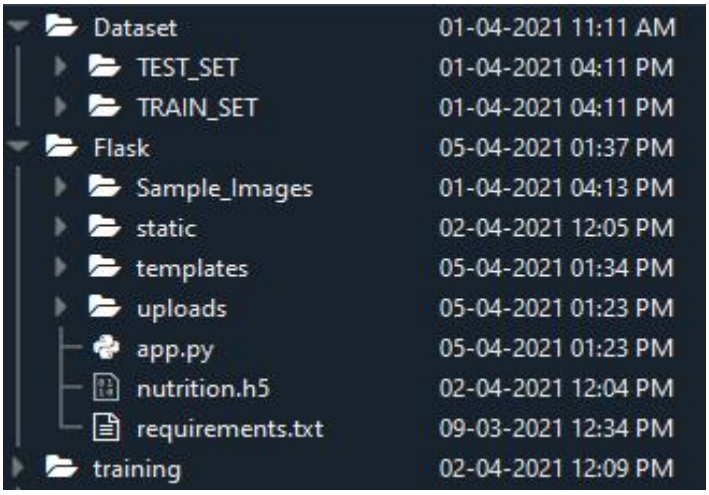
1. know fundamental concepts and techniques of Convolutional Neural Network.

gain a broad understanding of image data.

2. Knowhow to pre-process/clean the data using different data preprocessing techniques.
3. know how to build a web application using the Flask framework.

1.3 Project Structure

Create a Project folder that contains files as shown below



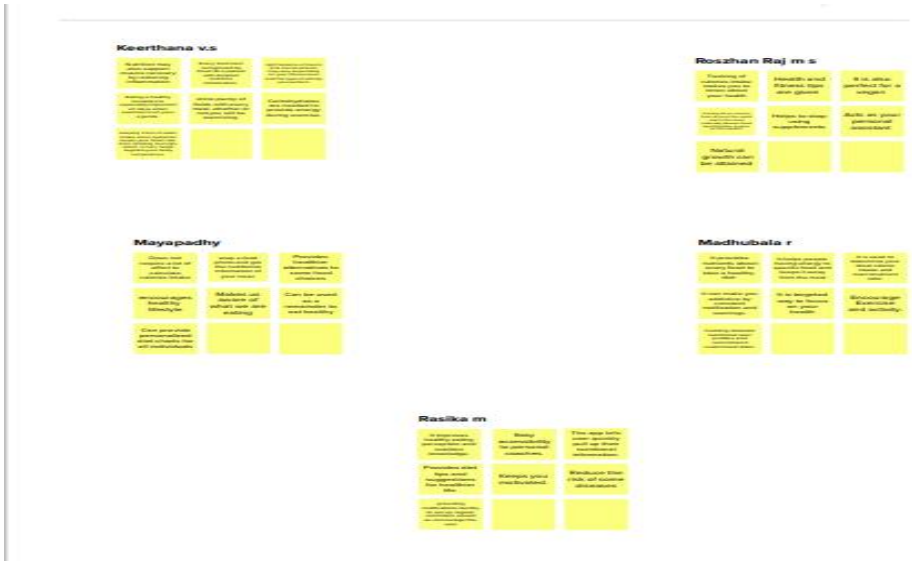
- Dataset folder contains the training and testing images for training our model.
- We are building a Flask Application that needs HTML pages stored in the templates folder and a python script app.py for serverside scripting
- we need the model which is saved and the saved model in this content is a nutrition.h5
- templates folder contains home.html, image.html, imageprediction.html pages.
- Statis folder had the css and js files which are necessary for styling the html page and for executing the actions.
- Uploads folder will have the uploaded images(which are already tested).
- Sample_images will have the images which are used to test or upload.
- Training folder contains the trained model file.

2.Ideation Phase

In this milestone you are expected to get started with the Ideation process.

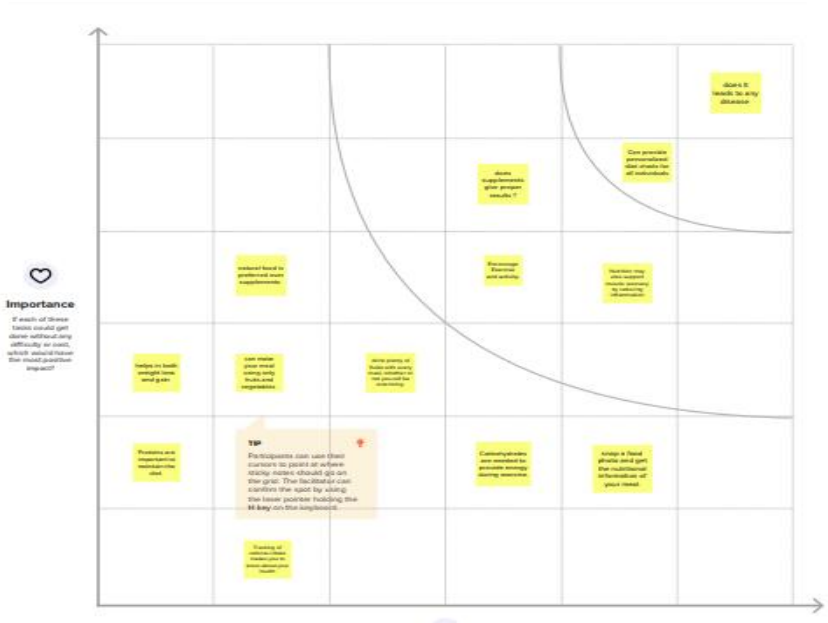
2.1 BRAINSTORM:

We have Written down our ideas that come to mind that address our problem statement.



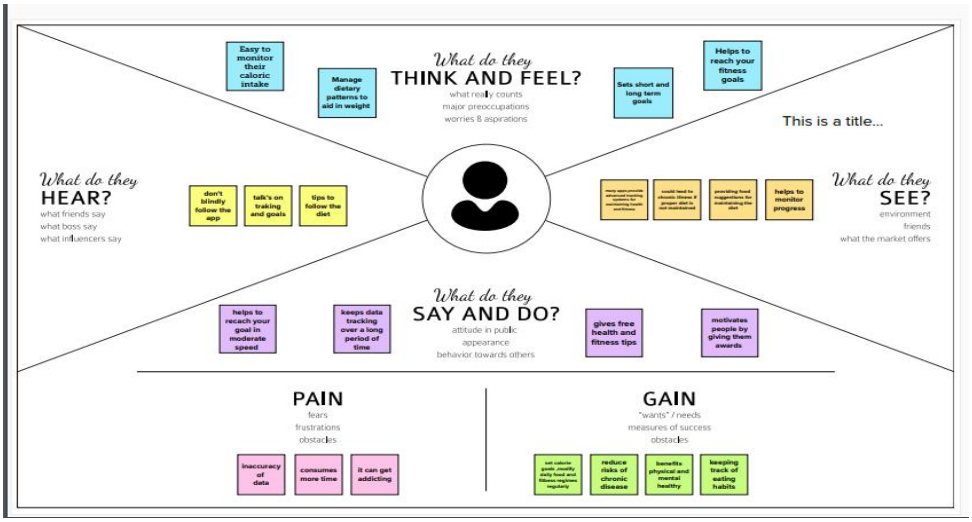
2.2 PRIORITIZE:

We have placed our ideas on this grid to determine which ideas are important and which are feasible



2.3 EMPATHY MAP:

In this activity we are expected to prepare the empathy map canvas to capture the user Pains & Gains, Prepare list of problem statements.



