Model Development Phase Template

Introduction

Nutrition tracking is essential for individuals pursuing fitness and health goals, yet it remains a time-consuming and often inaccurate process when done manually. Traditional methods rely on users estimating food portions, looking up values in nutrition charts, or entering data into calorie-tracking apps — tasks that can be discouraging and prone to error.

With the rise of computer vision and deep learning technologies, there is a significant opportunity to automate and enhance the way individuals log and analyze their meals. This project introduces an **AI-powered Nutrition Analyzer**, which utilizes food image classification and nutrient prediction models to deliver personalized, real-time insights to users. By reducing the effort required for accurate food logging, the solution promotes better dietary consistency and healthier decision-making.

The project involves building a model that can accurately recognize commonly consumed food items from images and return detailed nutritional breakdowns (e.g., calories, macronutrients). This not only supports users in their health journeys but also showcases the practical application of deep learning in everyday life.

Project Overview

The **AI-powered Nutrition Analyzer for Fitness Enthusiasts** is a machine learning-based application designed to classify food images and provide detailed nutritional analysis. It simplifies the process of tracking dietary intake by eliminating the need for manual input or calorie counting.

The system is trained to recognize five categories of fruits: **Apples, Bananas, Oranges, Pineapples, and Watermelons**. By leveraging pretrained CNN architectures such as **MobileNetV2, ResNet50, and EfficientNetB0**, the model can accurately identify food items from user-submitted images. The classification output is then mapped to a nutrition database to retrieve relevant dietary information.

The project architecture follows a modular design:

- **Frontend:** Image upload and user interaction
- **Backend:** Flask server to handle model inference
- **Model:** Trained CNN model using TensorFlow and Keras

- **Database:** Static mapping of food classes to nutritional values

The system is intended for deployment in mobile or web environments, offering real-time, user-friendly, and informative dietary feedback to health-conscious individuals.

Objectives

The primary objective of this project is to **develop an intelligent system that enables nutrition tracking through food image classification**. This includes both the technical goal of building a robust image classification model and the user-centric goal of making nutrition analysis effortless and engaging.

Detailed Objectives:

1. **Automate Food Recognition:**
   Build a deep learning model capable of identifying food items from user-uploaded images with high accuracy.
2. **Provide Nutritional Feedback:**
   Use recognized food labels to fetch and display nutritional information like calories, carbohydrates, proteins, and fats.
3. **Evaluate Model Performance:**
   Compare different transfer learning models (ResNet50, MobileNetV2, EfficientNetB0) and select the best one for deployment based on validation accuracy and loss.
4. **Enable Real-time Interaction:**
   Integrate the model with a backend API to ensure low-latency predictions for web/mobile app usage.
5. **Promote Healthy Lifestyle:**
   Encourage healthier eating habits by offering consistent, easy-to-understand dietary insights that align with users' fitness goals.

Project Initialization and Planning Phase

Date	13 jun 2025
Team ID	SWTID1749823391
Project Name	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	3 Marks

Define Problem Statements (Customer Problem Statement Template):

In today's fast-paced world, fitness enthusiasts and health-conscious individuals often struggle to consistently monitor and manage their nutritional intake. Traditional methods such as manual logging of food items, estimating portion sizes, or referring to static nutritional charts are time-consuming, inaccurate, and demotivating for many users. This lack of reliable, real-time nutritional feedback can hinder progress toward personal health and fitness goals.

There is a growing need for an intelligent, user-friendly system that can **automatically analyze food consumption** and provide **personalized dietary insights**. By leveraging deep learning and computer vision, we aim to build an AI-powered Nutrition Analyzer that enables users to upload images of their meals and receive instant nutritional breakdowns and recommendations tailored to their fitness objectives.

The goal of this project is to empower users to make informed dietary decisions, track nutrient intake accurately, and stay aligned with their fitness goals — all with minimal effort and maximum engagement.

