**Food Calories Estimation from Image**

Seat No : 22111066

Batch:2022

Project Guide: Nitin Patil

Signature (Students) Signature (Guide)

**Abstract**

This project focuses on developing a comprehensive system for food image detection and calorie estimation using Convolutional Neural Networks (CNNs) and object detection models. By integrating manual feature extraction with CNNs, the system aims to enhance food classification accuracy and provide calorie information for detected food items. The ultimate goal is to aid dietary monitoring and nutritional analysis through an automated approach.

**Domain/Technology/Hardware & Software**

* **Domain:** Computer Vision, Deep Learning, Nutrition
* **Technology:**
  + **Convolutional Neural Networks (CNNs):** For image classification.
  + **Object Detection Models:** Faster R-CNN for detecting and localizing food items in images.
  + **Manual Feature Extraction:** Using OpenCV and scikit-image for extracting color histograms and edge features.
* **Hardware:**
  + **Training:** High-performance computing with GPU support (optional for faster processing).
  + **Inference:** Standard CPUs or GPUs for running the trained models.
* **Software:**
  + **Programming Language:** Python
  + **Frameworks and Libraries:** TensorFlow, Keras, PyTorch, OpenCV, scikit-image, NumPy, Matplotlib

**Development Environment:** Jupyter Note

**Rationale and Significance**

Accurate food image classification and calorie estimation are crucial for dietary monitoring and management, especially for individuals with specific nutritional needs or health conditions. Existing models often lack accuracy and fail to provide comprehensive nutritional information. This project addresses these limitations by combining CNN-based image recognition with detailed manual feature extraction, aiming to enhance prediction accuracy and reliability

### Approach/Methodology Adopted with Justification

#### **Data Collection and Preprocessing**

* **Image Dataset:** A diverse set of food images was collected, covering various types and categories of food.
* **Manual Feature Extraction:** Key visual features such as color histograms and edges were extracted from images using OpenCV and scikit-image.
  + **Color Histogram:** Captures the color distribution in an image by converting it to HSV color space and computing histograms.
  + **Edge Detection:** Utilizes the Canny edge detection method to identify edges, providing information about the shape and structure of food items.
* **Feature Combination:** The extracted features were combined into a single feature vector for each image, which was used alongside the CNN model.

#### **Model Development Convolutional Neural Network (CNN)**

* **Architecture:**
  + **Input Layer:** Accepts preprocessed food images.
  + **Convolutional Layers:** Extracts hierarchical features from images.
  + **Pooling Layers:** Reduces dimensionality and retains essential features.
  + **Dense Layers:** Fully connected layers for classification.
  + **Output Layer:** Softmax activation for multi-class classification.
* **Training:**
  + **Data Augmentation:** Applied to increase the diversity of the training dataset.
  + **Optimization:** Adam optimizer used for training the model.
  + **Loss Function:** Categorical cross-entropy used to compute the loss.
  + **Evaluation Metrics:** Accuracy, precision, recall, and F1-score.

##### **Object Detection Model**

* **Model:** Pre-trained Faster R-CNN (FasterRCNN\_ResNet50\_FPN) for detecting food items in images.
* **Training:** Fine-tuning on the collected food image dataset to improve detection accuracy.
* **Inference:** Detects and localizes food items in images, providing bounding boxes and confidence scores.

#### **Implementation Steps**

1. **Data Preprocessing:** Images were resized, normalized, and augmented to ensure consistency and enhance model robustness.
2. **Feature Extraction:** Manual features (color histograms, edges) were extracted and combined with image data.
3. **Model Training:** The CNN model was trained on the preprocessed dataset, and the object detection model was fine-tuned.
4. **Model Integration:** Combined the outputs of the CNN and object detection models to predict food items and estimate calories.
5. **Visualization:** Detected food items were visualized with bounding boxes and calorie information overlayed on the images.

**Significant Results Obtained**

* **Classification Accuracy:** The CNN model achieved high accuracy in classifying various food items, as demonstrated by the training and validation metrics.
* **Feature Combination:** Integrating manual features with CNN outputs significantly improved classification performance.
* **Object Detection:** The Faster R-CNN model effectively detected and localized food items in images, providing accurate bounding boxes and labels.
* **Calorie Estimation:** The system successfully estimated calorie content for detected food items, displaying this information on the images for easy interpretation.
* **Visualization:** Provided clear visualizations of detected food items and associated calorie information, enhancing user understanding and interaction.

**Suggestions for Further Work**

* **Dataset Expansion:** Incorporate a larger and more diverse set of food images to improve model generalization.
* **Automated Feature Extraction:** Develop automated methods for extracting manual features, reducing the need for manual input and increasing efficiency.
* **Real-time Implementation:** Implement the model in a mobile or web application for real-time food classification and calorie estimation.
* **Additional Features:** Investigate the impact of other potential features, such as ingredient lists or nutritional information, on model performance.
* **Cross-Domain Applications:** Apply the methodology to other domains requiring detailed image classification and analysis, such as medical imaging or industrial quality control.