2.4 LITERATURE SURVEY

INTRODUCTION : In the literature survey, a brief summary of the different methods have been proposed for AI powered nutrition analysis over the period of 2010 to 2021. Each of these 10 papers has a unique approach towards AI-powered nutrition analyzer for fitness enthusiasts in some parameter or the other. The summaries of each of the papers are provided as follows:

1. **AN IMAGE ANALYSIS SYSTEM FOR DIETARY ASSESSMENT AND EVALUATION**
   There is a growing concern about chronic diseases and other health problems related to diet including obesity and cancer. Dietary intake provides valuable insights for mounting intervention programs for prevention of chronic diseases. Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields. In this paper, we describe a novel mobile telephone food record that provides a measure of daily food and nutrient intake. Our approach includes the use of image analysis tools for identification and quantification of food that is consumed at a meal. Images obtained before and after foods are eaten are used to estimate the amount and type of food consumed. The mobile device provides a unique vehicle for collecting dietary information that reduces the burden on respondents that are obtained using more classical approaches for dietary assessment. We describe our approach to image analysis that includes the segmentation of food items, features used to identify foods, a method for automatic portion estimation, and our overall system architecture for collecting the food intake information.

2. **ESTIMATION OF QUANTITY AND NUTRITIONAL INFORMATION OF FOOD USING IMAGE** PROCESSING Every edible food item—while sharing many common elements with others—uses a handful of ingredients that combine for unique flavor alongside providing calories and nutrients required for the functioning of human body. Our project aims at detecting and identifying the food item whose details are existing in the designed database and gives us certain details such as the calories and nutrients the food item consists of, and along with this, our project provides advice on how much amount of calories and nutrients the user must take according to the user's details and his/her required intake for healthy diet and fitness.
3. **CALORIE AND NUTRITION MEASURMENT FROM FOOD IMAGE BY USING SVM & KNN**
   Measuring daily food consumption for obese patients is one of the challenges in obesity management. In this project, a computer vision based system to estimate energy intake based on food pictures taken. I propose a Food Recognition System (FRS) for calories and

nutrient values assumption. The system then processes and classifies the images to detect the type of food and portion size, then uses the information to estimate the number of calories in the food. Emerging food classification methods play an important role in nowadays food recognition applications. For this purpose, a new recognition algorithm for food is presented, considering its shape, colour, size, and texture characteristics. Using various combinations of these features, a better classification will be achieved. Food calorie and nutrition measurement system that can help dietitians to measure and manage daily food intake. Here the food image is segmented into multiple segments by using the K-Means clustering. After that the texture, shape, and size features are extracted from the food image by using the Gabor filter. The performance improvement of food classification will be obtained by the combination of Support Vector Machine and K-Nearest Neighbor method. The better classification will be obtained by these combined method. The volume of the food is measured. After the mass calculation the calorie and nutrition of each food can be derived using nutritional tables.Images are an important source of data and information in the agricultural sciences. The use of image-processing techniques has outstanding implications for the analysis of agricultural operations. Fruit and vegetable classification is one of the major applications that can be utilized in supermarkets to automatically detect the kinds of fruits or vegetables purchased by customers and to determine the appropriate price for the produce. Training on-site is the underlying prerequisite for this type of arrangement, which is generally caused by the users having little or no expert knowledge. We explored various methods used in addressing fruit and vegetable classification and in recognizing fruit disease problems. We surveyed image-processing approaches used for fruit disease detection, segmentation and classification. We also compared the performance of state-of-the-art methods under two scenarios, i.e., fruit and vegetable classification and fruit disease classification. The methods surveyed in this paper are able to distinguish among different kinds of fruits and their diseases that are very alike in color and texture.

4. FOOD AND FORMALIN DETECTOR USING MACHINE LEARNING APPROACH

Unethical use of formalin, in the preservation of food items posturing threat to communal nutrition. Without chemical experts accurately Formalin detection is a time consuming and complicated task. Moreover, the presence of naturally occurring formalin in food items may interfere in detecting artificially added formalin. This paper presents a dynamic and reliable food and formalin detection technique based on machine learning approaches. Different machine learning algorithms i.e., Naïve Bayes, Logistic regression, Support Vector Machine, K-NN Classifier are applied to the experimental dataset to build a predictive model. Conductive properties were used to detect the type of foods. The designed system is able to detect 1-50 ppm of formalin using VOC HCHO gas sensor combining with arduino-uno. Several Tests are conducted and polynomial regression has been applied to presume the application of formalin.

5. FOOD IMAGE RECOGNITION AND FOOD SAFETY DETECTION METHOD BASED ON DEEP LEARNING

With the development of machine learning, as a branch of machine learning, deep learning has been applied in many fields such as image recognition, image segmentation, video segmentation, and so on. In recent years, deep learning has also been gradually applied to food recognition. However, in the field of food recognition, the degree of complexity is high, the situation is complex, and the accuracy and speed of recognition are worrying. This paper tries to solve the

above problems and proposes a food image recognition method based on neural network. Combining Tiny-YOLO and twin network, this method proposes a twostage learning mode of YOLO-SIMM and designs two versions of YOLOSiamV1 and YOLO-SiamV2. Through experiments, this method has a general recognition accuracy. However, there is no need for manual marking, and it has a good development prospect in practical popularization and application.

6. APPLICATION OF MACHINE VISION SYSTEM IN FOOD DETECTION

Food processing technology is an important part of modern life globally and will undoubtedly play an increasingly significant role in future development of industry. Food quality and safety are societal concerns, and food health is one of the most important aspects of food processing. However, ensuring food quality and safety is a complex process that necessitates huge investments in labor. Currently, machine vision system based image analysis is widely used in the food industry to monitor food quality, greatly assisting researchers and industry in improving food inspection efficiency. Meanwhile, the use of deep learning in machine vision has significantly improved food identification intelligence. This paper reviews the application of machine vision in food detection from the hardware and software of machine vision systems, introduces the current state of research on various forms of machine vision, and provides an outlook on the challenges that machine vision system faces.

7. DEEP LEARNING AND MACHINE VISION FOR FOOD PROCESSING:

A SURVEY The quality and safety of food is an important issue to the whole society, since it is at the basis of human health, social development and stability. Ensuring food quality and safety is a complex process, and all stages of food processing must be considered, from cultivating, harvesting and storage to preparation and consumption. However, these processes are often labour-intensive. Nowadays, the development of machine vision can greatly assist researchers and industries in improving the efficiency of food processing. As a result, machine vision has been widely used in all aspects of food processing. At the same time, image processing is an important component of machine vision. Image processing can take advantage of machine learning and deep learning models to effectively identify the type and quality of food. Subsequently, follow-up design in the machine vision system can address tasks such as food grading, detecting locations of defective spots or foreign objects, and removing impurities. In this paper, we provide an overview on the traditional machine learning and deep learning methods, as well as the machine vision techniques that can be applied to the field of food processing. We present the current approaches and challenges, and the future trends.

8. IMAGE PROCESSING METHODS FOR FOOD INSPECTION

With the advances in computer technology, signal processing techniques are widely applied to many food safety applications. In this thesis, new methods are developed to solve two food safety problems using image processing techniques.First problem is the detection of fungal infection on popcorn kernel images. This is a damage called blueeye caused by a fungus. A cepstrum based feature extraction method is applied to the kernel images for classification purposes. The results of this technique are compared with the results of a covariance based feature extraction method, and previous solutions to the problem. The tests are made on two different databases; reflectance and transmittance mode image databases, in which the method of the image acquisition differs. Support Vector Machine (SVM) is used for image feature classification. It is experimentally observed that an overall success rate of 96% is possible with the covariance matrix based feature extraction method

over transmittance database and 94% is achieved for the reflectance database. The second food inspection problem is the detection of acrylamide on cookies that is generated by cooking at high temperatures. Acrylamide is a neurotoxin and there have been various studies on detection of acrylamide during the baking process. Some of these detection routines include the correlation between the acrylamide level and the color values of the image of the cookies, resulting easier detection of acrylamide without the need of complex, expensive and time consuming chemical tests. Studies on the subject are tested on still images of the cookies, which are obtained after the cookies are removed from the oven. An active contour method is developed, that makes it possible to detect the cookies inside the oven or possibly on a moving tray, from the video captured .

