March 2024

Automatic Image-Based Nutritional Calculator App

Kejvi Cupa
University of South Florida

Automatic Image-Based Nutritional Calculator App

by

Kejvi Cupa

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Yu Sun, Ph.D.
Hariom Yadav, Ph.D.
John Templeton, Ph.D.
Zhao Han, Ph.D.

Date of Approval:
March 13, 2024

Keywords: Computer Vision, Ingredient Recognition,
Portion Estimation, Nutrition, Transformers

Dedication

In dedication to my family and friends that have supported me in this journey. I want to express my special gratitude to Dr. Yu Sun and Dr. Hariom Yadav who have guided and supported me, providing me with plenty of opportunities to challenge and grow myself. Finally, I want to thank Dr. John Templeton and Dr. Zhao Han, who served on the committee.

Table of Contents

List of Tables

List of Figures

Abstract

Nutrition plays a pivotal role in shaping an individuals' health and quality of life, making the evaluation of dietary intake crucial for promoting healthier lifestyle choices. Various solutions, particularly mobile apps, have been developed to facilitate the process of dietary estimation. Accurate nutritional intake assessment relies on two key components: ingredient recognition and food portion estimation. For a mobile app to offer a comprehensive solution for automatic nutritional assessment, it must address both components.

In this work, we focus on a mobile app pipeline: the semi-automatic pipeline which focuses on automatic food ingredient recognition. This pipeline integrates state-of-the-art models for ingredient recognition. We demonstrate that models such as BLIP-2 and GPT-3.5, when combined, can deliver precise ingredient recognition and great generalization capabilities. A fine-tuned GPT-3.5 model calculates the nutritional value of meals based on the recognized ingredients and portion sizes provided by users manually. Since the implementation of this app focuses more on the automatic ingredient recognition the paper will primarily focus on that component. The performance and outcomes of the semi-automatic pipeline are benchmarked against existing mobile apps. Our findings reveal that the semi-automatic pipeline holds significant promise for generalization across different cuisines and enabling individuals to record their nutritional intake accurately and efficiently.

Chapter 1: Introduction and Background

In recent years, public awareness of nutrition has significantly increased due to growing concerns over diet-related health issues. This heightened interest has coincided with technological advancements aimed at simplifying the monitoring of dietary intake. Among these advancements, we leverage state-of-the-art models such as BLIP-2 and GPT-3.5 for ingredient recognition. These technologies contribute significantly to advancements in automated dietary assessment, aiming to improve accuracy and efficiency.

A key challenge in this area is the accurate tracking and analysis of what people eat. Manual food logging is time-consuming and can be difficult due to unfamiliarity with certain ingredients. As a result, various automated methods have been developed to help with the recognition of ingredients and their portions, with mobile applications standing out as an effective tool. Most of these solutions have focused on ingredient recognition, with fewer efforts made toward portion estimation, which often rely on reference sizes rather than actual volume estimation, especially in commercial mobile apps.

The variety of meals and their ingredient compositions present significant challenges in accurately identifying ingredients and estimating portion sizes. In this paper, we present a mobile app designed to primarily address ingredient recognition effectively. To support our work, we conducted an extensive literature review to collect insights from existing mobile apps, and different approaches in ingredient recognition and food portion estimation ranging from traditional methods to more novel ones. Most of these studies have focused primarily on ingredient recognition, with less emphasis on portion estimation. Our goal with this review is to identify cutting-edge approaches and evaluate their effectiveness in comparison to our implementation and discuss potential advantages and limitations. Additionally, we examine the technological foundations behind ingredient recognition, ranging from traditional Convolutional Neural Networks (CNNs) [1, 2] and Deep Convolutional Neural Networks (DCNNs) [3, 4, 5] to more recent innovations like Transformer Models [6, 7]. We categorize portion estimation techniques into single-image and multi-image/video methods, with various strategies discussed. It is difficult to understand the challenging tasks that are ingredient recognition and portion estimation, which is why novel approaches are being developed to address them. Chapter 2: Literature Review provides a more detailed analysis of each method, providing more insight into the advantages and limitations of each approach.

The remaining sections of the thesis are organized as follows: Chapter 3 elaborates on our semi-automatic pipeline for ingredient recognition. This chapter discusses the chosen technologies, including BLIP-2 and GPT-3.5, the rationale behind their selection, and a careful description of each component of the pipeline. Chapter 4 outlines the experimental framework of our mobile app, detailing the experiments carried out to refine the pipeline and comparing the performance of the semi-automatic pipeline against baseline models from other mobile apps, focusing on the importance of guidelines for evaluating models of different architectures on traditional datasets. Chapter 5 provides an analysis of the results, offering insights into the outcomes and proposing areas for improvement. Finally, Chapter 6 concludes the thesis, summarizing the findings and proposing directions for future research and development.

Chapter 2: Literature Review

2.1 Related Work

In this section, we examine mobile applications that automate the food recognition aspect of nutritional assessments. For a fair comparison, we omit proprietary applications such as MyFitnessPal, LoseIt, etc, due to the lack of access to their network architecture and technology. Our focus is on mobile applications documented in research papers, where the details of the network architecture, technology used, and datasets evaluated are openly available. Additionally, we provide a more comprehensive review of approaches taken on Ingredient Recognition and Portion Estimation, ranging from the classical approaches to more novel ones.

A notable paper from the mobile apps is "MyDietCam" by Tahir et al. [8], which specializes in automatic food recognition. It leverages state-of-the-art deep learning networks, with Inception-Resnet-V2 identified as the most effective for feature extraction. For classification, it uses the Adaptive Reduced Class Incremental Kernel Extreme Learning Machine (ARCIKELM). The application's performance was assessed using the Food101, UECFOOD100, and UECFOOD256 datasets, achieving accuracies of 87.27%, 88.74%, and 76.51%, respectively.

Another application, "FoodTracker" by Sun et al. [9], focuses on food detection. It employs a deep convolutional neural network (DCNN) based on Mobilenet and integrates the advanced one-stage detection framework, YOLOv2, for simultaneous multi-object recognition and localization. Its performance on the UECFOOD100 and UECFOOD256 datasets resulted in mean average precision scores of 76.35% and 75.05%, respectively.

Additionally, "DietLens" by Ming et al. [10] targets food recognition using deep-based image recognition algorithms, notably ResNet-50. For portion estimation, it offers predetermined sizes, hence not estimating portions directly from images. It was evaluated on a diverse dataset comprising Chinese, Western, Indian, Malay, and other cuisines, sourced from Google. The application demonstrated a higher top-1 accuracy for Western foods at 78.6% compared to 64.2% for Chinese dishes. The average recognition accuracy for popular foods reached 75.2% for top-1 accuracy. Furthermore, when tested on a dataset of 4,697 real user meal photos, the model achieved a top-1 accuracy of approximately 47.6%.

2.2 Ingredient Recognition

In the development of an automatic image-based nutrition calculator app, understanding the advancements in ingredient recognition is crucial. Here we review different approaches that focus on ingredient recognition, categorizing them into Convolutional Neural Networks (CNNs), Deep Convolutional Neural Networks (DCNNs), Transformer Models and Advanced Architectures, and Innovative Techniques.

A widely adopted method for recognizing ingredients is through multi-label classification. This technique involves predicting several output labels or categories, each representing a different ingredient [11].

Focusing on Convolutional Neural Networks (CNN), Bolano [1] introduces a model adapted as a multi-label predictor, specifically designed for recognizing multiple food ingredients from single images. This approach enables the model to accurately identify a comprehensive list of ingredients in dishes it has never encountered before, leveraging two newly developed datasets for enhanced training and generalization. Similarly, Konstantakopoulos [2] applies a pre-trained

CNN, EfficientNetB2, for food image classification on the MedGRFood dataset, highlighting the adaptability of CNNs in processing complex food images.

Transitioning to Deep Convolutional Neural Networks (DCNN), the work by Chen and Ngo [3] proposes a novel deep learning framework for cooking recipe retrieval and ingredient recognition in food images. This approach adapts architectures from VGG to learn food categorization and ingredient recognition simultaneously, showing improved performance in handling the large visual variations of dishes. Chen et al. [4] introduce a multi-relational graph convolutional network (mRGCN) for zero-shot ingredient recognition. This novel approach integrates multiple relational graphs with a DCNN to predict classifiers for unseen ingredients, showing great performance on Chinese and Japanese food datasets such as UECFOOD-100 and VireoFood-172. J. Chen [5] further explores food ingredient recognition through multi-task and region-wise deep learning, using DCNNs to address the complexities of ingredient recognition due to varied appearances and demonstrating the benefits of leveraging food categories for enhanced accuracy.

In exploring Transformer Models and Advanced Architectures, the paper by Amaia Salvador [6] introduces an inverse cooking system that generates recipes from food images. This approach uses a novel architecture for predicting ingredients without a fixed order and crafting instructions by considering both the image and ingredients simultaneously. It employs ResNet-50 for image embeddings and transformers for decoding, evaluated on the Recipe1M dataset, demonstrating improved ingredient prediction and the ability to produce high-quality recipes. This showcases the integration of visual and textual analysis in recipe generation.

Further advancements in food image classification are achieved with the Vision Transformer model, AlsmViT, as presented by Gao et al [7]. This model is enhanced with data augmentation and feature enhancement techniques, including Augmentplus, LayerScale, and multi-layer perception mechanisms. AlsmViT achieves notable accuracies of 95.17% on the Food-101 dataset and 94.29% on the Vireo Food-172 dataset, surpassing other self-supervised methods in food image classification.

The FL-Tran model, introduced by Zhou. W [12], marks an improvement in multi-label image classification by effectively handling small-scale objects and discovering hidden features. This model integrates a Multi-Scale Fusion Module (MSFM), a Spatial Attention Module (SAM), and a Feature Enhancement and Suppression Module (FESM). FL-Tran addresses the challenge of recognizing small objects and uncovers useful features that are usually overshadowed by more prominent ones, outperforming existing methods on key datasets such as MS-COCO 2014, PASCAL VOC 2007, and NUS-WIDE.

Yunan Wang [13] employs a multi-label learning approach for mixed dish recognition, emphasizing region-level recognition and the effectiveness of Negative Sampling and targeted pre-trained models. This approach, demonstrated through experiments on two specially collected datasets, significantly reduces the need for manual labeling and improves performance over traditional multi-label classification methods.