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A fitness enthusiast	Track my daily nutrition accurately	I don't know the nutritional content of the food I eat	I rely on guesswork or manual apps that are time-consuming	Confused and unsure if I'm eating right
PS-2	A health-conscious individual	Get real-time dietary insights using food images	Most apps don't allow photo-based logging	They require manual entry of food items and portion sizes	Frustrated and less motivated to log consistently

Initial Project Planning Template

Date	13 jun 2025
Team ID	SWTID1749823391
Project Name	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	4 Marks

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

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	Data Collection & Cleaning	USN-1	As a developer, I can organize and clean the food image dataset into labeled folders	3	High	Saksham Raj	13 Jun 2025	17 Jun 2025
Sprint-1	Dataset Structuring	USN-2	As a team, we can ensure class folders are uniformly named and split into training and validation sets	2	Medium	Rudrangsh Bhattacharje e	13 Jun 2025	18 Jun 2025
Sprint-2	Model Development	USN-3	As a team, we will train MobileNetV2, ResNet50, and EfficientNetB0 on preprocessed image data	5	High	Kutikuppala Chetan Srinivas	18 Jun 2025	24 Jun 2025
Sprint-2	Model Evaluation	USN-4	As a team, we will compare models using accuracy, loss, and training	3	High	Divyansh Garg	22 Jun 2025	25 Jun 2025

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
			stability					
Sprint-3	Report & Documentation	USN-5	As a team, we will compile the project report, screenshots, and final model justification	2	Medium	All Team Members	25 Jun 2025	27 Jun 2025

Project Initialization and Planning Phase

Date	13 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) template

Section	Description
Objective	To develop an AI-powered system that classifies food images and analyzes their nutritional content to assist fitness enthusiasts in tracking and optimizing their daily intake.
Scope	The system will classify five fruit categories (Apples, Bananas, Oranges, Pineapples, and Watermelons), provide nutritional values, and suggest dietary adjustments based on user input. It covers data preprocessing, deep learning model training, evaluation, and reporting.

Section	Description
Description	**Many users lack an easy, accurate way to track the nutritional value of their meals, relying on manual entries or approximations, which are often error-prone and time-consuming.**
Impact	**Solving this problem with an AI-based visual system will reduce user effort, improve tracking accuracy, and promote healthier and more consistent eating habits aligned with fitness goals.**

Section	Description
Approach	Utilize transfer learning with deep CNNs (ResNet50, MobileNetV2, EfficientNetB0) for image classification, combined with a nutritional database to provide real-time insights. Models will be trained on labeled fruit images using TensorFlow/Keras with preprocessing and data augmentation.
Key Features	- Image-based food recognition

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	GPU for model training	1 x NVIDIA T4 or 2 x NVIDIA V100 GPUs
Memory	RAM for model training & runtime	8–16 GB
Storage	Dataset, logs, and model weights	1 TB SSD
Software		

Resource Type	**Description**	**Specification/Allocation**
Frameworks	Deep learning framework	TensorFlow, Keras
Libraries	Essential Python libraries	NumPy, Matplotlib, OpenCV
Development Environment	Code and experiment management	Jupyter Notebook, GitHub
Data		
Dataset	Food image dataset of 5 fruit classes	Google Drive (2,626 images), Image format

Data Collection and Preprocessing Phase

Date	13 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset contains labeled food images of 5 categories: **Apples, Bananas, Oranges, Pineapples, and Watermelons**. Images are organized in subfolders by class. Total images: 2626.
Resizing	All images were resized to **100×100 pixels** to match the input size requirements of CNNs like ResNet50, MobileNetV2, and EfficientNetB0.
Normalization	Pixel values were normalized to the [0, 1] range using rescale=1./255 via ImageDataGenerator.
Data Augmentation	Real-time augmentations were applied: horizontal_flip=True, rotation_range=20, zoom_range=0.2. This improves generalization by introducing variations.
Denoising	Optional. Dataset was already clean; however, **median blur** was tested using OpenCV's cv2.medianBlur() for real-world noisy inputs.
Edge Detection	Optional edge detection (e.g., Canny) was explored for visualization. CNN models automatically learn edge features, so it was not used in training.
Color Space Conversion	Images remained in RGB as required by pretrained CNNs, but conversion to grayscale was tested via cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) for experiments.