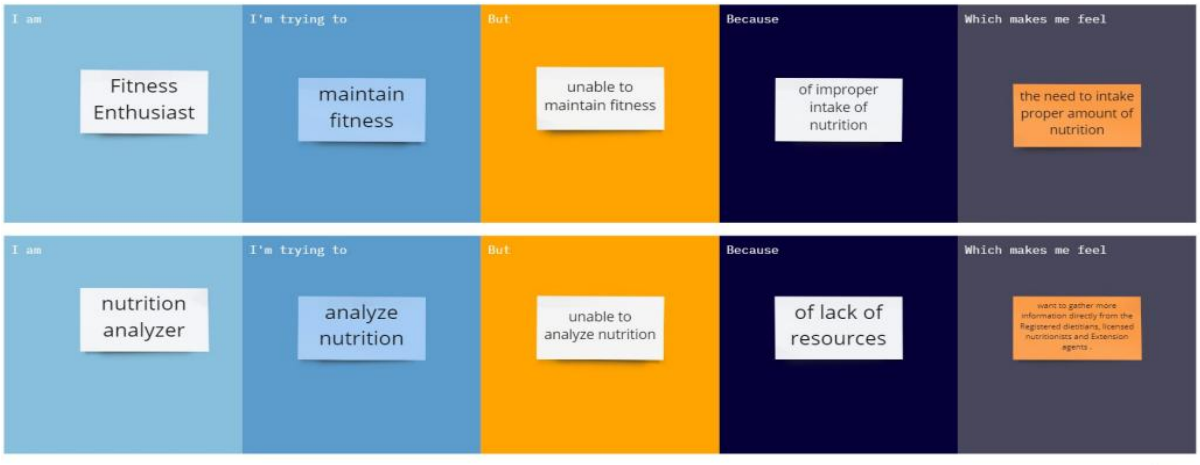
8. **QUALITY DETECTION OF FRUITS BY USING ANN TECHNIQUE**

Grading and classification of fruits is based on observations and through experiences. The system utilizes image-processing techniques to classify and grade quality of fruits. Two dimensional fruit images are classified on shape and colour based analysis methods. However, different fruit images may have similar or identical colour and shape values. Hence, using colour or shape features analysis methods are still not effective enough to identify and distinguish fruits images. Therefore, we used a method to increase the accuracy of the fruit quality detection by using colour, shape, and size based method with combination of artificial neural network (ANN). Proposed method grades and classifies fruit images based on obtained feature values by using cascaded forward network. The proposed system starts the process by capturing the fruit's image. Then, the image is transmitted to the processing level where the fruit features like colour, shape and size of fruit samples are extracted. After that by using artificial neural network fruit images are going through the training and testing. In this proposed paper neural network is used to detect shape, size and colour of fruit and with the combination of these three features the results obtained are very promising.

9. **STUDY FOR FOOD RECOGNITION SYSTEM USING DEEP LEARNING**

Accurate dietary appraisal has been found by literature to be very significant in the evaluation of weight loss treatments. Most current methods of dietary evaluation, however, depend on recollection. The development of a modern computer-based food recognition system for reliable food evaluation is now possible across comprehensive mobile devices as well as rich Cloud services. Fixing the problem of food detection and identification in photos of different kinds of foods. Given the variety of food products with low inter-and high intra-class variations and the limited information in a single picture, the problem is complicated. By propose the overall application of multiple fusiontrained classifiers to achieve increased identification and recognition capabilities on characteristics obtained from various deep models. This paper studied various techniques of food recognition using different approaches and based on several variables, compared their effectiveness. Our study results demonstrate that deep learning overcomes other strategies like manual feature extractors, standard ML algorithms, as well as DL as a practical tool for food hygiene and safety inspections. Keywords: Food Recognition, Neural Networks, Deep Learning, Classification, Clustering, Feature Selection.

2.5PROBLEM STATEMENT:

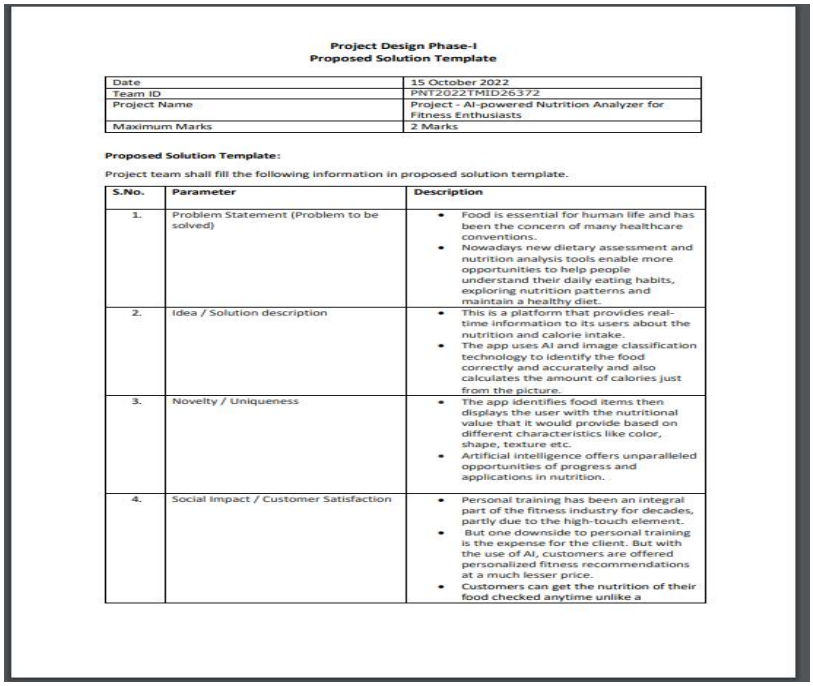


3.PROJECT DESIGN PHASE– I

From this milestone we will be starting the project design phase. We are expected to cover the activities given.

3.1 PROPOSED SOLUTION:

In this activity we are expected to prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.



Project Design Phase-I
Proposed Solution Template

Date	15 October 2022
Team ID	PNT2022TMID26372
Project Name	Project - AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	2 Marks

Proposed Solution Template:

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

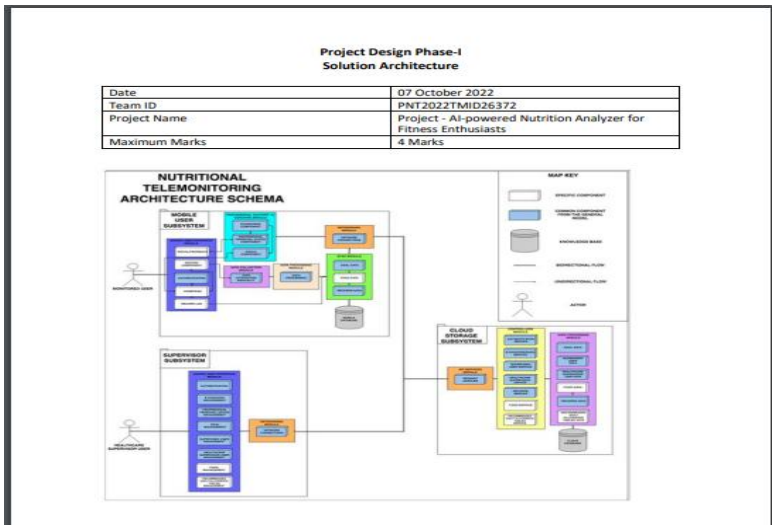
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Food is essential for human life and has been the concern of many healthcare conventions. • Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet.
2.	Idea / Solution description	• This is a platform that provides real-time information to its users about the nutrition and calorie intake. • The app uses AI and image classification technology to identify the food correctly and accurately and also calculates the amount of calories just from the picture.
3.	Novelty / Uniqueness	• The app identifies food items then displays the user with the nutritional value that it would provide based on different characteristics like color, shape, texture etc. • Artificial intelligence offers unparalleled opportunities of progress and applications in nutrition.
4.	Social Impact / Customer Satisfaction	• Personal training has been an integral part of the fitness industry for decades, partly due to the high-touch element. • But one downside to personal training is the expense for the client. But with the use of AI, customers are offered personalized fitness recommendations at a much lesser price. • Customers can get the nutrition of their food checked anytime unlike a