Building on innovative techniques, Zhang [14] incorporates cooking logic into ingredient recognition, using a sequential learning approach. This method, which enhances feature extraction through a double flow feature fusion module and applies reinforcement learning with a hybrid loss function, shows promise in improving food cognition and offers potential for diet recommendations and cooking assistance.

Liu and Yang [15] introduce a method that merges an Attention Fusion Network with a Food-Ingredient Joint Learning module, targeting fine-grained recognition of food and ingredients.

This approach is especially effective for complex Chinese dishes, achieving high accuracy on the VIREO Food-172 dataset by focusing on discriminative regions and addressing ingredient imbalance, laying the foundation for broader applications in dietary management and automated food systems.

Expanding the scope of food recognition, Min and Liu [16] present the Ingredient-Guided Cascaded Multi-Attention Network (IG-CMAN). Leveraging multi-scale, sequential localization of informative regions from category to ingredient level, this method shows impressive performance on popular food datasets and the newly created ISIA Food-200. By focusing on discriminative, ingredient-related regions, IG-CMAN enhances food image analysis, promising advancements in multimodal food logging and personalized healthcare.

Pellegrini [17] discusses Food2Vec and FoodBERT, models designed for ingredient embeddings, which are extended to multimodal versions to assist in identifying ingredient substitutes. FoodBERT proves most effective in substitute recommendations, suggesting its embeddings capture food knowledge more accurately than traditional models, paving the way for personalized dietary recommendations and improved understanding of food items through targeted image training.

In an integration of deep learning and natural language processing, Mezgec [18] outlines an automated dietary assessment methodology that combines fake food image recognition with food item standardization. Achieving classification accuracies of 92.18% and 93%, this approach marks a significant leap in dietary assessment automation, reducing manual data handling and enabling real-time analysis via smartphone applications. This research highlights the potential for extending the methodology to include automatic measurements of food weight or volume, aiming for a fully automated dietary assessment process.

Chhikara [19] introduces FIRE, a novel multimodal methodology for generating recipes from food images. Utilizing the BLIP model for title generation, a Vision Transformer for ingredient extraction, and the T5 model for cooking instruction generation, FIRE shows substantial improvements over existing methods. This approach enables recipe customization and automatic cooking processes, setting a direction for future research in recipe correctness metrics, recipe diversity through knowledge graphs, and addressing the challenge of hallucination in recipe generation.

Further advancing ingredient recognition techniques, Zhu et al [20] propose a novel CNN-based framework for segmenting ingredients in food images. This approach, which does not require pixel-level annotations, uses a standardized biological-based hierarchical ingredient structure to develop a Single-Ingredient Classification Model, significantly enhancing ingredient identification within food images as demonstrated on the FoodSeg103 dataset.

In the context of recipe generation and food recognition, Wang et al [21] introduce a Structure-aware Generation Network (SGN) for generating cooking instructions from food images and ingredients. This method highlights the utility of combining CNN and RNNs for feature extraction and classification, particularly in the absence of structural annotations in long paragraphs of recipes.

Addressing food recognition and calorie calculation, Minija. J [22] presents a method that combines image segmentation with a feed-forward neural network classifier to enhance classification accuracy. Employing techniques like salient region detection and multi-scale segmentation, this approach not only identifies food items in images but also calculates calorie values based on food volume and nutrition content, showcasing improved classification performance.

To provide a better consolidation of these different approaches for Ingredient Recognition a summarization of these approaches has been provided in Tables 2.1, 2.2, 2.3, 2.4, and 2.5.

2.3 Portion Estimation

Automatically calculating food volume presents multiple challenges, such as the diversity of food compositions, an expanding variety of ingredients, and different preparation methods. The quality of photos used for estimating food volume is crucial, with clear and well-lit images generally yielding more accurate estimates than those that are blurry or poorly lit. To date, several methods have been developed to estimate food volume accurately, ranging from simple pixel counting to advanced 3D image reconstruction techniques. These methods fall into two categories: single image view and multi-image/video view. Single-image-view methods estimate food volume using just one photo. These techniques are more user-friendly than multi-image view methods, as they don't require taking multiple photos from different angles. However, this simplicity can lead to less precision in volume estimation compared to methods that use multiple views.

2.3.1 Single-Image-View Methods

Single-image-view methods for estimating food volume utilize just one photo, simplifying the process by eliminating the need for multiple shots from different angles. This convenience, however, often comes at the cost of accuracy compared to multi-image or video-based approaches.

Among the simplest techniques for portion estimation are traditional manual methods, which involve the use of rulers and adjustable wedges [23]. These approaches, while straightforward, require direct interaction with the meal, potentially limiting their applicability in non-intrusive settings.

Advancements in technology have introduced more sophisticated methods for single-image food volume estimation. Notably, mobile augmented reality and virtual reality [24], as highlighted in a comprehensive review [25], offer dynamic and interactive means for assessing portion sizes. These methods are complemented by visual assessment techniques [25], which, despite their susceptibility to human error, provide valuable insights into portion estimation through subjective evaluation. Automatic food volume estimation methods offer significant advantages for individuals managing chronic diseases by facilitating dietary monitoring without the need for direct expert intervention, providing faster results than traditional approaches that typically require sending food images to a dietitian. The conventional process demands ongoing dietitian involvement, limiting their capacity to promptly address the needs of a large patient base. However, a notable drawback of automated systems is the lack of standardization, with no clear expert guidelines addressing acceptable error rates. Accuracy and usability vary across different estimation techniques, broadly categorized into single-image-view and multi-image-view methods. While single-image methods are more user-friendly, they often sacrifice accuracy, unlike multi-image methods that achieve higher precision using images from various angles.

Generative Adversarial Networks (GANs) have been explored by Fang et al. [26] for estimating the energy content of food portions from single-view images. They introduced the concept of "energy distribution" within each image, using a dataset with detailed annotations including segmentation masks and energy values. Their GAN-based approach shows promise with an average energy estimation error rate of 10.89%, highlighting GANs' potential in dietary assessment.

Miyazaki et al. [27] propose a method that leverages an image-analysis technique for calorie estimation from food images. This method is distinct in that it does not rely on identifying specific food items. Instead, it utilizes visual similarities across various image features, such as color histograms, color correlograms, and SURF features, to rank images by similarity. Calorie content is then estimated through linear estimation based on these rankings, utilizing a dataset of 6,512 food images with expert-estimated calories from the FoodLog web service.

The work by Lan et al [28] introduces FoodSAM, a development of the Segment Anything Model (SAM) for food image segmentation. FoodSAM addresses SAM's limitations by combining coarse semantic masks with SAM-generated masks to enhance segmentation quality. It applies instance segmentation to recognize ingredients as separate entities and includes panoptic segmentation with an object detector to identify non-food objects. Furthermore, FoodSAM incorporates promptable segmentation for food images, supporting different prompt types. As the first framework to achieve instance, panoptic, and promptable segmentation for food imagery, FoodSAM shows strong performance in extensive experiments, underscoring SAM's potential in the field of food image analysis.

Beijbom et al. [29] introduce the Menu-Match, a system designed to identify and estimate the calorie content of restaurant meals from images. By creating a database of restaurant-specific menu items, Menu-Match employs advanced computer vision techniques, comparing input images to known items using features like color histograms, HOG, SIFT, LBP, and MR8 filter responses. This novel approach simplifies the calorie estimation process by transforming it into an identification challenge, facilitating easier food logging.

Ma et al [30] introduce an improved encoder-decoder framework for estimating food energy from single monocular images, addressing the challenge of extracting limited energy information. This method transforms images into a format where food energy information is more accessible, facilitated by a high-quality dataset verified by dietitians, including images, segmentation masks, and calorie values. Demonstrating a significant advancement over prior methods, their approach reduces the Mean Absolute Percentage Error (MAPE) and Mean Absolute

Error (MAE) by over 10% and 30 kcal, respectively, showcasing its potential in automatic image-based dietary assessment.

Another critical aspect of portion estimation is the precise extraction of the food region from an image. Kirii. D [31] employs TransalNet and KernelCut to refine the border between the food item and the background in meal images. Applied to a set of 1,027 images from the UNIMIB2016 dataset, this method combines state-of-the-art DNN-based saliency detection with graph theory to enhance the accuracy of food region extraction.

2.3.2 Multi-Image and Video-Based Methods

Multi-Image and video-based approaches significantly enhance the accuracy of food portion estimation by leveraging images from various angles. The use of 3D geometric models and shape templates is a common strategy for determining portion sizes [32,33,34,23,35]. These methods depend on the accurate classification and segmentation of food items, which allows for the effective application of geometric models to ascertain precise portion sizes.

Xu et al. [36] introduce a method that utilizes 3D modeling and pose estimation to estimate food volume from images. This technique identifies and quantifies food items within single images, improving the assessment of nutrient content through accurate volume estimations.

Augmented Reality (AR) systems, such as Eat AR [37] and Serv AR [38], apply fiducial markers or standard-sized objects within the image frame. These systems compute food volume by overlaying 3D shapes on the captured scene, hence enabling precise volume measurements. Dense 3D model reconstruction methods developed by Joachim Dehais [39] and Wen Wu [40] construct detailed 3D models from multiple viewpoints. This approach allows for a highly accurate estimation of food volumes.

Konstantakopoulos [2] explores Stereo vision techniques for volume estimation, requiring at least two images taken from smartphones. With a reference card placed next to the dish for scale estimation, this method facilitates accurate 3D reconstruction and subsequent volume estimation.

In a different approach, Xu et al [41] developed a novel LiDAR-based machine vision system, incorporating a custom-designed roller conveyor to facilitate non-destructive, online volume measurement of sweet potatoes. Utilizing advanced analysis techniques, including multiple linear regression and neural networks, the system achieved an impressive 97.9% accuracy in volume estimation, showcasing its potential for high-throughput agricultural applications. While this system has not been tested on other foods yet, it nonetheless shows great promise.

These multi-image and video-based methods, along with the latest in segmentation and energy estimation technologies, contribute to the evolving field of dietary assessment. They offer more accurate and user-friendly options for food portion estimation, aligning with the needs of individuals managing their dietary intake. By utilizing cutting-edge machine learning and computer vision, these approaches significantly improve the accuracy of identifying and measuring food portions. This advancement is particularly beneficial as it minimizes errors that are common with older, more manual techniques. Moreover, the integration of these sophisticated technologies into dietary assessment tools has the potential to make tracking food intake simpler and more efficient for users. This can in turn encourage individuals to stay consistent with their dietary tracking, fostering better nutritional habits and informed eating decisions.

