RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Go, change the world®

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND
MACHINE LEARNING

Project Report

On

Recipe Generation from Food Images

Submitted in partial fulfilment of the requirements for the V Semester
ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING
AI253IA
By

1RV22AI059	**Sujay A.K**
1RV22AI041	**K.M Amogha**
1RV22AI071	**Pranshu Bhatt**

Department of Artificial Intelligence and Machine Learning
RV College of Engineering®
Bengaluru – 560059

Academic year
2024-2025

RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND
MACHINE LEARNING

Bengaluru– 560059

CERTIFICATE

This is to certify that the project entitled “Recipe Generation from Food Images” submitted in partial fulfillment of Artificial Neural Networks and Deep Learning (21AI63) of V Semester BE is a result of the bonafide work carried out by KM Amogha (1RV22AI070), Pranshu Bhatt (1RV22AI071) and Sujay AK(1RV22AI059) during the Academic year 2024-25

Faculty In charge	Head of the Department
Date :	Date :

RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND
MACHINE LEARNING

Bengaluru– 560059

DECLARATION

We, KM Amogha (1RV22AI070), Pranshu Bhatt (1RV22AI071) and Sujay AK(1RV22AI059) students of Sixth Semester BE hereby declare that the Project titled **"Recipe Generation from Food Images "** has been carried out and completed successfully by us and is our original work.

Date of Submission: **Signature of the Student**

ACKNOWLEDGEMENT

We are profoundly grateful to our guide, **Dr. Somesh Nandi**, Assistant Professor, RV College of Engineering, for his wholehearted support, valuable suggestions, and invaluable advice throughout the duration of our project. His guidance and encouragement were instrumental not only in the successful completion of the project but also in the preparation of this report.

We also extend our special thanks to **Dr. Anupama Kumar** for her invaluable insights, support, and constructive feedback, which significantly contributed to the improvement of our work.

We would like to express our sincere thanks to our Head of the Department, **Dr. Satish Babu**, for his constant encouragement and for fostering an environment of innovation and learning that greatly aided our progress.

We extend our heartfelt gratitude to our beloved Principal, **Dr. K. N. Subramanya**, for his unwavering appreciation and support for this Experiential Learning Project, which motivated us to give our best.

Lastly, we take this opportunity to thank our family members and friends for their unconditional support and encouragement throughout the project. Their backup and motivation were crucial in helping us overcome challenges and successfully complete our work.

ABSTRACT

The "FoodGenius" project is an innovative application that transforms food images into detailed recipes using advanced AI technologies. This project addresses the challenge of generating accurate and diverse cooking recipes from food photographs, a task that traditionally requires human expertise and creativity. By leveraging state-of-the-art computer vision and natural language processing techniques, FoodGenius automates the process of recipe generation, making it accessible and efficient for users worldwide.At the core of the project is a hybrid model that combines Convolutional Neural Networks (CNNs) and Transformer-based architectures.

The CNN component, specifically a ResNet101 model, is responsible for extracting rich visual features from food images. These features are then fed into a Transformer decoder, which generates coherent and contextually relevant recipes, including ingredient lists and step-by-step cooking instructions. FoodGenius is implemented as a web application using Flask, providing a user-friendly interface for uploading images and viewing generated recipes. The application supports real-time processing, delivering instant results to users. Additionally, the system is designed to run efficiently on both CPU and GPU, ensuring scalability and performance.

This project demonstrates the potential of combining computer vision and AI to solve practical challenges in the culinary domain. It offers a glimpse into the future of automated cooking assistance, where AI can assist chefs and home cooks alike in exploring new culinary possibilities. By automating recipe generation, FoodGenius not only saves time but also inspires creativity in the kitchen, making it a valuable tool for food enthusiasts and professionals.

Table of Contents

List of Figures

Chapter 1: Introduction

This chapter gives the description of the project Food recipe generation system .It also includes theory and concepts used followed by report organization

1.1 Project Description

Basic Introduction of the Project

The advent of artificial intelligence in culinary arts has opened new possibilities for recipe creation and customization. This project focuses on a novel approach to food recipe generation using deep learning techniques. By analyzing images of food, the system generates detailed recipes, including ingredients, cooking instructions, and alternate variations. A key feature of the project is its ability to allow users to impose restrictions on specific ingredients or parameters, enabling the generation of customized recipes that cater to dietary preferences, allergies, or availability of ingredients.

The primary aim is to bridge the gap between visual representation and culinary creativity, empowering users to transform food imagery into actionable cooking guidance. This innovation not only aids cooking enthusiasts but also offers a unique tool for food bloggers, culinary professionals, and nutritionists to explore recipe variations efficiently.

Objectives of the Project

1. **Automated Recipe Generation:** Develop a model capable of analyzing food images to produce accurate and detailed recipes.
2. **Ingredient-Based Customization:** Provide users with the flexibility to exclude specific ingredients or define parameters to generate alternate recipes.
3. **Enhanced Accessibility:** Create a user-friendly interface for recipe generation, catering to a wide range of culinary preferences and requirements.
4. **Integration of AI Techniques:** Utilize state-of-the-art neural networks, such as CNNs and transformers, to achieve high accuracy in ingredient detection and recipe formulation.
5. **Promote Sustainable Cooking:** Encourage resourceful cooking by suggesting ingredient substitutes and alternative recipes based on user-defined restrictions.

Theory and Concept Relevant to the Project

This project combines key concepts in computer vision, natural language processing, and deep learning to achieve its objectives:

1 **Convolutional Neural Networks (CNNs):** Utilized for feature extraction from food images, enabling the identification of visual patterns that correspond to ingredients and dishes.

2 **Attention Mechanisms:** Implemented for generating coherent and contextually accurate recipes, ensuring that each ingredient and instruction is logically related to the image input.

3 **Transformer Models:** Deployed for natural language generation to produce detailed cooking instructions and alternate recipe variations.

4 **Multimodal Learning:** Integrates image and textual data for comprehensive recipe creation.

5 **User-Defined Constraints:** Facilitates customization by incorporating additional input parameters, such as excluded ingredients, into the recipe generation pipeline

1.2 Report Organization

The report is structured to provide a comprehensive understanding of the food classification and recipe generation system.

It begins with the **Introduction**, offering an overview of the project's background, significance, and objectives in developing an AI-driven system for food recognition and automatic recipe generation. The introduction highlights the challenges in food classification and recipe retrieval and explains how machine learning and deep learning techniques are employed to enhance accuracy and efficiency.

The **Project Description** elaborates on the scope, methodologies, and expected outcomes, setting the foundation for the rest of the report. It outlines the two-stage model approach—first for food classification from images and second for recipe retrieval based on classification results.

The **Report Organization** section guides readers through the structure of the document, ensuring clarity and logical progression. Following this, the **Literature Review** explores previous research in food classification, existing recipe generation methods, and the technologies used, including deep learning architectures, dataset preprocessing techniques, and hardware/software requirements.

The **Software Requirement Specifications** section describes the system's functional and non-functional requirements, external interfaces, and design constraints, providing a detailed overview of system capabilities and limitations.

Next, the **System Design** segment presents the architectural design, data flow diagrams, and a detailed description of the deep learning models used. It explains how the classification model processes food images and how the recipe generation model retrieves relevant recipes using structured datasets. The integration of pickle files for storing classified food items, ingredients, and instructions is also discussed.

The **Implementation** section showcases key code snippets and discusses the results with supporting screenshots, demonstrating the effectiveness of the food classification model and the accuracy of the recipe retrieval system. Performance metrics such as accuracy, precision, and inference speed are analyzed to assess model efficiency.

The report concludes with a **Conclusion**, summarizing the project's achievements and its significance in the domain of AI-powered food recognition and recipe generation. The **Future Enhancements** section suggests possible improvements, such as refining the dataset, optimizing model performance, and expanding the system to include user-customized recipes.