		nutritionist who cannot be available all the time.
5.	Business Model (Revenue Model)	• Users can subscribe with the app to get various rewards on eating healthy. • The subscription gives the customer an edge over others by getting personalized food recommendations and exclusive recipes that promotes healthy eating. • With the help of partnerships from various other companies based on fitness sector, the app can advertise a range of health related products such as smart wearables, protein shakes etc • It will provide users with these products at a discounted price which will be a key way for monetization and helps the partnership companies in targeted marketing.
6.	Scalability of the Solution	• Tracking a member's activity lets you know what their interests are, and you can instantly provide additional information on a specific service the member was interested in. • Virtual assistants coupled with motion sensing technologies can offer even more specific instructions and movement suggestions to users to improve their form or change certain habits.

3.2 PROBLEM SOLUTION:

3.3 SOLUTION ARCHITECTURE:

In this activity we are expected to prepare solution architecture document and submit for review.



Project Design Phase-I
Solution Architecture

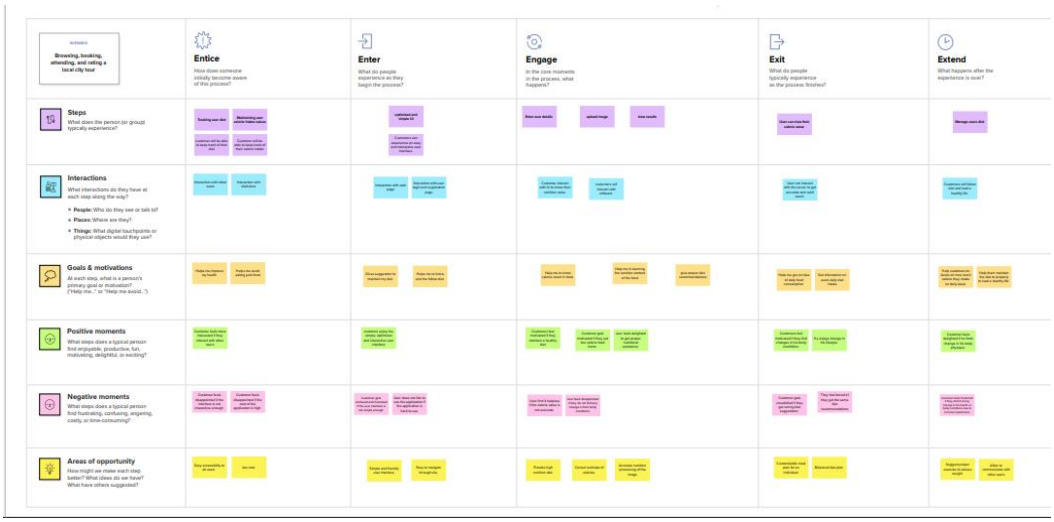
Date	07 October 2022
Team ID	PNT2022TMID26372
Project Name	Project - AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	4 Marks

4.PROJECT DESIGN PHASE-II

From this milestone we will be continue working on the project design phase. We are expected to cover the activities given.

4.1 **CUSTOMER JOURNEY**:

Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).



	Entice	Enter	Engage	Exit	Extend
Steps					
Interactions					
Goals & motivations					
Positive moments					
Negative moments					
Areas of opportunity					

4.2 DATA FLOW DIAGRAM:

In this activity we are expected to prepare the data flow diagrams and submit for review.

PROJECT DESIGN PHASE-II

DATA FLOW DIAGRAM

Date	28-10-2022
Team ID	PNT2022TMID26372
Project name	AI-powered Nutrition Analyzer for Fitness Enthusiasts

Simplified diagram

DFD Level 0 (Industry Standard)

4.3 **FUNCTIONAL REQUIREMENT**:

In this activity we are expected to prepare the functional requirement document.

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	20 October 2022
Team ID	PNT2022TMID26372
Project Name	Project - AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image Acquisition	Capture the Image and Check the Top and Side View of Image
FR-4	Object Detection	Get a series of Bounding Boxes, which means objects are located.
FR-5	Image Segmentation	Get a series of food images stored in matrix with values of background pixels replaced by zeros.
FR-6	Volume Estimation	To estimate the volume, calculate the scale factors on calibration objects.
FR-7	Calorie Estimation	After estimating the volume, the next step is to estimate each food's mass

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Informs you how nutrient dense your food is.
NFR-2	**Security**	The information is visible to user only and image was secured highly.
NFR-3	**Reliability**	The food packages are important for calculate the calories
NFR-4	**Performance**	It is based on the package of food used for the calorie calculation
NFR-5	**Availability**	It is available for all users to calculate the calorie of the foods
NFR-6	**Scalability**	Increasing the calculation of the calorie in foods

4.4 TECHNOLOGY ARCHITECTURE:

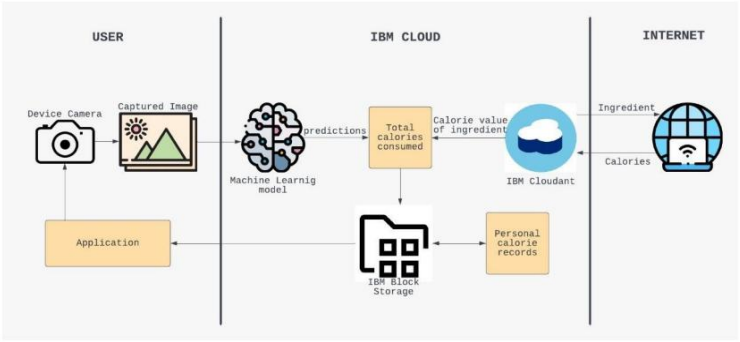
Project Design Phase-II Technology Stack (Architecture &Stack)

Date	25 October 2022
Team ID	PNT2022TMID26372
Project Name	Project - AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Convolution layers are used to process images	Python
3.	Application Logic-2	Developing a size analysis and tech stack for the backend	Python, HTML
4.	Application Logic-3	Analyzing texture and colour based on input	IBM Assistant
5.	Database	Various datasets and configurations	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant, etc.
7.	File Storage	Data Storage On Cloud	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Developing a model on IBM and integrating it with a flask application	CNN IBM Deployment
9.	External API-2	A Flask application receives input parameters from an HTML page	Python Flask, HTMl
10.	Deep Learning Model	By using deep learning and artificial intelligence in nutrition analysis, superior performance can be achieved for predicting and demonstrating the feasibility of using these technologies	Image Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Pycharm, Python,Anaconda Navigator, Flask,HTML	Deep Learning
2.	Security Implementations	Strong passwords and two-factor authentication	Encryptions
3.	Scalable Architecture	Provides support for higher workloads without sacrificing performance	Python
4.	Availability	Inputs, for example: datasets	Kaggle
5.	Performance	Adding layers to the convolution network to increase its capacity inputs	Artificial Neural Network

5.PROJECT PLANNING PHASE:

In this milestone we are expected to prepare milestones & tasks, sprint schedules.

5.1 PREPARE MILESTONE & ACTIVITY LIST:

In this activity we are expected to prepare the milestones & activity list of the project.