To provide a better consolidation of these different approaches for Portion Estimation a summarization of these approaches has been provided in Tables 2.6, 2.7, 2.8, 2.9, 2.10, and 2.11.

Table 2.1: Comparative summary on ingredient recognition architecture/approaches

Paper	**Architecture/Approach**
Bolano [1]	CNN adapted for multi-label prediction
Konstantakopoulos [2]	Pre-trained CNN (EfficientNetB2)
Chen and Ngo [3]	DCNN with architectures from VGG
Chen et al. [4]	Multi-relational graph convolutional network (mRGCN)
J. Chen [5]	Multi-task and region-wise deep learning with DCNNs
Amaia Salvador [6]	Inverse cooking system with ResNet-50 and transformers
Gao et al [7]	Vision Transformer model (AlsmViT)
Zhou. W [12]	FL-Tran model with MSFM, SAM, FESM
Yunan Wang [13]	Multi-label learning with Negative Sampling
Zhang [14]	Sequential learning with double-flow feature fusion
Liu and Yang [15]	Attention Fusion Network with Food-Ingredient Joint
Min and Liu [16]	IG-CMAN for food recognition
Pellegrini [17]	Food2Vec and FoodBERT for ingredient embeddings
Mezgec [18]	Deep learning and NLP for dietary assessment
Chhikara [19]	FIRE for recipe generation from images
Zhu and Dai [20]	CNN framework for ingredient segmentation
Wang, Lin, Hoi, and Miao [21]	Structure-aware Generation Network (SGN)
Minija. J [22]	Image segmentation with neural network classifier

Table 2.2: Simplified overview on ingredient recognition architecture/approaches

Paper	Architecture
[1], [2], [20]	CNN
[3], [5], [22]	DCNN
[6], [7]	Transformer
[4]	GCN
[13], [5], [15], [16]	Multi-Task/Region-wise
[14], [12]	Sequence/Fusion
[17], [18], [19]	Embedding/NLP

Table 2.3: Metric-based comparison of our pipeline with related work

Paper	Generalization	Precision (>75%)	Complex Images	Fine-Grained Recognition	Multi Label	Not trained Fine-tuned
Bolano [1]	✓	X	✓	✓	✓	X
Chen and Ngo [3]	X	X	✓	X	✓	X
Gao et al [7]	---	✓	✓	X	X	X
Chen et al. [4]	✓	X	✓	X	✓	X
Wang [13]	X	✓	✓	X	✓	X
Zhang [14]	X	X	✓	✓	✓	X
Mezgec [18]	---	✓	X	X	✓	X
[8]	X	✓	✓	X	X	X

Table 2.3 (Continued)

Paper	Generalization	Precision (>75%)	Complex Images	Fine-Grained Recognition	Multi Label	Not trained Fine-tuned
[9]	X	✓	✓	X	✓	X
Ours	✓	✓	✓	✓	✓	✓

Table 2.4: Comparative summary on ingredient recognition approaches limitations

Paper	Limitations
Bolano [1]	1. Limited by transfer learning for unique ingredient features 2. Ingredient list simplification may reduce complex dish accuracy
Konstantakopoulos [2]	Limited to estimating nutrition in shallow dishes
Chen and Ngo [3]	1. Ignores cooking methods in architecture 2. Struggles with invisible ingredients (e.g., oil)
Chen et al. [4]	BERT's averaging reduces performance compared to Word2Vec
J. Chen [5]	1. Multi-scale processing doesn't aid ingredient recognition 2. Conflicting objectives in image-level categorization and ingredient recognition 3. Vireo Food-251 dataset issues with balance and ingredient co-occurrence
Amaia Salvador [6]	Transformer model penalizes ingredient order prediction
Gao, Xinle, and Xiao [7]	Model not tested on general image datasets

Table 2.4: (Continued)

Paper	**Limitations**
Zhou. W [12]	1. Less flexible dropout mechanism 2. Ineffective in long-tailed class distribution datasets
Yunan Wang [13]	Difficulty with very small objects
Zhang [14]	1. Poor performance on non-cuisine matched images 2. Suggests neural graph networks and external knowledge for improvement
Liu and Yang [15]	Needs expansion to more multi-attribute and multi-modal tasks
Min and Liu [16]	1. Fails to localize semantic regions in complex ingredient cases 2. Limited to five predefined regions
Pellegrini [17]	1. Biased dataset creation 2. Arbitrary ingredient-substitute sampling
Mezgec [18]	Lower detail in validation and test image predictions
Chhikara [19]	1. Lacks a reliable grounding mechanism 2. Recipe diversity affected by location 3. Challenges in recipe hallucination
Zhu and Dai [20]	1. Ingredient-background segmentation issues 2. Calls for more segmentation model results
Wang, Lin, Hoi, and Miao [21]	1. SGN models may produce redundant sentences 2. Difficulty aligning generated recipe nodes with ground truth
Minija. J [22]	1. Inaccuracy in non-food images 2. Low accuracy in cutlery-rich images

Table 2.5: Comparative summary on ingredient recognition approaches advantages

Paper	**Advantages**
Bolano [1]	1. High generalization capability 2. Effective multi-label recognition
Konstantakopoulos [2]	Effective in complex image processing
Chen and Ngo [3]	Simultaneous categorization and recognition
Chen et al. [4]	Zero-shot ingredient recognition
J. Chen [5]	Addresses varied ingredient appearances
Amaia Salvador [6]	Integrates visual and textual analysis
Gao, Xinle, and Xiao [7]	Surpasses self-supervised methods
Zhou. W [12]	Effective in multi-label classification
Yunan Wang [13]	Efficient in mixed dish recognition
Zhang [14]	Incorporates cooking logic
Liu and Yang [15]	Fine-grained recognition
Min and Liu [16]	Focuses on discriminative regions
Pellegrini [17]	Captures food knowledge accurately
Mezgec [18]	Automates dietary assessment
Chhikara [19]	Enables recipe customization
Zhu and Dai [20]	Pixel-level ingredient recognition
Wang, Lin, Hoi, and Miao [21]	Combines CNN and RNNs for recipe generation
Minija. J [22]	Enhanced classification accuracy

Table 2.6: Comparative summary on portion estimation architecture/approaches

Paper	Architecture/Approach
Traditional manual methods [23]	Rulers and adjustable wedges
AR and VR [24]	Mobile augmented reality and virtual reality
Visual assessment [25]	Visual assessment techniques
Miyazaki et al. [27]	Image-analysis for calorie estimation
Beijbom et al. [29]	Menu-Match system for calorie estimation
Fang et al [26]	GANs for energy content estimation
Kirii. D [31]	TransalNet and KernelCut for food region extraction
3D geometric models [32,33,34,23,35]	3D geometric models and shape templates
Xu et al. [36]	3D modeling and pose estimation
Eat AR and Serv AR [37, 38]	Augmented Reality systems for volume computation
Joachim Dehais [39] and Wen Wu [40]	Dense 3D model reconstruction
Konstantakopoulos [2]	Stereo vision techniques for volume estimation
FoodSAM [28]	Food image segmentation with FoodSAM
Ma et al [30]	Improved encoder-decoder framework for food energy estimation
Xu et al [41]	LiDAR-based machine vision with roller conveyor, MLR, SNN, DNN analysis

Table 2.7: Simplified overview on portion estimation architecture/approaches

Paper	**Approach**
[23]	Traditional methods
[24]	AR and VR technologies
[25]	Visual assessment techniques
[27,29]	Image-analysis
[26,35]	GAN
[32, 36, 33, 34, 23, 35, 39, 40, 2, 41]	3D modeling and reconstruction
[31, 37, 38, 28, 30]	Food region extraction

Table 2.8: Comparative summary on portion estimation approaches limitations

Paper	**Limitations**
Traditional manual methods [23]	Requires direct interaction
AR and VR [24]	1. Sensitivity to camera rotation angle impacts volume estimation accuracy for food 2. Limited accuracy for irregularly shaped foods due to reliance on virtual objects of regular shapes
Visual assessment [25]	Limited to hospital meals with standardized portions
Miyazaki et al. [27]	Ineffective for images with multiple dishes

Table 2.8: (Continued)

Paper	**Limitations**
Beijbom et al. [29]	Only effective for uniform portion sizes in restaurant dishes
Fang et al [26]	Cannot capture occluded food items
Kirii. D [31]	Unsuitable for images with multiple food items
3D geometric models [32,33,34,23,35]	Requires camera calibration and multiple images
Xu et al. [36]	Requires camera calibration and multiple images
Eat AR and Serv AR [37, 38]	Focused solely on rice and similar shaped foods, lacks generalization
Joachim Dehais [39] and Wen Wu [40]	Need for camera calibration and multiple images
Konstantakopoulos [2]	Need for camera calibration and multiple images
FoodSAM [28]	Fails to capture occluded food items
Ma et al [30]	Lack generalization to unseen images
Xu et al [41]	Limited to sweet potatoes, conveyor speed not optimized

Table 2.9: Simplified overview on portion estimation limitations

Paper	Limitations
[23]	Requires direct interaction
[24]	Sensitivity to camera orientation
[25,37,38, 41]	Limited application scope
[27,29,31,28]	Ineffective for complex scenes
[32,36,33,34,23,35,41,39,40,2]	Requires camera calibration
[35, 30]	Lack of generalization

Table 2.10: Comparative summary on portion estimation approaches advantages

Paper	Advantages
Traditional manual methods [23]	Simple and straightforward
AR and VR [24]	Dynamic and interactive assessment
Visual assessment [25]	Provides valuable insights
Miyazaki et al. [27]	Leverages visual similarities for estimation
Beijbom et al. [29]	Simplifies calorie estimation process
Fang et al [26]	Minimizes energy estimation error rate
Kirii. D [31]	Enhances accuracy of food region extraction
3D geometric models [32,33,34,23,35]	Accurate portion size determination

Table 2.10: (Continued)

Paper	Advantages
Xu et al. [36]	Improves nutrient content assessment
Eat AR and Serv AR [37, 38]	Enables precise volume measurements
Joachim Dehais [39] and Wen Wu [40]	Highly accurate food volume estimation
Konstantakopoulos [2]	Facilitates accurate 3D reconstruction
FoodSAM [28]	Achieves advanced segmentation
Ma et al [30]	Improves image-based dietary assessments
Xu and Lu [41]	Significantly reduces estimation errors
Xu et al. [36]	Very high accuracy, rapid, and high throughput

Table 2.11: Simplified overview on portion estimation advantages

Papers	Main Advantage
[23]	Simple and straightforward methods
[24]	Dynamic and interactive assessment
[25,31, 39,40]	Accuracy and insightfulness
[27,29,26,35,30]	Efficient estimation processes
[32,36,33,34,23, 35,41,2,37,38]	Accurate size and volume determination

Chapter 3: Semi-Automatic Pipeline

3.1 Semi-Automatic Pipeline Overview

The semi-automatic pipeline focuses on ingredient recognition. Here we are going to describe the overview of the pipeline flow and then go more in-depth about each component. The pipeline starts with the user uploading an image of a meal or capturing one using their camera. Then, an image description is generated by prompting the BLIP-2 model. Once the description is generated the ingredients are extracted from it using GPT-3.5. The detected ingredients are then populated in the app where the user will be able to input the portion sizes. Once the portion sizes have been provided for the ingredients, then you can calculate the nutritional profile of the meal using a GPT-3.5 model that has been fine-tuned on USDA nutritional dataset [42]. The nutritional profile is exhaustive as it includes both macronutrients and micronutrients. The data is stored internally in the FirebaseDB for further analysis, which is not covered in this paper. The entire pipeline is shown in Figure 3.1. We are now going to describe each component more in-depth.