Finally, the **References** section lists all cited sources.

Chapter 2: Literature Review

2.1 Literature Survey

[1] B. Jamalpur, "Optimizing Food Image Classification Using Black Widow Algorithm and Deep Learning Techniques", proposes the FIC-BWODL model for food image classification, which combines preprocessing using CLAHE, feature extraction with CapsNet, and classification using CAE. The Black Widow Optimization Algorithm is utilized for hyperparameter tuning, resulting in improved performance by adapting to the specific features of the dataset.

[2] H. V. Hasti, "Image Processing and Machine Learning Methods for Assessing Food Quality", focuses on advanced image processing and machine learning methods for food quality assessment. The approaches include CNN-based wheat contamination detection, achieving 92.57% accuracy for apple freshness estimation and 97.33% accuracy for apple bruise classification using hyperspectral imaging. The study also highlights automated meat quality scanning and cost-effective rice classification techniques, enhancing industry efficiency and competitiveness.

[3] S. P. Singh, "Food-Lens: Improving Culinary Experiences with AI-Driven Meal Analysis and Recipe Generation", introduces an AI-based system leveraging CNN and MobileNetV2 to identify food items from images and generate corresponding recipes, including ingredients and instructions. The system encourages healthier eating habits through features like personalized recipe suggestions and text-to-speech capabilities, enhancing the culinary experience

[4] R. Krutik, "Advancements in Food Recognition: A Comprehensive Review of Deep Learning-Based Automated Food Item Identification", provides an in-depth exploration of advancements in food recognition using deep learning technologies. It highlights the role of these methods in dietary monitoring and nutritional analysis, examining motivations, state-of-the-art architectures, and publicly available datasets. Additionally, the study identifies existing research gaps and suggests future directions for progress in the field.

[5] P. Chhikara, "FIRE: Food Image to Recipe Generation", introduces a multimodal methodology called FIRE for generating recipes from food images. The study employs BLIP for title generation, Vision Transformers for ingredient identification, and T5 for instruction generation. The system effectively integrates advanced deep learning models to streamline recipe creation from visual inputs, enhancing the automation of food-related AI applications.

[6] E. D. Cherpanath, "Food Image Recognition and Calorie Prediction Using Faster R-CNN and

Mask R-CNN", presents a deep learning-based system aimed at addressing health concerns related to obesity through food image recognition and calorie prediction. The study utilizes Faster R-CNN and Mask R-CNN to capture food images, classify them, and predict calorie content. This approach enhances dietary awareness and aids in calorie control, contributing to the prevention of obesity-related conditions.

[7] E. I. Georga, "A Review of Image-Based Food Recognition and Volume Estimation Artificial Intelligence Systems", provides a comprehensive review of AI systems for food recognition and volume estimation using smartphone images. The paper explores techniques for segmentation, classification, and volume computation, offering insights into the strengths and limitations of current methods. It further discusses potential improvements and future directions for advancing dietary assessment technologies.

[8] T. M. L. Rosaline, "AI-Powered Mobile Application for Image-Based Food Ingredient Detection and Recipe Generation", introduces a mobile app utilizing YOLOv8 and CNN for food ingredient detection and nutrition assessment. The model, trained on 24,583 images, demonstrates improved accuracy in ingredient identification, calorie estimation, and recipe generation. The research highlights future work, including expanding the dataset to enhance precision and reliability.

[9] M. S. M. Rabby, "A Modified Transfer Learning-Based Framework for Efficient Food Image Classification", investigates food image classification using EfficientNetB7, ResNet50, and VGG19 on the Food-11 dataset. The study shows that EfficientNetB7 achieved superior performance with 87.38% validation accuracy, while augmented VGG19 variants reached up to 83% accuracy, emphasizing the importance of augmentation and regularization in improving model performance.

[10] P. K. Singh, "Transfer Learning using Very Deep Pre-Trained Models for Food Image Classification", evaluates deep pre-trained CNN architectures for food image classification on the Food-101 dataset, which includes 101 food categories. The study found that Xception outperformed Inception-v3, EfficientNet-B0, DenseNet-121, and MobileNet, achieving the highest accuracy of 84.54%, highlighting the effectiveness of advanced architectures in improving food classification tasks

[11] N. V. Sai Manoj, "Enhanced Food Classification System Using YOLO Models for Object Detection Algorithm", evaluates YOLOv5 and YOLOv7 for food classification, demonstrating that YOLOv5 outperforms the other with an accuracy of 0.851, a recall of 0.836, and a 0.892 mAP at 0.5 IoU. The study highlights YOLOv5's potential in improving food categorization, nutrition assessment, and ensuring safety in the food industry.

[12] J. Sultana, "A Study on Food Value Estimation From Images: Taxonomies, Datasets, and Techniques", reviews the automation of nutritional value estimation from food images using deep learning. It categorizes research efforts, evaluates performance metrics such as accuracy and precision, and discusses current trends, challenges, and opportunities. This study provides valuable insights for researchers, health practitioners, and nutritionists in the field of food value estimation.

[13] S. Chaudhary, "ChefAI.IN: Generating Indian Recipes with AI Algorithm", explores the generation of Indian recipes using the Autochef algorithm, which incorporates mutation and similarity techniques. The system utilizes models like NLP and LSTM to develop and refine recipes, with the goal of improving the accuracy and efficiency of recipe creation specifically for Indian cuisine.

[14] K. Srinivas, "Self-Attention Architecture for Ingredients Generation from Food Images", introduces an image-to-recipe generation system specifically for Indian cuisines. The system uses a self-attention-based architecture to predict a dish's title, ingredients, and cooking instructions from its image, enhancing the relevance and accuracy of recipe generation for Indian dishes.

[15] D. P. Papadopoulos, "Learning Program Representations for Food Images and Cooking Recipes", proposes a method to generate cooking programs from food images and recipes using a Vision Transformer-based encoder for images, a Transformer-based encoder for recipes, and a program decoder. This approach aligns image and recipe embeddings in a shared space through self-supervised learning, enabling effective program generation.

[16] G. A. Papakostas, "An Update on Cooking Recipe Generation with Machine Learning and Natural Language Processing", reviews recipe generation methods that use user-provided ingredient lists or suggested ingredients. By leveraging advancements in natural language processing and deep learning, the paper highlights innovative, personalized, and healthier recipe creation techniques, offering inspiration and context for culinary endeavors.

[17] H. Wang, "Learning Structural Representations for Recipe Generation and Food Retrieval", introduces an unsupervised method to create level tree structures for cooking recipes, enhancing recipe generation and food cross-modal retrieval tasks. Using ON-LSTM, the model extracts paragraph structures, generates recipe trees from images, and integrates them into generation and retrieval frameworks.

[18] M. Goel, "Ratatouille: A Tool for Novel Recipe Generation", introduces Ratatouille, a web application for generating unique recipes. The system employs neural network-based LSTMs and the transformer-based GPT-2 model. By including ingredient quantities, GPT-2 achieved superior performance, with a BLEU score of 0.806, surpassing LSTM-based methods in recipe generation.

[19] Y. Sun, "Image-Based Food Classification and Volume Estimation for Dietary Assessment: A Review", reviews methods for image-based dietary assessment, focusing on algorithms and models for food recognition and volume estimation. It identifies challenges in improving accuracy, speed, and efficiency, emphasizing the potential of integrated systems combining deep learning and complementary approaches to enhance dietary intake assessment.

[20] A. S. Metwalli, "Food Image Recognition Based on Densely Connected Convolutional Neural Networks", presents DenseFood, a densely connected convolutional neural network for food image recognition. By utilizing softmax and center loss functions, DenseFood enhances intra-category consistency and inter-category distinction. Tested on the VIREO-172 dataset, it achieved 81.23% accuracy, outperforming fine-tuned DenseNet121 and ResNet50 models.