PROJECT PLANNING PHASE
MILESTONE & ACTIVITY LIST

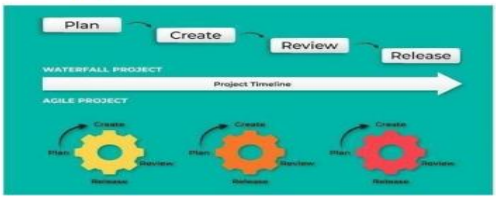
DATE	19 October 2022
TEAM ID	PNT2022TMID26372
PROJECT NAME	AI-powered Nutrition Analyzer for Fitness Enthusiasts

Milestone:

Machine learning and AI in nutrition analyze raw data to identify competitive traits that are useful for forecasting improved dietary plans. Artificial intelligence and machine learning have become primary components of daily workouts. 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.)..

Activity List:
In Project Management Planning is an Important task to scheduling the phase of the projectto the Team Member. In this Acti vity can shows the various activity are allocated and done by the Team Members! In Project we can Split into the Four Step of Phases are,

Phase 1: Information Collection and Requirement Analysis.
Phase 2: Project Planning and Developing Modules.
Phase 3: Implementing the High Accuracy Deep Learning Algorithm to Perform. Phase 4: Deploying the Model on Cloud and Testing the Model and UI Performance



5.2 SPRINT DELIVERY PLAN

In this activity we are expected to prepare the sprint delivery plan.

Project Planning Phase
Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	22 October 2022
Team ID	PNT2022TMID26372
Project Name	AI – Powered Nutrition Analyzer for fitness Enthusiasts
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download Food Nutrition Dataset	2	Medium	Roszhan Raj M S
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low	Rasika M
Sprint-1		USN-3	Handling Missing Data	3	Medium	Roszhan Raj M S
Sprint-1		USN-4	Feature Scaling	3	Low	Madhubala R
Sprint-1		USN-5	Data Visualization	3	Medium	Maya Padhy
Sprint-1		USN-6	Splitting Data into Train and Test	4	High	Keerthana V S
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High	Keerthana V S ,Maya Padhy

Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium	Rasika M
Sprint-2		USN-9	Initializing The Model	1	Medium	Keerthana V S
Sprint-2		USN-10	Adding LSTM Layers	2	High	Roszhan Raj M S
Sprint-2		USN-11	Adding Output Layers	3	Medium	Keerthana V S
Sprint-2		USN-12	Configure The Learning Process	4	High	Maya Padhy
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2		USN-13	Train The Model	2	Medium	Roszhan Raj M S
Sprint-2		USN-14	Model Evaluation	1	Medium	Maya Padhy
Sprint-2		USN-15	Save The Model	2	Medium	Rasika M
Sprint-2		USN-16	Test The Model	3	High	Madhubala R
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium	Roszhan Raj M S
Sprint-3		USN-18	Build Python Code	4	High	Keerthana V S
Sprint-3		USN-19	Run The App in Local Browser	4	Medium	Rasika M
Sprint-3		USN-20	Showcasing Prediction On UI	4	High	Maya Padhy
Sprint-4	Train The Model On IBM	USN-21	Register For IBM Cloud	4	Medium	Roszhan Raj M S
Sprint-4		USN-22	Train The ML Model On IBM	8	High	Keerthana V S
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High	Maya Padhy

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

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

6.PROJECT DEVELOPMENT PHASE

In this milestone we will start the project development and expected to perform the coding & solutioning, acceptance testing, performance testing based as per the sprint and submit them.

6.1 PROJECT DEVELOPMENT - DELIVERY OF SPRINT-1

In this activity we are expected to develop & submit the developed code by testing it.

6.1.1DATA COLLECTION:

Collect images of different food items organized into sub directories based on their respective names as shown in the project structure.
Create folders of types of food items that need to be recognized.

In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', 'pineapple' and 'watermelon', they are saved in the respective sub directories with their respective names.

For more accurate results we can collect images of high resolution and feed the model with more images.

DataSet:/content/drive/MyDrive/IBM

6.1.2 DATA PROCESSING:

IMPORTING LIBRARY

```
from keras.preprocessing.image import ImageDataGenerator
```

IMAGE DATA GENERATOR

```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
text_dataset=ImageDataGenerator(rescale=1./255)
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale= 1./255,horizontal_flip = True,vertical_flip = True,zoom_range = 0.2)
test_datagen = ImageDataGenerator(rescale= 1./255)
```

```
x_train = train_datagen.flow_from_directory("/content/drive/MyDrive/IBM/Dataset/TRAIN_SET",target_size = (64,64),
                                            class_mode = "categorical",batch_size = 24)
```

```
Found 4184 images belonging to 5 classes.
```

```
x_test = test_datagen.flow_from_directory("/content/drive/MyDrive/IBM/Dataset/TEST_SET",target_size = (64,64),
                                          class_mode = "categorical",batch_size = 24)
```

```
Found 939 images belonging to 5 classes.
```

```
import cv2
```

```
img = cv2.imread("/content/drive/MyDrive/IBM/Dataset/TEST_SET/ORANGE/n07749192_10081.jpg")
```

```
img
img.ndim
img.shape
```

```
(256, 256, 3)
```

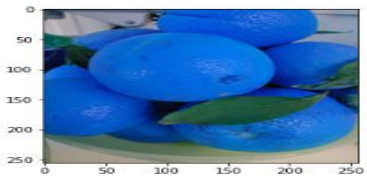
```
img_flag = cv2.imread("/content/drive/MyDrive/IBM/Dataset/TEST_SET/ORANGE/n07749192_10081.jpg")
```

```
img_flag
```

```
array([[[ 46,  65,  73],
        [ 44,  62,  73],
        [ 41,  63,  75],
        ...,
        [141, 155, 177],
        [141, 155, 177],
        [140, 154, 176]],

       [[ 28,  40,  52],
        [ 39,  52,  66],
        [ 45,  61,  77],
        ...,
        [141, 155, 177],
        [141, 155, 177],
        [140, 154, 176]],

       [[ 36,  41,  50],
        [ 21,  30,  39],
        [ 24,  36,  46],
        ...,
        [143, 157, 179],
        [142, 156, 178],
        [141, 155, 177]],

       ...,

       [[ 99, 108,  82],
        [104, 113,  86],
        [107, 114,  87],
        ...,
        [197, 181, 164],
        [196, 180, 163],
        [196, 180, 163]],

       [[ 98, 107,  80],
        [103, 112,  85],
        [105, 113,  83],
        ...,
        [195, 179, 162],
        [195, 179, 162],
        [194, 178, 161]],

       [[ 97, 106,  79],
        [102, 112,  82],
        [103, 111,  81],
        ...,
        [194, 178, 161],
        [193, 177, 160],
        [193, 177, 160]]], dtype=uint8)
```

In [69]:
```
import matplotlib.pyplot as plt
plt.imshow(img)
```

Out[69]:



performing the following :

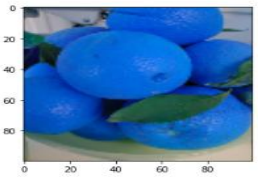
1. resizing of the image
2. coversion of color
3. cropping of the image
4. blurring of the image
5. Gaussian blur on the image
6. canny edge detection
7. binary thresholding
8. writing text on the image

In [70]:
```
plt.imshow(img_flag)
```

Out[70]:

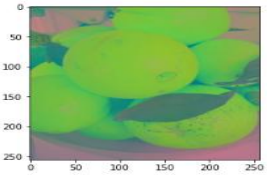
In [71]:
```
resized_img = cv2.resize(img,(100,100))
resized_img.shape
plt.imshow(resized_img)
```

Out[71]:

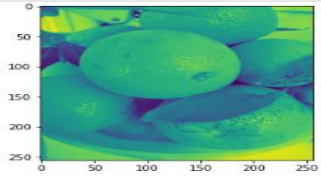


In [72]:
```
cv_img = cv2.cvtColor(img,cv2.COLOR_BGR2YCR_CB)
plt.imshow(cv_img)
```

Out[72]:

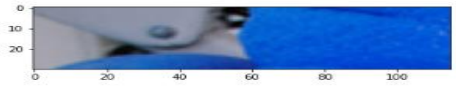


In [73]:
```
cv_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(cv_img)
```



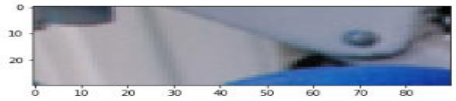
[74]:
```
roi_img = img[50:280,35:150]
roi_img = img[10:40,35:150]
plt.imshow(roi_img)
```

[74]:

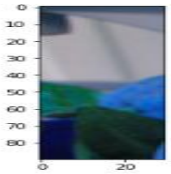


[75]:
```
roi_img = img[10:40,0:90]
plt.imshow(roi_img)
```

[75]:

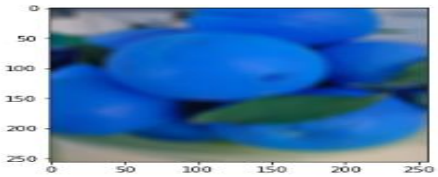


[76]:
```
roi_img = img[0:90,10:40]
plt.imshow(roi_img)
```



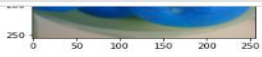
n [77]:
```
img_bl = cv2.blur(img,(10,10))
plt.imshow(img_bl)
```

ut[77]:



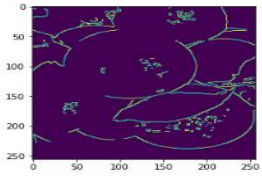
n [78]:
```
img_gbl = cv2.GaussianBlur(img,(5,5),0)
plt.imshow(img_gbl)
```

ut[78]:



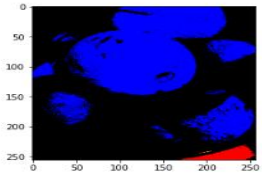
79]:
```
img_edge = cv2.Canny(img,230,350)
plt.imshow(img_edge)
```

79]:



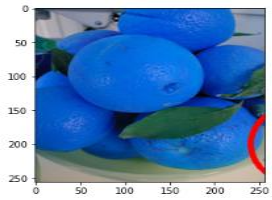
80]:
```
thresh, thresh_img = cv2.threshold(img, 200, 255, cv2.THRESH_BINARY) #img
plt.imshow(thresh_img)
```

80]:



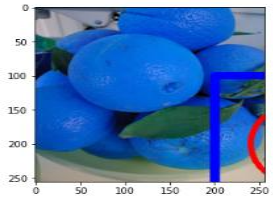
```
[81]: circle = cv2.circle(img,(300,200),60,(255,0,0),5)
      plt.imshow(img)
```

[81]:

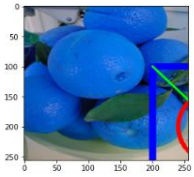


```
[82]: rectangle = cv2.rectangle(img,(200,100),(400,300),(0,0,255),10)
      plt.imshow(img)
```

[82]:

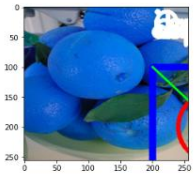


```
[83]: line = cv2.line(img,(200,100),(400,300),(0,255,0),3)
      plt.imshow(img)
```



```
]: text = cv2.putText(img,"orange",(200,50),cv2.FONT_HERSHEY_SCRIPT_COMPLEX,2,(255,255,255),5)
   plt.imshow(img)
```

]:



```
]: 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).
```

6.2 PROJECT DEVELOPMENT - DELIVERY OF SPRINT-2

In this activity we are expected to develop & submit the developed code by testing it

IMPORTING LIBRARY

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.preprocessing.image import ImageDataGenerator
```

DATA AUGMENTATION

TESTING AND TRAINING

```
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/IBM/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/IBM/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 4184 images belonging to 5 classes.
Found 939 images belonging to 5 classes.
```

```
print(x_train.class_indices)
```

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

```
print(x_test.class_indices)
```

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

```
from collections import Counter as c
c(x_train .labels)
```

```
Counter({0: 995, 1: 1354, 2: 1039, 3: 321, 4: 475})
```

INITIALIZATION OF CNN

INITIALIZATION OF CNN

PERFORMING THE FOLLOWING:

1. Adding the convolution layer
2. Adding maxpooling layer
3. Second Maxpooling snf convolution layers
4. Flattening of layers
5. Adding Dense layer

```
classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Flatten())
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

```
classifier.summary()
```

```
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 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 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

```
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

FITTING THE MODEL

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=10, validation_data=x_test,validation_steps = len(x_test))
```

```
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
on. Please use `Model.fit`, which supports generators.
  This is separate from the ipykernel package so we can avoid doing imports until
837/837 [==============================] - 63s 75ms/step - loss: 0.4014 - accuracy: 0.8494 - val_loss: 0.4496 - val_accuracy: 0.8328
Epoch 2/10
837/837 [==============================] - 50s 59ms/step - loss: 0.3665 - accuracy: 0.8602 - val_loss: 0.4661 - val_accuracy: 0.8285
Epoch 3/10
837/837 [==============================] - 52s 62ms/step - loss: 0.3486 - accuracy: 0.8647 - val_loss: 0.4653 - val_accuracy: 0.8190
Epoch 4/10
837/837 [==============================] - 51s 61ms/step - loss: 0.3205 - accuracy: 0.8762 - val_loss: 0.3912 - val_accuracy: 0.8562
Epoch 5/10
837/837 [==============================] - 52s 62ms/step - loss: 0.3067 - accuracy: 0.8776 - val_loss: 0.4102 - val_accuracy: 0.8509
Epoch 6/10
837/837 [==============================] - 51s 61ms/step - loss: 0.3085 - accuracy: 0.8855 - val_loss: 0.3579 - val_accuracy: 0.8722
Epoch 7/10
837/837 [==============================] - 48s 58ms/step - loss: 0.2729 - accuracy: 0.8944 - val_loss: 0.4144 - val_accuracy: 0.8605
Epoch 8/10
837/837 [==============================] - 52s 62ms/step - loss: 0.2618 - accuracy: 0.8984 - val_loss: 0.3602 - val_accuracy: 0.8733
Epoch 9/10
837/837 [==============================] - 52s 62ms/step - loss: 0.2398 - accuracy: 0.9085 - val_loss: 0.4673 - val_accuracy: 0.8168
Epoch 10/10
837/837 [==============================] - 51s 61ms/step - loss: 0.2354 - accuracy: 0.9125 - val_loss: 0.6300 - val_accuracy: 0.8381
```

SAVING THE MODEL

```
classifier.save('nutrition.h5')
```

NUTRITION ANALYSIS RESULT PREDICTION

IMPORTING LIBRARY

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

```
img = image.load_img("/content/drive/MyDrive/IBM/Dataset/TRAIN_SET/ORANGE/n07749192_10454.jpg",target_size= (64,64))
img
```

```
x=image.img_to_array(img)
x
```

```
array([[[229., 236., 246.],
        [235., 239., 250.],
        [237., 241., 252.],
        ...,
        [ 56.,  58.,  53.],
        [ 28.,  32.,  18.],
        [ 40.,  41.,  33.]],

       [[234., 238., 249.],
        [235., 239., 250.],
        [237., 241., 252.],
        ...,
        [ 23.,  26.,   9.],
        [ 21.,  24.,   7.],
        [ 38.,  39.,  31.]],

       [[235., 239., 250.],
        [237., 241., 252.],
        [238., 242., 251.],
        ...,
        [ 15.,  19.,   2.],
        [ 13.,  17.,   3.],
        [ 30.,  31.,  25.]],

       ...,

       [[ 61.,  60.,  39.],
        [ 62.,  62.,  36.],
        [ 62.,  65.,  38.],
        ...,
        [ 29.,  32.,  11.],
        [ 27.,  30.,  11.],
```