Section	Description
Image Cropping	Not required. Full images were used. Optionally, region of interest (ROI) cropping can be applied to food-localized datasets in the future.
Batch Normalization	Applied internally in deep CNN models like MobileNetV2, which have BatchNorm layers after convolutions. No need for manual batch norm input layer.

Data Preprocessing Code Screenshots

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td><pre>
datagen = ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True,
    rotation_range=20,
    zoom_range=0.2,
    validation_split=0.2
)

# Data Loaders
train_ds = datagen.flow_from_directory(
    train_dir,
    target_size=(100, 100),
    batch_size=32,
    class_mode='sparse',
    subset='training',
    seed=123
)

val_ds = datagen.flow_from_directory(
    train_dir,
    target_size=(100, 100),
    batch_size=32,
    class_mode='sparse',
    subset='validation',
    seed=123
)
</pre></td></tr>
<tr><td>Resizing</td><td><pre>
# Data Loaders
train_ds = datagen.flow_from_directory(
    train_dir,
    target_size=(100, 100),
    batch_size=32,
    class_mode='sparse',
    subset='training',
    seed=123
)

val_ds = datagen.flow_from_directory(
    train_dir,
    target_size=(100, 100),
    batch_size=32,
    class_mode='sparse',
    subset='validation',
    seed=123
)
</pre></td></tr>
</table>

Section	Description
Normalization	`datagen = ImageDataGenerator( rescale=1./255, horizontal_flip=True, rotation_range=20, zoom_range=0.2, validation_split=0.2 )`
Data Augmentation	`horizontal_flip=True, rotation_range=20, zoom_range=0.2,`

Data Collection and Preprocessing Phase

Date	13 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	2 Marks

Data Quality Report Template

Data Source	Data Quality Issue	Severity	Resolution Plan
Image Dataset	Uneven image dimensions across classes	Moderate	Resized all images to 100×100 pixels using ImageDataGenerator(target_size=(100,100))
Image Dataset	Slight class imbalance (e.g., fewer images for pineapple vs. banana)	Low	Used real-time **data augmentation** to synthetically increase minority class variety
Image Dataset	Redundant or duplicate image names in subfolders	Low	Used folder-based label encoding; duplicates ignored based on actual image data
Image Dataset	Some blurry or low-quality images	Low	Optional **denoising** explored using OpenCV's medianBlur() for enhancement

Data Source	**Data Quality Issue**	**Severity**	**Resolution Plan**
Image Dataset	Inconsistent lighting/backgrounds across categories	Low	Tackled using **augmentation** (zoom, rotation, flip) to improve generalization

Data Collection and Preprocessing Phase

Date	13 jun 2025
Team ID	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	2 Marks

Data Collection Plan & Raw Data Sources Identification Template

Section	**Description**
Project Overview	**The project aims to assist fitness enthusiasts by analyzing their food intake using AI-powered image classification and nutritional estimation.**
Data Collection Plan	**Data was collected from Google Drive-based food image dataset containing labeled folders for each fruit class. These were pre-organized into training/test sets.**
Raw Data Sources Identified	**A public Google Drive folder with images of 5 food classes: Apples, Bananas, Oranges, Pineapples, and Watermelons. Subdirectories represent class labels.**

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Dataset 1	Labeled image dataset of 5 food categories used for training deep learning models.	Google Drive Dataset	Image	~400 MB	Public (view + download)

