National College of Ireland

Food Recognition Tool for Dietary Management

MSc Research Project
Research of Computing

Gautam Bhatia
Student ID: x22171118

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gautam Bhatia
Student ID:	x22171118
Programme:	Research of Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	11/08/2024
Project Title:	Food Recognition Tool for Dietary Management
Word Count:	6874
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gautam Bhatia
Date:	15th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Food Recognition Tool for Dietary Management

Gautam Bhatia
x22171118

Abstract

International students usually face challenges with food when they move abroad, especially if they have specific dietary restrictions or allergies. This research aims to help these students by developing a tool that uses advanced deep learning models such as CLIP (Contrastive Language-Image Pretraining) and MobileNetV2 to recognise food ingredients and allergens. The goal is to assist students in identifying what's in their food and finding recipes that suit their dietary needs. We compared the performance of both models to the recognition of food. our results show that CLIP performs better than MobileNetV2 and CLIP recognize various food categories. This study contributes to the food computing field by showing the best tool that can address real-life dietary issues faced by international students. This tool was developed to improve food safety and diet management for users. Future work will focus on improving the models and adding more features to make the tool even more useful. This research highlights the use of deep learning models to create solutions for everyday problems.

1 Introduction

The journey of pursuing a master's from abroad is an important milestone for many students worldwide. This experience brings numerous challenges for international students; one of the most difficult is the adaptation to a new culinary environment. Cooking and meal prep are valuable for my daily life as a student. Most people are unfamiliar with local food products and culinary practices. This issue is even greater for students with food allergies and specific dietary needs. Recognising this need, it is important to develop a helpful tool that assists these students in navigating their new food environments safely and confidently.

International students usually struggle with identifying local ingredients, understanding different cooking methods, managing their diet restrictions, and having food allergies This is a huge problem to solve for students who live alone and are far from home. One more difficulty is that most people don't go to eat food from outside due to a lack of understanding of their local food labels and this will not always meet their nutritional needs. In this problem, we need innovative solutions that can help students manage their food diet effectively. Another problem for international students is the language barrier, which makes reading and understanding food labels difficult because different countries have different ways of labelling their allergens information and nutritional information. we need a system that can help this student to identify dish names, ingredients and allergens in their food.

1.1 Research Motivation

This research aims to create a tool that helps international students manage their dietary needs by recognizing food ingredients and allergens information. Deep learning and computer vision technologies, help to build this kind of tool, specifically CLIP (Contrastive language image pretraining and MobileNetV2 these two models give the solution to this problem. These technologies can be used to develop a system that identifies food names, ingredients and allergens from food images, which helps students choose ingredients and allergens for food to meet their preferences.

Deep learning models such as CLIP and MobileNetV2, have shown great success in image recognition tasks. CLIP can understand images and text together and MobileNetV2 is known for its efficiency in mobile and embedded vision applications, Both models provide an excellent base for developing this food recognition system. By using these technologies, this research aims to provide an accessible solution for international students who facing dietary challenges.

1.2 Research Question and Objectives

The main research question that should be addressed is: *How can CLIP models enhance the recognition of food ingredients and allergens information of dish recognition and recipe retrieval assistant for users with specific dietary restrictions?* To answer this question, the research has some objectives:

1. **Model Training**: Train and fine-tune the CLIP and MobileNetV2 models to accurately identify food ingredients and potential allergens.

2. **Performance Metrics:** Evaluate the models on the test set to measure the performance of the models we used these metrics: accuracy, precision, recall and F1-Score to see that the model is recognizing food.

3. **User Interface:** Create a user-friendly interface that allows users to input food images, and receive ingredient and allergen information.

4. **Practical Utility:** Assess the suitability and usefulness of the developed system in the real-life world, focusing on its impact on dietary management for international students.

This research holds important potential in addressing a critical need among international students. By providing a tool that helps in food preparation and ensures the safety of those who have dietary restrictions. The ability to recognize and categorize ingredients present in the food also supports students in maintaining a balanced and safe diet.

1.3 Report Structure

The structure of the report are going this process: First related works in this section reviews previous researches on food recognition, deep learning models , datasets and their applications for food recognition systems these all are the things that previously done with authors. Now next is the methodology in this we discuss the methods and processes which i used to implement our system and describe the models that used which is MobileNetV2 and CLIP after the methodology next is design specification in this

section the author create a architecture diagram and framework which used to make a system of dish image to recipe retrieval.Then implementation in this section the author discusses how user will interact with this project then evaluation in this section author evaluate the model and check the performances of the evaluation of models on test set and shows the effectiveness of the model.The final section is conclusion and future work which summarizes the key findings and discuses the research objectives and suggests future directions for improving the system of dish image-to-recipe retrieval.

2 Related Works

2.1 Introduction

Food recipe recognition, a subset of image classification tasks, plays a vital role in various fields like dietary monitoring, restaurant management, and the analysis of food nutrition with advancements in technologies in deep learning. So first, understand what deep learning is. It is a subfield of artificial intelligence and machine learning inspired by the human brain's structure. Basically, deep learning is performed with the help of deep networks, which are called neural networks with multiple hidden layers. The architecture of deep learning is based on biological neurons so, in the deep neural network, the first layer is an input layer where we give the input to our neural network, the second is a hidden layer and the third is the output layer, so deep learning algorithms are complex. There are different types of deep learning models to solve specific problems. CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), LSTMs (Long Short–Term Memory Networks), GANs (Generative Adversarial Networks) and lastly, autoencoders. For this food recognition task, basically image classification, we use CNN (Convolutional Neural Networks). This literature review explores the methods that authors used previously, datasets, challenges, and future directions in the domain of food recognition using deep learning.