```
        [ 27.,  30.,  11.],
        [ 23.,  25.,  11.]],

       [[ 60.,  59.,  38.],
        [ 58.,  61.,  34.],
        [ 59.,  62.,  35.],
        ...,
        [ 36.,  37.,  19.],
        [ 17.,  22.,   2.],
        [ 26.,  30.,  16.]],

       [[ 57.,  59.,  37.],
        [ 58.,  60.,  38.],
        [ 58.,  61.,  34.],
        ...,
        [ 14.,  18.,   4.],
        [ 22.,  24.,  10.],
        [ 26.,  28.,  15.]]], dtype=float32)
```

```
x.ndim
x=np.expand_dims(x,axis=0)
x.ndim
```

```
4
```

```
predict_x=classifier.predict(x)
classes_x=np.argmax(predict_x,axis = -1)
classes_x
```

```
1/1 [==============================] - 0s 116ms/step
array([2])
```

```
predict_x
```

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

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

```
'ORANGE'
```

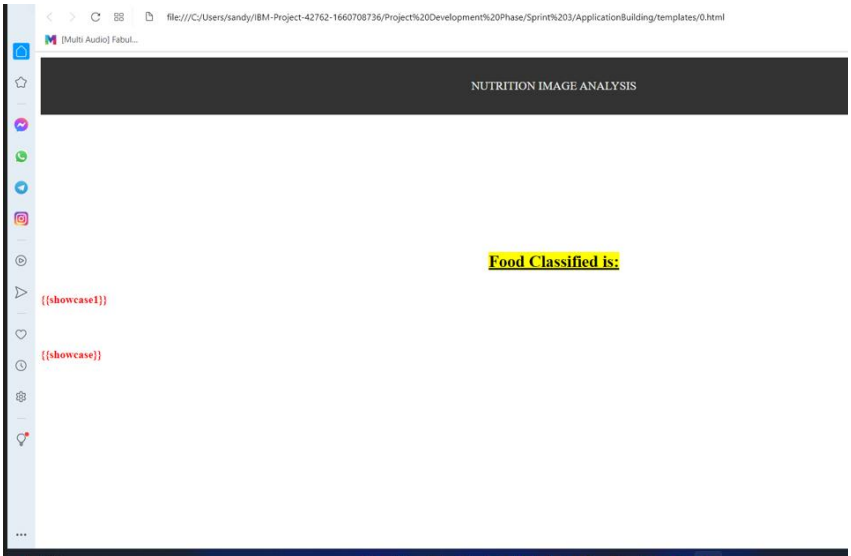
6.3 PROJECT DEVELOPMENT - DELIVERY OF SPRINT-3

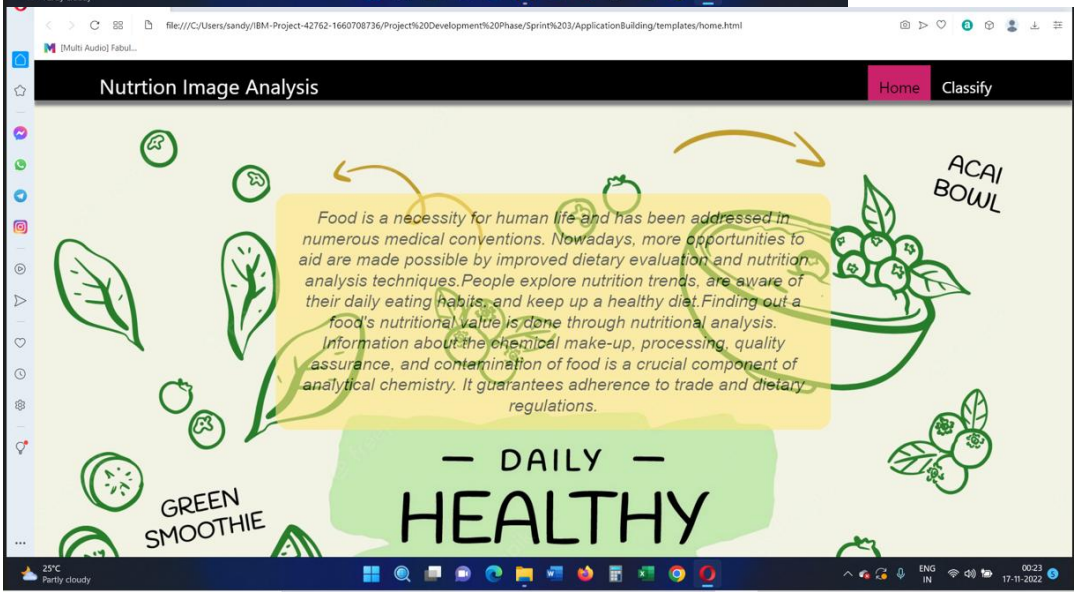
In this activity we are expected to develop & submit the developed code by testing it.

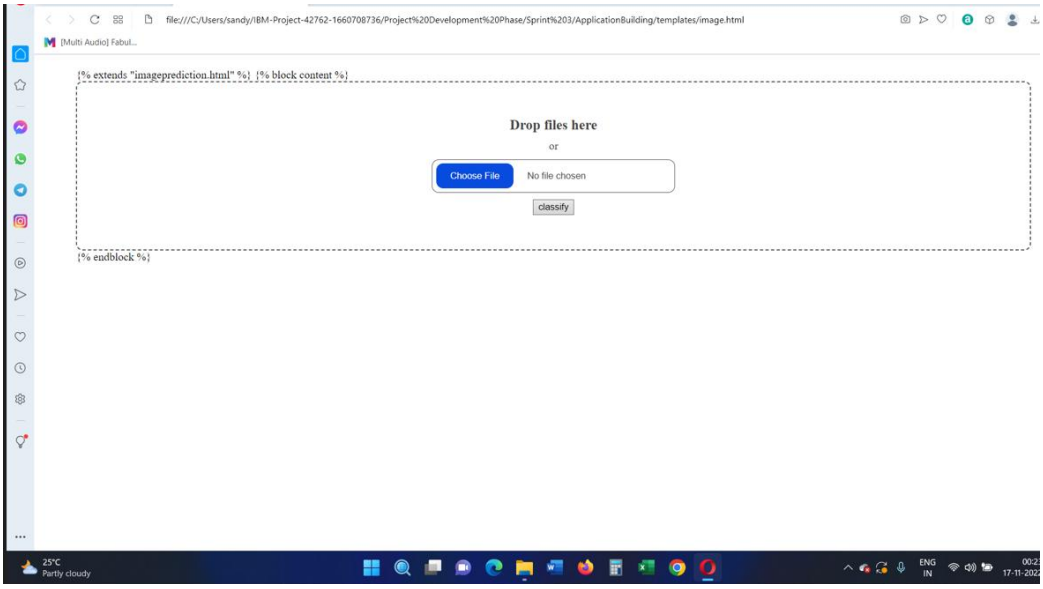
```
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
import requests


app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")


@app.route('/')# route to display the home page
def home():
    return render_template('home.html')#rendering the home page

@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("image.html")



@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to an array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred)#printing the prediction
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']

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

        x=result
```

```
        print(x)
        result=nutrition(result)
        print(result)

        return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):


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

    querystring = {"query":index}

    headers = {
        'x-rapidapi-key': "5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
        'x-rapidapi-host': "calorieninjas.p.rapidapi.com"
        }

