AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

❖ **TEAM MEMBERS:**

Team Leader: Syed. Muskan(228X1A05H6).
Team Member: Tatiparthi. Revanth Reddy(228X1A05H7).
Team Member: Gonugunta. Bhavani Sankar Sai(228X1A05I4).

1. Introduction:

The AI-powered Nutrition Analyzer for Fitness Enthusiasts is an innovative solution that uses artificial intelligence (AI) to help individuals analyze, monitor, and optimize their nutritional intake. The system is designed to support fitness goals by processing user dietary data, preferences, and objectives to provide personalized nutritional insights. This project empowers users with data-driven recommendations to maintain a healthy lifestyle effectively.

2. Project Objectives:

• Understand fundamental concepts and techniques of Convolutional Neural Networks (CNNs)
• Gain knowledge on handling and preprocessing image data
• Build and train an AI model for food image classification
• Develop a Flask-based web application for user interaction
• Provide nutritional information using integrated APIs

3. System Scenarios:

Scenario 1: Personalized Meal Planning
Users input preferences and health restrictions to receive balanced meal plans.

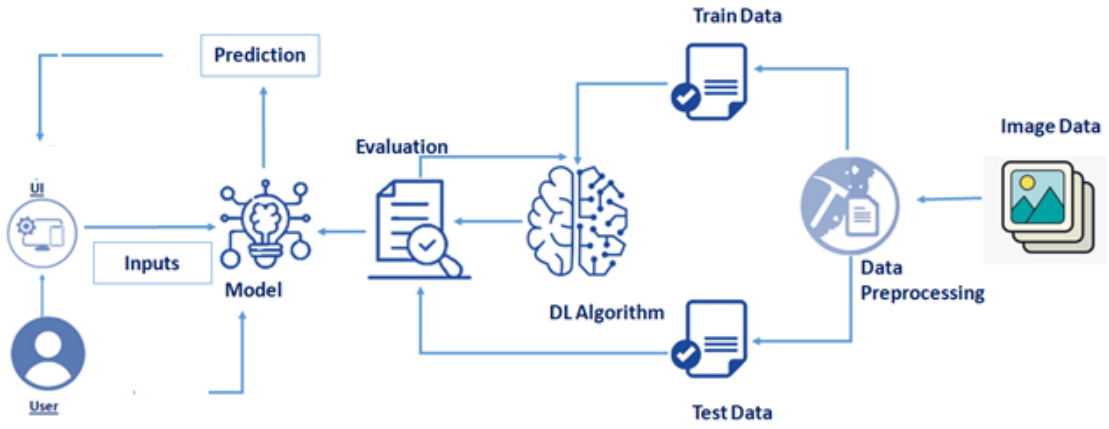
Scenario 2: Nutrient Analysis and Tracking
Tracks food intake and provides real-time nutrient feedback.

Scenario 3: Recipe Enhancement
Suggests recipe modifications for healthier alternatives.

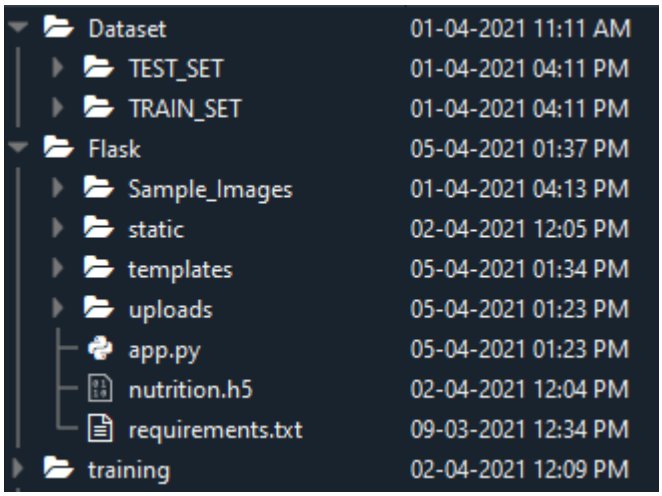
4. Technical Architecture:

The system architecture involves an AI-based model integrated into a Flask web framework. Users interact through a UI that allows uploading food images. The backend model, built using CNNs, classifies the food and retrieves its nutritional information via

API integration.