2.2 Overview of Deep Learning for Food Recognition

The explosion of work related to computer vision and deep learning has extremely transformed how we interact with and understand the culinary world. It gave birth to an emerging and growing field in food computing. The key to increasing computation is that there are already a lot of tasks, including monitoring dietary intake, analyzing food nutrition and many more. Such problems can be solved with the image classification task, including food recognition. This task does the automatic extraction of features with complex patterns from a large dataset. This overview will help you to understand the challenges that exist in the image classification task and their solutions. Deep learning is a subset of machine learning that uses deep neural networks and deep learning is highly effective in image classification tasks. There are some advantages to the use of deep learning it can handle large datasets and it is flexible to adapt specific tasks such as food recognition. Deep learning has learning capabilities, and it can automatically extract the features. We can clearly say that deep learning is well-suited for image classification tasks. Many Researchers have researched the food recognition task so there are several deep-learning architectures that have been used to build the food recognition, each contributing to advancements in accuracy and efficiency. Convolutional Neural Networks (CNNs) this is a class of deep learning that is a specialized type of algorithm which is mainly used

in object recognition, image classification, image segmentation and detection. The CNN architecture is a hierarchical structure with 3 main types of layers Convolutional layer, the Pooling layer and the Fully connected layer. All 3 layers complete the architecture of CNN. Models Like GoogleLeNet, AlexNet and VGGNet have been majorly used for food recognition tasks of these papers Singla et al. (2016); Rahmat and Kutty (2021). Another architecture by Howard et al. (2017) using MobileNetV2, which is designed for mobile and embedded vision applications. MobileNetV2 is used for depthwise separable convolutions to reduce the number of parameters and computational costs while keeping high accuracy. This model MobileNetV2 is easy to use and requires less computational resources and this is suitable for real-time applications on mobile devices. According to *ML — Introduction to Transfer Learning* (2023) Transfer learning is a technique in deep learning where a model trained on one task is used as the starting point for a model on a second task.This is very useful when the 2nd task is the same as the 1st task, or when limited data is available for the second task. This can also help to prevent overfitting in the model. There are two types of transfer learning where first is feature extraction and the second is fine-tuning. In feature extraction, we freeze the convolutional layer of our architecture replace the dense layer with our dense layers and then retrain our model on our dataset that's called feature extraction. In fine-tuning, we freeze the convolutional layer of our architecture except for the last convolutional layer we can retrain the previous convolutional layer and dense layer on our dataset. The best way to apply the fine-tuning is by using pre-trained models on large datasets like ImageNet and fine-tuning them for specific tasks when we have a specific task to solve then we apply the fine-tuning. This approach influences the knowledge learned from a large collection of data, improving the accuracy of food recognition models while reducing the need for extensive training on task-specific datasets Tai et al. (2022), Kagaya et al. (2014)).

2.3 Existing Food Recognition Systems

Tai et al. (2022) worked on dish recognition using transfer learning and it shows the efficiency of pre-trained models in improving recognition accuracy and reducing training time, Similarly Kagaya et al. (2014) the authors also used Convolutional Neural Networks (CNNs) for food detection and recognition. Their approach includes training CNNs to distinguish between food and non-food items which shows the effectiveness of CNN architectures in capturing features of food images.

Singla et al. (2016) the authors used a pre-trained GoogleLeNet model for food/non-food image classification and food categorization. This model is known for its deep architecture and inception modules that give high accuracy in the detection of various types of food items. Yanai and Kawano (2015) the authors worked on food image recognition using deep convolutional networks with pre-training and fine-tuning. This approach shows the importance of fine-tuning pre-trained models to adapt to specific food datasets and resulting in improved performance.

Islam et al. (2018) and Howard et al. (2017) give an overview of deep learning techniques for food image classification. They discussed the application of various models, including CNNs and provided their success in recognizing complex food items.These authors introduced MobileNetV2 an easy and efficient CNN model designed for mobile and embedded applications. MobileNetV2 uses depthwise separable convolutions to reduce the number of parameters and computational cost while maintaining high accuracy and it is suitable for the task of real-time food recognition on mobile devices.

H. Lee et al. (2020) introduced RecipeGPT, a system for generating and evaluating cooking recipes using generative pre-training. This approach shows very diverse changes in the field of deep learning models in food-related tasks outside of image classification; it shows the ability of generative models in culinary applications. Louro et al. (2024) these authors explored recognizing food ingredients using machine learning techniques and showing challenges in accurately identifying ingredients from an image.

O. M.Salim et al. (2021) and team members presented a food recognition system using deep learning, their system architecture and training methodology are unique and impressive because they focused on dataset quality and preprocessing to achieve high accuracy. Zhenlin He and Yi (2022) and colleagues gave us a dish-recognizing system based on deep learning they discussed their system's design and performance in recognizing different dishes. Rahmat and Kutty (2021) author used AlexNet CNN and transfer learning for Malaysian Food recognition it demonstrates the effectiveness of transfer learning in adaption of pre-trained models to specific food datasets.