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

    print(response.text)
    return response.json()['items']
if __name__ == "__main__":
  # running the app
    app.run(debug=False)
```

FRONT END:







6.4 PROJECT DEVELOPMENT - DELIVERY OF SPRINT-4

In this activity we are expected to develop & submit the developed code by testing it.

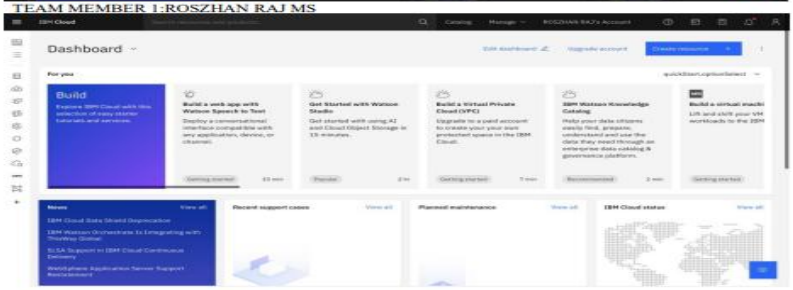
6.4.1 IBM CLOUD:

IBM CLOUD-Software

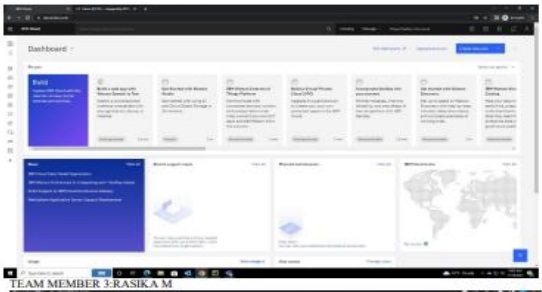
Date	12 November 2022
Team ID	PNT2022TMID26372

TEAM LEADER:KEERTHANA VS

TEAM MEMBER 1:ROSZHAN RAJ MS



TEAM MEMBER 2:MAYA PADHY



TEAM MEMBER 3:RASIKA M

TEAM MEMBER 4:MADHUBALA R

6.4.2 TRAIN MODEL ON IBM:

EPBL/IBM-Project-22536-1659853629/blob/main/Project%20Development%20Phase/sprint%204/Train%20Model%20on%20IBM/IBM_...

IMPORTING LIBRARY

[9]:
```
pwd
```

[9]:
```
'/home/wsuser/work'
```

72]:
```
!pip install keras
!pip install tensorflow
```

```
Requirement already satisfied: keras in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.0)
Requirement already satisfied: tensorflow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.2)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.37.0)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.2.1)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.23.
1)
Requirement already satisfied: absl-py>=0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.12.0)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.15.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: tensorboard~=2.7 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: keras<2.8,>=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.20.3)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: setuptools>=41.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (58.0.
4)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7-
>tensorflow) (0.6.1)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.0.2)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tens
orflow) (0.4.4)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (1.
23.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorf
low) (1.6.0)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.2
6.0)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (3.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboar
```

```
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboa
rd~=2.7->tensorflow) (4.2.2)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.7->
tensorflow) (4.7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.
4.1->tensorboard~=2.7->tensorflow) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth
<3,>=1.6.3->tensorboard~=2.7->tensorflow) (0.4.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~
=2.7->tensorflow) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.
7->tensorflow) (2022.9.24)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorbo
ard~=2.7->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->ten
sorflow) (3.3)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-o
authlib<0.5,>=0.4.1->tensorboard~=2.7->tensorflow) (3.2.1)
```

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.preprocessing.image import ImageDataGenerator
```

DATA AUGMENTATION

TESTING AND TRAINING

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='5Lj0H6wwDuTqeL545tX4VeIBqI9_mpPA6hiunt9ltdjb',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
```

```
bucket = 'imageclassification-donotdelete-pr-2slksiubpgncif'
object_key = 'Dataset.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
```

```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```
pwd
```

```
'/home/wsuser/work'
```

```
import os
filesnames = os.listdir("/home/wsuser/work/Dataset/TRAIN_SET")
```

```
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train = train_datagen.flow_from_directory(
    r'/home/wsuser/work/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/home/wsuser/work/Dataset/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.
```

```
print(x_train.class_indices)
```

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

```
print(x_test.class_indices)
```

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

```
from collections import Counter as c
```

```
c(x_train .labels)
```

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

INITIALIZATION OF CNN

PERFORMING THE FOLLOWING:

1. Adding the convolution layer
2. Adding maxpooling layer
3. Second Maxpooling snf convolution layers
4. Flattening of layers
5. Adding Dense layer

```
classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Flatten())
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

```
classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 128)               802944

 dense_3 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

COMPILING THE MODEL

```
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

FITTING THE MODEL

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=10, validation_data=x_test,validation_steps = len(x_test))
```

```
/tmp/wsuser/ipykernel_164/3242859618.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Mode
l.fit`, which supports generators.
  classifier.fit_generator(
Epoch 1/10
526/526 [==============================] - 13s 24ms/step - loss: 0.1902 - accuracy: 0.9212 - val_loss: 0.1300 - val_accuracy: 0.9517
Epoch 2/10
526/526 [==============================] - 14s 26ms/step - loss: 0.0204 - accuracy: 0.9943 - val_loss: 0.1104 - val_accuracy: 0.9592
Epoch 3/10
526/526 [==============================] - 12s 24ms/step - loss: 6.9652e-04 - accuracy: 1.0000 - val_loss: 0.1573 - val_accuracy: 0.9517
Epoch 4/10
526/526 [==============================] - 12s 23ms/step - loss: 0.0278 - accuracy: 0.9909 - val_loss: 0.1126 - val_accuracy: 0.9763
Epoch 5/10
526/526 [==============================] - 13s 25ms/step - loss: 2.2342e-04 - accuracy: 1.0000 - val_loss: 0.0651 - val_accuracy: 0.9773
Epoch 6/10
526/526 [==============================] - 13s 24ms/step - loss: 4.3394e-05 - accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9773
Epoch 7/10
526/526 [==============================] - 12s 23ms/step - loss: 3.9238e-04 - accuracy: 1.0000 - val_loss: 0.0528 - val_accuracy: 0.9773
Epoch 8/10
526/526 [==============================] - 13s 24ms/step - loss: 2.9002e-04 - accuracy: 1.0000 - val_loss: 0.0250 - val_accuracy: 0.9867
Epoch 9/10
526/526 [==============================] - 12s 23ms/step - loss: 1.4737e-05 - accuracy: 1.0000 - val_loss: 0.0194 - val_accuracy: 0.9943
Epoch 10/10
526/526 [==============================] - 12s 23ms/step - loss: 9.3164e-06 - accuracy: 1.0000 - val_loss: 0.0232 - val_accuracy: 0.9915
```

SAVING THE MODEL

```
classifier.save('nutrition.h5')
```

```
!tar -zcvf image-classification.model_new.tgz nutrition.h5
```

```
nutrition.h5
```

```
ls -1
```

```
Dataset/
image-classification.model_new.tgz
nutrition.h5
```

```
!pip install watson-machine-learning-client --upgrade
```

```
Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
-client) (0.5.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-c
lient) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learnin
g-client) (1.21.41)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->
boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>
=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-le
arning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-mach
ine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client)
(3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-lear
ning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2
021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(1.20.3)
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials= {"url":"https://us-south.ml.cloud.ibm.com","apikey":"sFYJM2RQbIt39atcqI8BbnhEEJs8l9nBHWLeyeKC0vdq"}
```

```
Space UID = c69b5cda-6240-47d0-9324-3f683ca61ce2
```

```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```
client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c388ddbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
```


```
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
```


```
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ffd6ca2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc          5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
------------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

```
[33]: software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_1.15-py3.6")
      software_spec_uid
```

```
[33]: '2b73a275-7cbf-420b-a912-eae7f436e0bc'
```

```
[34]: software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
      software_spec_uid
```

```
[34]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
[ ]: model_details = client.repository.store_model(model= "image-classification-model_new.tgz",meta_props={
      client.repository.ModelMetaNames.NAME:"imageclassification",
      client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
      client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid}
                                                          )
     model_id = client.repository.get_model_id(model_details)
```

7 CONCLUSIONI:

This webapplication helps people to maintain their health diet,so it help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet

GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-22536-1659853629

DEMO VIDEO LINK:
https://youtu.be/SHMlIZD8Yzw