Food Classification Using CNN

1. Introduction

** Project Objective:

The goal of this project is to build a **deep learning-based food classification system** that can accurately classify food images into **34 categories** using **multiple deep learning models**. The project follows a structured pipeline that includes **dataset preprocessing, model training, evaluation, deployment using Flask, and documentation**.

** Key Highlights:

- Dataset balancing and preprocessing.
- Training **three different models**: **Custom CNN, VGG16, and ResNet**.
- Model validation using **accuracy, precision, recall, and F1-score**.
- Flask-based **web application** for real-time predictions.
- JSON-based **nutritional information storage** for food items.(Food_data.json)
- Well-structured **GitHub repository** and project documentation.

2. Dataset Details

- **Total Classes:** 34 food categories.
- **Dataset Source:** [Food Image Classification Dataset] (provide link if available).
- **Preprocessing Steps:**
  - **Balance the dataset** by selecting an equal number of images per class.
  - **Use 200 images per class** to ensure uniform distribution.
  - **Upload selected images to Google Drive** for easy access.

3. Model Development & Training

We train three different models and evaluate their performance.

A. Custom Deep Learning Model

- **Uses pre-trained ResNet architecture** to classify food images.
- **Fine-tuned to enhance classification accuracy.**
- **Training Parameters:**
- **Epochs:** 30
- **Optimizer:** Adam
- **Performance Metrics:**
  - Accuracy, TP, TN, FP, FN
  - Precision, Recall, F1-Score
  - **Validation Report Saved As:** CustomModel_validation_report.json

B. VGG16 Model (Transfer Learning)

- **Uses pre-trained ResNet architecture** to classify food images.
- **Fine-tuned to enhance classification accuracy.**
- **Training Parameters:**
- **Epochs:** 30
- **Optimizer:** Adam
- **Performance Metrics:**
  - Accuracy, TP, TN, FP, FN
  - Precision, Recall, F1-Score
  - **Validation Report Saved As:** VGG16_validation_report.json

C. ResNet Model (Transfer Learning)

- **Uses pre-trained ResNet architecture** to classify food images.
- **Fine-tuned to enhance classification accuracy.**
- **Training Parameters:**
- **Epochs:** 30
- **Optimizer:** Adam
- **Performance Metrics:**
  - Accuracy, TP, TN, FP, FN
  - Precision, Recall, F1-Score
  - **Validation Report Saved As:** ResNet_validation_report.json

4. Coding Standards & Best Practices

- **Object-Oriented Programming (OOP):**
  - Implement **Classes, Objects, and Constructors** for structured code.
  - **Exception Handling:**
  - Use **try-except** blocks to handle errors smoothly.
  - **Code Readability & Efficiency:**
- **Development Environment:**
  - **PyCharm IDE** for development.

5. Data Collection & JSON Creation

Nutritional Information Extraction

Each food category includes the following **nutritional attributes:**

- **Protein (g)**
- **Fiber (g)**
- **Calories (kcal)**
- **Carbohydrates (g)**
- **Fat (g)**

All data stored in a structured JSON file (Food_data.json).

6. Model Deployment (Flask & Web Interface)

Frontend Features:

- **User Input:**
  - **Upload an image** for classification.
  - **Dropdown selection** for choosing a trained model (**Custom, VGG16, ResNet**).

Backend Processing (Flask API):

- **Load the selected deep learning model dynamically**.
- **Process uploaded images** and perform classification.
- **Return classification results** .

Output Display:

- **Predicted Food Class** based on the input image.
- **Nutritional Information**:
  - **Protein (g)**
  - **Fiber (g)**
  - **Calories (kcal)**
  - **Carbohydrates (g)**
  - **Fat (g)**

7. Submission

Final Deliverables:

- **Trained Deep Learning Models**: intern_task_CustomModel.ipynb, intern_task_VGG16.ipynb, intern_task_ResNET.ipynb.
- **Validation Reports**: CustomModel_validation_report.json, VGG16_validation_report.json, ResNet_validation_report.json.
- **JSON File**: Nutritional data for all 34 classes(Food_data.json).
- **Flask Web Application**: User-friendly prediction system.
- **GitHub Repository**: Well-structured project with README.md.
- **Documentation & PPT**.