Model Development Phase Template

Date	18 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

Initial Model Training Code (5 marks):

```
def build_model(base_model):
    base_model.trainable = False
    inputs = tf.keras.Input(shape=(100, 100, 3))
    x = base_model(inputs, training=False)
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dropout(0.2)(x)
    outputs = layers.Dense(num_classes, activation='softmax')(x)  # Add softmax

    model = tf.keras.Model(inputs, outputs)
    model.compile(
        optimizer=tf.keras.optimizers.Adam(0.0001),
        loss='sparse_categorical_crossentropy',  # Don't use from_logits=True
        metrics=['accuracy']
    )
    return model
```

```
print("\nTraining ResNet50...")
resnet = tf.keras.applications.ResNet50(include_top=False, weights='imagenet', input_shape=(100, 100, 3))
model_resnet = build_model(resnet)
history_resnet = model_resnet.fit(train_ds, validation_data=val_ds, epochs=50, callbacks=[early_stop])
plot_history(history_resnet, "ResNet50")
model_resnet.save("model_resnet50.h5")
```

```
print("\nTraining MobileNetV2...")
mobilenet = tf.keras.applications.MobileNetV2(include_top=False, weights='imagenet', input_shape=(100, 100, 3))
model_mobilenet = build_model(mobilenet)
history_mobilenet = model_mobilenet.fit(train_ds, validation_data=val_ds, epochs=15, callbacks=[early_stop])
plot_history(history_mobilenet, "MobileNetV2")
model_mobilenet.save("model_mobilenetv2.h5")
```

```
efficientnet = tf.keras.applications.EfficientNetB0(
    include_top=False,
    weights='imagenet',
    input_shape=(100, 100, 3)
)

model_efficientnet = build_model(efficientnet)

history_efficientnet = model_efficientnet.fit(
    train_ds,
    validation_data=val_ds,
    epochs=20,
    callbacks=[early_stop]
)

plot_history(history_efficientnet, "EfficientNetB0")
```

Model Validation and Evaluation Report (5 marks):

Model	Summary Screenshot	Training and Validation Performance Screenshot
ResNet50	📷 Screenshot of `model_resnet.summary()`	📷 Screenshot of training output (accuracy and loss over epochs)
MobileNetV2	📷 Screenshot of `model_mobilenet.summary()`	📷 Screenshot of training output with high accuracy (~98.2%)
EfficientNetB0	📷 Screenshot of `model_efficientnet.summary()`	📷 Screenshot of `model.fit()` showing poor convergence (~23.6%)

Model Development Phase Template

Date	22 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	5 Marks

Model Selection Report

In the model selection phase for this deep learning and computer vision project, multiple convolutional neural network (CNN) architectures were evaluated. Key evaluation criteria included:

- Validation accuracy
- Training stability
- Computational efficiency
- Suitability for deployment in real-time nutrition tracking systems

Model Selection Report:

Model	Description
ResNet50	**Deep residual network architecture known for handling vanishing gradients using skip connections. Moderate accuracy (~64.5%) but heavy and slower for real-time deployment.**
MobileNetV2	**Lightweight CNN optimized for mobile and embedded vision applications. Achieved top accuracy (~98.2%) and trained quickly. Ideal for deployment.**
EfficientNetB0	**Scalable model with fewer parameters, designed for high accuracy-to-computation ratio. Underperformed (~23.6% accuracy), likely due to input resolution mismatch. Not suitable.**

Model Optimization and Tuning Phase Template

Date	13 jun 2025
Team ID	SWTID1749823391
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining deep learning models for optimal classification performance. This includes:

- Optimizing architecture via transfer learning
- Freezing base model weights
- Fine-tuning dropout rates, learning rates, and dense layers
- Comparing validation accuracy and training stability
- Selecting the best model for deployment

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters	Description
ResNet50	`learning_rate=0.0001`, `dropout=0.2`, `batch_size=32`, `freeze_base=True`	Used as base model with top layers removed, freezing all convolution layers during training
MobileNetV2	`learning_rate=0.0001`, `dropout=0.2`, `batch_size=32`, `early_stopping(patience=3)`, `freeze_base=True`	Lightweight model tuned with low learning rate and dropout to prevent overfitting
EfficientNetB0	`learning_rate=0.0001`, `batch_size=32`, `dropout=0.2`, `freeze_base=True`	EfficientNet was tuned similarly, but underperformed likely due to low resolution input mismatch