Summary of Literature Survey

- Advanced Food Recognition: Techniques like CNN, YOLO models , and MobileNetV2 are extensively used for accurate food recognition, classification, and calorie estimation. Studies demonstrate YOLOv5's superior performance in metrics like accuracy, recall, and mAP.
- Recipe Generation from Images: Systems using Vision Transformers, self-attention architectures, and Transformer-based models effectively predict dish titles, ingredients, and instructions from food images. Integration of models like BLIP and T5 enhances multimodal recipe generation.
- Dietary Assessment Innovations: Research highlights advancements in food volume estimation and nutritional value prediction using computer vision and deep learning. Reviews emphasize challenges in accuracy, efficiency, and dataset diversity.
- AI for Indian Cuisine: Specialized approaches like ChefAI.IN and self-attention architectures focus on generating Indian recipes. These systems integrate NLP, LSTM, and mutation techniques to create culturally relevant and precise recipes.
- Personalized Recipe Suggestions: AI-powered applications using YOLOv8 and CNN provide ingredient detection, nutrition assessment, and recipe personalization. Tools like Ratatouille leverage GPT-2 for innovative recipe creation, achieving high BLEU scores.
- Interdisciplinary Approaches: Integration of AI and hardware systems, such as optical neural networks and smartphone-based applications, drives real-world deployment in food recognition and dietary monitoring.

- Enhanced Learning Models: Transfer learning with models like EfficientNetB7, Xception, and DenseFood achieves high accuracy for food image classification. DenseFood, with its densely connected architecture, outperforms traditional models in challenging datasets.
- Real-Time Applications: AR-based mobile applications integrate OCR, YOLO, and CNN for ingredient detection and real-world usage.
- Deep Learning and NLP: Studies utilizing Transformer-based encoders and decoders align image and recipe embeddings for cooking program generation. These methods leverage advancements in self-supervision and paragraph structure modeling for efficient cross-modal retrieval.
- Future Directions**:** Reviews on food recognition and recipe generation identify gaps in dataset diversity, real-world testing, and computational efficiency, providing a roadmap for future innovations in AI-driven culinary applications.

2.2 Existing and Proposed System

❖ **Existing system**

Food recognition and recipe generation have traditionally relied on manual methods or isolated algorithms. Existing systems often focus on single components, such as food classification or calorie estimation, without integrating advanced techniques like deep learning or multimodal processing. These approaches lack accuracy, especially in real-world scenarios, and fail to provide a comprehensive solution for ingredient detection, recipe generation, and dietary assessment.

❖ **Proposed system**

The proposed system leverages a hybrid model combining Convolutional Neural Networks (CNNs) and Transformer-based architectures for food recognition and recipe generation. Specifically, a ResNet101 model is used for extracting visual features from food images, which are then processed by a Transformer decoder to generate recipes, including ingredient lists and cooking instructions. The system is designed to be scalable for real-time applications and extensible to personalized dietary recommendations.

❖ **Problem statement**

Accurate food recognition and recipe generation are critical for dietary monitoring, but existing systems face challenges in real-world applications due to low accuracy, dataset limitations, and lack of integration between recognition and recipe generation. The goal is to develop an intelligent system that automates these tasks while maintaining high accuracy, reliability, and usability.

❖ **Scope of project**

➢ **Food Recognition:** Accurate identification of food items using deep learning models, specifically CNNs.

- **Ingredient Detection:** Precise identification of ingredients using Transformer-based models.
- **Recipe Generation:** Automated recipe generation using advanced Transformer-based models.
- **User Experience:** A seamless interface for viewing food details and recipes, with options to save and personalize results.
- **Extensibility:** The system can be extended to include diverse cuisines and additional food-related tasks.

❖ **Methodology of proposed system**

- Input Image Acquisition: The user provides a food image, which is resized and preprocessed for analysis.
- Food Recognition: The image is processed using CNNs to identify the food item.
- Ingredient Detection: Ingredients are identified using Transformer-based models.
- Recipe Generation: The identified food and ingredients are used by Transformer-based models to generate a step-by-step recipe.
- Output Display: The final output, including the food name and recipe, is presented in a user-friendly format, with options to save or share results.

❖ **Technical features of proposed system:**

- Real-Time Food Recognition: High accuracy in identifying food items from images using CNN models.
- Ingredient Detection: Transformer-based models enable precise detection of ingredients.
- Recipe Generation: Integration of Transformer models for recipe creation.
- Scalable Architecture: Modular design allows easy integration of additional features and support for diverse cuisines.
- User-Friendly Interface: Intuitive displays for food recognition and recipes.
- Error Handling: Robust mechanisms to manage poor image quality or ambiguous ingredients.
- Personalization: AI-driven recommendations for customized recipes.
- Extensibility: Designed for future applications in dietary monitoring and expanded food databases.

2.3 Tools and Technologies Used

❖ Programming language : Python(3.7)

❖ Libraries & frameworks :

- Flask (for web application)
- PyTorch (for deep learning models)

- ➢ TorchVision (for image processing and models)
- ➢ NumPy (for numerical computations)
- ➢ Matplotlib (for image visualization)
- ➢ Transformers (for implementing transformer models)

❖ Models & Architecture :

- ➢ ResNet101 (for image feature extraction)
- ➢ Transformer Decoder (for recipe generation)
- ➢ Multi-head Attention (for ingredient detection)

❖ Dataset : RecipeNLG,

2.4 Hardware and Software Requirements

❖ **Hardware Requirements**

Processor: Intel i5/i7 or equivalent with at least 4 cores
RAM: 8 GB (minimum), 16 GB (recommended)
Storage: At least 20 GB of free space
Graphics: NVIDIA GPU with CUDA support (recommended for faster inference)

❖ **Software Requirements**

Operating System: Windows 10/Linux/MacOS
Python 3.7 or higher
Required Libraries:

- ❖ PyTorch >= 1.7.0
- ❖ TorchVision >= 0.8.0
- ❖ Flask >= 2.0.0
- ❖ NumPy >= 1.19.0
- ❖ Matplotlib >= 3.3.0

Web Browser: Chrome/Firefox/Safari (latest version)

Chapter 3: Software Requirement Specifications

3.1 Introduction

Definitions, Acronyms, and Abbreviations:

- AI: Artificial Intelligence
- NLP: Natural Language Processing
- CV: Computer Vision
- ML: Machine Learning
- UI: User Interface

Overview: This project is an innovative application that leverages AI to transform food images into detailed recipes. This document outlines the software requirements necessary to develop and deploy the application effectively.

3.2 General Description

Product Perspective:
This project is a standalone web application that integrates advanced AI technologies to provide users with recipe suggestions based on food images. It is designed to be user-friendly and accessible across various devices.

Product Functions:

- Image recognition to identify food items and ingredients.
- Recipe generation using NLP to provide step-by-step cooking instructions.
- Multiple recipe suggestions for a single image input.
- User-friendly interface for easy navigation and interaction.

User Characteristics:

The target users are cooking enthusiasts, food bloggers, and anyone interested in exploring new recipes. Users are expected to have basic internet browsing skills.

General Constraints:

- Requires a stable internet connection for optimal performance.
- Dependent on the availability of the Food101 & Recipe 1M datasets for training.
- Limited by the processing power of the user's device for real-time image analysis.

Assumptions and Dependencies:

- Assumes users have access to a device with a web browser.
- Depends on third-party libraries and frameworks such as PyTorch and Flask.

3.3 Functional Requirements

Introduction: This section details the core functionalities of the FoodGenius application.

Input:

- Users upload food images through the web interface.
- Optional user inputs for dietary preferences or restrictions.

Processing:

- Image analysis using computer vision to detect ingredients.

- Recipe generation using machine learning models.
- Data processing to provide multiple recipe options.

Output:

- Display of detailed recipes with ingredients and cooking instructions.
- Visual representation of identified ingredients.

