NutriScan AI – Final Project Report

Project Overview:

NutriScan AI is a deep learning powered system that can classify fruit and vegetable images and provide nutrition-related insights.

The system is built using Python and a MobileNet model, enabling fast and reliable predictions. It offers a modern web interface

implemented using HTML, CSS, and JavaScript, and uses Flask as the backend API to handle model requests. The project reduces

manual effort and allows instant identification of food items. The model supports 36 different fruit and vegetable classes.

Objective:

The project's objective is to create a real-time system that predicts the name, category, and calorie content of a fruit or vegetable

from its image. Users can upload an image through the web interface and quickly receive results on the same page. The model

outputs the name of the item and whether it belongs to the fruit or vegetable category. It also provides calorie information and

basic advice. This makes the system useful for nutrition planning, food education, and dietary management.

Technologies Used:

The technologies used include Python, TensorFlow, Keras, NumPy, PIL, Flask, HTML, CSS, and JavaScript. The MobileNet architecture

is employed due to its lightweight nature, which allows it to run efficiently even without high-end computing hardware. Flask handles

API calls seamlessly, while the browser interface makes the system accessible to non-technical users.

Model Description:

The MobileNet model used in this project is pre-trained on a wide dataset and then fine-tuned to classify 36 different fruit

and vegetable images. The input image is resized to 224x224 pixels to match the model's expected input shape. The predictions

are generated using softmax probability distribution, selecting the highest-scoring class. This combination ensures high accuracy.

Dataset:

The dataset consists of apples, bananas, mangoes, potatoes, tomatoes, onions, carrots, and many other fruits and vegetables.

Data augmentation was performed to improve model robustness. The model was trained with Adam optimizer using categorical

cross■entropy loss. Multiple epochs were used to reach satisfactory training accuracy.

System Workflow:

1. User uploads an image from the browser.

2. JavaScript sends the image to the Flask backend using fetch().

3. The backend preprocesses the image and performs model prediction.

4. The result is returned as JSON to the browser.

5. The user sees the name, calories, and category instantly.

Advantages:

The system saves time and automates identification. It can also be extended to mobile applications.

Results and Conclusion:

The system successfully identifies fruit and vegetable images and provides calorie information and category classification.

Users can visually confirm results through preview images on the web interface. The system architecture allows integration

into diet applications, kitchens, schools, or restaurants. It is reliable, fast, and user-friendly.

Future Scope:

The project may be improved to include health charts, storage recommendations, and personalized diet suggestions.

Additional features such as multilingual support, voice output, and smartphone app deployment are also possible.

Final Remarks:

NutriScan AI integrates computer vision and deep learning into a real-world nutrition support system. It demonstrates how

technology can simplify daily tasks such as food identification and calorie tracking. The project offers scalable application

potential and serves as a strong foundation for future AI-based nutrition research.