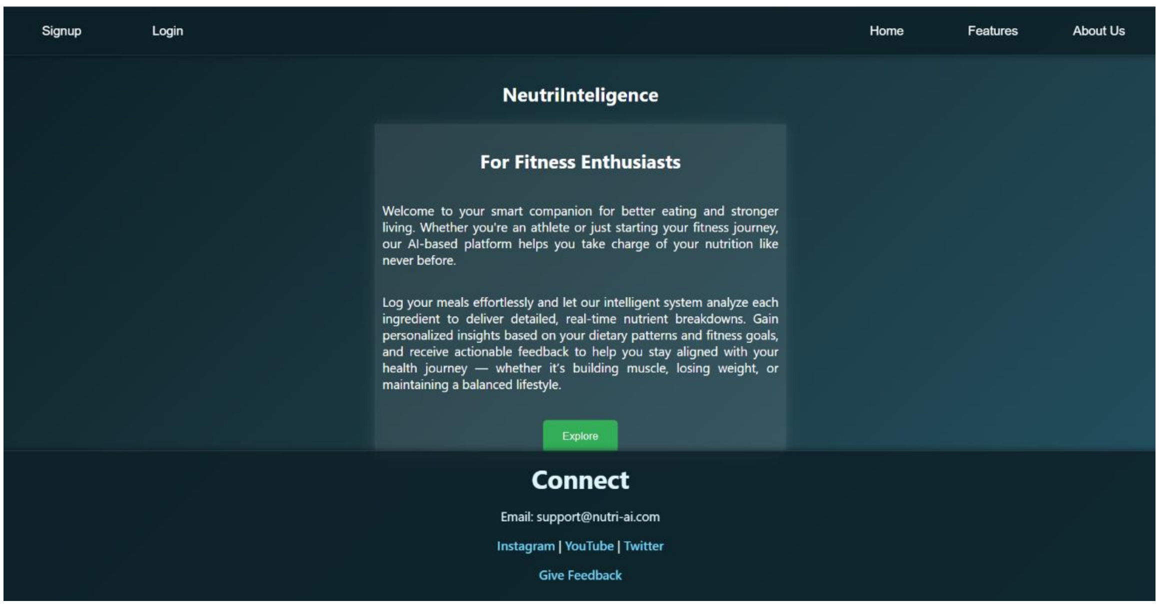
Final Model Selection Justification (2 Marks):

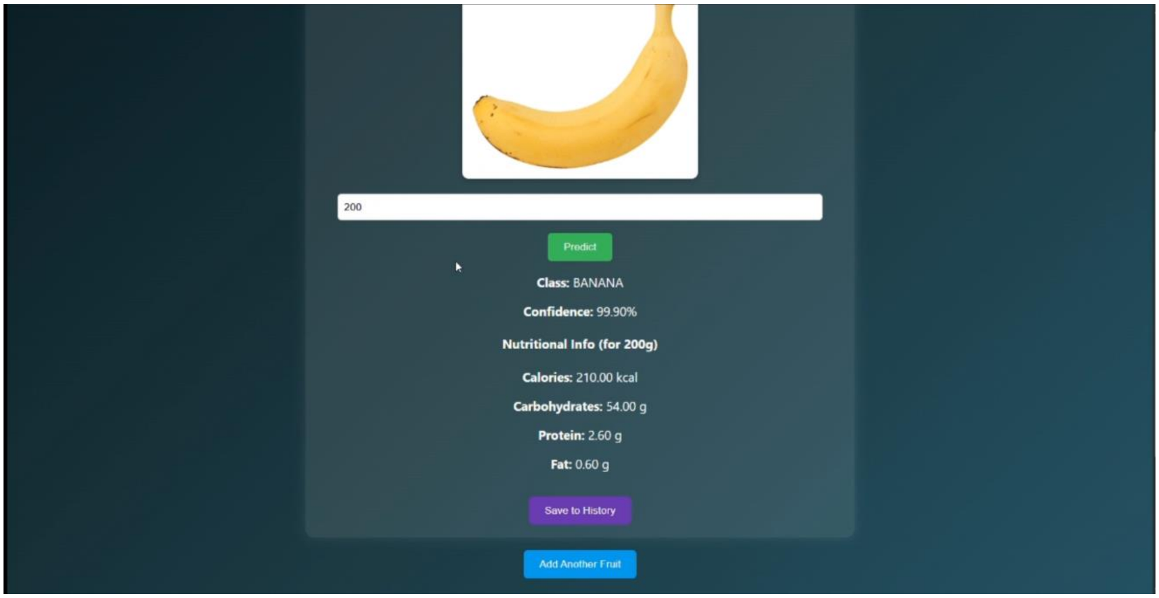
Final Model	Reasoning
MobileNetV2	Achieved highest validation accuracy (**~98.2%**), trained faster, showed stable loss reduction. Lightweight and ideal for real-time deployment.