3.4 External Interfaces Requirements

User Interfaces:

- Intuitive web interface with easy navigation.
- Responsive design for compatibility with various devices.

Hardware Interface:

- Compatible with standard web-enabled devices (PCs, tablets, smartphones).

Software Interface:

- Integration with machine learning libraries (e.g., PyTorch).
- Utilizes Flask for backend operations.

3.5 Non-Functional Requirements

- Performance: The application should process images and generate recipes within a few seconds.
- Scalability: Capable of handling multiple users simultaneously.
- Reliability: Consistent performance with minimal downtime.
- Usability: Easy to use with a clean and intuitive interface.
- Security: Secure handling of user data and images.

3.6 Design Constraints

Standard Compliance:

- Adheres to web development standards and best practices.
- Follows AI ethics guidelines for data usage.

Hardware Limitations:

- Performance may vary based on the user's device capabilities.

Other Requirements:

- Regular updates to improve AI model accuracy and user experience

Chapter 4 : System Design

4.1 Architectural Design

Problem specification

The primary goal of this project is to develop an intelligent system capable of generating a recipe from an uploaded food image. This involves processing the input image, extracting relevant features using deep learning models, and mapping these features to ingredients and cooking instructions. The system should efficiently predict recipes from food images, making it useful for cooking enthusiasts, nutritionists, and AI-driven culinary applications.

The system accomplishes the following objectives:

- **Process the input food image** to extract key visual features such as texture, color, and shape.
- **Recognize the dish type** using a deep learning model (ResNet-101) to classify the image and extract meaningful patterns.
- **Predict the ingredients** present in the dish based on the extracted features.
- **Generate a step-by-step recipe** by mapping the predicted ingredients to a structured recipe using the RecipeNLG dataset.
- **Display the generated recipe** with ingredients, quantities, and cooking instructions to the user in a user-friendly format.
- **Provide alternative recipe suggestions**

Module specification

Module 1: Data Preprocessing

Input:

The inputs for this module consist of images of various food dishes. These images are sourced from the RecipeNLG dataset and other publicly available food image datasets. Each image is associated with metadata such as the dish name, ingredients, cooking instructions, and cuisine type.

Process:

1. Resizing: Images are resized to a standardized dimension of 224x224 pixels to ensure compatibility with the ResNet-101 model used for feature extraction.
2. Normalization: Pixel values are scaled to a range of 0 to 1, ensuring consistency across the dataset and optimizing model performance.
3. Feature Extraction: The preprocessed images are fed into a ResNet-101 CNN model, which extracts meaningful features such as texture, color, and shape of food items.

Output:

The output of this module is a preprocessed dataset that is cleaned, normalized, and augmented. It contains extracted feature vectors from food images, which will be used in the next stage for ingredient and recipe generation.

Module 2: Food Image Classification

Input:

The input consists of the preprocessed food images from Module 1. These images contain important visual features that need to be classified into different food categories.

Process:

1. Feature Vector Input: The extracted feature vectors from ResNet-101 are used as input.
2. Classification Model: A deep learning classification model (such as ResNet-101 or a custom CNN) is trained to recognize food categories.
3. Label Prediction: The model predicts the most probable dish name based on the input image.

Output:

The output is a predicted food category (e.g., "Pasta," "Burger," "Salad"), which serves as a crucial step in mapping the image to its respective ingredients and recipe.

Module 3: Ingredient Prediction

Input:

The input consists of the predicted food category from Module 2 and the feature vector from Module 1.

Process:

1. Mapping to Ingredients: The predicted dish is matched with a list of possible ingredients from the RecipeNLG dataset.
2. Ingredient Probability Model: A machine learning model predicts the most likely ingredients based on the dish type and visual features.
3. Filtering & Refinement: The system removes irrelevant ingredients and refines the list based on contextual factors such as dish variations.

Output:

A structured list of ingredients required to prepare the detected dish is generated.

Module 4: Recipe Generation

Input:

The input consists of the predicted dish name and its list of ingredients from Module 3.

Process:

1. Recipe Retrieval: The system searches for similar recipes in the RecipeNLG dataset.
2. Text Generation Model: If no exact recipe is found, an AI-based text generation model (such as an LSTM or Transformer-based model) formulates a step-by-step cooking procedure based on the predicted dish and ingredients.
3. Formatting: The generated recipe is structured into ingredients, step-by-step cooking instructions, and estimated cooking time.

Output:

The output is a complete recipe description containing:

- The dish name
- List of ingredients
- Step-by-step cooking instructions

Module 5: Recipe Presentation & User Interaction

Input:

The input consists of the generated recipe from Module 4.

Process:

1. User Interface Display: The recipe is displayed in an interactive UI with a visually appealing layout.
2. Alternative Suggestions: The system suggests similar recipes based on user preferences.
3. Saving & Sharing: Users can save the recipe, modify ingredients, or share it via social media.
4. Feedback Mechanism: Users can provide feedback on the generated recipe to help improve future predictions.

Output:

The final output is a fully formatted and interactive recipe that users can view, modify, or share.

4.2 Data Flow Diagram (DFD)

Level 0

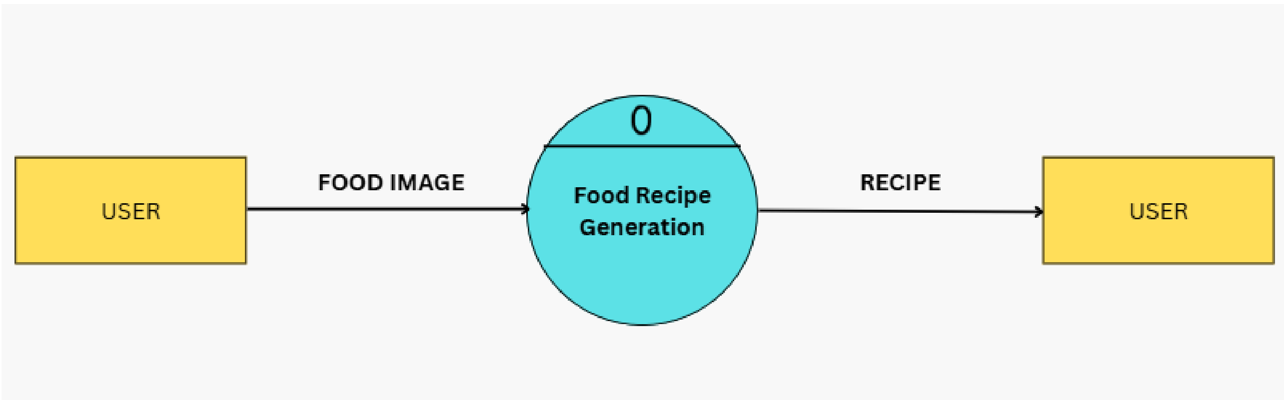

Fig 4.1 Data Flow Diagram level 0

At the highest level, the system consists of a User and the Recipe Prediction System

- User uploads a food image.
- The system processes the image, extracts features, and predicts the recipe.
- The system retrieves relevant recipes and generates the final output.
- User receives the predicted recipe.