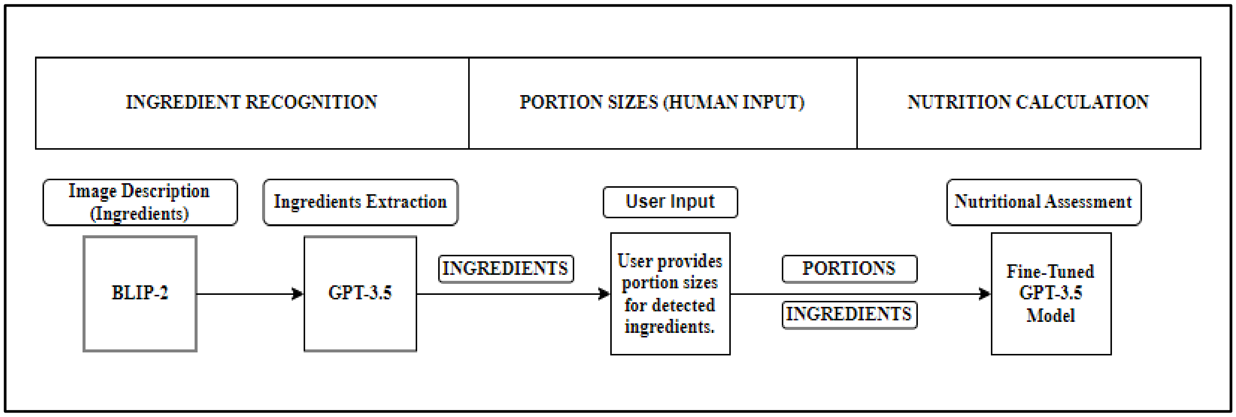


Figure 3.1: Visualization of the semi-automatic pipeline and its main components

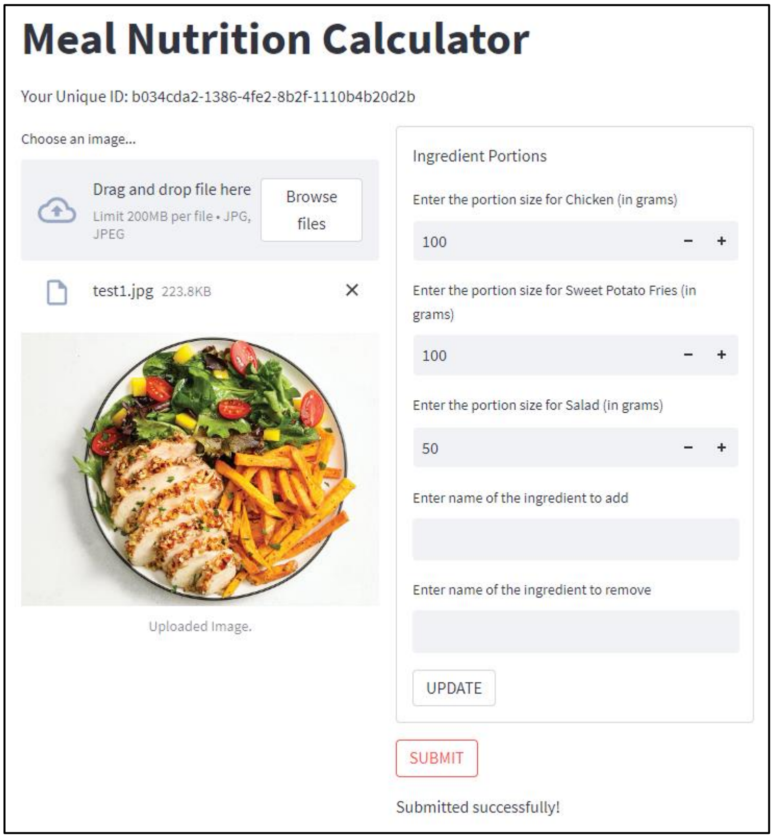
3.2 Ingredient Recognition

The initial stage of our pipeline utilizes a vision-language model BLIP-2 [43] to create a detailed description of the meal from an image provided by the user. This model integrates the capabilities of large language models, more specifically leveraging the Vicuna LLM [44]. As a result, BLIP-2 is promptable, allowing for prompt engineering to refine the prompt and enhance the quality of the descriptions produced. The final prompt we used is: "*List the main food ingredients shown in the image. Provide a concise answer without additional comments about the meal or its surroundings*."

Following the generation of the meal description, we utilize another language model, GPT-3.5, to parse and identify the ingredients. This step is crucial to minimize the occurrence of hallucinations, a common challenge with large language models. Through careful prompt engineering, we designed a prompt that ensures the accurate extraction of ingredients into a structured list format. The prompt used was: *"From the provided sentence, {resp}, extract the food ingredients, formatting the output as a list (e.g., [ingredient1, ingredient2, ingredient3]). For instance, if given 'The recipe includes apples, oranges, and chicken,' the response should be [apple, orange, chicken], focusing solely on actual food components."* This extraction process feeds into the subsequent phase of our pipeline, ensuring a smooth transition and accurate ingredient identification. Multiple meal images uploaded to the app are shown in Figure 3.3 and Figure 3.4 at the end of this chapter.

3.3 User Input

Once the ingredients have been extracted, they are populated in the app and now the user is able to adjust the portion sizes in grams. Grams is the metric used in our calculations for simplicity, but additional support for other metrics and conversions will be implemented in the future. An ingredient addition/removal functionality has been implemented to allow the user to manually change ingredients if an ingredient is incorrectly classified, or an ingredient hasn't been detected. Once the user has verified the ingredients and provided the portion sizes for each of them, the user can hit "UPDATE" and the data has been processed and nutritional calculation can begin. When the user is ready to get the nutritional profile of the meal provided, he can hit "SUBMIT" and the data is passed to the next step in the pipeline where the nutritional assessment takes place. The nutritional data can later be used for further analysis in building health trends so basically expand the functionality of the app further, but this is outside the scope of this thesis. The app's user interface is shown in Figure 3.2. The extensive iterations needed for the development of the app's user interface to provide a smooth consumer experience are outside the scope of this thesis.



Meal Nutrition Calculator

Your Unique ID: b034cda2-1386-4fe2-8b2f-1110b4b20d2b

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, JPEG

Browse files

test1.jpg 223.8KB ×

Uploaded Image.

Ingredient Portions

Enter the portion size for Chicken (in grams)

100 − +

Enter the portion size for Sweet Potato Fries (in grams)

100 − +

Enter the portion size for Salad (in grams)

50 − +

Enter name of the ingredient to add

Enter name of the ingredient to remove

UPDATE

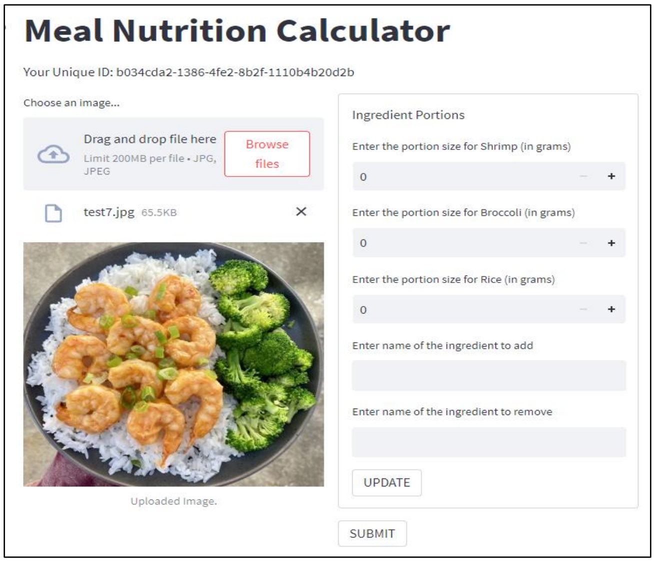
SUBMIT

Submitted successfully!

Figure 3.2: User interface of the app showcasing the "UPDATE" and "SUBMIT" buttons

3.4 Nutrition Calculation

The nutrition calculation step utilizes a GPT-3.5 model, fine-tuned on USDA nutritional dataset. This model processes the list of ingredients and their respective portions to output comprehensive nutritional information for the meal. The nutritional data is not exhaustive as there are certain ingredients or variations of ingredients that are not in the nutritional data that the model has been trained on. To account for that, whenever a detected ingredient is not identified in the nutritional data database, that ingredient is pushed to the Firebase Database under "Unknown Ingredients" for the medical team to review it and update the nutritional database with the respective nutritional information. Then the model is fine-tuned again on the updated database, and the ingredient will be found next time, and its nutritional value will be considered for the final meal nutritional value calculation. If all ingredients are found, the model calculates the nutritional profile for the meal and stores it in the database or visualizes it in the app for the user. In this paper, we store the nutritional profile of the meal internally for review. A snippet of the nutritional profile is shown in Figure 3.5.