2.4 Datasets

In this section, we discuss the datasets that authors used to build a food recognition and recipe retrieval task Tai et al. (2022) this authors used a dataset of Vietnamese dish images named UEH-VDR this dataset contains 7848 images collected from multiple sources on the internet. All images are in RGB colour format with various sizes. This dataset was used to explore the Vietnamese culinary culture. Kagaya et al. (2014) This authors build a dataset of ordinary food images. The author used food-logging apps to build food detection and recognition tasks this app is used for the general public where this app has its food recording using both photos and text. The user takes a photo of a meal and puts the name of the food item. This author took 2 months to record data from food logging and the data is from everyday meals logged by the general public and it was 1,70,000 images. Singla et al. (2016) In this paper, the authors have created two image datasets which were Food-5K and Food-11 used for the experiments on food/Non-food classification and category of food. The first dataset is Food-5K which contains 2500 food images and 2500 non-food images and the second dataset is Food-11 which contains 16,643 images grouped into 11 categories that cover the various types of food that people consume in day-to-day life the 11 categories are: Bread, Dairy products, Dessert, Egg, Fried food, Meat, Noodles/Pasta, Rice, Seafood, Soup and Vegetable/Fruit. Yanai and Kawano (2015) In this paper, the authors used the Japanese food image datasets, which are UEC-Food 100 and UEC-Food 256. These datasets contain 100 classes of Japanese food images. Islam et al. (2018) In this Paper, the authors proposed a method that can classify food categories with a food image dataset. The author used the Food-11 dataset for his research, This data contains 16643 images that were grouped into 11 food categories: bread, dairy products, eggs, fried food, meat, pasta, rice, seafood, soup, and vegetables. Then the author further divided the dataset into training, validation, and testing for image preprocessing. H. Lee et al. (2020), the authors proposed a novel online recipe generation model, which is RecipeGPT. The system provides two methods of text generation. The first is instruction generation from a given recipe and ingredient generation from title and cooking instructions. The author used a generative pre-trained language model, which is GPT-2. The author fine-tuned the model on a large cooking recipe dataset. Louro et al. (2024), the authors presents a paper on the recognition of Food ingredients, The author used the Food 101 dataset to build a food ingredient

recognition model, and the author used the ResNet-50 convolutional neural network, which achieved 90% accuracy for the validation set and 97% accuracy in training so you can see that in this field, ResNet50 is the best-performing model, which gives better accuracy. Rahmat and Kutty (2021), This authors worked on Malaysian food recognition using Alexnet CNN and Transfer learning. The author presents 6 datasets and they have six different types of Malaysian food that have been collected. This includes three food types that have proteins, two food types that have carbohydrates and one food type that represents fibre. In this Malaysian food dataset, the author applied the AlexNet CNN model and achieved 91.43% accuracy. Ma et al. (2024), this authors have a novel approach to deep image-to-recipe translation, The author employs the food-101 dataset as the primary source of food images number of images 1,01,000 images and 101 food categories and also uses the Recipe dataset which is 1 million recipes in this dataset has each recipe entry contains ingredients cooking instructions and associated images. In this project author used two types of methods: ingredient prediction and the second is recipe generation for ingredient prediction author used the ResNet-50 (CNN) model and for recipe generation author used RNN with pre-trained Glove word embeddings to generate contextually correct recipe steps.

In the next section, we will talk about the CLIP model which is the Contrastive Language Image pretraining model by Open AI which I used for my task of dish image to recipe retrieval.

2.5 CLIP for Image-to-Recipe Retrieval

In current years, the field of computer vision and NLP which is natural language processing, has improved and introduced a new model like CLIP which is contrastive Language image pretraining by OpenAI. This CLIP model's ability to understand and associate visual and textual information has opened new possibilities in multimodal AI apps. The goal is to retrieve recipes based on images of food. This section of the literature review explores how CLIP has been useful in this domain and highlights studies and contributions.

2.5.1 CLIP Overview

CLIP represented by Radford et al. (2021), is a model designed to connect the visual images and text by learning from large datasets of image-text pairs. The model architecture consists of an image encoder and a text encoder, which are trained together using a contrastive loss function. Clip can generalize across various vision and language tasks without the need for task-specific tuning and that makes it suitable for applications such as image-to-recipe generation.

2.5.2 Image-to-recipe Generation: The Role of CLIP Model.

1. The base of image-to-recipe models: The task of generating a recipe from a food image requires understanding the visual content of the image and translating it into a recipe. This understanding and knowledge enable you to do this task easily and properly. The traditional way of doing this task involves extracting features from food images using CNN which we previously discussed is a convolutional neural network and then mapping these features to recipe ingredients using additonal

models. This approach can be limited by their dependence on task-specific training and feature engineering.

2. CLIP Contribution: For doing this task the contribution of CLIP is that it is flexible in handling both images and text which makes it a powerful tool for image-to-recipe. Clip can be adapted to this task by pairing visual inputs with textual recipe descriptions. Some researchers have explored various approaches for utilizing the CLIP model.

- Zero-Shot Recipe Generation: Radford et al. (2021) This author represents the method of zero-short capabilities. By Training the clip on a large dataset of image-text pairs, the model can generate textual information about recipes from images without the need for any additional training on a specific recipe dataset. This zero-shot learning approach is great and suitable for my task.

- Fine-Tuning for Recipe Generation- When CLIP's Zero-shot performance is outstanding, some studies have researched fine-tuning the model on food-specific datasets to enhance the performance of the model. I also used this method for fine-tuning in my task where I Fine-tuned the CNN MobileNetV2 architecture on the dataset of food which is Food-101 and Indian Food image dataset.

3 Methodology

This methodology section will start with understanding the dataset that I used to build my task because understanding the data is a crucial step for every task. So I took 2 datasets, the Indian Food Image dataset and the Food-101 dataset.

1. **Indian Food Image Dataset:** In this dataset, we have images of different traditional Indian dishes, which are high-resolution. This dataset includes a wide scope of food from various parts of India. We have 4000 Indian food pictures in 80 different categories here. All the images are manually labelled so that there will be ground truth for training and evaluation purposes to make a dish name Each image has been tagged by hand with its dish name to create ground truth for learning and testing. This dataset is about Indian cuisine, which incorporates a variety of regional and traditional cuisines originating from the Indian subcontinent. The dataset contains only 80 labels, consisting of different categories These are all categories I used when training my model on predicting the dish's name.

2. **Food 101 Dataset:** The most widely used food recognition benchmark is Food-101, which includes over 101,000 images belonging to 101 food classes. It was developed as an extension of the previous version. using the same data split as before, each category consists of exactly one thousand examples It has been arranged such that it can be easily loaded into TensorFlow for training purposes. Almost every image in this dataset can be found on several online platforms, with some being taken professionally and others being uploaded by users themselves.

3. **Combining Datasets:** The Indian Food images dataset and Food-101 dataset were combined to build dish images for recipe retrieval. I combined the food-101 dataset with Indian food images because I wanted some Indian dishes to be suitable

adhirasam	aloo_gobi	aloo_matar	aloo_methi
aloo_shimla_mirch	aloo_tikki	anarsa	ariselu
bandar_laddu	basundi	bhatura	bhindi_masala
biryani	boondi	butter_chicken	chak_hao_kheer
cham_cham	chana_masala	chapati	chhena_kheeri
chicken_razala	chicken_tikka	chicken_tikka_masala	chikki
daal_baati_churma	daal_puri	dal_makhani	dal_tadka
dharwad_pedha	doodhpak	double_ka_meetha	dum_aloo
gajar_ka_halwa	gavvalu	ghevar	gulab_jamun
imarti	jalebi	kachori	kadai_paneer
kadhi_pakoda	kajjikaya	kakinada_khaja	kalakand
karela_bharta	kofta	kuzhi_paniyaram	lassi
ledikeni	litti_chokha	lyangcha	maach_jhol
makki_di_roti_sarson_da_saag	malapua	misi_roti	misti_doi
modak	mysore_pak	naan	navrattan_korma
palak_paneer	paneer_butter_masala	phirni	pithe
poha	poornalu	pootharekulu	qubani_ka_meetha
rabri	ras_malai	rasgulla	sandesh
shankarpali	sheer_korma	sheera	shrikhand
sohan_halwa	sohan_papdi	sutar_feni	unni_appam

Table 1: List of Dishes in the Indian Food Image Dataset

for my task. After all, the main focus was on Indian food images. The combined dataset was carefully balanced to make sure that no single category dominated the training process.

3.1 Data Preprocessing

It is the methods and procedures applied to take raw data into a form that can be analysed or modelled. This would improve the data quality, increase the model performance and reduce computational complexity.

3.1.1 Image Preprocessing

Before we feed our image to the model, there are a few steps for processing the images.

1. **Image Resizing:** Image Resizing is valuable to ensure that all input images are compatible with the model, Each image was resized to a constant size of 224*224 pixels. This step is crucial, as it standardises the input dimensions and enables the model to process the image efficiently and consistently.

2. **Normalization:** In this second step, we normalized the pixel values to ensure that they are scaled-produced in input data is made available within the range [0, 1]. This step is crucial to speeding up the training part of your Neural Network This way, we make sure the data distribution has zero mean, which works better for stabilization of my training process.

3. **Data Augmentation:** This method is helpful when you have a smaller dataset of images in my case this data augmentation was implemented to boost the model's power. so that the model sees some unseen pictures in the training phases, the following is the approach that was used to perform data augmentation.

- **Rotation:** Rotated the images by a random degree of up to 20 degrees to make sure it help in viewing them from different angles.

- **Width and Height Shift:** Horizontal and vertical transformations of up to 20% of the image sizes were applied to simulate shifts in the camera's position.

- **Shear:** shear transformations up to 0.2 were applied so that it pretends the change in perspective of the image.

- **Zoom:** Random zoom of up to 20% is applied in an attempt at a distance from the camera.

- **Horizontal Flip**: Images were randomly flipped horizontally to account for the balance in food presentations and increase variability.

After data preprocessing, image data is ready to implement the model and make predictions. Model training is a crucial step that involves training the model on a specific dataset to improve the performance of the prediction.

3.2 Model Training

For the prediction of the Food, I applied the MobileNetV2 model, which is Pre-trained on the ImageNet dataset, It was used as the backbone for our food image classification task. The model architecture was adjusted to suit our specific dataset and the training of the model. Steps of Models:

1. **Base Model:** The MobileNetV2 Base model was loaded with pre-trained weights from ImageNet. The Top layers of the model which are specific to the ImageNet classification task were removed and custom layers for our food classification task.

   Here you can see the architecture of MobileNetV2:

2. **Custom Layer Addition:** For the model training we didn't use the top layer it has been false which means we did not use the top layers of the model and then make our architecture at the end of the model we used the weights of the imagenet dataset and then we add our architecture first define the Sequential to add the layers in the model. First, add the base layer of the mobilenetv2 model and then add some layers to the model.

   - **Global Average Pooling Layer:** This layer was added to reduce the spatial dimensions of the feature maps the purpose of this layer is to replace the fully connected layers that are typically at the end of the CNN. This layer gives benefits to reduce overfitting by minimizing the number of parameters in the model.

- **Dense Layer:** A fully connected layer with 512 units and a RELU activation function was added to learn complex representations from the features of the model.
- **Output Layer:** The final layer was a dense layer with units equal to the number of food categories in our dataset which means 180. This layer used the softmax activation function to output a probability distribution over the classes.

3.3 Model Compiling

The model was compiled using the Adam optimizer, this optimizer is known for efficiency and low memory requirements. The loss function used was sparse categorical cross-entropy, suitable for multiclass classification problems.

3.4 Training Process

The model was split into training, validation and testing this splitting will help me to train my model to get better accuracy and also help us prevent overfitting. The model was trained for an initial 10 epochs using the training dataset with data augmentation techniques applied to improve the model performance and the validation dataset was used to monitor the model performance and prevent overfitting.

Additionally, I extended the training process to further improve the model performance in this extended version the model was trained for 20 epochs and added the dropout layer with a 0.3 rate after the dense layer. This dropout layer helped to prevent overfitting and randomly disabling the fraction of the neurons during each training process. In these 20 epochs, I used the learning rate which is 0.001 and this helped me to minimise the loss function. In addition, I changed the dense layer to a fully connected layer with 1024 units and a RELU activation function.

3.5 CLIP Model

The CLIP which is the Contrastive Language-Image pre-training model, which developed by Open AI. In this CLIP model, there is the ability to perform zero-shot classification by learning visual concepts from natural language supervision. In this clip model, there is no need for additional training on your specific dataset. First, the clip model (openai/clip-vit-base-patch32) and its processor were loaded using the Hugging Face Transformers library. The processor was necessary for preprocessing both images and text inputs and making sure it would be compatible with the clip model. The model was configured to run on a GPU if the GPU is available because GPU allows us to enhance computational efficiency. The combined dataset of Indian food images and Food101 was split into training and test datasets while keeping a balanced class distribution. The test set was 10% of the overall dataset and the remaining 90% was used for training. After splitting the dataset Images were preprocessed to match the input requirements of the clip model, Preprocessing of images includes resizing and normalization. After this preprocessing label mapping was done a list of food class labels which is 180 was extracted from

the dataset and each label was mapped to a unique index. After this label mapping image inputs were processed through the model to generate image embeddings and furthermore text inputs were processed into text embeddings using the model processor. After this process, each image of image embedding was compared with the text embedding of the class label. After this process model was evaluated on the test set for prediction and to measure the accuracy of the model. True labels and predicted labels were collected for all test samples and the confusion matrix and classification report were developed to analyze the model performance. To measure the accuracy of the model author used a confusion matrix, classification report and Top-5 accuracy to see the model performance. I want to say one thing once the model predicted the dish, the ingredients were retrieved using the Edaman API. The API fetched ingredient data based upon on the predicted dish name. After predicting the dish name of the image the model gives the ingredients and the potential allergens so the user can see what allergens are present in the dish.

In the comparison of MobileNetV2 and CLIP, I would like to say that we will see the evaluation section to analyze which performs best in this task of dish image to recipe retrieval.

4 Design Specification

System architecture and frameworks for the food image-to-recipe prediction system are to be described in this section. To achieve this, MobileNetV2 and the CLIP model were used by the author so that both models could be compatible with this task and work fluently. Now let's go to see the techniques and architecture used for building it.

- MobileNetV2 is a lightweight deep-learning model meant for mobile and embedded vision applications. By using depthwise convolution, this architecture reduces the number of parameters and computational complexity while still maintaining the accuracy of the model.
- Techniques: The technique that was used by the author to finish this was utilizing transfer learning, where the author employed a pre-trained model (MobileNetV2) on the ImageNet dataset that enabled me to reduce training time as well as enhance our food recognition task model's performance and another technique were used which is data augmentation in this we were applied rotation, width shift, height shift, shear, zoom, and horizontal flip to our dataset while training. TensorFlow and Keras libraries were used for model implementation.
- CLIP Model Architecture: CLIP is a Contrastive Language-Image pre-training which is a multimodal network in deep learning this model learns the images and text data. It is the concept of vision Transformer and text transformer that enable it to understand and classify images based on textual descriptions. The CLIP model has Zero-short learning where we do not require training on specific labelled datasets. This method allows us to implement this model on new tasks without any fine-tuning. Also, the model generates embedding for both images and textual labels and then classification is performed by seeing

the similarity between these embeddings. PyTorch and Transformers both libraries were used for model implementation and Edamam API for retrieving the ingredients present in the predicted dish.

4.1 System Architecture Diagram

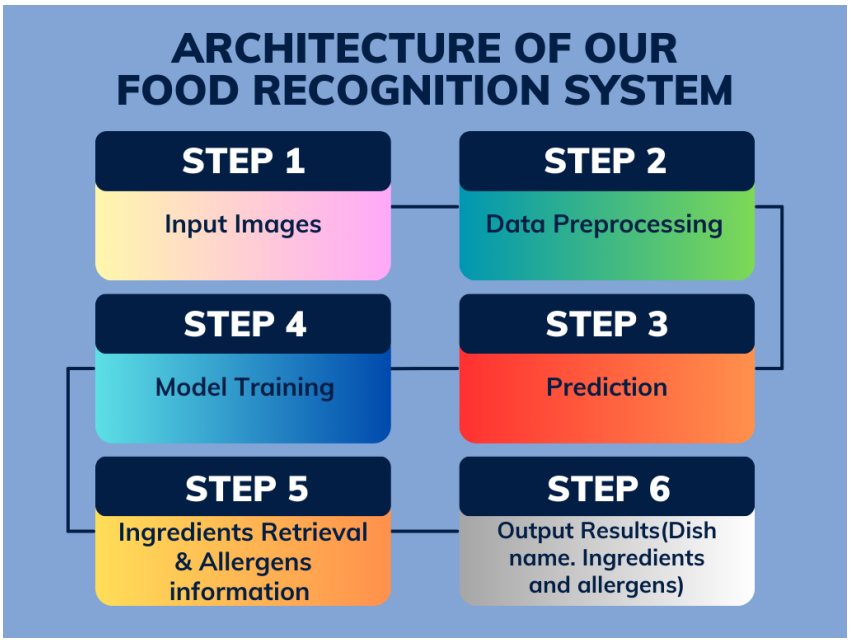


Figure 1: Architecture of our food recognition system

The diagram shows the architecture of our food recognition system, We aim to accurately classify food images into their labels and provide ingredients of the predicted dish and allergen information that are present in the ingredients.

(a) **Input:** This input is in the form of food images in various formats (JPEG, PNG, JPG), a user will upload the image of food because user wants the information related to their food.

(b) **Processes:** After the user uploads the image in our system, the system will start the process for food recognition. First is the data preprocessing such as resizing and normalizing the image. The next step is that the image will go into the model for training purposes then the prediction will done by the given input image, and the system will predict the dish name. The next step is using the predicted dish name system will fetch the ingredients and identify potential allergens present in the dish.

(c) **Output:** As an output user will get three things first the predicted dish name, the list of ingredients and potential allergens.

5 Implementation

For the implementation of the food recognition system, the author developed the User Interface by using Streamlit, which is an open-source app framework that helps to create a visual of your task. The Streamlit was chosen for its simplicity and ease of use.

Here is my Streamlit UI:

(a) **Image Upload:** Users can easily upload an image of a dish through the Streamlit interface. The upload widget allows users to select and upload an image from their local drive.

(b) **Dish Prediction:** After uploading the image as input the system processes the image using the trained model to predict the dish name. In this prediction system, the author used MobileNetV2 and CLIP models to predict the dish name.

(c) **Ingredient Retrieval and Allergen Information:** Once the dish is identified, the system uses the Edamam API to fetch information about dish-like ingredients and the system checks for potential allergens present in the ingredients this will help the user to see the allergens that are present in the dish.

(d) **Result:** The system will give the output to the user which is the predicted dish name, ingredients and allergen information. This visual output helps the user to enhance their experience in the field of food computing.

The benefits of using this Streamlit UI are that it is easy to use and allows us to create an interactive web app with minimal coding knowledge. Streamlit supports real-time interactivity which helps the user to see immediate results also it is compatible with Python allows us to directly use it to make web applications for machine learning and deep learning models streamlit applications can easily deployed and shared with others, So this is very user-friendly.

6 Evaluation

In this evaluation section, the food recognition system was conducted using a test set to ensure the accuracy and reliability of the models. The evaluation metrics included test accuracy, Top-K accuracy, Confusion Matrix and Classification report to analyze the model's performance. First start with the model training, the author trained the model which is mobilenetv2 where the first case was that the model was trained on initially 10 Epochs. We can see this Figure 2 graph of model accuracy and model loss where at epoch 1, the model starts with a training accuracy of 35.74% and a validation accuracy of 46.32% these values show the baseline for the model's performance and also see the loss of model at 1 epoch is 2.7524. Now analysing the graph the model shows a noticeable improvement in accuracy by the second epoch training accuracy was improved to 44.89% and the validation accuracy increased to 49.81%. These rapid changes in the model show that the model quickly

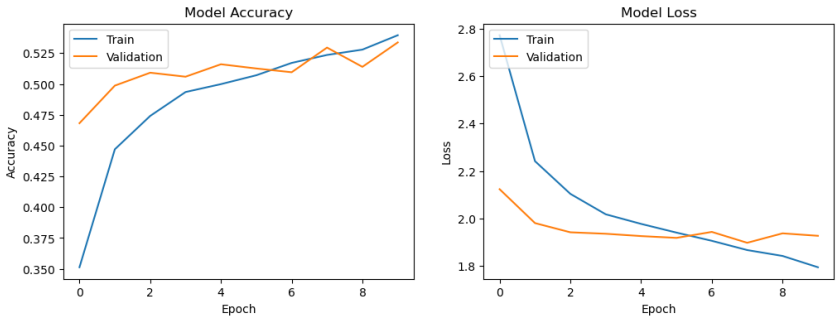


Figure 2: Model Accuracy and Model Loss of 10 epochs

learns and adapts the dataset. As training accuracy progresses, the learning curve shows an upward trend by epoch 5, where the training accuracy reaches 50.17% and validation accuracy of 50.86% In the later epochs, the accuracy increases and reaches to 53.55% and validation 53.20% also the loss trend decreases from 2.7524 to 1.8131 that indicates the model is minimizing errors. The graph suggests that MobileNetV2 has reached its peak performance with the given dataset and the given parameters.

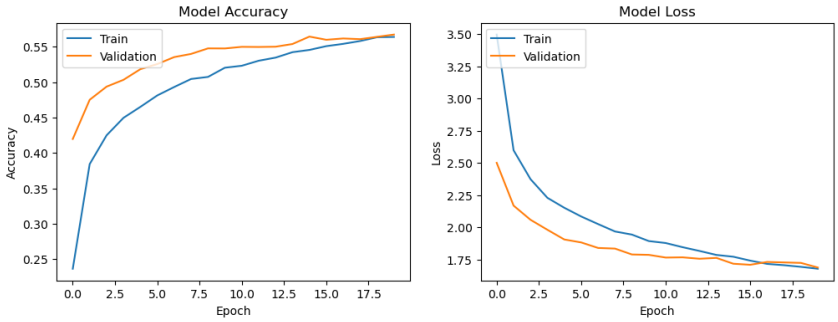


Figure 3: Model Accuracy and Model Loss of 20 epochs

Model accuracy and model loss of 20 epochs are shown in the figure 3 graph. Though this trend continues, it becomes a lot more gradual after the tenth epoch. The training accuracy at epoch 20 is 56.39% with the validation accuracy at 56.73% and the loss curve keeps decreasing from 3.4975 to 1.6787 meaning that the model is reducing mistakes. Once training was complete, the author ran a test set on his model where he obtained an overall accuracy of 52.30%. Concerning my case, I had a multi-class classification problem which consisted of around one hundred eighty classes thus it will be harder to achieve good results compared to binary classification.

After implementing MobileNetv2, the author applied another model to increase the performance of the task, which is CLIP, This model was evaluated on a test set containing 10,494 images of various food categories.

To see figure 4 the overall performance of the model so used evaluation metrics to analyze the model performance. Top-5 Accuracy- 91% Overall Accuracy- 71% The top-5 accuracy of 91% shows that the 91% of cases the correct label was among

Top-5 accuracy on test set: 0.91
Classification Report:

	precision	recall	f1-score	support
adhirasam	1.00	0.33	0.50	3
aloo_gobi	0.00	0.00	0.00	7
aloo_matar	0.00	0.00	0.00	4
aloo_methi	0.06	0.60	0.12	5
aloo_shimla_mirch	0.00	0.00	0.00	8
aloo_tikki	0.00	0.00	0.00	3
anarsa	0.00	0.00	0.00	5
apple_pie	0.67	0.52	0.59	102
ariselu	0.00	0.00	0.00	6
baby_back_ribs	0.82	0.67	0.74	105
baklava	0.91	0.63	0.75	98
bandar_laddu	0.11	0.50	0.17	4
basundi	0.11	0.40	0.17	5
beef_carpaccio	0.59	0.86	0.70	102
beef_tartare	0.61	0.78	0.68	101
beet_salad	0.73	0.73	0.73	102
beignets	0.74	0.90	0.81	87
bhatura	0.43	0.75	0.55	4
bhindi_masala	0.45	0.83	0.59	6
bibimbap	0.92	0.94	0.93	103
biryani	0.38	0.60	0.46	5
...				
accuracy			0.71	10494
macro avg	0.51	0.54	0.48	10494
weighted avg	0.77	0.71	0.71	10494

Figure 4: Classification Report

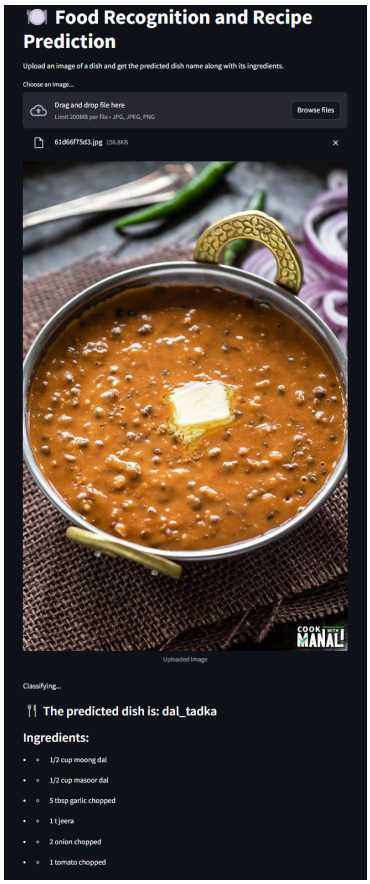
the top-5 predictions made by the model. This shows the strong model's capability to capture relevant features. The overall accuracy of 71% that the model's top prediction matches the true label 71% of the time. The classification report shows the view of model performance in the individual categories. Key metrics are precision, recall and F1-Score. Precision- The ratio of true positive prediction among all positive predictions made. Recall- The ratio of true positives among all actual positives. F1-Score- The mean of precision and recall providing a single metric to balance both parts. Support- The number of samples in each class. Key insights of the CLIP model: High-performing Categories: Adhirasam, Caesar_salad, Bibimbap, Chicken wing, Cannoli, edamame, waffles, takoyaki, tacos, sushi, spaghetti_carbonara, Sandesh, samosa, risotto, ramen, pho, pancakes, oysters, omelette, mussels, malapua, macarons, lasagna, hot_dog, hamburger, gyoza, guacamole, falafel, deviled_eggs, churros. These all are the food categories where the clip model predicts high precision which means that the model accurately predicted this dish in the range of 90-100%. Some categories of Indian foods predict low performance which means that the precision is low which indicates failure in identifying these dishes Aloo Gobi, Aloo Matar, Aloo Tikki, Anarsa, Ariselu, Bandar Laddu and some Indian sweets. The low performance in these categories is because of insufficient training data. The other way of evaluating the model is through macro-average. It is also a mixture of precision is 0.51, recall is 0.54 and F1-score is 0.48. These statistics were used to assess the performance of unweighted mean across all categories. Conversely, weighted average precision (0.77), recall (0.71), and F1-score were used to evaluate the mean performance per category for sample size, with more preference being awarded to categories with many samples in them. The resulting measure indicates better performance due to valid outcomes in categories having high numbers of samples.

6.1 Discussion

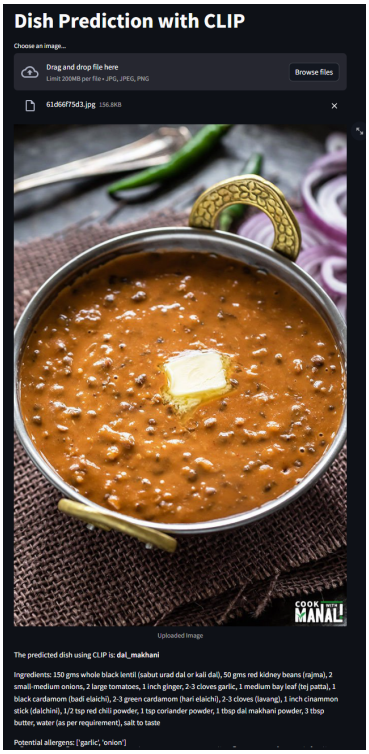
In this study, we compared the performance of two deep learning models,CLIP and MobileNetv2, on multi-class classification task.Where we have image of food and the model predict the dish name, ingredients and allergens information so for this task CLIP model performs well and got the accuracy of 71% so overall for tasks the clip is superior model and gives highest classification accuracy.

6.2 Case Study 1: Accurate Prediction and Ingredient Retrieval

In this case study we can see our prediction of food recognition and ingredients. Here we took a photo of food which is Dal Makhani, a popular and rich north Indian dish known for its creamy texture and spiced lentils. so our primary objective is to see wheather our model can correctly classify the food item and provide ingredients prediction based on the image.



(a) Prediction of MobileNetV2 Model



(b) Prediction of CLIP Model

Figure 5: Model Predictions

See this figure 5(a) MobileNetV2 is predict Dal tadka instead of Dal makhani this is not accurate but yes this is category of Dal.

See this figure 5(b) it shows that the CLIP is performing well and gives accurate prediction of Dal Makhani.

6.3 Case Study 2 Misclassification

In this case study 2 we can see the misclassification between two models. we can figure out which model will perform best and gives accurate prediction

Figure 6: Prediction of MobileNetV2 Model

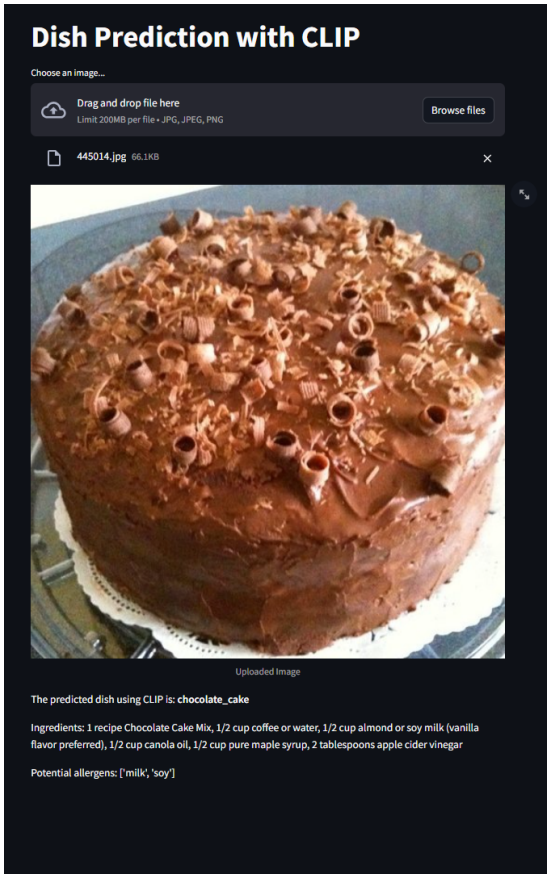


Figure 7: Prediction of CLIP Model

Here we can clearly see this figure 6 that the MobileNetv2 is predict apple_pie instead of chocolate cake, which means there is misclassification on the MobileNetv2 model. Next, we will see that if the CLIP model will predict this correct or not. Now see this figure 7 the CLIP model perform best and shows the prediction of

Chocolate cake. Because the CLIP model has zero-short learning and the accuracy I got on test set is 71%.

7 Conclusions & Future Works

This research expected to examine the effictiveness of CLIP models for enhancing the recognition of food ingredients and potential allergens by developing dish recognition and recipe retrieval. The objective was to evaluate CLIP model to improve their performance compared to the MobileNetV2 model. we trained and tested the both models on food classification task and compare the performances. The main focus was on their ability to accurately identify the dish name, ingredients and allergen information. The performance of the model were mesaured using these metrics like accuracy, top-5 accuracy, precision, recall, and F1-score. This study successfully showed that the clip model offers a better performance than MobileNetV2 for recognising various food categories.

This research is useful in various domains, including accurate food ingredient recognition and allergen detection. One application can be useful in the development of personalised dietary assistance tools for people who maintain dietary restrictions. These types of technologies will help to develop mobile apps, smart kitchen appliances, and health management platforms, which could help users enhance their experience in the food industry. This research contributes to enhancing food safety, dietary management, and personalised allergen solutions for both consumers and commercials.

In the future, this research will work to enhance the user experience of the application by adding some advanced features that aim to personalise recipe customisation based on present allergens in dishes. One approach is to develop an intelligent system that not only recognises food ingredients using the CLIP model but also allows the user to tailor their recipes according to their specific allergens. Using this information, the system would recommend alternative ingredients and updates to recipe. so overall enhancing the user experience for personalisation of recipes. It could be developed in the future to transform the dish image-to-recipe assistant into a powerful tool for people who maintain their dietary restrictions, promote healthy eating habits and enhance the enjoyable culinary experience of tailored recipes.

References

H. Lee, H., Shu, K., Achananuparp, P., Prasetyo, P. K., Liu, Y., Lim, E.-P. and Varshney, L. R. (2020). Recipegpt: Generative pre-training based cooking recipe generation and evaluation system, *Companion Proceedings of the Web Conference 2020*, WWW '20, ACM.
URL: *http://dx.doi.org/10.1145/3366424.3383536*

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.
URL: *https://arxiv.org/abs/1704.04861*

Islam, M. T., Karim Siddique, B. N., Rahman, S. and Jabid, T. (2018). Image recognition with deep learning, *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Vol. 3, pp. 106–110.

Kagaya, H., Aizawa, K. and Ogawa, M. (2014). Food detection and recognition using convolutional neural network, pp. 1085–1088.

Louro, J., Fidalgo, F. and Oliveira, (2024). Recognition of food ingredients—dataset analysis, *Applied Sciences* **14**: 5448.

Ma, J., Mawji, B. and Williams, F. (2024). Deep image-to-recipe translation.
URL: *https://arxiv.org/abs/2407.00911*

ML — Introduction to Transfer Learning (2023).
URL: *https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/*

O. M.Salim, N., Zeebaree, S., M.Sadeeq, M., Radie, A., Shukur, H. and Najat, Z. (2021). Study for food recognition system using deep learning, *Journal of Physics: Conference Series* **1963**: 012014.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I. (2021). Learning transferable visual models from natural language supervision.
URL: *https://arxiv.org/abs/2103.00020*

Rahmat, R. A. and Kutty, S. B. (2021). Malaysian food recognition using alexnet cnn and transfer learning, *2021 IEEE 11th IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pp. 59–64.

Singla, A., Yuan, L. and Ebrahimi, T. (2016). Food/non-food image classification and food categorization using pre-trained googlenet model, pp. 3–11.

Tai, T. T., Thanh, D. N. H. and Hung, N. Q. (2022). A dish recognition framework using transfer learning, *IEEE Access* **10**: 7793–7799.

Yanai, K. and Kawano, Y. (2015). Food image recognition using deep convolutional network with pre-training and fine-tuning, pp. 1–6.

Zhenlin He, Zixuan Zhang, G. F. Z. Y. L. Y. and Yi, Z. (2022). Dishes recognition system based on deep learning, *Academic Journal of Computing Information Science* **5**: 48–53.