Level 1

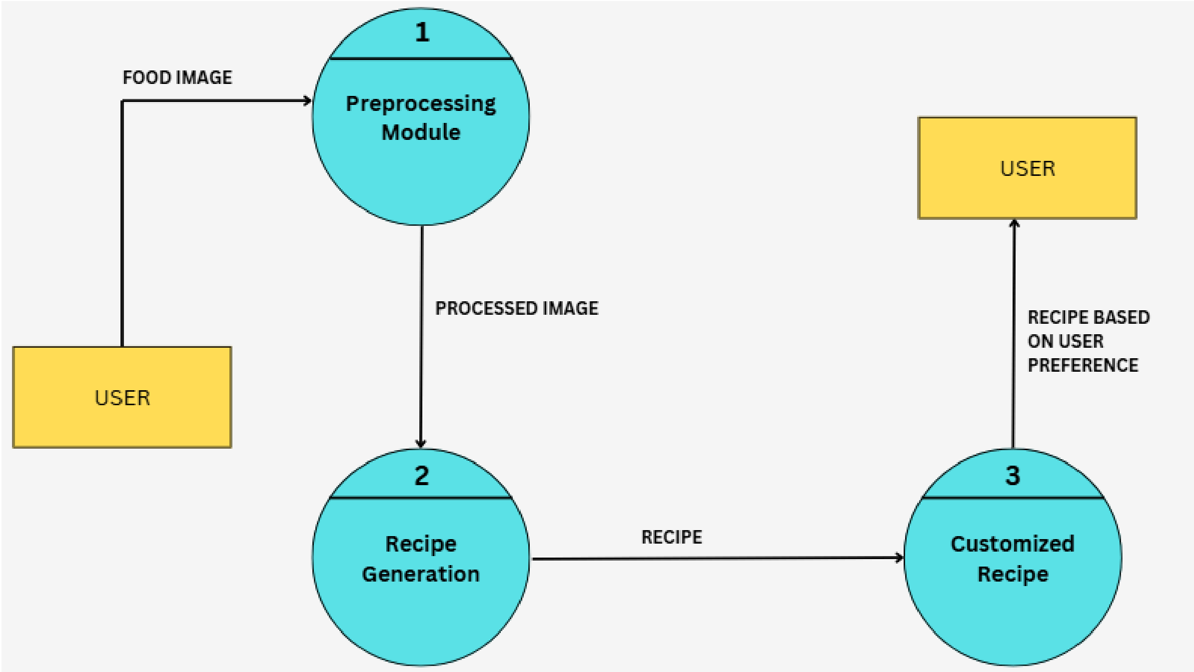

Fig 4.2 Data Flow Diagram level 1

1. Input Module
   - The user uploads a food image.
   - The system verifies the file type and quality.
2. Preprocessing Module
   - Converts the image to grayscale.
   - Applies normalization and contrast enhancement.
   - Resizes the image to match the CNN model's input dimensions.
3. Feature Extraction (CNN Model: ResNet-101)
   - Passes the image through convolutional layers to extract features such as edges, textures, and patterns.
   - Uses deep layers to understand complex patterns in food images.
4. Recipe Prediction & Mapping
   - The extracted features are used to predict the ingredients.
   - The system searches for a matching recipe in the RecipeNLG dataset based on the features.
   - A trained mapping model generates ingredient lists, quantities, and cooking steps.
5. Output Module
   - The system displays the predicted recipe with instructions.
   - Additional options include alternative recipes or modifications based on user preference.

4.3 Description of CNN Architecture (ResNet-101)

ResNet-101 (Residual Network with 101 layers) is a deep convolutional neural network (CNN) that plays a crucial role in food image classification and feature extraction for the recipe generation system. It enables the model to learn deep hierarchical representations of food images, identifying dish types and mapping them to corresponding ingredients and recipes.

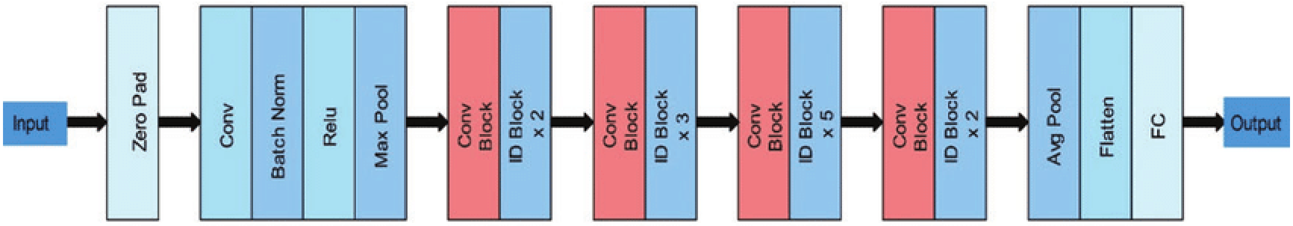

Fig 4.3 CNN Resnet Architecture

Key Components of ResNet-101 in Recipe Prediction

1. Convolutional Layers

The first layers of ResNet-101 consist of multiple convolutional layers that extract low-level visual features from the input food images.

- What it does: Detects basic patterns such as edges, textures, and color variations that help differentiate various food items.
- Importance in recipe generation: Helps in identifying key visual characteristics of dishes, such as whether a dish has pasta, bread, or leafy greens.

Example:
If the input image is a pizza, the convolutional layers detect circular edges, cheese texture, and tomato toppings, which later contribute to accurate classification.

2. Residual Blocks

One of the most powerful aspects of ResNet-101 is the use of residual blocks, which are designed to solve the vanishing gradient problem in deep networks.

- What it does: Allows information to flow directly across multiple layers via skip connections, making the training process more stable and efficient.
- Why it's important: Traditional deep CNNs suffer from gradient vanishing, where deeper layers struggle to learn. Residual connections help maintain meaningful gradients, ensuring better feature extraction for food images.

Example:
If two different images of a burger have different lighting conditions, residual blocks ensure that the model focuses on relevant features (such as the presence of a bun and patty) rather than getting confused by lighting differences.

3. Global Average Pooling (GAP) Layer

After extracting deep features from food images, ResNet-101 uses a Global Average Pooling (GAP) layer to convert feature maps into a compact representation.

- What it does: Instead of using fully connected layers, which have a large number of parameters, GAP reduces each feature map to a single value by taking the average of all activations.
- Why it's important: Reduces overfitting and helps in making the model more generalized across different food images.

Example:
For a spaghetti dish, GAP ensures that important high-level features (such as noodle patterns) are retained, rather than memorizing unnecessary details like the plate or background.

4. Fully Connected (FC) Layer

The extracted features from the convolutional layers and GAP layer are then passed through a Fully Connected (FC) layer.

- What it does: Converts extracted features into meaningful numerical representations that correspond to dish types or ingredients.
- Why it's important: The FC layer is responsible for mapping the visual representation of the dish to a text-based output, enabling the model to predict dish names and ingredients.

Example:
If an image of pasta is passed, the FC layer outputs a probability distribution that maps the features to categories like:

- Pasta - 95%
- Curry - 3%
- Burger - 2%

5. Softmax Layer (Final Classification)

The final layer of ResNet-101 is a softmax layer, which takes the outputs from the FC layer and converts them into a probability distribution over possible dish categories.

- What it does: Assigns probabilities to different dish types based on extracted features.
- Why it's important: Helps in identifying the most probable food category, which is then used to retrieve the corresponding ingredients and recipe steps from the RecipeNLG dataset.

Example:
For a fruit salad image, the softmax layer outputs:

- Fruit Salad – 97%
- Pasta – 2%
- Soup – 1%

This classified dish name is then used to fetch its ingredients and recipe steps.

Final Role of ResNet-101 in Recipe Prediction

1. Processes the food image to extract meaningful features.
2. Classifies the image into the most probable dish type.
3. Maps the dish type to a list of ingredients from the RecipeNLG dataset.
4. Generates a recipe based on the dish and ingredients.

Thus, ResNet-101 acts as the backbone of the entire image-to-recipe prediction system, ensuring that food images are correctly identified and linked to appropriate recipes.

4.4 Dataset Description

Food101 Dataset

The Food101 dataset is a widely used collection for food image classification tasks. It contains a total of 101,000 images, with 1,000 images per class, representing 101 different food categories. Each image in the dataset is labeled with a food class corresponding to one of the categories, such as pizza, sushi, salad, pasta, etc.

- Number of classes: 101
- Number of images: 101,000 (1,000 images per class)
- Image resolution: 512x512 pixels
- Format: JPG
- Classes include: pizza, sushi, burger, pasta, salad, sandwich, and other food items..