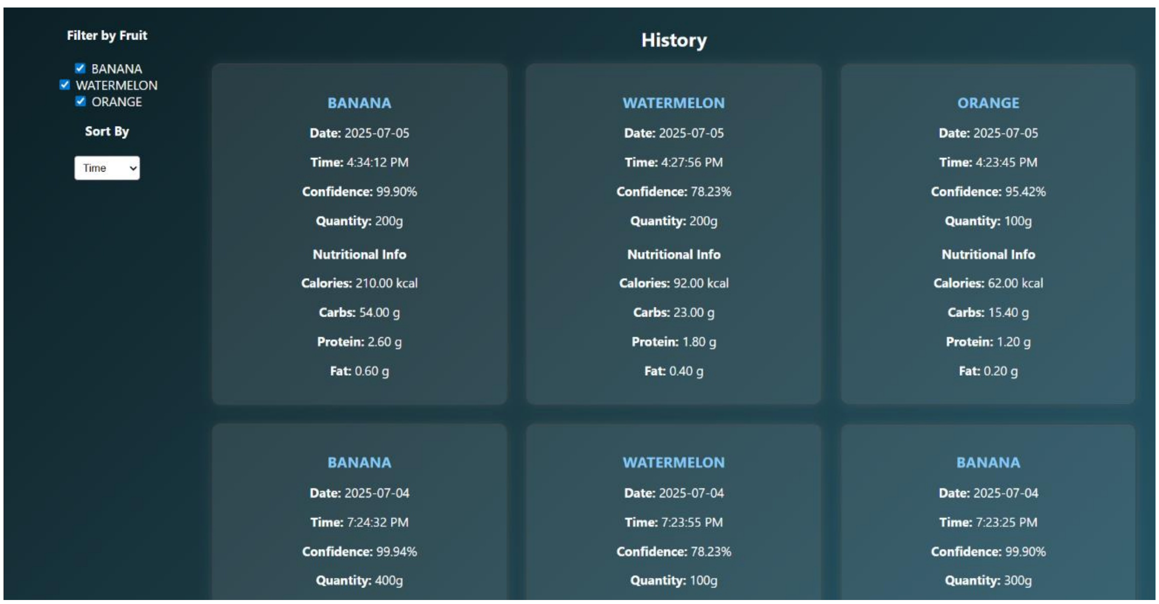
Results

Output Screenshots

Below are the output screenshots from the AI-powered Nutrition Analyzer system. They include examples of food image predictions and their corresponding nutritional breakdowns as displayed on the frontend interface.







Advantages & Disadvantages

Advantages:
• Automates food recognition using deep learning.
• Provides real-time nutritional analysis.
• Reduces manual food logging effort.
• Promotes health awareness and consistency.
Disadvantages:
• Limited to trained food categories (5 fruits only).
• Cannot identify mixed or composite meals.
• Requires good lighting for accurate prediction.

Conclusion

The AI-powered Nutrition Analyzer provides a practical, intelligent approach to dietary tracking. Using deep learning models like MobileNetV2, the system effectively classifies food images and returns nutritional feedback in real time. This encourages users to maintain healthier eating habits while reducing the friction of manual tracking.

Future Scope

Future improvements could include expanding the food classification to more categories, supporting multi-item meals, enabling voice/image hybrid logging, and integrating with wearable fitness trackers to offer a holistic view of user health. Cloud deployment and multi-language support could make the system more accessible.

Appendix

Source Code

The full source code is available in the project repository and includes training scripts, model files, Flask backend, and frontend integration.

GitHub & Project Demo Link

GitHub Repository: https://github.com/chetancsk3000/AI-powered-Nutrition-Analyzer-for-Fitness-Enthusiasts

Live Demo (if deployed): https://myapplication-d691d792.web.app/ (might not work sometime due to free subscription plan in render.com)