Meal Nutrition Calculator

Your Unique ID: b034cda2-1386-4fe2-8b2f-1110b4b20d2b

Choose an image...

Drag and drop file here
Limit 200MB per file • JPG, JPEG

test7.jpg 65.5KB

Uploaded Image.

Ingredient Portions

Enter the portion size for Shrimp (in grams)
0

Enter the portion size for Broccoli (in grams)
0

Enter the portion size for Rice (in grams)
0

Enter name of the ingredient to add

Enter name of the ingredient to remove

Figure 3.3: Example meal image containing rice, broccoli, and shrimp

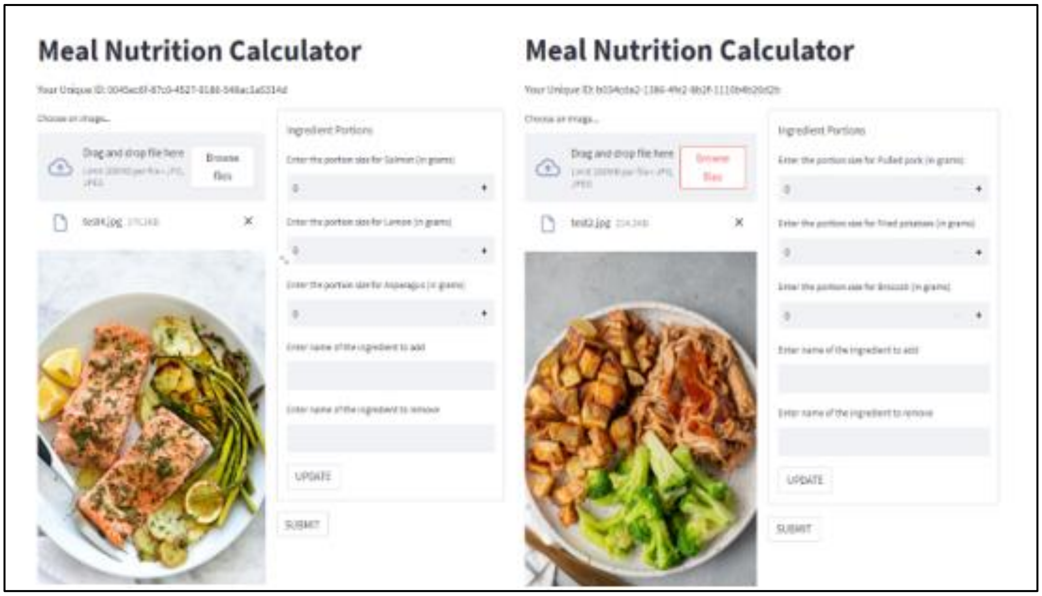
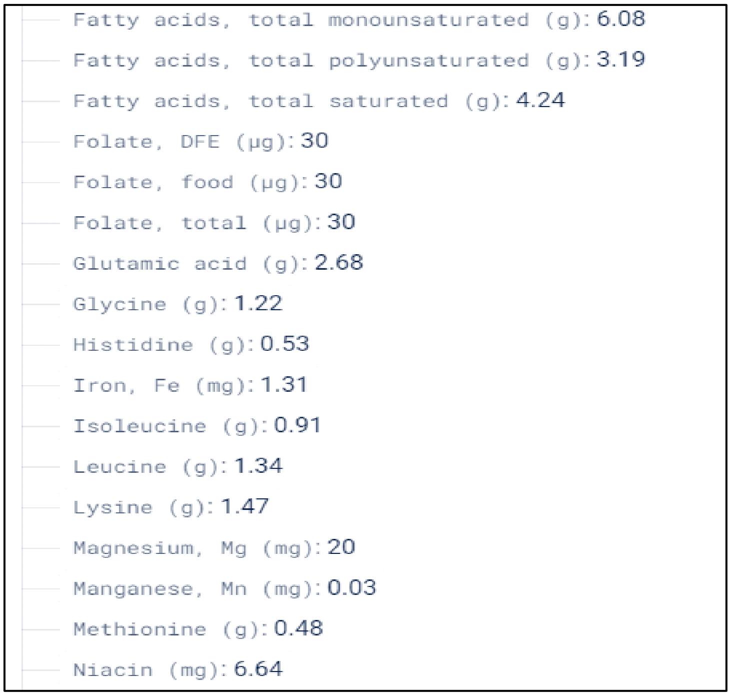

Meal Nutrition Calculator
Choose an image...
Drag and drop file here
Browse files
test4.jpg
Ingredient Portions
Enter the portion size for Salmon (in grams)
0
Enter the portion size for Lemon (in grams)
0
Enter the portion size for Asparagus (in grams)
0
Enter name of the ingredient to add
Enter name of the ingredient to remove
UPDATE
SUBMIT
Meal Nutrition Calculator
Choose an image...
Drag and drop file here
Browse files
test2.jpg
Ingredient Portions
Enter the portion size for Pulled pork (in grams)
0
Enter the portion size for Fried potatoes (in grams)
0
Enter the portion size for Broccoli (in grams)
0
Enter name of the ingredient to add
Enter name of the ingredient to remove
UPDATE
SUBMIT

Figure 3.4: Example meal image containing potatoes, asparagus, salmon, pork, and fries



```
Fatty acids, total monounsaturated (g): 6.08
Fatty acids, total polyunsaturated (g): 3.19
Fatty acids, total saturated (g): 4.24
Folate, DFE (µg): 30
Folate, food (µg): 30
Folate, total (µg): 30
Glutamic acid (g): 2.68
Glycine (g): 1.22
Histidine (g): 0.53
Iron, Fe (mg): 1.31
Isoleucine (g): 0.91
Leucine (g): 1.34
Lysine (g): 1.47
Magnesium, Mg (mg): 20
Manganese, Mn (mg): 0.03
Methionine (g): 0.48
Niacin (mg): 6.64
```

Figure 3.5: Snippet of the nutritional profile of a meal

Chapter 4: Experiments and Results

4.1 Environment Setup

Here we are going to provide a description of our development environment. Firstly, we will describe our hardware configuration. We were hardware restricted as we only had access to a server with a RTX 3090 GPU with 24 GB of VRAM. The CPU is a 24-core AMD Threadripper 3960WX. The model's memory requirement was 18.7 GB of VRAM using 4-bit quantization. The model inference was run on that server. As for the software development side, the main app components are the database, the backend, and the frontend. The development language was Python. We built 2 APIs, one for connecting the frontend to the backend and one for connecting the backend to the server running model inference. This implementation makes our system very modular, and it is easier to switch hardware for model inference if needed. The frontend was built using Streamlit which is a simple Python frontend package. The app was hosted on AWS servers. The system architecture is shown in Figure 4.1, and a summary of technical specifications is shown in Table 4.1

Table 4.1: Overview of critical hardware & software components

Component	**Specification**
GPU	RTX 3090 with 24 GB of VRAM
CPU	24-core AMD Threadripper 3960WX
Memory Requirement	18.7 GB of VRAM with 4-bit quantization
Development Language	Python

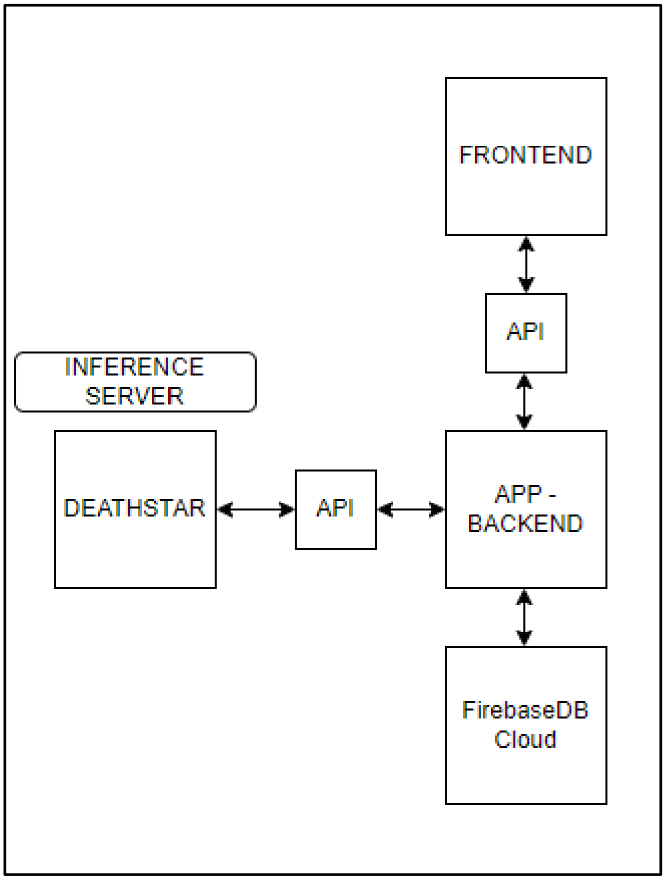


Figure 4.1: Nutrition app system architecture

4.2 Datasets

To assess our pipeline's efficiency, particularly the ingredient recognition part, we chose the UECFOOD100 and UECFOOD256 datasets. This decision was influenced by the fact that the nutrition apps "MyFoodCam" and "FoodTracker," mentioned in our Related Work section, also tested their performance using these datasets. The UECFOOD100 dataset comprises 13,920 images across 100 categories, focusing on Japanese foods, specifically popular dishes in Japan. The images, captured via mobile cameras, make it an ideal dataset for testing a mobile nutrition app in realistic scenarios. Examples of these images are presented in Figure 4.3. Meanwhile, the

UECFOOD256 dataset contains 30,290 images in 256 categories, also emphasizing Japanese foods taken with mobile cameras, with some images showcased in Figure 4.4.

Additionally, we utilized an Internal Dataset, developed in collaboration with our medical team. This dataset includes 38 meal images from various, mainly Western, cuisines. The medical team manually labeled the ingredients in these images, which were captured using a smartphone under natural lighting conditions, closely mimicking real-life situations. A significant portion of these images is displayed in Figure 4.2. An overview of the datasets is shown in Table 4.2.

Table 4.2: Overview of datasets

Dataset	Image Counts	Cuisine	Captured by
Internal	38	Mix, Western	Mobile camera
UECFOOD100	13920	Japanese	Mobile camera
UECFOOD256	30290	Japanese	Mobile camera

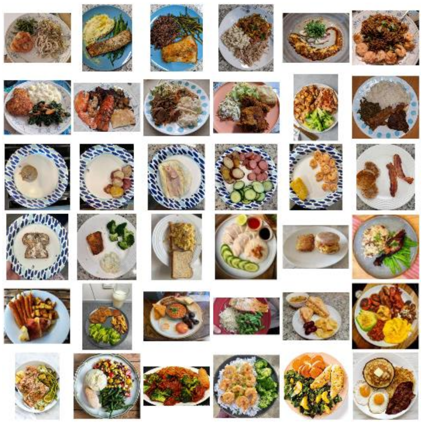


Figure 4.2: Meal images from the internal dataset

Figure 4.3: Meal images from the UECFOOD100 dataset

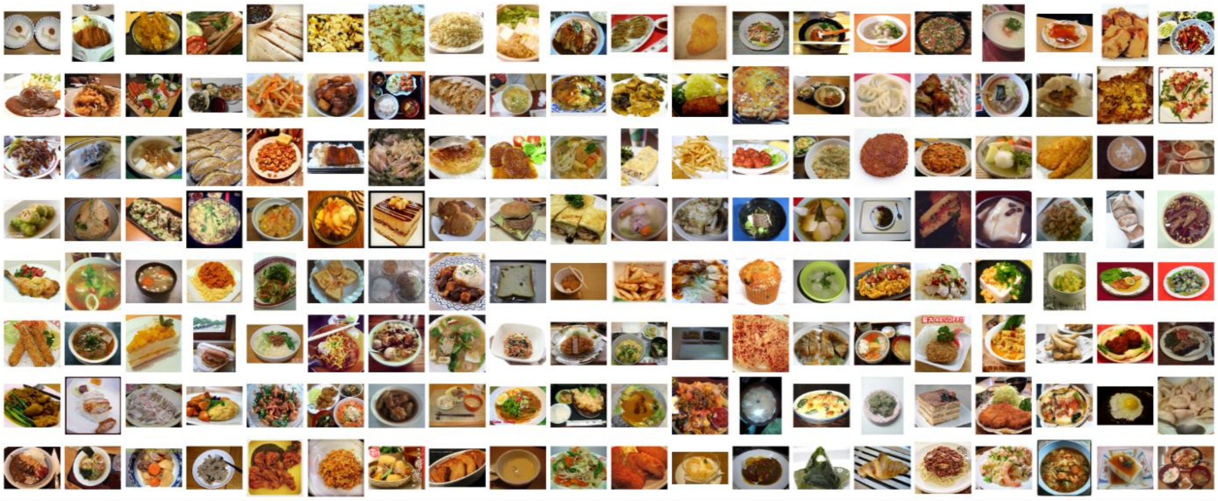


Figure 4.4: Meal images from the UECFOOD256 dataset

4.3 Experiments

In our experiments, we organized the process into two main stages. The first stage focused on fine-tuning the hyperparameters of the BLIP-2/Vicuna model. Through comprehensive testing of various configurations and prompt engineering, we pinpointed the optimal hyperparameters for the large language model (LLM) and the most effective prompts for generating image descriptions and extracting ingredients. We also made specific adjustments to other hyperparameters to refine the output further. Given the reliance on a language model for description generation, which results in output variation with each execution, we concentrated on tweaking the *temperature* and *max_tokens* parameters to manage this variability. The temperature parameter controls the level of unpredictability in the model's predictions—lower values lead to less variation, while the *max_tokens* parameter sets the maximum length of the model's responses. We lowered the temperature to decrease the variability of responses and reduced *max_tokens* to speed up the generation process.

Specifically, we adjusted the key hyperparameters, temperature and *max_length*, to 0.5 and 800, respectively, to minimize the unpredictability of LLM outputs and improve the speed of generation. The details of this optimal hyperparameter configuration can be found in Table 4.3.

For the task of generating meal descriptions, after prompt engineering this was the prompt, we found to be clear and to the point: *"List the main food ingredients shown in the image. Provide a concise answer without additional comments about the meal or its surroundings."* This prompt was designed to yield straightforward descriptions focusing on the meal's ingredients.

Following this, we utilized GPT-3.5 for the precise and efficient extraction of ingredients from the generated descriptions. The prompt we used for this step was: *"From the provided sentence, {resp}, extract the food ingredients, formatting the output as a list (e.g., [ingredient1, ingredient2, ingredient3])." An example provided for clarity was, "If the input is 'The recipe includes apples, oranges, and chicken,' the output should be [apple, orange, chicken]," emphasizing the focus on actual food ingredients."*

In the second stage of our experiments, we concentrated on evaluating our semi-automatic pipeline using the UECFOOD100, UECFOOD256, and our own internal dataset. We applied our ingredient recognition pipeline to the entirety of the internal dataset. Notably, our model was neither pre-trained nor fine-tuned using this dataset, and for each image, we generated a single prediction. This approach yielded a classification accuracy of 96% and a detection accuracy of 86%. To derive an overall accuracy measure, we multiplied these two scores, resulting in an average accuracy of 83%. This result was very good.

Following the internal dataset tests, we proceeded to evaluate our pipeline on the UECFOOD100 and UECFOOD256 datasets. Recognizing that comparable apps mentioned in our Related Work section, such as "MyDietCam" and "FoodTracker," were trained on these datasets and assessed their models on about 10% of the data, we opted to test our model on a similar 10% sample. This approach aimed to maintain consistency with the testing environments used by others. However, our testing methodology diverged in several key aspects.

Firstly, unlike the models in the Related Work, ours had not been pre-trained or fine-tuned on these datasets. Consequently, the classes our model predicted sometimes differed from those in the datasets, partly because the datasets primarily consist of Japanese foods, including classes not represented in English.

To ensure a fair and accurate testing environment despite these differences, we implemented specific measures. We randomly selected 10% from each class to create a balanced dataset and avoid the risk of including sequential images that might represent the same item from different angles. This random sampling was intended to introduce stochasticity. Additionally, aware of the potential for indeterminacy in large language model outputs, we decided to limit our predictions to a single run per image. This approach might slightly underestimate our model's true performance, but it was considered necessary. Despite our efforts to sample randomly, manual inspection revealed occasional instances of the same image captured from various viewpoints.

Given that the benchmark set by related works utilized ingredient recognition accuracy for performance validation, we adopted the same metric for our evaluation. This consistent measure allows for a direct comparison of our results against those established in the field.

Table 4.3: Optimal Vicuna-LLM hyperparameter configuration

Hyperparameter	**Optimal Value**
Temperature	0.5
top_p	0.9
Repetition_penalty	1.0
Length_penalty	1
Max_new_tokens	400
Max_length	800

4.4 Baseline

We'll use the performance outcomes of mobile apps discussed in the Related Work section as our benchmark. Specifically, "MyDietCam" achieved food recognition accuracies of 88.74% on the UECFOOD100 dataset and 76.51% on the UECFOOD256 dataset. Meanwhile, "FoodTracker" reported mean average precision scores of 76.35% for UECFOOD100 and 75.05% for UECFOOD256. These results set the baseline for our comparison, allowing us to directly evaluate how our pipeline stacks up against these established performances.

4.5 Evaluation

Evaluating a model on datasets where it hasn't been trained or fine-tuned presents a significant challenge due to discrepancies in class predictions. These classes may not align perfectly or might represent variations of each other. To address this, we employed semantic similarity to assess performance accurately. By comparing the ground truth with the model's predictions, we can consider a prediction correct if the ingredients are variations of one another or are semantically similar. This approach was straightforward for our small Internal dataset, which comprised only 38 images with manually labeled ingredients. We evaluated the model using two primary metrics: ingredient classification accuracy (the percentage of ingredients correctly classified) and ingredient detection accuracy (the percentage of ingredients correctly detected). An aggregate score, derived from the product of these two metrics, provided the final score.

However, manual evaluation becomes significantly more challenging with larger datasets, especially those representing complex cuisines like UECFOOD100 and UECFOOD256, which also lack labeled ingredients. The primary challenges included adapting to the UECFOOD datasets without prior training, handling disparities in dataset sizes, and overcoming expertise limitations in semantic similarity assessments and dish-to-ingredient conversions. These challenges inspired the need for a comprehensive evaluation framework. This framework would enable comparative analysis against other food recognition models, focusing on performance assessment, model comparison, recognition focus differentiation, and evaluation of generalizability.

To address these challenges, we proposed using GPT-4, fine-tuned through prompt engineering, to assess and compare the generalizability and performance of our ingredient recognition model against others. This strategy aims to mitigate the discrepancies in dataset focuses and training backgrounds by leveraging GPT-4's ability for semantic similarity assessments and the conversion of dish names to ingredient lists. This ensures more accurate comparisons when direct matches are impossible by missing specific ingredients or food classes.

We chose GPT-4 for its advanced semantic comparison capabilities and its continuously updated knowledge base, which almost guarantees relevance and accuracy. The deterministic outcomes provided by precise prompt engineering offer a novel evaluation approach. We formulated a method where, for each image classification task, the ground truth and prediction are given in a specified format. This allows GPT-4 to assess semantic similarity and, when necessary, compare the ingredients of dishes or meals. The prompt we used is: *“I have an image classification task. I will provide you with the ground truth and my prediction in this format: image name | ground truth | prediction. For each image, indicate 'yes' if they are semantically similar, otherwise 'no'. Additionally, if the ground truth involves a dish or meal, identify some ingredients and compare them with my prediction. If semantic similarity exists, consider it a correct prediction. Calculate the percentage of 'yes' responses out of all responses given.”*

This percentage serves as our metric for ingredient recognition accuracy, aligning with the evaluation metrics used by other apps. This provides a fair comparison between our pipeline and theirs. By integrating GPT-4 into our evaluation framework, we address the identified challenges and set a new benchmark for assessing food and ingredient recognition models, enabling effective performance comparisons across any dataset.

4.6 Results

Within our internal dataset, we achieved a combined accuracy of 83%, with our classification accuracy at 96% and detection accuracy at 86%. Next, we applied our pipeline to the UECFOOD100 dataset, using the previously outlined evaluation framework, and got an ingredient recognition accuracy of 81%. In comparison, “MyDietCam” achieved an accuracy of 88.74%, while “FoodTracker” achieved 76.35%.