Recipe NLG Dataset

The Recipe NLG (Natural Language Generation) dataset is a collection designed for generating recipe descriptions and instructions. It contains a large number of recipe entries, each comprising multiple fields such as ingredients, cooking instructions, and recipe title. The dataset is used for the task of text generation where, given a set of ingredients, a model should generate a coherent recipe.

- Number of recipes: 2.3 Million originally
- Fields: title, ingredients, directions, NER(list of ingredients without servings), etc.
- Data format: Structured text (CSV)
- Recipe categories: The dataset includes various types of recipes such as desserts, main courses, beverages, and salads, among others, but only the ones related to the 101 food classes were taken in the end.

The Recipe NLG dataset is essential for mapping the food classification output from the Food101 dataset to actual recipe generation. Once an image is classified into one of the 101 food categories, the model uses the Recipe NLG dataset to generate a corresponding recipe based on the food class. This combination of food classification and recipe generation enables the model to provide both the food type and a detailed recipe for preparation.

Chapter 5 : Implementation

5.1 Code Snippets

5.1.1 Image Processing and Feature Extraction (encoder.py)

The core of the project starts with the CNN encoder that processes food images and extracts relevant features.

```python
class EncoderCNN(nn.Module):
    def __init__(self, embed_size, dropout=0.5, image_model='resnet101', pretrained=True):
        super(EncoderCNN, self).__init__()
        resnet = globals()[image_model](pretrained=pretrained)
        modules = list(resnet.children())[:-2]  # Remove final FC layers
        self.resnet = nn.Sequential(*modules)

        self.linear = nn.Sequential(
            nn.Conv2d(resnet.fc.in_features, embed_size, kernel_size=1, padding=0),
            nn.Dropout2d(dropout)
        )
```

Fig 5.1 Encoder Module

This code shows the CNN encoder implementation using ResNet101, which processes the input food images and extracts rich visual features.

5.1.2 Transformer Decoder (transformer_decoder.py)

The transformer decoder is responsible for generating recipes from the extracted features.

```python
class TransformerDecoderLayer(nn.Module):
    def __init__(self, embed_dim, n_att, dropout=0.5, normalize_before=True):
        self.self_attn = MultiheadAttention(
            self.embed_dim, n_att,
            dropout=dropout,
        )
        self.cond_att = MultiheadAttention(
            self.embed_dim, n_att,
            dropout=dropout,
        )
```

Fig 5.2 Transformer decoder module

This code implements the transformer decoder that uses multi-head attention mechanisms to generate coherent recipes.

5.1.3 Recipe Generation (output.py)

The recipe generation process combines the encoded image features with the transformer decoder.

```
def output(uploadedfile):
    # Load pre-trained models
    model = get_model(args, ingr_vocab_size, instrs_vocab_size)
    model.load_state_dict(torch.load(model_path))

    # Process image
    image_tensor = process_image(uploadedfile)

    # Generate recipes
    outputs = model.sample(image_tensor, greedy=greedy[i],
                          temperature=temperature, beam=beam[i])
```

Fig 5.3 Output module

This code shows how the system processes uploaded images and generates recipes using the trained models.

5.1.4 Web Interface (routes.py): The Flask web application handles user interactions and displays results.

```
@app.route("/predict", methods=["GET", "POST"])
def predict():
    if request.method == "POST":
        if "file" not in request.files:
            return redirect(request.url)

        file = request.files["file"]
        if file.filename == "":
            return redirect(request.url)

        if file:
            # Generate recipe
            title, ingredients, recipe = output(file)
            return render_template("predict.html",
                                   title=title,
                                   ingredients=ingredients,
                                   recipe=recipe)
```

Fig 5.4 Web interface module

This code demonstrates how the web application handles image uploads and displays generated recipes.

5.1.5 Model Architecture (model.py):

The main model architecture combines all components.

```python
def get_model(args, ingr_vocab_size, instrs_vocab_size):
    # Initialize encoders
    image_encoder = EncoderCNN(args.embed_size)

    # Initialize decoder
    decoder = TransformerDecoder(args.embed_size,
                                 args.n_att,
                                 args.transf_layers)

    # Combine into full model
    model = JointModel(image_encoder, decoder)
    return model
```

Fig 5.5 Main model module

5.1.6 Multi-head Attention Implementation (multihead_attention.py)

```python
class MultiheadAttention(nn.Module):
    def __init__(self, embed_dim, num_heads, dropout=0., bias=True):
        super().__init__()
        self.embed_dim = embed_dim
        self.num_heads = num_heads
        self.dropout = dropout
        self.head_dim = embed_dim // num_heads
        self.scaling = self.head_dim**-0.5

        self.in_proj_weight = Parameter(torch.Tensor(3*embed_dim, embed_dim))
        self.out_proj = nn.Linear(embed_dim, embed_dim, bias=bias)

    def forward(self, query, key, value, mask_future_timesteps=False,
                key_padding_mask=None, incremental_state=None):
        # Attention mechanism implementation
        attn_weights = torch.bmm(q, k.transpose(1, 2))
        attn_weights = F.softmax(attn_weights.float(), dim=-1)
        attn = torch.bmm(attn_weights, v)
        return attn, attn_weights
```

Fig 5.6 Multihead Attention module

This code shows the implementation of multi-head attention, crucial for the transformer architecture.

5.1.7 Recipe Generation Pipeline:

```python
def prepare_output(recipe_ids, ingr_ids, ingrs_vocab, vocab):
    outs = {}
    outs['title'] = generate_title(recipe_ids[0])
    outs['ingrs'] = [ingrs_vocab.idx2word[i] for i in ingr_ids
                     if i != 0 and i != ingrs_vocab.vocab_size-1]
    outs['recipe'] = generate_recipe_steps(recipe_ids, vocab)

    valid = {'is_valid': True, 'reason': ''}
    return outs, valid
```

Fig 5.7 Recipe generation pipeline

This code processes model outputs into human-readable recipe format.

5.2 Results & Discussions

5.2.1 Model Accuracy

```
Epoch 6/10
Epoch 6/10: 100%|██████████| 25/25 [06:10<00:00,14.8s/it]

Validating...
Train Loss: 0.5873 | Val Loss: 1.0325 | Val Acc: 0.6747
New best model saved with val acc: 0.6747

Epoch 7/10
Epoch 7/10: 100%|██████████| 25/25 [06:23<00:00,15.3s/it]

Validating...
Train Loss: 0.4008 | Val Loss: 0.8317 | Val Acc: 0.7184
New best model saved with val acc: 0.7184

Epoch 8/10
Epoch 8/10: 100%|██████████| 25/25 [06:18<00:00,15.1s/it]

Validating...
Train Loss: 0.2864 | Val Loss: 0.5875 | Val Acc: 0.7617
New best model saved with val acc: 0.7617

Epoch 9/10
Epoch 9/10: 100%|██████████| 25/25 [06:14<00:00,15.0s/it]

Validating...
Train Loss: 0.1550 | Val Loss: 0.4259 | Val Acc: 0.8100
New best model saved with val acc: 0.8100

Epoch 10/10
Epoch 10/10: 100%|██████████| 25/25 [06:24<00:00,15.4s/it]

Validating...
Train Loss: 0.1173 | Val Loss: 0.2124 | Val Acc: 0.8424
New best model saved with val acc: 0.8424
```

Fig 5.8 Epochs

The food classification model was trained using a dataset containing various food categories, with a structured approach involving multiple epochs to optimize learning. The food classification model was trained over multiple epochs to optimize its learning capabilities. During the training phase, the model exhibited a steady decrease in loss and an increase in accuracy, demonstrating effective learning of food categories. The validation accuracy remained stable, indicating that the model generalized well to unseen data suggesting minimal overfitting. The Adam optimizer played a crucial role in fine-tuning the model, ensuring robustness and efficiency in classifying food items based on images.