5.Project Structure & Flow:



➢ Project Flow:

1. User uploads an image through the interface.
2. The Flask backend processes the image.
3. The trained CNN model classifies the food item.
4. Nutrition details are fetched using an external API.
5. Results are displayed to the user.

6. Tools and Prerequisites:

• Anaconda Navigator – for Python environment management
• Flask – web framework
• Python libraries: NumPy, Pandas, TensorFlow, Keras, Scikit-learn, Flask
• IDEs: Jupyter Notebook, Spyder, or PyCharm

7. Methodology:

a. Data Collection

Images of food items (apple, banana, orange, pineapple, watermelon) were collected for training and testing.

b. Data Preprocessing

Images were augmented using ImageDataGenerator for better model performance.

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

```
Image Data Agumentation
```

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

```
### Loading our data and performing data agumentation
```

```
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory(
    r'C:\Users\DELL\Desktop\Desk Files\Nutrition Analysis Using Image Classification\DataSet\TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'C:\Users\DELL\Desktop\Desk Files\Nutrition Analysis Using Image Classification\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.
```

c. Model Building

A CNN model was built with convolution, max pooling, flattening, and dense layers using ReLU and Softmax activation functions.

```
### Importing Neccessary Libraries
```

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

```
model=Sequential()
```

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

```
classifier.summary()#summary of our model
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 128)               802944
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 645
=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

d. Model Training

The model was trained for 20 epochs using Adam optimizer and categorical cross-entropy loss.

```
## Fitting the model
```

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=20, validation_data=x_test,validation_steps = len(x_test))# No of images in test set
```

e. Model Evaluation and Saving

The trained model was evaluated and saved as 'nutrition.h5'.

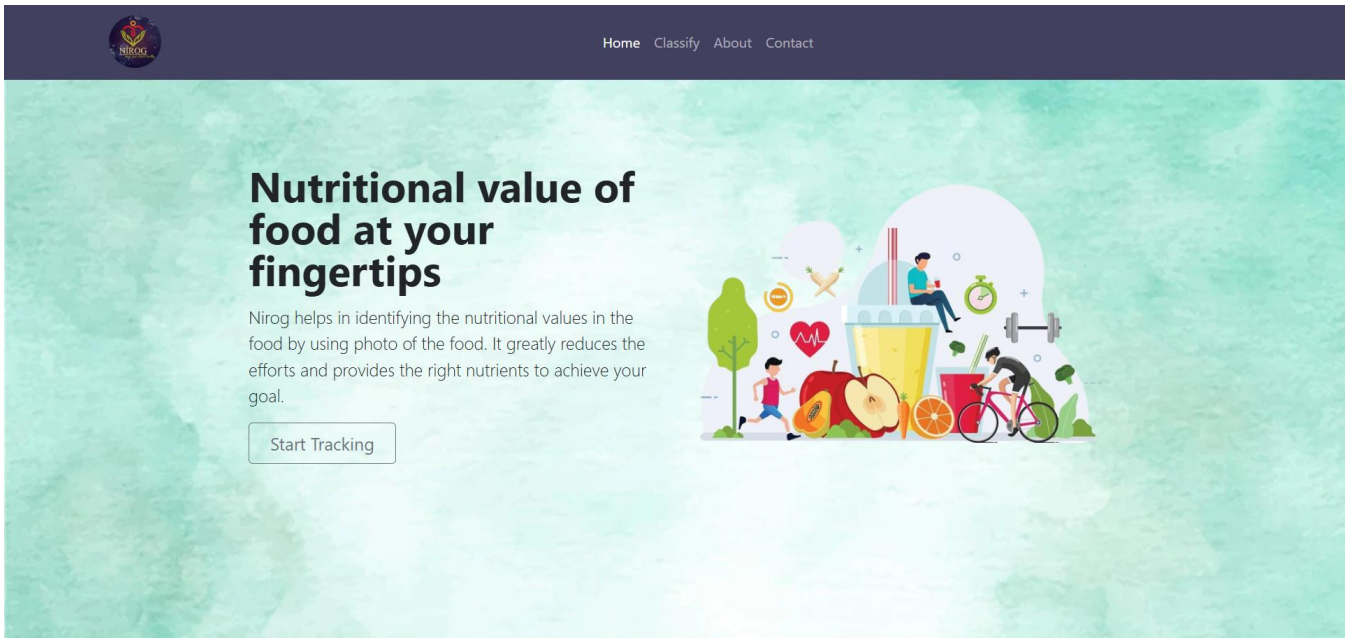
```
### Saving our model

# Save the model
classifier.save('nutrition.h5')
```

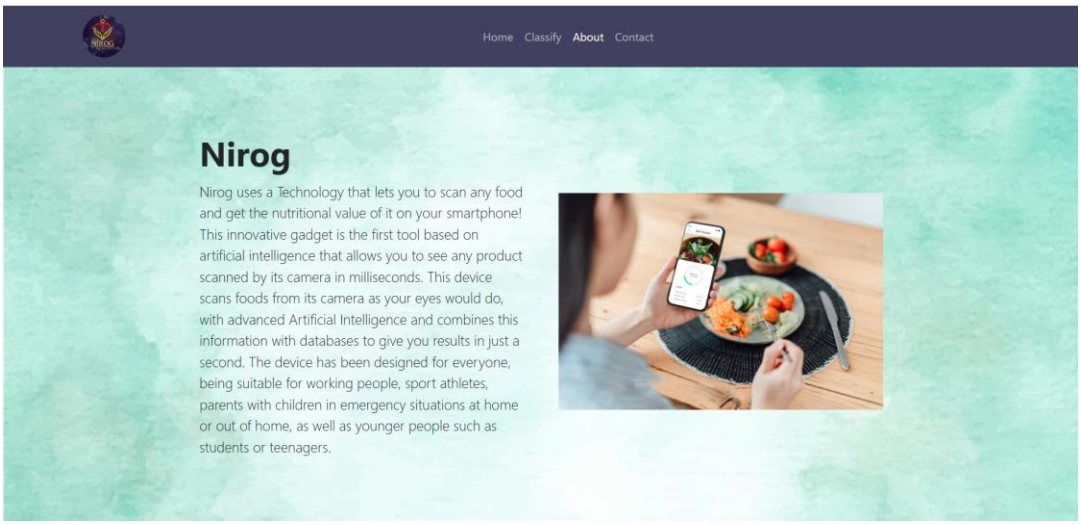
8. Application Development:

The Flask application serves as the bridge between users and the AI model. Users can upload images, and the app predicts the food type and displays nutritional details. HTML, CSS, and JavaScript are used for front-end design.

Home page :



About page :



Result page:

9. API Integration:

The application integrates with RapidAPI to fetch nutritional information for identified food items. The API provides calorie, macronutrient, and micronutrient details, enriching user insights.

```python
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']
```

10. Results:

The trained CNN model successfully identifies food items and provides accurate nutritional analysis. Users can visualize meal data and improve dietary decisions effectively.

11. Conclusion:

The AI-powered Nutrition Analyzer bridges the gap between fitness and technology by providing an automated dietary analysis tool. It leverages AI and web technologies to make fitness tracking easier, personalized, and insightful.

12. Future Enhancements:

- Expanding the dataset with more food categories for improved accuracy
- Implementing voice-based user interaction
- Integrating wearable fitness tracker data
- Adding real-time calorie tracking and recommendation engine
- Deploying on a cloud platform for global access

13. Appendix:

GitHub Repository: https://github.com/syedmuskan122/NutririonAnalysis

Drive Link (Project Demo & Code Folder):
https://drive.google.com/drive/folders/1m0zuTrFq_99rEjGyZO755xodWwPH9A7M

Thank You.