Next, we tested our pipeline on the UECFOOD256 dataset and achieved an ingredient recognition accuracy of 75%. For comparison, "MyDietCam" scored 76.51%, and "FoodTracker" scored 75.05%. These tests show how our pipeline competes with the established benchmarks in the field, shown in Table 4.4. Some of the results of the pipeline in each dataset are shown in Tables 4.5, 4.6 and 4.7.

Table 4.4: Results of our pipeline in different datasets compared to baseline

Dataset	**Our Pipeline**	**MyDietCam**	**FoodTracker**
Internal	83%	N/A	N/A
UECFOOD100	81%	88.74%	76.36
UECFOOD256	75%	76.51%	75.05

Table 4.5: Snippet of the results of the semi-automatic pipeline's performance on internal dataset

Meal	**Ingredients (Ground Truth)**	**Prediction**	**Ingredient Classification Accuracy**	**Ingredient Detection Accuracy**
Figure 4.5	fried chicken, french fries, milk, broccoli	french fries, fried chicken, broccoli	100%	75%
Figure 4.6	noodles, shrimp, green onions	noodles, shrimp, green onions	100%	100%

Table 4.6: Snippet of the results of the semi-automatic pipeline’s performance on UECFOOD100 dataset

Meal	Ground Truth	Prediction	Result
Figure 4.7	ramen noodle	bowl, ramen noodles, boiled egg	correct
Figure 4.8	rice, miso soup	rice, miso soup rice, vegetables, meat, soy sauce, ginger, garlic	correct

Table 4.7: Snippet of the results of the semi-automatic pipeline’s performance on UECFOOD256 dataset

Meal	Ground Truth	Prediction	Result
Figure 4.9	goya chanpuru	bowl, ramen noodles, boiled egg	correct
Figure 4.10	pork cutlet on rice	rice, miso soup rice, vegetables, meat, soy sauce, ginger, garlic	incorrect

Below are provided examples of meals from the UECFOOD100 and UECFOOD256 datasets which were used to evaluate our model's performance. These are complex dishes for which even the ground truth may not encapsulate the complexity of the dish.

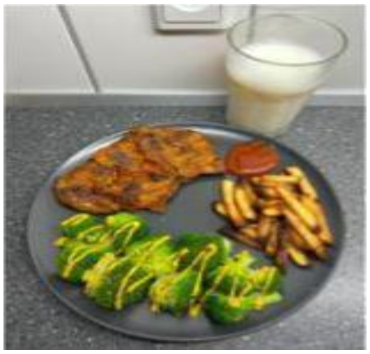


Figure 4.5: Meal from internal dataset containing broccoli, french fries, milk, and fried chicken

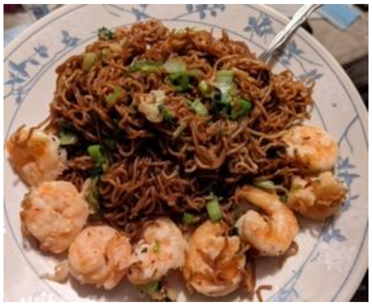


Figure 4.6: Meal from internal dataset containing noodles, shrimp, green onions

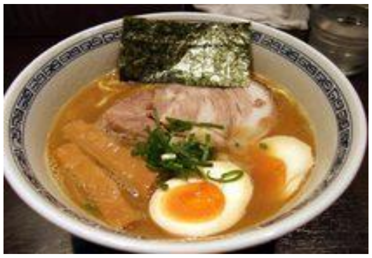


Figure 4.7: Meal from UECFOOD100 dataset which contains ramen noodle

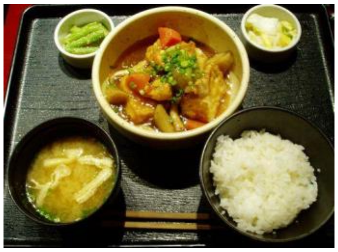


Figure 4.8: Meal from UECFOOD100 dataset which contains rice, miso soup

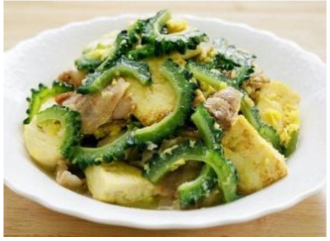


Figure 4.9: Meal from UECFOOD256 dataset which contains goya chanpuru

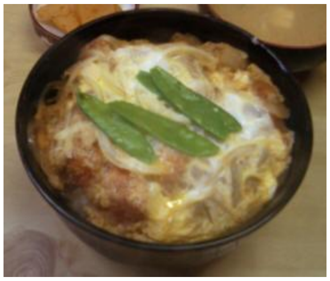


Figure 4.10: Meal from UECFOOD256 dataset which contains pork cutlet on rice

Chapter 5: Discussion

Our pipeline achieved a combined accuracy of 83% on the internal dataset, which includes both detection and classification accuracies. If we focus only on classification, the accuracy reaches about 96%. However, given the small size of this dataset, we can't solely rely on these results to judge our pipeline's overall performance and ability to generalize. On the UECFOOD100 dataset, our accuracy was 81%, surpassing the "FoodTracker" baseline of 76.35% but falling short of "MyDietCam's" 88.74%. For the UECFOOD256 dataset, our accuracy dropped to 75%, which is understandable given the increase to 256 classes from 100. Our performance is comparable to the established baselines, which is encouraging. However, these comparisons need careful consideration. Our pipeline's accuracies come from datasets it wasn't specifically trained or fine-tuned on.

For the internal dataset, inaccuracies primarily arose from BLIP-2's generated descriptions. With UECFOOD100 and UECFOOD256, the errors stemmed from both BLIP-2 and GPT-4's evaluations. Differences in class predictions could lead to false negatives. Despite some predictions marked as incorrect by GPT-4 potentially being correct upon manual review, we chose not to adjust these to maintain evaluation consistency.

These results suggest our model's strong generalization capability across various datasets, particularly with language variations and unseen classes. Further accuracy improvements seem possible with model fine-tuning or additional prediction iterations. Had our model been fine-tuned on 80-90% of the datasets like the other models, our accuracy likely would have significantly surpassed the baselines. This underscores the impressive generalization ability of vision-language models.

Another key observation is the advantage of detecting ingredients over specific dishes for accurate nutrition estimation. Meals can vary greatly, so focusing on ingredients allows for broader recognition with fewer classes and lower computational demands compared to traditional CNN models for meal recognition. This approach improves the app's versatility across different cuisines, facilitating accurate ingredient recognition even in foreign dishes.

Chapter 6: Conclusion and Future Work

This pipeline demonstrated strong performance in ingredient recognition accuracy across the UECFOOD100, UECFOOD256, and our internal datasets. Despite not being fine-tuned or trained on these specific datasets, it showcased impressive generalization capabilities, making it well-suited for a variety of cuisines and complex dishes. However, the absence of certain ingredients in the evaluation could still affect the model's performance negatively. To improve, expanding the range of ingredients the model recognizes through further training or fine-tuning is essential. The broader the ingredient recognition, the more accurate the nutrition estimation for various dishes will become. Based on these outcomes, we plan to fine-tune the model on additional ingredients and food datasets to enhance its accuracy and generalization further.

As our goal is to perform an accurate nutritional estimation for a user-provided meal, we understand that it is important to account for the different state of the ingredients, and different ways of cooking. Spices, sauces, and oils are key in cooking, and they may sometimes contribute to most of the nutritional profile of a meal. Being able to recognize the type of sauce, and ways of cooking an ingredient is the next step in ensuring the app can accurately calculate the nutrition. To make this possible, significant training and fine-tuning on large datasets will be essential. We will utilize prompt engineering to explore the current limitations of our model, and future ones that we intend to utilize. Reaching that level of fine-grained recognition will require a lot of resources and fine-tuning and it is important to try and address such limitations in the short term. We can achieve this by introducing additional functionality to our nutritional calculator app. Many of the development iterations of the app’s user interface and functionality were outside the scope of this thesis, as we primarily focused on the ingredient recognition pipeline. However, for future work we intend to build a large database of ingredient states and cooking methods and allow the user to select the corresponding state or cooking method for a detected ingredient in the app. To make the process smoother we will introduce autocomplete and suggestions, with the goal of providing more accuracy in the final nutritional estimation the app performs.

Furthermore, in the user interface side we intend to expand the steps in how the user interacts with the interface. To facilitate a seamless experience, we are going to separate certain specific ingredient state/cooking selections into different pages so that the user is not overwhelmed with a lot of information all at once. This approach could also help the elderly and people who are less tech savvy to have a better experience interacting with the app.

The nutritional profile calculated by the app is very exhaustive containing information about macronutrients and micronutrients. That information may not be very useful to all our users so they will be able to specify the information that will be presented to them. Additionally, the data will be recorded to generate metrics for them so they can better understand their eating habits and meet their diet-specific requirements.

It is our goal to provide the most accurate nutritional calculation, which is why we are going to work on developing a Fully Automatic pipeline that combines automatic ingredient recognition with automatic portion size estimation. This aims to enhance the user experience by minimizing manual inputs completely. We're considering using technologies like Grounding-DINO and SAM-HQ for region-based segmentation to create accurate masks for ingredients, which, combined with height estimation or another method, will help calculate the meal's volume accurately. These are tentative models that we are going to test. Ultimately, we will use the latest state-of-the-art models and fine-tune them to our needs. An important element to consider is the fact that people may not always finish their meal. We intend to address that by allowing the user to take a photo of the final meal, and the actual nutritional consumption will be the difference between the initial meal and the final meal.

In the meantime, we plan to roll out the app to a select group of users for thorough testing. Their feedback and data-driven insights will be invaluable in refining the model's accuracy, generalization, and overall user experience. Through this iterative process of improvement, we aim to align the nutrition calculator app with evolving user needs and push forward the technology for dietary analysis.

References

[1] Marc, B.; Ferrà, A.; Radeva, P. "Food ingredients recognition through multi-label learning." In Proceedings of the International Conference on Image Analysis and Processing, Catania, Italy, 11–15 September 2017; Springer: Cham, Switzerland, 2017.

[2] Konstantakopoulos, F.S.; Georga, E.I.; Fotiadis, D.I. "An Automated Image-Based Dietary Assessment System for Mediterranean Foods." IEEE Open Journal of Engineering in Medicine and Biology, vol. 4, pp. 45-54, 2023. doi: 10.1109/OJEMB.2023.3266135.

[3] Chen, J.; Ngo, C. "Deep-based Ingredient Recognition for Cooking Recipe Retrieval." In Proceedings of the 24th ACM International Conference on Multimedia (MM'16), Amsterdam, The Netherlands, 15–19 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 32–41.

[4] Chen, J.; Pan, L.; Wei, Z.; Wang, X.; Ngo, C.-W.; Chua, T.-S. "Zero-Shot Ingredient Recognition by Multi-Relational Graph Convolutional Network." In Proceedings of the AAAI Conference on Artificial Intelligence 2020, New York, NY, USA, 7–12 February 2020; vol. 34, pp. 10542–10550.

[5] Chen, J.; Zhu, B.; Ngo, C.-W.; Chua, T.-S.; Jiang, Y.-G. "A Study of Multi-Task and Region-Wise Deep Learning for Food Ingredient Recognition." IEEE Transactions on Image Processing, 2021, vol. 30, pp. 1514–1526.

[6] Salvador, A.; Drozdzal, M.; Giro-i-Nieto, X.; Romero, A. "Inverse cooking: Recipe generation from food images." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.

[7] Gao, X.; Xiao, Z.; Deng, Z. "High Accuracy Food Image Classification via Vision Transformer with Data Augmentation and Feature Augmentation." Journal of Food Engineering, 2024, vol. 365, 111833. [CrossRef] [DOI: 10.1016/j.jfoodeng.2023.111833]

[8] Tahir, G.A.; Loo, C.K. "An open-ended continual learning for food recognition using class incremental extreme learning machines." IEEE Access, 2020, vol. 8, pp. 82328–82346.

[9] Sun, J.; Radecka, K.; Zilic, Z. "FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks." arXiv [cs.CV], 2019. Retrieved from http://arxiv.org/abs/1909.05994

[10] Ming, Z.-Y.; Chen, J.; Cao, Y.; Forde, C.; Ngo, C.-W.; Chua, T. "Food Photo Recognition for Dietary Tracking: System and Experiment." 2018. doi:10.1007/978-3-319-73600-6_12

[11] Tsoumakas, G.; Katakis, I. "Multi-label classification: An overview." International Journal of Data Warehousing and Mining, 2006, vol. 3, pp. 1–3.

[12] Zhou, W.; Dou, P.; Su, T.; Hu, H.; Zheng, Z. "Feature Learning Network with Transformer for Multi-Label Image Classification." Pattern Recognition, 2023, vol. 136, 109203. [CrossRef] [DOI: 10.1016/j.patcog.2022.109203].

[13] Wang, Y.; Chen, J.-J.; Ngo, C.-W.; Chua, Y.-S.; Zuo, W.; Ming, Z. "Mixed Dish Recognition through Multi-Label Learning." In Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities (CEA'19), Ottawa, ON, Canada, 10 June 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–8.

[14] Zhang, M.; Tian, G.; Zhang, Y.; Liu, H. "Sequential Learning for Ingredient Recognition From Images." IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 5, pp. 2162-2175, May 2023, [DOI: 10.1109/TCSVT.2022.3218790].

[15] Liu, C.; Liang, Y.; Xue, Y.; Qian, X.; Fu, J. "Food and Ingredient Joint Learning for Fine-Grained Recognition." IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 6, pp. 2480-2493, June 2021, [DOI: 10.1109/TCSVT.2020.3020079].

[16] Min, W.; Liu, L.; Luo, Z.; Jiang, S. "Ingredient-Guided Cascaded Multi-Attention Network for Food Recognition." In Proceedings of the 27th ACM International Conference on Multimedia (MM '19), Association for Computing Machinery, New York, NY, USA, 2019, pp. 1331–1339. [https://doi.org/10.1145/3343031.3350948].

[17] Pellegrini, C.; Özsoy, E.; Wintergerst, M.; Groh, G. "Exploiting Food Embeddings for Ingredient Substitution." 2021, pp. 67-77, [DOI: 10.5220/0010202000670077].

[18] Mezgec, S.; Eftimov, T.; Bucher, T.; Koroušić Seljak, B. "Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment." Public Health Nutrition, vol. 22, no. 7, pp. 1193–1202, 2019, [DOI: 10.1017/S1368980018000708].

[19] Chhikara, P.; Chaurasia, D.; Jiang, Y.; Masur, O.; Ilievski, F. "FIRE: Food Image to REcipe generation." arXiv:2308.14391 [cs.CV], 2023, [https://doi.org/10.48550/arXiv.2308.14391].

[20] Zhu, Z.; Dai, Y. "A New CNN-Based Single-Ingredient Classification Model and Its Application in Food Image Segmentation." Journal of Imaging, 2023, vol. 9(10), 205. [https://doi.org/10.3390/jimaging9100205].

[21] Wang, H.; Lin, G.; Hoi, S.C.; Miao, C. "Structure-Aware Generation Network for Recipe Generation from Images." European Conference on Computer Vision, 2020.

[22] Minija, S.J.; Emmanuel, W.R.S. "Food recognition using neural network classifier and multiple hypotheses image segmentation." The Imaging Science Journal, 68:2, 100-113, 2020, DOI: 10.1080/13682199.2020.1742416.

[23] Godwin, S.; Chambers, E.T.; Cleveland, L.; Ingwersen, L. "A new portion size estimation aid for wedge-shaped Foods." J. Am. Diet. Assoc., 2006, 106: 1246–1250.

[24] Zhang, Z.; Yang, Y.; Yue, Y.; Fernstrom, J.D.; Jia, W.; Sun, M. "Food volume estimation from a single image using virtual reality technology." In Proceedings of the 2011 IEEE 37th Annual Northeast Bioengineering Conference (NEBEC), Troy, NY, USA, 1 April 2011; pp. 1–2.

[25] Comber, R.; Weeden, J.; Hoare, J.; Lindsay, S.; Teal, G.; Macdonald, A.; Methven, L.; Moynihan, P.; Olivier, P. "Supporting visual assessment of food and nutrient intake in a clinical care setting." In Proceedings of the Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 919–922. [CrossRef] [PubMed]

[26] Fang, S. et al. "Single-View Food Portion Estimation: Learning Image-to-Energy Mappings Using Generative Adversarial Networks." 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 251-255, DOI: 10.1109/ICIP.2018.8451461.

[27] Miyazaki, T.; de Silva, G.C.; Aizawa, K. "Image-based Calorie Content Estimation for Dietary Assessment." In Proceedings of the 2011 IEEE International Symposium on Multimedia, Dana Point, CA, USA, 5–7 December 2011; pp. 363–368.

[28] Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; Dollár, P.; Girshick, R.B. "Segment Anything." 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 3992–4003.

[29] Beijbom, O.; Joshi, N.; Morris, D.; Saponas, S.; Khullar, S. "Menu-Match: Restaurant-Specific Food Logging from Images." In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 844–851.

[30] Ma, J.; He, J.; Zhu, F.M. "An Improved Encoder-Decoder Framework for Food Energy Estimation." Proceedings of the 8th International Workshop on Multimedia Assisted Dietary Management, 2023.

[31] Kirii, D.; Futagami, T. "Improved Food Region Extraction Using State-of-the-Art Saliency Detection." 精密工学会誌, 2023, 89, 12, 949–955. [CrossRef] [DOI: 10.2493/jjspe.89.949]

[32] He, Y.; Xu, C.; Khanna, N.; Boushey, C.J.; Delp, E.J. "Food image analysis: Segmentation, identification and weight estimation." In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME), San Jose, CA, USA, 15–19 July 2013; pp. 1–6.

[33] Khanna, N.; Boushey, C.J.; Kerr, D.; Okos, M.; Ebert, D.S.; Delp, E.J. "An Overview of The Technology Assisted Dietary Assessment Project at Purdue University." In Proceedings of the 2010 IEEE International Symposium on Multimedia, ISM 2010, Taichung, Taiwan, 13–15 December 2010; pp. 290–295.

[34] Jia, W.; Yue, Y.; Fernstrom, J.D.; Zhang, Z.; Yang, Y.; Sun, M. "3D localization of circular feature in 2D image and application to food volume estimation." In Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, USA, 29 August 2012; pp. 4545–4548.

[35] Fang, S.; Liu, C.; Zhu, F.; Delp, E.J.; Boushey, C.J. "Single-View Food Portion Estimation Based on Geometric Models." In Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM), Miami, FL, USA, 14–16 December 2015; pp. 385–390.

[36] Xu, C.; He, Y.; Khanna, N.; Boushey, C.J.; Delp, E.J. "Model-based food volume estimation using 3D pose." In Proceedings of the 2013 IEEE International Conference on Image Processing, Melbourne, VIC, Australia, 15–18 September 2013; pp. 2534–2538.

[37] Stütz, T.; Dinic, R.; Domhardt, M.; Ginzinger, S. "Can mobile augmented reality systems assist in portion estimation? A user study." In Proceedings of the 2014 IEEE International Symposium on Mixed and Augmented Reality—Media, Art, Social Science, Humanities and Design (ISMAR-MASH’D), Munich, Germany, 10–12 September 2014; pp. 51–57.

[38] Rollo, M.E.; Bucher, T.; Smith, S.P.; Collins, C.E. "ServAR: An augmented reality tool to guide the serving of food." International Journal of Behavioral Nutrition and Physical Activity, 2017, 14: 65.

[39] Dehais, J.; Anthimopoulos, M.; Shevchik, S.; Mougiakakou, S. "Two-View 3D Reconstruction for Food Volume Estimation." IEEE Transactions on Multimedia, 2017, 19: 1090–1099.

[40] Wu, W.; Yang, J. "Fast food recognition from videos of eating for calorie estimation." In Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, New York, NY, USA, 28 June–3 July 2009; pp. 1210–1213.

[41] Xu, J.; Lu, Y.; Olaniyi, E.; Harvey, L. "Online volume measurement of sweetpotatoes by A LiDAR-based machine vision system." Journal of Food Engineering, 2024, 361, 111725. doi:10.1016/j.jfoodeng.2023.111725

[42] U.S. Department of Agriculture, Agricultural Research Service. "USDA FoodData Central." U.S. Department of Agriculture, 2023. https://fdc.nal.usda.gov

[43] Li, J.; Li, D.; Savarese, S.; Hoi, S. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models." arXiv [Cs.CV], 2023. Retrieved from http://arxiv.org/abs/2301.12597

[44] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., … Stoica, I. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv [Cs.CL]. Retrieved from http://arxiv.org/abs/2306.05685