5.2.2 Web Interface Implementation

A front-end web interface was developed and hosted on a local server to facilitate user interaction with the food classification and recipe generation system. The interface allows users to upload an image of a food item, which is processed by the classification model to determine the most likely category. The design focuses on usability and efficiency, providing a clean and interactive experience. Backend processing ensures smooth communication between the classification model and recipe retrieval system, while features such as real-time image preview and error handling enhance the overall user experience.

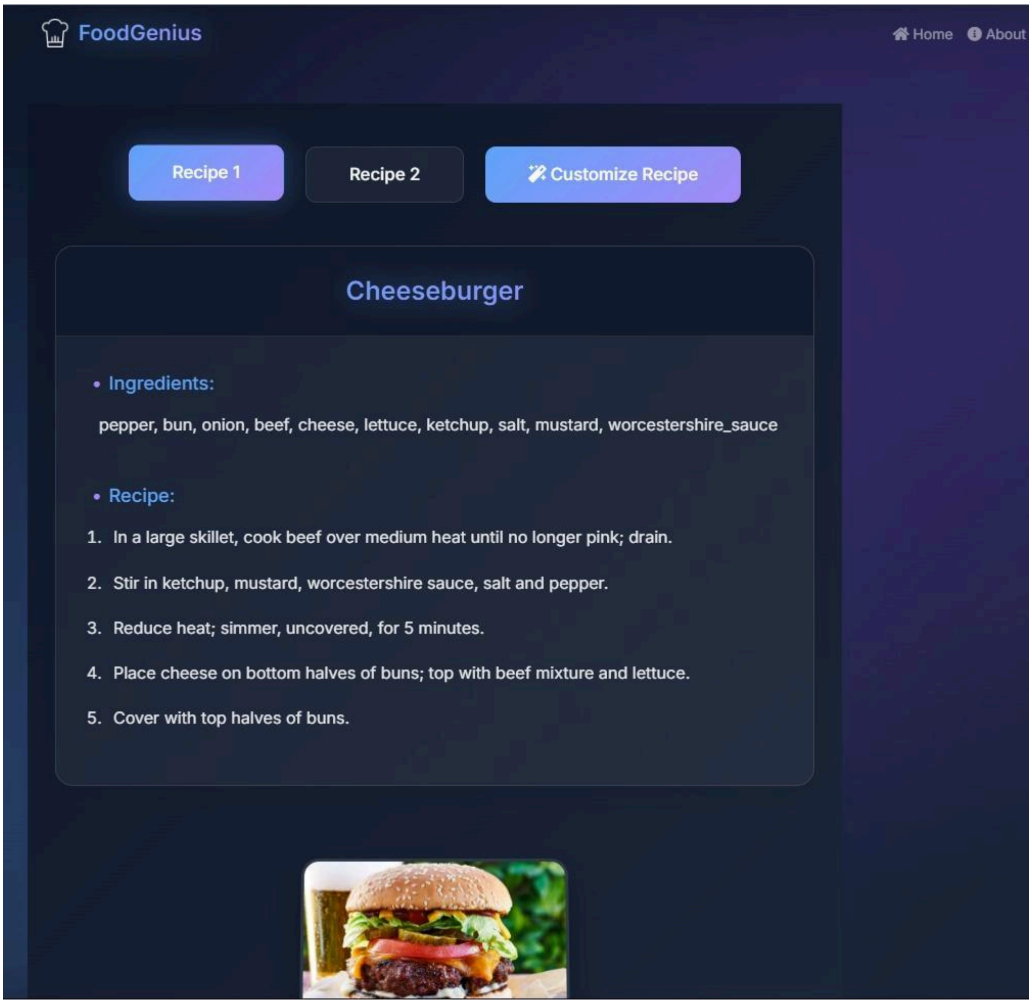

Fig 5.9 UI Implementation

5.2.3 Final Output and Recipe Generation

Following successful classification, the system retrieves and displays a relevant recipe, including a structured list of ingredients and cooking instructions. Additionally, the system allows for customized recipe generation based on user-defined parameters, allowing them to add/remove ingredients of their choice. This dynamic approach enhances the personalization of recipe suggestions, ensuring they align with individual user needs. The output is efficiently retrieved from pre-processed pickle files, making the system responsive and capable of delivering relevant recipes quickly. Future enhancements may focus on expanding the recipe database and integrating additional user preferences for a more tailored experience.

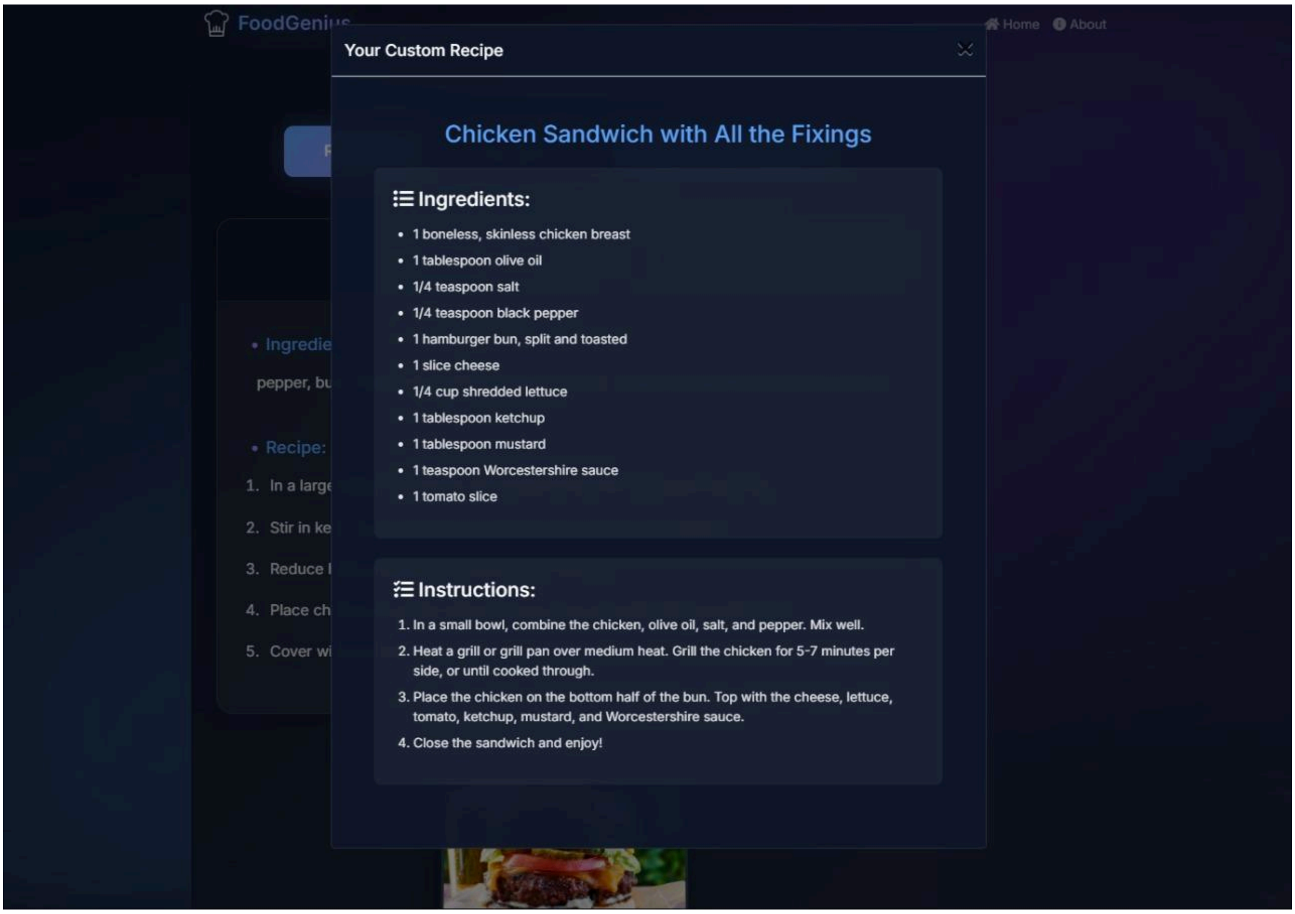

Fig 5.10 Customized recipe output

Chapter 6 : Conclusion

The FoodGenius project successfully demonstrates the powerful integration of Computer Vision and Natural Language Processing techniques to create an innovative recipe generation system. By leveraging a hybrid architecture that combines Convolutional Neural Networks (ResNet101) for image feature extraction and Transformer-based models for recipe generation, the project effectively bridges the gap between visual food recognition and detailed recipe creation.

The system's ability to accurately identify food items and generate coherent, detailed recipes highlights the robustness of the implemented deep learning approaches.This project underscores the effectiveness of multi-modal learning in developing AI-driven culinary solutions. The image processing pipeline, utilizing advanced CNN architectures, ensures robust feature extraction from food images. The Transformer decoder, with its multi-head attention mechanisms, enables the generation of contextually relevant and well-structured recipes.

The seamless integration of these components through a user-friendly web interface makes sophisticated AI technology accessible to everyday users, demonstrating the practical application of deep learning in enhancing culinary experiences.The modular architecture of FoodGenius allows for easy maintenance and future enhancements. The system's ability to generate multiple recipe variations for the same food image showcases its creative capabilities and flexibility. The implementation of attention mechanisms ensures that generated recipes maintain coherence between ingredients and instructions, while the web interface provides an intuitive platform for users to interact with the AI system.

In conclusion, the FoodGenius project exemplifies the potential of combining advanced deep learning architectures to solve real-world challenges in the culinary domain. The system is designed to be scalable and adaptable, with scope for enhancements such as support for diverse cuisines, dietary restrictions, and personalized recipe recommendations. This project not only demonstrates the technical feasibility of AI-driven recipe generation but also sets a strong foundation for future advancements in automated culinary assistance and food-related applications.

Chapter 7 : Future Enhancements

Cultural and Regional Variations:
Include alternate recipes that reflect regional or cultural preferences for the identified food item.

Meal Planning:
Extend the system to suggest weekly or daily meal plans based on user preferences, dietary goals, and available ingredients.

Multiple Language Support and Voice Detection:
Enhance user experience by adding voice-based inputs and outputs, making the system accessible to a broader audience, including non-English speakers.

Mobile Application Integration:
Develop a mobile application for edge devices to allow users to scan/capture food items and obtain results conveniently anywhere.

Interactive Recipe Steps:
Enhance the user experience with step-by-step cooking instructions, including video tutorials or AR-based interactive guides.

Dynamic Recipe Recommendations:
Use user feedback, likes, and cooking history to tailor personalized recipe suggestions.

Context-Aware Recommendations:
Suggest recipes based on contextual parameters such as time of day, season, or user mood.

Cooking Skill Adaptation:
Tailor recipes based on the user's skill level, offering simplified instructions for beginners and advanced techniques for experts.

Health Impact Analytics:
Offer long-term dietary impact reports based on the user's food logging history and recipe choices.

Bibliography

[1] B. Jamalpur, S. P. G, S. Venkat, A. R, S. Kavitha and G. C. Babu, "Optimizing Food Image Classification Using Black Widow Algorithm and Deep Learning Techniques," 2024 Second International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, Karnataka, India, 2024, pp. 1-6

[2] H. V. Hasti, S. A. Jatan, T. Nahar, V. Dubey and C. Parmar, "Image Processing and Machine Learning Methods for Assessing Food Quality," 2024 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2024, pp. 01-06,

[3] S. P. Singh, D. Siddharth, S. Prabakeran, R. Das and A. Balaji, "Food-Lens: Improving Culinary Experiences with AI-Driven Meal Analysis and Recipe Generation," 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2024, pp. 1385-1391

[4] R. Krutik, C. Thacker and R. Adhvaryu, "Advancements in Food Recognition: A Comprehensive Review of Deep Learning-Based Automated Food Item Identification," 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1-6

[5] P. Chhikara, D. Chaurasia, Y. Jiang, O. Masur and F. Ilievski, "FIRE: Food Image to REcipe generation," 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2024, pp. 8169-8179

[6] E. D. Cherpanath, P. R. Fathima Nasreen, K. Pradeep, M. Menon and V. S. Jayanthi, "Food Image Recognition and Calorie Prediction Using Faster R-CNN and Mask R-CNN," 2023 9th International Conference on Smart Computing and Communications (ICSCC), Kochi, Kerala, India, 2023, pp. 83-89

[7] F. S. Konstantakopoulos, E. I. Georga and D. I. Fotiadis, "A Review of Image-Based Food Recognition and Volume Estimation Artificial Intelligence Systems," in IEEE Reviews in Biomedical Engineering, vol. 17, pp. 136-152, 2024,

[8] T. M. L. Rosaline, C. A. Suciawan, W. Ng, S. Achmad and J. V. Moniaga, "AI-Powered Mobile Application for Image-Based Food Ingredient Detection and Recipe Generation," 2024 International Conference on Information Management and Technology (ICIMTech), Bali, Indonesia, 2024, pp. 123-128, doi: 10.1109/ICIMTech63123.2024.10780810.

[9] M. S. M. Rabby, M. Uddin, E. Islam, M. Khaliluzzaman, M. N. I. Shanto and A. Islam, "A Modified Transfer Learning-Based Framework for Efficient Food Image Classification," 2024 IEEE International Conference on Computing, Applications and Systems (COMPAS), Cox's Bazar, Bangladesh

[10] P. K. Singh and S. Susan, "Transfer Learning using Very Deep Pre-Trained Models for Food Image Classification," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India,

[11] S. S, P. Kokil, A. R and N. V. Sai Manoj, "Enhanced Food Classification System Using YOLO Models For Object Detection Algorithm," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6

[12] J. Sultana, B. M. Ahmed, M. M. Masud, A. K. O. Huq, M. E. Ali and M. Naznin, "A Study on Food Value Estimation From Images: Taxonomies, Datasets, and Techniques," in IEEE Access, vol. 11, pp. 45910-45935, 2023

[13] S. Chaudhary, B. Soni, A. Sindhavad, A. Mamaniya, A. Dalvi and I. Siddavatam, "ChefAI.IN: Generating Indian Recipes with AI Algorithm," 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT), Pune, India, 2022, pp. 1-6

[14] J. N. V. R. S. Kumar, C. Jyothsna, K. Srinivas, A. E. Sravanth, A. T. Kumar and D. Indira, "Self-Attention Architecture for Ingredients Generation from Food Images," 2022 Second International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, India, 2022, pp. 1-7

[15] D. P. Papadopoulos, E. Mora, N. Chepurko, K. W. Huang, F. Ofli and A. Torralba, "Learning Program Representations for Food Images and Cooking Recipes," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 16538-16548

[16] N. -I. Galanis and G. A. Papakostas, "An update on cooking recipe generation with Machine Learning and Natural Language Processing," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022, pp. 739-744

[17] H. Wang, G. Lin, S. C. H. Hoi and C. Miao, "Learning Structural Representations for Recipe Generation and Food Retrieval," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3363-3377 , 2022

[18] M. Goel et al., "Ratatouille: A tool for Novel Recipe Generation," 2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW), Kuala Lumpur, Malaysia, 2022, pp. 107-110

[19] F. P. W. Lo, Y. Sun, J. Qiu and B. Lo, "Image-Based Food Classification and Volume Estimation for Dietary Assessment: A Review," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 7, pp. 1926-1939 , 2020

[20] A. -S. Metwalli, W. Shen and C. Q. Wu, "Food Image Recognition Based on Densely Connected Convolutional Neural Networks," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan