Comparative Analysis and Implementation of a Food Recipe Generation Model

Submitted April 2023, in partial fulfillment of
the conditions for the award of the degree **BSc Computer Science.**

20215622

Supervised by Dr Andrew French

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature __________L.Y.C.__________

Date ___24___ / ___04___ / ___2023___

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.

Abstract

According to a Meta report [4], 52% of people use their platforms to acquire their knowledge of delicious food and drinks, while more than 300 million Instagrammers share and view food-related content each month. However, accesses to food recipes are still limited to cooking websites or related tutorials.

This paper presents the development and implementation of different food recognition methods, aimed at enriching the cooking information available behind the vast number of food photos on the Internet. The method is divided into two processes: model analysis and practical application. The model analysis process is detailed in this paper, which discusses four food image recognition models, including traditional indexing methods and their optimised approaches. As a foundation of the project, Recipe 1M+ [18] is used as a dataset for the traditional method and Inverse cooking [27] provided a brand new approach to recipe generation. Addressed the flaws left by these two models, I continuously came up with new ideas and adjusted the model's logic while a comparative analysis is conducted to evaluate the evolution of essential logic chains and the effectiveness of newly proposed ideas. The practical application process involves the implementation of the optimized food recognition method in a software system with a user-friendly interface, allowing users to quickly search for cooking information based on uploaded food images. The system has already been tested with various food images, and the results demonstrate its accuracy and usability in most cases. This work contributes to the development of optimised food recognition methods and provides a practical solution for users to learn recipes from photos more conveniently.

Keywords

Deep Learning, food recipes, image recognition, model analysis, software system, accuracy

Acknowledgements

I would like to express my sincere gratitude to the developers of the Recipe1M+ dataset for providing valuable data for my project. I would also like to extend my deepest appreciation to my supervisor, Dr Andrew French, for his invaluable guidance and encouragement throughout the project. Without his persistent help and support, I would not have been able to make such significant progress. I feel fortunate to have him as my supervisor, as he consistently ensured that I fully understood the subject matter and provided suggestions to inspire me on this journey. Additionally, I would like to thank my parents for their unwavering support and belief in me. Their encouragement has been instrumental in helping me to achieve my goals.

Contents

Chapter 1

Introduction

Food is an essential aspect of human life that not only impacts our physical health, but also plays a significant role in defining our identity, social status, and culture [14], As the famous French gourmet Brillat-Savarin said, "Tell me what you eat, and I will tell you who you are." As the world becomes more health-conscious, people are looking for more nutritious food options, which has led to an increased focus on food-related research, particularly in the area of food choice.

1.1 Motivation

The popularity of sharing food on social media has led to an increased interest in the underlying recipes for food products. Additionally, people often face the challenge of how to cook different kinds of dishes every day using staple ingredients. An application that generates a set of recipes when users upload pictures of ingredients or ready-made food can satisfy both requirements.

The most common cases are that when people find something delicious, they will take a picture and manage to cook a similar one at home. However, recognition of recipes from an image is a challenging task due to the variability in food appearance caused by various factors such as lighting, background, and serving style. Moreover, there is a vast amount of variability in cooking practices across cultures and regions, making it even more difficult to design an effective recipe recognition system. Fortunately, recent advances in computer vision and deep learning techniques have shown promising results in recognizing food images and identifying the constituent ingredients.

1.2 Aims and Objectives

The software should be able to solve the problem that people usually have no idea about recipes even if they have the ingredients. According to the software's design, users can choose desired recipes freely when using this application and when they see something delicious, they can go home and learn how to make it through photos. The application will generate appreciated ingredients and recipes based on what food is in the image. The main aim of the project is to build a system capable of accurately identifying ingredients and recipes from an uploaded food image. The output of the algorithm could be used to design a proper health-focused meal plan, as well as calculate the total nutritional intake based on the recipe.

The key objectives of the project are divided into two stages. The first one is to investigate and evaluate the performance of different food recipe recognition models. After determining which one is optimised, a cross-platform application with a user-friendly interface and attractive guidance will be designed as an implementation of the model. In the final stage of the project, I plan to make a user satisfaction survey to gather more ideas for software improvement.

Chapter 2

Background and Related Work

Numerous studies have been conducted on various aspects of food, including diet preferences [22], nutrition analysis [29], eating behaviour assumptions [24], food production safety [7], and food culture [11]. However, as food-related research covers several scientific areas, it often lacks a systematic set of investigations [19]. In 2019, Min and colleagues[19] proposed an advanced framework for food calculation, which includes goals related to sensation, recognition, detection, recommendation, medicine, biology, agriculture, food industry, nutritional health, and other fields [19]. One of the essential tasks in this framework is food image recognition.

As a subfield of fine-grained image classification (FGIC) [12, 25, 34, 35], food image recognition has significant research value. With the proliferation of mobile devices such as phones and cameras, as well as wearables such as the DJI Action 2 Power Combo [3], food image recognition has become a promising application of artificial intelligence technology. For example, the Amazon Fresh Go Store [32] utilizes food recognition technology to enable automatic payment and settlement, and the system also analyzes food nutrients and assesses user eating habits to ensure daily healthy intake. Additionally, FoodAI [26], a new food image recognition technology based on deep learning for smart food logging, has been deployed as an API service and is one of the components powering Healthy 365 [2], a mobile app developed by Singapore's Health Promotion Board. Moreover, food image recognition has the potential to provide food recommendations and indexing in social networks.

Food image recognition involves categorizing food in a given image or indexing different ingredients and cuisines. This technology has a history that can be traced back to 1977 when Parrish et al. conducted a visual-based fruit and vegetable identification study [23]. Kitamura's team developed a food logging system to provide diet suggestions based on ingredient and calorie analysis [15]. In 2014, Bossard et al. released the first large-scale Western cuisine image dataset "Food-101" and used deep learning for food image recognition [5]. With the rapid development of deep learning technologies and the increase of extensive food image datasets, related research has proliferated. The history of food image processing inspires the idea of combining food category recognition and nutrition analysis and implementing them in a practical application.

The original project was initially undertaken by a research group at the Universitat Politecnica de Catalunya, Massachusetts Institute of Technology and Qatar Computing Research Institute. It provides the largest dataset named Recipe1M+ [18] for the traditional indexing method. The dataset is a new large-scale, structured corpus of over one million cooking recipes and 13 million food images. As the largest publicly available collection of recipe data, Recipe1M+ affords the ability to train high-capacity models on aligned, multimodal data.

Generating recipes from images requires a deep understanding of both the constituent ingredients and the process of preparation (e.g., slicing, mixing with other ingredients, etc.). Traditional methods treat this problem as a retrieval task, retrieving recipes from a fixed dataset based on a similarity calculation between the input images and the dataset images [28]. Obviously, the traditional approach will fail in cases where the dataset is missing a certain food preparation method. In fact, the model will return a randomly selected value from several groups with lower correlation, which decreases the accuracy.

In the report, I provide an advanced method to overcome this data limitation, based on the project of Inverse cooking [27]. It is to treat the picture-to-recipe problem as a conditional generation task. Rather than getting the recipe directly from the picture, it would be better to first predict the ingredients of the food and then generate a food preparation method based on the image and the ingredients. Although it might lead to the error caused by multi-layer operations, this would allow for some additional information to be obtained using the intermediate process of images and ingredients, which makes the prediction more explainable and accurate.

Chapter 3

Methodology

There are several methods for food image recognition, including traditional computer vision methods and deep learning-based methods. The traditional methods utilise visual features, such as colour, texture, and shape to classify food images. These methods have limitations in accuracy due to the complexity of food images, and the difficulty in pre-setting optimal features for each food class. While people are seeking food image recognition methods with multiplex functionality and accuracy, deep learning-based methods have shown significant improvement in these fields.

Moreover, as mentioned in the related work section, the deep learning model can be implemented in two different ways. The most common one is the Image-to-Recipe Model. This model takes a food image as input and directly generates recipes as output. This approach requires a quite large dataset of food images and corresponding recipes for training and testing. The models are only able to learn to analyse the relationships between dishes images and recipes and then link them together in the dataset. According to Dr. Mane's analysis [17], the advantage of this approach is its high efficiency, as it can directly link the food to the recipe which has a highly similar food image. However, it requires a large amount of time and resources to train the model with a vast image dataset.

Another kind of food image recognition model can be called the Ingredient Detection Model [21]. This type of model uses deep learning to detect ingredients in food images and then searches for recipes that match the detected ingredients or generates recipes based on them. This method reduces the required database capacity and increases the accuracy of recipe organization. This approach has advantages in that it only needs to classify whether the ingredients exist in the image, which reduces the amount of required data and computation resources.

3.1 Image to Recipe

3.1.1 Similar Image Search

The most common approach for food image recognition is to calculate the similarity between the uploaded image and pre-stored images in the database. However, the accuracy of the model depends a lot on the quantity and quality of the dataset used for training the model. Therefore, the original project of Recipe 1M+ was started up by a research group at the Universitat Politecnica de Catalunya, Massachusetts Institute of Technology and Qatar Computing Research Institute. It divides the recipe into two major components: its ingredients and cooking instructions. The pre-trained embedding vector obtained by the word2vec algorithm was used with a bi-directional LSTM (since the ingredient list is an unordered set, a bi-directional LSTM model was chosen, which considers both forward and inverse ordering) [33], where the LSTM performs logistic regression on each word in the ingredient text. At the same time, a two-stage LSTM model is designed to encode the sequence of the sequences since the simple model can not contain a too-long sequence of instructions as a whole. Each cooking step is first represented as a vector, and then an LSTM is trained with a sequence of these vectors to obtain a vector characterising all the steps. For the food image representation, the im2recipe model uses two deep convolutional networks, VGG-16 and Resnet-50, removing the last 'softmax' classification layer and connecting the rest to a joint embedding model.

As shown in Figure 3.1 below, the recipe model consists of two encoders: one for the ingredients and the other for the cooking instructions. The outputs connecting the two encoders are embedded in a shared space of recipes-images. The image representations are also mapped into the same space by a simple linear transformation.

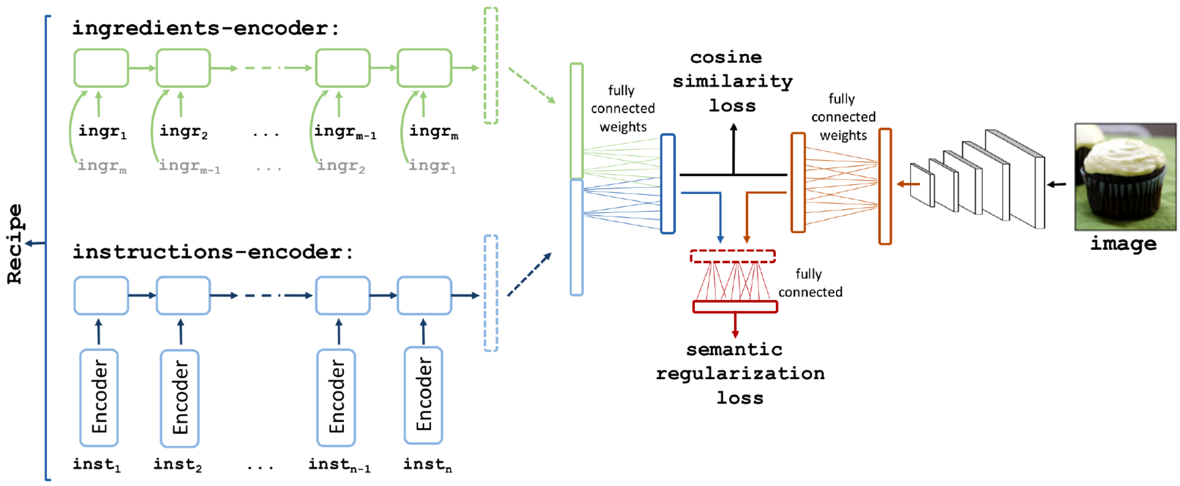


Figure 3.1: Similar Image Search Method Layers

The effectiveness of this model depends a lot on the storage capacity of the database and the computing power of the computer [18]. Therefore in this work, I can address data limitations by introducing the large-scale Recipe1M dataset which contains one million structured cooking recipes and their images. Using these data, I am able to train a neural network to learn a joint embedding of recipes and images that can be used as a classification model.

3.1.2 Improved Similar Image Search

The main limitation of the previous model's performance is that, when searching for the corresponding recipes of an image, the model has to look through all images in the database, calculating the value of similarity or possibility. It greatly influences the speed of which the model returns a result. To overcome the limitation, I come up with this idea.

The improved model is similarly divided into two main processes: data import and recipe retrieval. In contrast, the essential improvement of this method is to pre-process the original dataset before the recipe data is loaded into the model. The original data will be stored in a ". json" file, and the Word2Vec algorithm [10] will be used to gather the pre-trained vectors containing ingredients and cooking instructions from it. The model will take these parameters as input, cooking instructions, the number of steps, ingredients and the number of them. Then, these vectors will be loaded into the database corresponding to their recipe IDs.

In this case, the image-to-recipe model will first convert the physical image into a vector, and it can utilise Cosine Similarity [16] or Euclidean Distance [31] methods to search for similar recipe vectors and their ID. According to the id, the model obtains the recipe's details, such as name, ingredients and instructions.

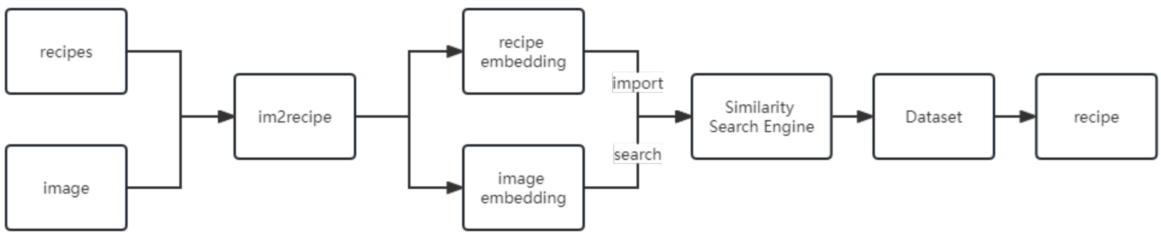


Figure 3.2: Improved Similar Image Search Method

It is believed that searching for the pixel values in an image is more complex than an ID [13], so it directly compares the similarity of images rather than vectors. Not to mention the faster processing speed and the lower memory usage, the vector representations reduce the dimensionality of the data and image noise. Moreover, the model is still on paper, but it can be further improved by implementing more functional similarity comparison algorithms and more advanced deep learning algorithms.

3.2 Image to Ingredient to Recipe

3.2.1 Individual Attempts

Although there are a lot of ways to raise the speed performance of the food recognition models simply, the limitation of the database's capacity and the precision of the food image remains to be guaranteed. One way to overcome the issue is to treat the picture-to-recipe problem as a conditional generation task. Rather than getting the recipe directly from the picture, it is likely better to first predict the ingredients of the food and then generate a food preparation method based on the image and the ingredients. This would allow some additional information to be obtained using the intermediate process of images and ingredients [27].

This model was designed by myself and based on the idea of recognising the ingredients and then generating a recipe. As the plan, I firstly would learn to use the popular algorithm, transformers to develop image classification to locate each food on a plate and assign labels, which is taught in the "Hugging face" [9]. The next step is using the NLTK module as a natural language processor to associate each ingredient together as a completed recipe.

In the evaluation part 5.1, this classification model works not well because there are quite large scales of recipes and cuisines in the collected data and most of the ingredients have less difference between each other than the distinction between an apple and an egg.

3.2.2 Inverse Cooking

Based on my design, I found a feasible approach for this. According to the definition by Dr. Salvador teams, inverse cooking [27] is an approach to recognise the food images and generate recipes, in contrast to the traditional approach, where a recipe is retrieved from a database based on the similarity between the input image and the stored images. The process first identifies the ingredients in a food image and then generates a recipe based on those ingredients. Inverse cooking reduces the required database capacity and improves the accuracy of recipe organization. It starts by classifying whether the ingredients exist in the food image, rather than trying to identify the specific dish or recipe directly. This method has been shown to have a higher success rate than traditional search methods in user experiments.

The model in Figure 3.3 consists of two main components. Firstly, the researchers pretrain an image encoder and an ingredients decoder to extract the visual features of the input image to predict the ingredients. An ingredient encoder and an instruction decoder are then trained to generate food names and cooking procedures based on the image features of the input image and the predicted ingredients. As the structure of the model is shown below, the input to the model is a picture of the food, the output is a sequence of cooking methods, and the intermediate step is to generate a list of ingredients based on the image.

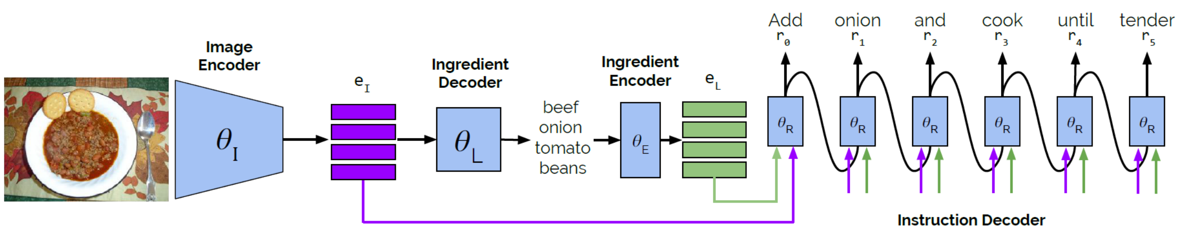


Figure 3.3: Recipe generation model

During the text generating in Figure3.4, the RNN [8] is completely abandoned but uses the powerful Transformer [6] instead. Each Transformer block consists of two attention layers and a linear layer. In order to combine image and ingredient information for cooking process generation and to achieve multimodal attention, the attention layer has been changed: the first attention layer performs self-attention on the previous output, which is consistent with the original Transformer. The second attention layer is changed to conditional attention in order to fine-grained extraction of the first layer of self-attention. This level of attention is subject to two constraints: image features and ingredient embeddings. In order to perform multimodal fusion simultaneously, three fusion structures were tried in the paper [27], which are shown in the diagram below as (b), (c), (d). The final experiment found that the Concatenated approach worked best.

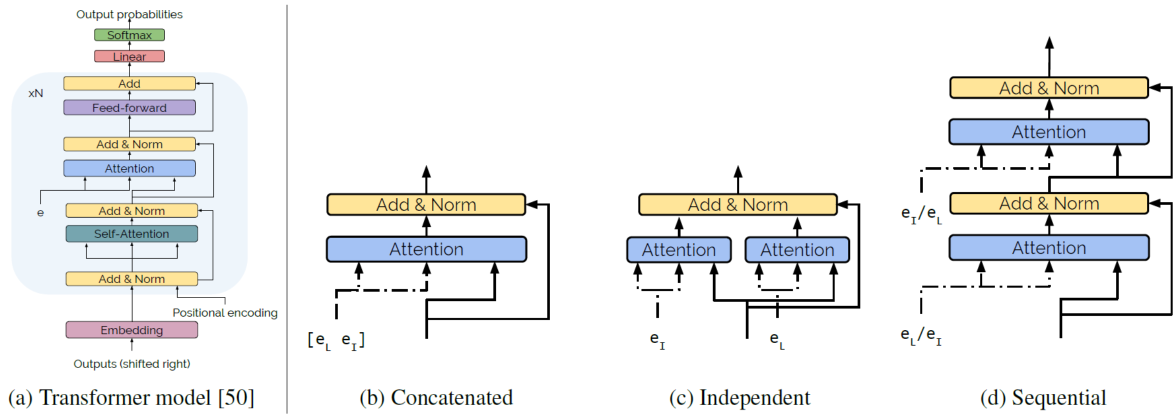


Figure 3.4: **Attention strategies for the instruction decoder.** The attention module in the transformer (a) is replaced, with three different attention modules (b-d) for cooking instruction generation using multiple conditions.

3.2.3 Improving Inverse Cooking

There are some more advanced methods for improving the inverse cooking model. For example, learning cross-modal embeddings with adversarial networks [30]. The model is purposed to use adversarial training to enforce cross-modal similarity in the learned embedding space. The objective is to enable efficient retrieval of recipes based on food images and vice versa.

The proposed model consists of three parts: an image encoder, a recipe encoder, and a discriminator. The image encoder takes in food images and generates a high-level feature representation. The recipe encoder takes in recipe texts and generates a sequence of ingredient embeddings and a sequence of step embeddings. The discriminator takes in pairs of image and recipe embeddings and determines whether they are real or fake. The joint embedding space is learned by minimizing a combination of adversarial loss and embedding similarity loss.

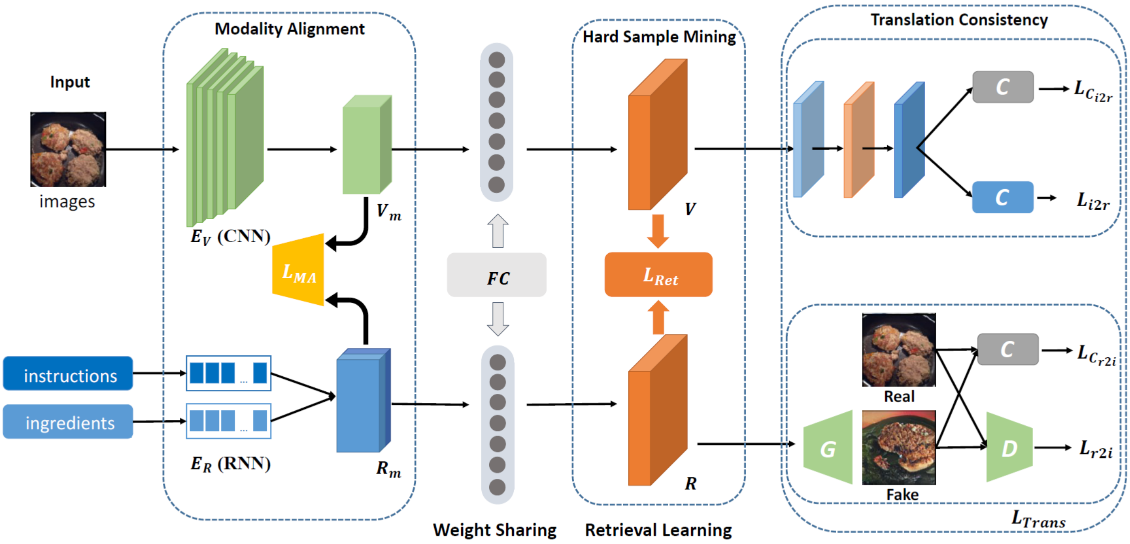


Figure 3.5: Adversarial Cross-Modal Embedding

Chapter 4

Implementation

During my second term, I concentrated on designing and developing a fully functional program, named “Food Master”. It took me about two months to finish developing the interface of the software and another month for implementing the chosen algorithm in the back end. This is a cross-platform program, so it can be accessed through the web, PC application and Android software. Up to now, the software has already been able to achieve most of the tasks and passed the program test as well as the user satisfaction survey. The software’s detail will be explained below.

4.1 Functionality and Layout

The software aims to proffer users the highest speed of image process as well as the accurate result of food recognition and recipe generation, which are the essential purpose to implement the optimised model mentioned before. Users only need to upload their favourite food images in the interface, the back end will get the image data posted from the front end. Then, the model implemented in the server will take the image into processing, and return the title, ingredients, and recipes of the food. During the processing, the front end will keep requesting the results until getting them from the back end. In the final stage, the interface will show all the details of the recipes generated by the model. Users can provide feedback, give likes and add favourites on the recipe details page.

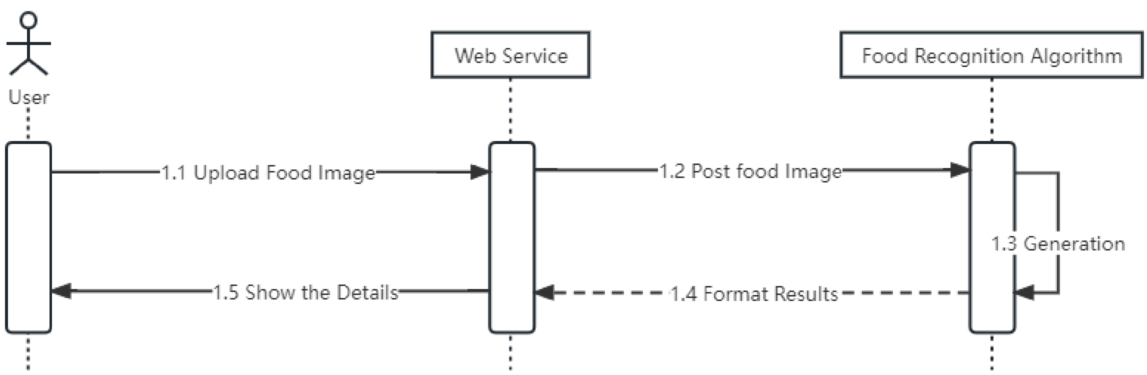


Figure 4.1: Dishes Image Recognition

Moreover, the algorithm of recipe generation is implemented further to obtain different functions. For instance, I developed a search tool that enables users to find recipes using any keyword related to ingredients or dishes. The software is also capable of accepting images of ingredients as input for the search function, because of its ability to convert images into ingredient lists.

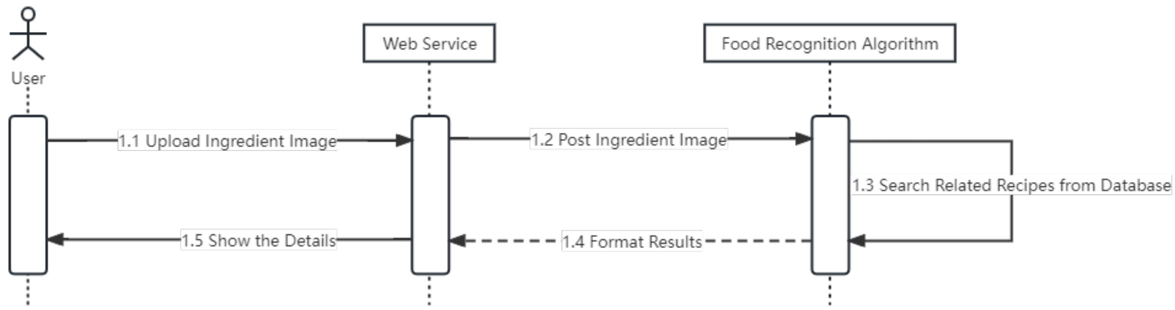


Figure 4.2: Ingredient Image Recognition

4.2 Front-end Development

4.2.1 Designing the Interface

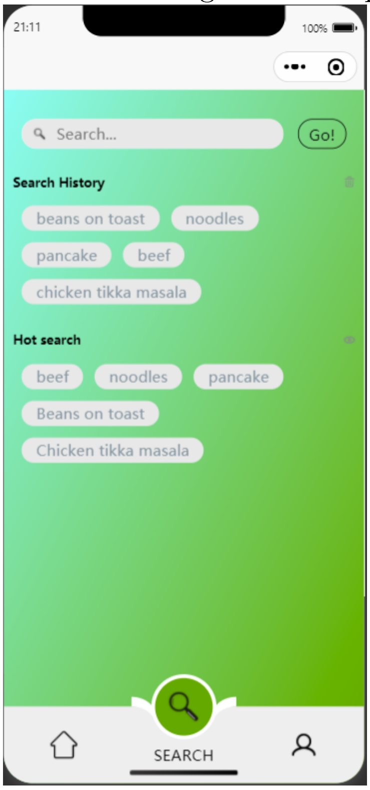
The software is designed to obtain a user-friendly interface and make it the most convenient to navigate and use. While I was taking internships as an E-commerce software engineer in a company, I gathered a lot of experience in developing a web-based program using Vue as a main programming language. Vue is a popular open-source JavaScript framework for building user interfaces and single-page applications. It is always flexible, modular and easily adaptable, but also Vue supports many platforms such as iOS and Windows. Therefore, I kept utilizing it on the dissertation project, and then package the code into a platform-specific executable.

The main colour scheme of the software is green, symbolizing freshness, health, and vitality. The interface is divided into three main pages in the beginning: the index page, search page, and user page. There is an intelligent navigation bar on the bottom of the main pages, which can swipe to one another page if the user clicks the related navigation button on it.

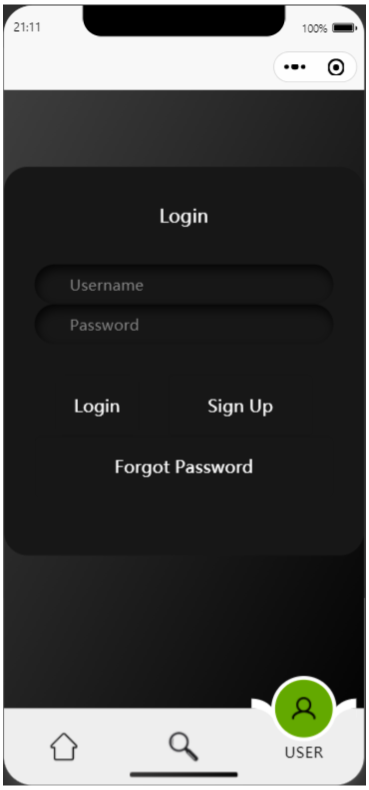
The search page has been implemented with the search tool developed by myself, which allows the user to search for specific recipes based on keywords, such as the name of a dish or an ingredient. The search results are displayed in a list format, with the recipe name, image, and brief description. The user can click on a recipe to view the detailed recipe page. Moreover, the page will also keep the search records list and show a list of the most popular or trending search queries in the “Hot Search”, if implemented the user authentication functions (mentioned in future development). These search terms often reflect current events, popular culture, or the interests of users at that particular time. Hot searches can be a useful tool for discovering new content or staying up-to-date with the latest trends and discussions.

Table 4.1: The Main Pages of the Application

Index Page



Search Page



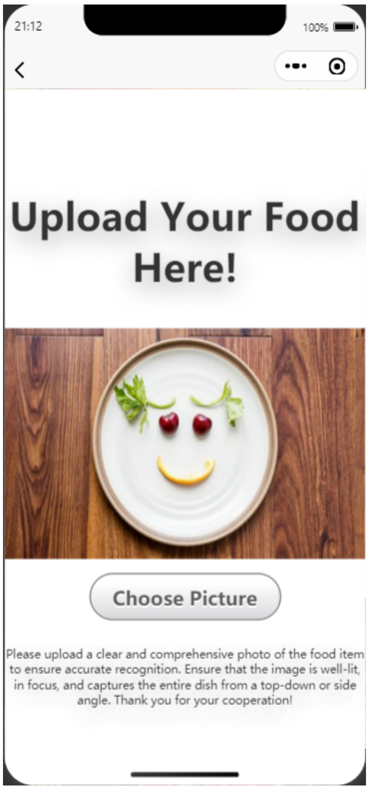
User Page

So far, I have only developed the interface of the user page, whose login page has two input fields for username and password. Users can also choose the sign-up button when they do not have an account or click the forget password button to change to a new one. After verifying the user's account, the user page will display the user's personal information and allows them to manage their account settings. The user can view their saved recipes, upload their own recipes, and update their profile information.

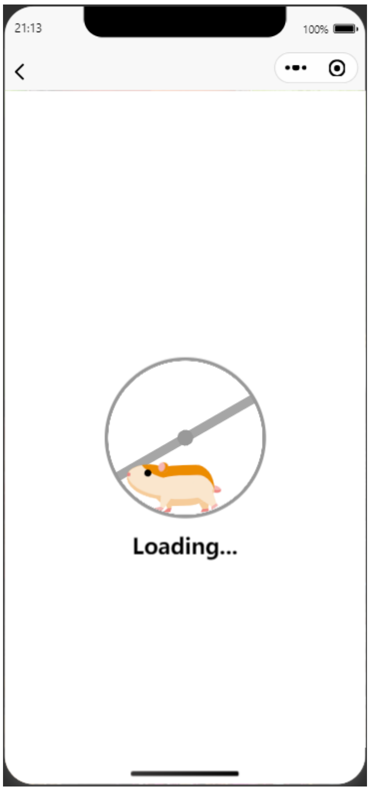
4.2.2 Integrating Image Uploading Functionality

The index page is the home page of the software, which displays the title of the software and two large buttons as the entrance of the main functions, that recognise recipes and ingredients. If users click the button, they can access different functions corresponding to the title of the button. The operation steps of them are quite the same, so I take the most common function, recognising recipes from food images, as an example: If users click the button of recognise recipe, users will first jump to the page which has a title, a default image (an empty plate), and a button with some guidance below. Users are expected to click the "Choose picture" button, which invokes the file system function, allowing them to select images from their local disk. Once the user confirms the image, the program automatically uploads the image file to the server and waits for processing. During this time, a cute hamster appears on the loading page, running in a wheel until the front end receives the results.

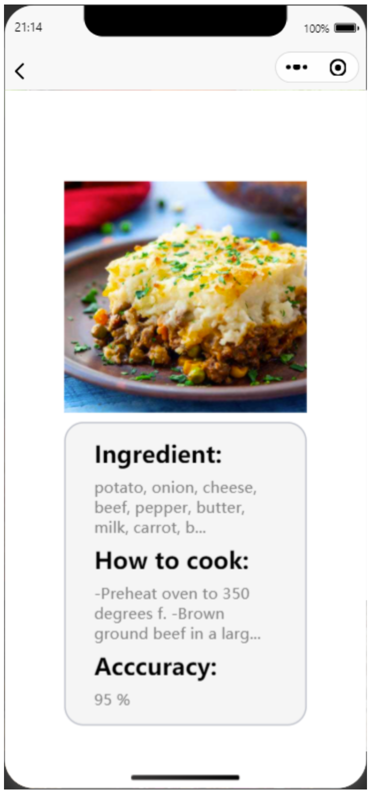
Table 4.2: The Upload Page



Upload Page



Loader

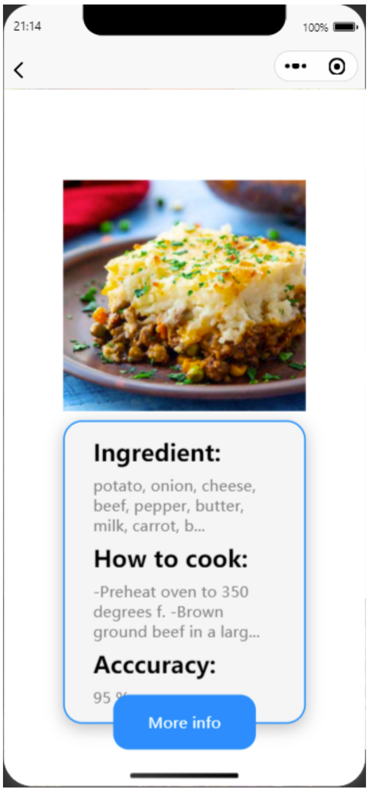


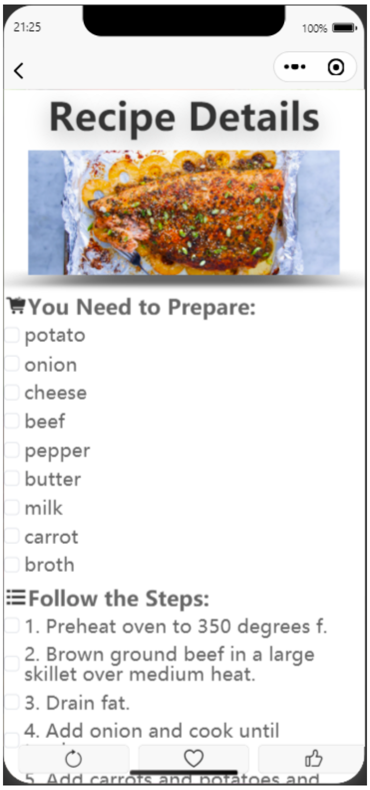
Recipe Page

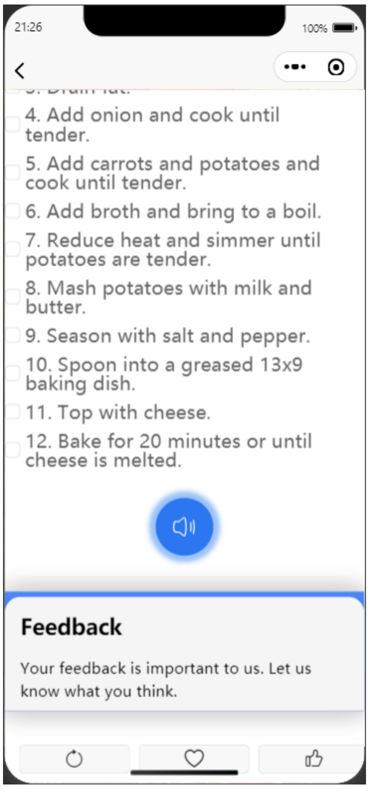
On the recipe page, there will be the image that the user uploads displaying on the top of the page, allowing the user to check whether it is the right one or not conveniently. In the box below, it shows a list of possible recipes related to the food image, ranking from high to low. In this version of the program, I only put the most likely result in the box and also omit the redundant content of the recipe since that has no space to display them all. If users would like to view the detail of the recipe, they can click the box where a button of "More info" will raise and enter the recipe details page. On that page, there are all the detailed ingredients users need to prepare and cooking instructions people could follow. Furthermore, I have thoughtfully designed a text-to-speech feature and check-boxes function for users, which allows them to check off each completed step as they go through the recipe. On the bottom of this page, people will see three buttons for re-uploading images, adding favourites and giving likes. A large blue input box jumps out when users swipe down on the feedback block.

Overall, the interface is designed to be intuitive and user-friendly, with clear navigation and easy-to-use features. The combination of image recognition technology and traditional search methods provides users with a comprehensive recipe database and allows them to easily find and access the recipes they need.

Table 4.3: The Recipe Page







More info Button

Details Page

4.3 Back-end Development

4.3.1 Previous Ideas

My previous plan was to develop the whole server and database with the framework of Spring Boot and Mybatis. Not only I was familiar with these, but also Spring Boot could simplify the process of developing, deploying, and running Java-based applications by providing default configurations and easy-to-use tools for common tasks, while Mybatis is advanced in mapping between Java objects and SQL databases.

On the other hand, when used together, Spring Boot and Mybatis can streamline the process of building web applications with a focus on simplicity and ease of use. Spring Boot handles the overall structure and configuration of the application, including features like dependency injection, auto-configuration, and embedded web servers.

At the same time, Mybatis is responsible for managing database connections, executing SQL queries, and mapping the results to Java objects. It provides a simple, flexible, and efficient way to handle database operations without the need for writing a lot of boiler-plate code.

Unfortunately, the idea is pretty great and easy to implement with Java, but I found it is much more complex to integrate the server functions with the food image recognition model, which is written by Python. Indeed, there are some approaches to connect two modules together, allowing the back end to get the food image, to send it to the model

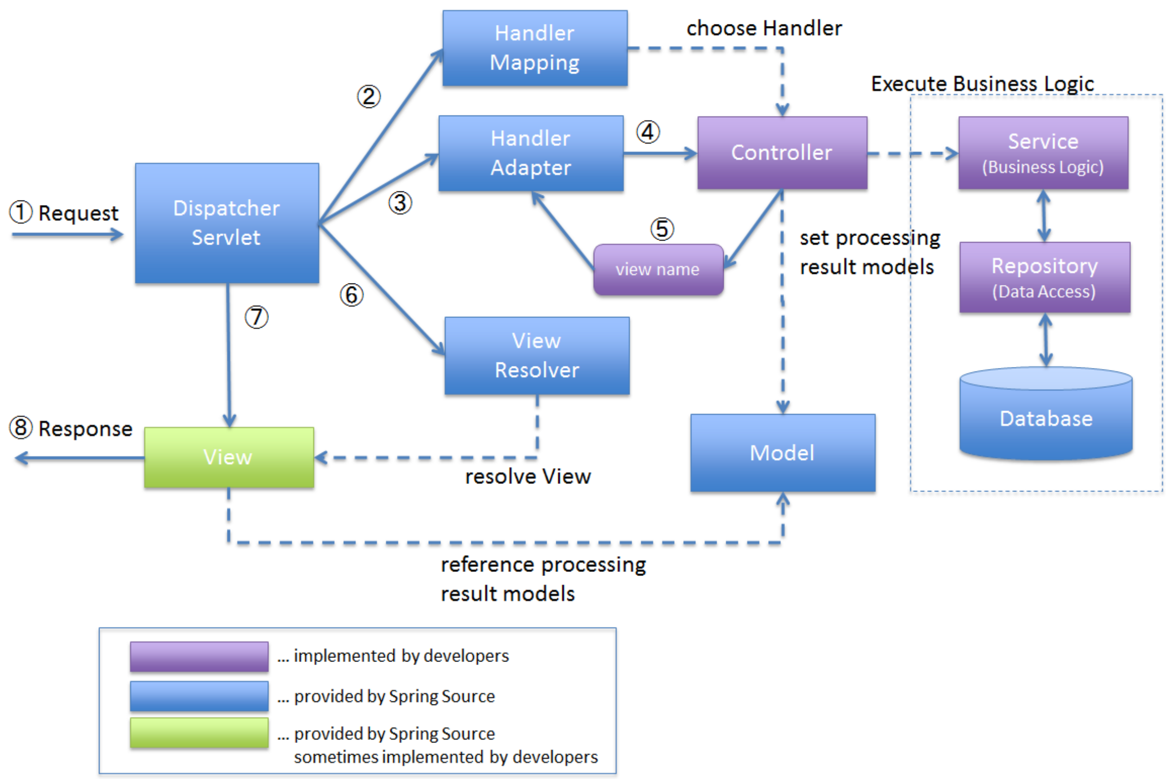


Figure 4.3: Request Life Cycle [1]

and to post the recipes to the front end after it gets the output. In spite of that, some common issues may arise.

First and foremost, the individual threads of GET, POST, and algorithm controller operations are hard to schedule or pose a threat to thread safety. Because the data packages containing recipe information are a shared resource, I have to set an asynchronous lock to ensure only one thread can access it and set a time period for the model to generate the results, which the GET and POST threads should be blocked. After evaluating its performance, it was found that the time of the model's processing that depends on the performance of the CPU or the size of the images is uncertain. Not to mention the potential problems like deadlock, determining whether to block the server for 5 seconds or 10 seconds is challenging, as setting a fixed blocking time can result in sub-optimal performance, such as unnecessary waiting or potential timeouts. During the evaluation, there were instances of null returns as the model got stuck for even 21 seconds.

Furthermore, the optimized classification model is deployed using a machine learning library called PyTorch, which is capable of loading the pre-trained model into the program to save processing time. It means the model is definitely impossible to convert to the Java version, and the threads schedule will go across two programming languages. There are other problems met in my developing stage, I gave Spring Boot up and turned to looking for another way to overcome it.

4.3.2 Setting Up the Flask Framework For Server-Side Development

Therefore, I found an easier solution, the Flask framework. it is a popular Python web framework used for building web applications. Flask is known for its simplicity and flexibility, making it an ideal choice for developing the back end of food image recognition recipe software.

However, these two different frameworks have similar working methods. Relatively, the flask will keep listening to port 8080 when the program is running. Once it has caught any post data from this port, which means the front end sends the food image to the back end, the flask will store the data in a variable for further processing.

I created a function called ”submit” that can request an image file, validate it, call the recognition model, and post the recipe back if the output of the model is not empty. I implemented encapsulation by creating a new Python file named ”demo” within the model project, instead of starting a new standalone project, to incorporate the “submit” function.

4.3.3 Integrating the Food Image Recognition Model into Back End

One of the key functionalities of the back end is integrating the food image recognition model into the system. The food image recognition model is responsible for analysing the uploaded food images and extracting relevant features from them. These features are then used to identify the food and generate a set of recipes that can be made with the identified food. The model calculates the possibility of a recipe according to the similarity between images, and then sorts them all, picking the highest one as the result.

In addition to the food image recognition model, the back end also handles the communication between the front-end interface and the server. This includes uploading images, evaluating the image quality, and returning the recipe results back to the front-end interface.

4.4 Future Development

4.4.1 Pre-Process: Developing image evaluation algorithms

Since it is not able to ensure the quality of the image that the user uploaded, the program needs to measure the quality of the image and determined a reasonable threshold. However, I won’t have a non-distorted image to calculate the quality of the distorted image. The only input obtained is the image whose quality is to be measured, there is no image at all that can be used as a reference. In this case, we need a reference-free IQA metric called the Reference-free Image Spatial Quality Assessor (BRISQUE). BRISQUE has been shown to outperform other popular image quality assessment methods such as PSNR and SSIM in terms of accuracy and consistency with human perception of image quality.

BRISQUE will assign the image a mean quality score [20] between 0 (best) and 100 (worst). It stands for blind or no reference image spatial quality evaluator, a reference-free spatial domain image quality evaluation algorithm. The overall principle of the algorithm is to extract mean subtracted contrast normalized (MSCN) coefficients from an image, fit the MSCN coefficients to an asymmetric generalized Gaussian distribution (AGGD), extract the features of the fitted Gaussian distribution and input them to a support vector machine SVM for regression to obtain the image quality evaluation results. In other words, the model is used to estimate the amount of naturalness between adjacent pixels in natural images, which is then used to determine the image quality. The quality score given by BRISQUE is based on how much the image deviates from the model of natural images.

Listing 4.1: Sample Python code using BRISQUE

```python
import imquality.brisque as brisque
import PIL.Image

Path = 'imagepath'
image = PIL.Image.open(path)
brisque.score(img)
```

Actually, I have introduced this approach in my interim report and planned to implement it in the final project. However, I was busy in implementing other functions in the back end, such as determining which framework I should integrate, which took me too much time to trial and error. Therefore, I removed this part in the process of my dissertation, and am ready to develop it in the future.

Based on the design, BRISQUE is utilized to assess the image quality, preventing the model from accepting blurry or low-quality images uploaded by users. The algorithm will start first when the "submit" function gets the image data and calculates the scores of the image quality. A reasonable threshold will be set after evaluation. For example, if the threshold is set at 20, images with scores below 20 are considered good samples and stored back in the stack. While those with scores above 20 will be rejected, the system will send back the warning to the front end and pop a window to alert users to upload a new image.

4.4.2 Interaction: Recipe Results Processing

In addition to these features, the software allows users to filter search results based on dietary preferences, cooking time, or difficulty level. This customization ensures that users can find recipes that align with their specific needs and preferences. Additionally, the software offers a social component where users can share their creations, exchange cooking tips, and follow other users with similar tastes. This fosters a sense of community and encourages the exchange of ideas and inspiration within the platform.

Besides, implementing data validation and error handling is also significant to improve this software. As tested in the evaluation part, the model has no ability to deal with something that is not a real dish, such as an empty plate or a trash bin. Therefore, it is necessary to use an item recognition model first to make sure there is food in images and then put the image into the model for further processing.

4.4.3 Server: Implementing user authentication and authorization

As mentioned before, I would like to implement a user authentication and authorization system by developing the framework of Spring Boot and Mybatis. Although these functions were not the primary focus of developing the "Food Master" software, their inclusion will enhance the user experience.

First of all, a login function will be implemented on the user page. I plan to build a database using MySQL for storing recipe data and user information. A function to verify the user identity should be built for users to create and log in to their own accounts on different devices, making it possible for cloud synchronization to keep users' favourite recipes and food images online. In the approach, it attaches great importance to enhancing system security as the database stores all the user information which will pose a threat to users' privacy if the information leakage happens.

Chapter 5

Evaluation

In this section, I provided a detailed evaluation of the proposed food image recognition model. The model was tested for accuracy using multiple food images from various cuisines, and user satisfaction was assessed through a survey with multiple participants.

5.1 Model Accuracy Test

To evaluate the model's comprehensive accuracy, I collected a set of diverse food images from different cuisines, such as Chinese, Japanese, American and especially European. This dataset was carefully designed to ensure that the images captured the variety of dishes and presentation styles found in each cuisine. Moreover, the dataset incorporated images taken from different angles, and with varying degrees of image quality, simulating real-world scenarios in which the model would be utilised.

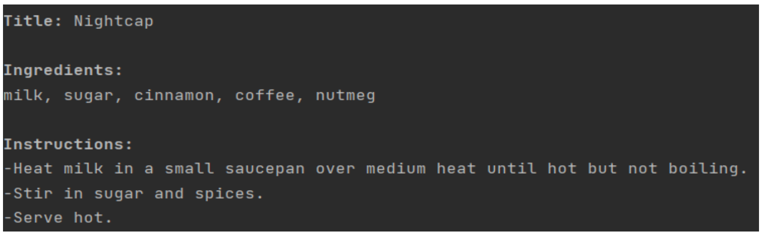
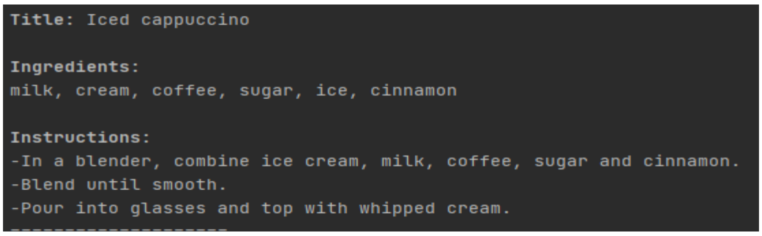
To compare the results generated by the model with human performance, I prepared draft recipes by myself in advance and collected a set of more official recipes online as the true answers for comparison. This approach aimed to emphasize the objectivity and reasonableness of the calculated data, providing a fair basis for assessing the model's performance.

5.1.1 Difficulties

In practice, it is challenging to determine how precise the model's output results are because various factors can affect the accuracy standards of a food image recognition model, such as dataset limitations as well as subjectivity and ambiguity. A biased or limited dataset might not cover the full range of food items and cuisines, leading to ambiguity in the correct categorisation.

For example, in Table 5.1, the machine recognised a rice pudding as a nightcap, while considering an actual nightcap as an iced cappuccino. Of course, there could be other reasons for this phenomenon as well. The boundary between certain types of dishes might be unclear, or the characteristics of dishes shown in uploaded images are limited by complex visual features, such as colours, textures, and shapes.

Table 5.1: Cases of Hard to Distinguish

Rice Pudding	**Generated Results**
	 **Title:** Nightcap **Ingredients:** milk, sugar, cinnamon, coffee, nutmeg **Instructions:** -Heat milk in a small saucepan over medium heat until hot but not boiling. -Stir in sugar and spices. -Serve hot.
Actual Night cup	**Generated Results**
	 **Title:** Iced cappuccino **Ingredients:** milk, cream, coffee, sugar, ice, cinnamon **Instructions:** -In a blender, combine ice cream, milk, coffee, sugar and cinnamon. -Blend until smooth. -Pour into glasses and top with whipped cream.

5.1.2 Measurement

In order to standardise the measurement of the model's accuracy, several strategies are employed. I develop a set of custom metrics to evaluate the quality of generated results.

For quantifying the evaluation results, I have designed a standardized algorithm to evaluate the accuracy of models. Learning from the user satisfaction survey, most people value the recipe results more highly. Therefore, for the three items, title, ingredients, and recipes, assign the following weights: 20%, 40%, and 40%, respectively. Next, I designed some thresholds to determine whether the ingredients and the recipes are nearly correct and meaningful.

1. Low accuracy (the score is 0): the cases with fewer than 2 of the same main ingredients and fewer than 3 steps in the recipe. This tier indicates that the generated recipe has low similarity to the actual recipe and may not successfully complete the recipe. To improve the user experience and ensure the quality of the results, any bad or inaccurate results should be promptly discarded by the user in real-time. Therefore, 0 is assigned as the final score for such unsatisfied results to distinguish them from reliable ones.

2. Medium accuracy (the score is 0.5): the cases with 2-3 of the same main ingredients. 4-5 similar steps in the recipe. This tier indicates that the generated recipe is similar to the actual recipe but still has some discrepancies. Such results should be treated with scepticism because some of the ingredients and instruction steps in the generated results are different from those shown in the uploaded photos. While some of the information may be useful as a reference, others may be inaccurate. It is difficult to determine the proportion of each part, therefore, these results have been given a score of 0.5.

3. High accuracy (the score is 1): the cases with more than 3 of the same main ingredients and more than 5 similar steps in the recipe. This tier indicates that the generated recipe is very close to the actual recipe and can effectively complete the recipe. To simplify the scoring process, I assign a score of 1 to reliable results, indicating 100% of accuracy.

After comparing the data generated by the models, I designed a formula to calculate the final score for a kind of cuisine. The sum of these weighted results (LA, MA, and HA) is divided by the total number of results (SUM), which is the sum of the individual counts of LA, MA, and HA. The resulting value, "Accuracy", provides a measure of the overall performance of the system, taking into account the different levels of accuracy. The equation calculates the average accuracy of model results generated from one type of cuisine, such as burger or sushi, which not only count the number of each kind of results but also will take their weight (scores) into consideration.

$$\text{Accuracy} = \frac{(0 \times \text{LA}) + (0.5 \times \text{MA}) + (1 \times \text{HA})}{1 \times \text{SUM}} \tag{5.1}$$

LA = The number of low accuracy results

MA = The number of medium accuracy results

HA = The number of high accuracy results

SUM = **LA** + **MA** + **HA**

5.1.3 Downsides

It should be noted that these methods are not perfect and may be subject to subjective judgment and varying evaluation criteria. First of all, the thresholds will be varied among different food photos and cuisines. Pre-setting every level of results could lead to data bias. Besides, depending on the specific application, the weights of 0, 0.5, and 1 might be inaccurate to represent each parameter. Likewise, if the values of LA, MA, or HA are prone to outliers or extreme values, the calculated accuracy might be heavily influenced by these extreme values. Depending on the context, this might lead to a less accurate representation of the overall accuracy.

Therefore, when assessing the accuracy of a recipe generation system, the data can only be provided for reference and comparison purposes only and should not be taken as having actual significance.

5.1.4 Result

Following the completion of a model accuracy test, the results have been analysed to gain insights into the performance of the machine learning model. The test outcomes are vital for understanding the model's strengths and weaknesses, as well as informing future improvements and refinements.

		Title(20%)	Ingredient(40%)	Recipe(40%)	Mean
Western cuisine	Burger	0.67	0.67	1.00	0.80
	Meat	0.60	0.40	0.40	0.44
	Dessert	0.57	0.71	0.71	0.69
	Sides	0.40	0.40	0.40	0.40
	Shepherd's pie	1.00	1.00	1.00	1.00
Eastern cuisine	Rice	0.67	0.83	0.83	0.80
	Sushi	1.00	1.00	0.67	0.87
	Mean	**0.70**	**0.72**	**0.72**	**0.71**

Figure 5.1: Table for Test Results

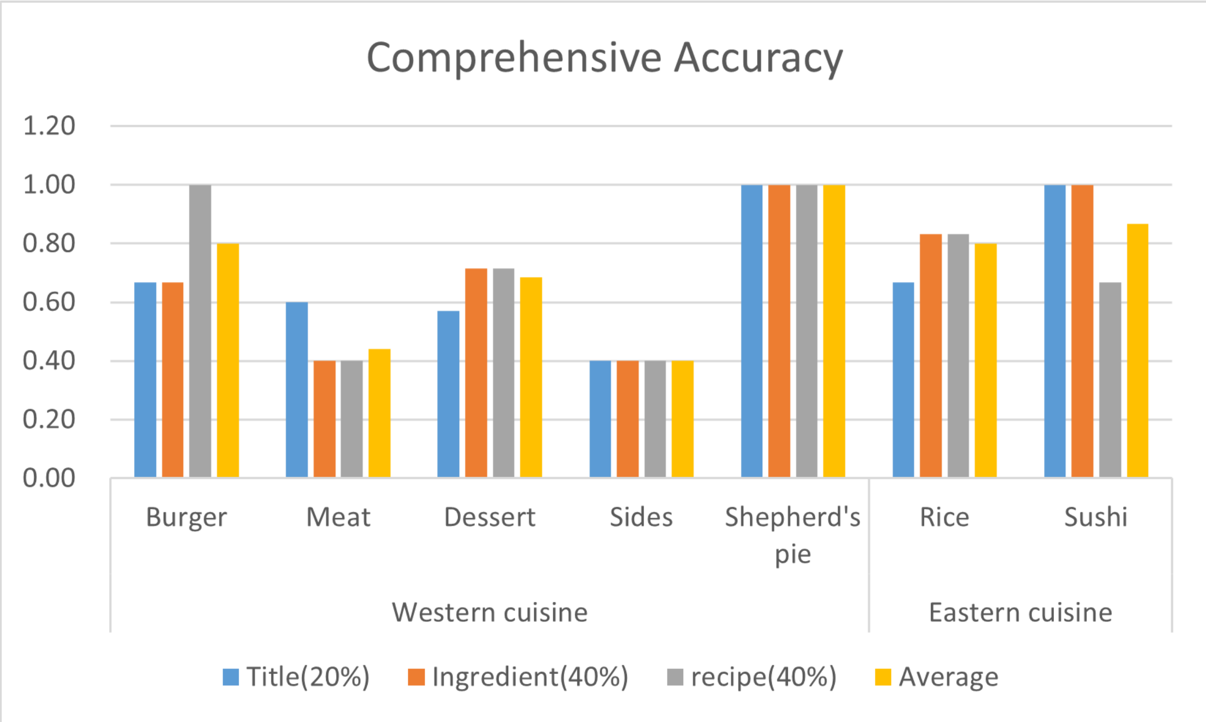


Figure 5.2: Chart for Showing Accuracy of Test Results of Different Cuisine

Overall, the model has an average accuracy of 0.71, which demonstrates that the model performs well in many cases, successfully generating relevant and coherent outputs. This indicates that the model can effectively analyse photos of a wide variety of cuisines and generate reasonable recipe suggestions. However, there are certain areas, especially in Western cuisine, where the model can be improved.

For Western cuisine, the model performs exceptionally well for Shepherd's Pie, with a perfect score in all categories. It also shows strong performance in identifying Burgers, with an average accuracy of 0.80. The model has some difficulty with Meat and Side dishes, scoring 0.44 and 0.40 respectively, which indicates that there is room for improvement in these categories. While for Eastern cuisine, the model achieves high accuracy for Sushi, with an average score of 0.87. It also performs well in identifying Rice dishes, with an average accuracy of 0.80. The model demonstrates consistent performance across Eastern cuisine categories, with only a slight variation in accuracy between them.

		Title(20%)	Ingredient(40%)	Recipe(40%)
Western cuisine	Burger	±0.15	±0.2	0.00
	Meat	±0.1	±0.12	±0.12
	Dessert	±0.12	±0.14	±0.14
	Sides	±0.88	±0.1	±0.1
	Shepherd's pie	0.00	0.00	0.00
Eastern cuisine	Rice	±0.15	±0.17	±0.17
	Sushi	0.00	0.00	±0.24

Figure 5.3: The Deviation of Test Results

I also calculated the deviation of each category of cuisines. Except for the sides' title with the greatest deviation ± 0.88, the deviations for other items are usually around $0.1 - 0.3$, and some even reach 0. It indicates that in the test dataset, the range of error made by the model is not very large, so the model can still provide relatively accurate recognition results in most cases. However, ± 0.88 deviation means the model is quite unstable in recognising sides, while 0 deviation supports the model can correctly identify food titles and recipes without any error, including Shepherd's pie and sushi.

5.1.5 Analysis

There could be several reasons for the inaccuracy of a food image recognition model. First and foremost, a model's performance is heavily dependent on the quality and quantity of the training data. If the dataset used for training is not diverse, comprehensive, or representative of real-world scenarios, the model may struggle to generalize and accurately recognize different types of food.

Likewise, some food items may have similar appearances, making it challenging for the model to differentiate between them. For example, distinguishing between various types of pasta or closely related dishes can be difficult even for humans. Conversely, food items in photos are often presented in various environments, with different ingredients and cooking methods. This complexity can make it challenging for the model to accurately identify and classify the food items in an image.

In addition to the food itself, other compositional elements in the image can also affect food recognition. When food items are partially obscured or overlapping in an image, it can be difficult for the model to recognize them correctly. The model may struggle to differentiate between the individual items or accurately estimate their proportions. Besides, poor lighting or low image quality can affect the model's ability to recognize food items. Shadows, glare, and noise in the images can lead to misinterpretations and reduced accuracy.

The choice of the model architecture and its complexity can also impact the accuracy of the food image recognition model. If the model is not capable of capturing the intricate patterns and relationships present in the data, it may struggle to achieve high accuracy.

Moreover, the model has "perfect" accuracy in classifying some kinds of food such as shepherd's pie. Even though I intentionally zoomed in on the original image and only selected a small portion of it, the results are still very impressive. Obviously, this is highly unlikely to occur in real life. Based on speculation, it is possible that this is due to bias in the training data of the model, which makes the model more prone to return certain cases.

5.1.6 Possible Improvement

Considering the challenges mentioned above, I have devised several effective strategies to address these issues:

The software can analyse the clarity of user-uploaded images before processing. If the image clarity is insufficient, the software can prompt the user to upload a higher-quality image. This step ensures that the food recognition model works with clear images, reducing the chances of misinterpretation due to poor image quality.

By incorporating an algorithm to identify the food item before generating the recipe, the software can narrow down the scope of recipe generation, thereby improving the recipe's accuracy. This approach also helps prevent the processing of non-food images, such as empty plates or the sun.

To further improve the model's performance, context-based recognition can be employed. This method uses additional information from the image, such as the presence of utensils, serving size, or the surrounding environment, to help the model make more informed decisions about the food items.

Allowing users to provide feedback and correct the model's output can help improve its accuracy over time. By incorporating user feedback, the model can learn from its mistakes and adapt to better recognize various food items.

Utilizing data augmentation techniques, such as image rotation, scaling, and flipping, can help increase the diversity of the training data and improve the model's generalization.

Combining multiple food recognition models using ensemble methods can enhance the overall performance. By leveraging the strengths of different models, the ensemble can provide a more accurate and robust food recognition system.

Implementing these strategies can help improve the accuracy and reliability of the food image recognition model, ultimately enhancing the user experience and the software's effectiveness.

5.2 User Satisfaction Survey: See Appendix A for Survey Details

5.2.1 Content

To assess user satisfaction with the food image recognition model, I conducted a survey involving 20 participants. The participants included different groups of people with various cuisines. As the data collection method was limited, most of them are my friends. They come from different regions and some of them are excellent cooks while others are not. This diverse group allowed me to gather a wide range of perspectives on the model's performance and usability.

At the beginning of the survey, the participants are asked to choose one photo and upload it to utilise the application by themselves to simulate the realistic usage environment. The survey never requires participants to upload their own photos, which avoids the possibility of accidental leakage of personal privacy. Otherwise, the photos provided in a dataset which contains some randomly selected food images are uncertain whether they have already been trained or not. It ensures a fair evaluation of the model's performance and provides opportunities for identifying special cases.

The survey consisted of some simplified questions, focusing on rating the model's ease of use, speed, and accuracy on a predefined scale, and the latter aimed to collect open-ended feedback on the participants' experiences. Enough space is also left for participants to provide suggestions for identifying potential areas of improvement which are not captured by ratings. For example, "Which of the following situations are you most likely to use this program?" or "Which colour do you prefer as the primary theme colour for the entire software?"

5.2.2 Survey Results

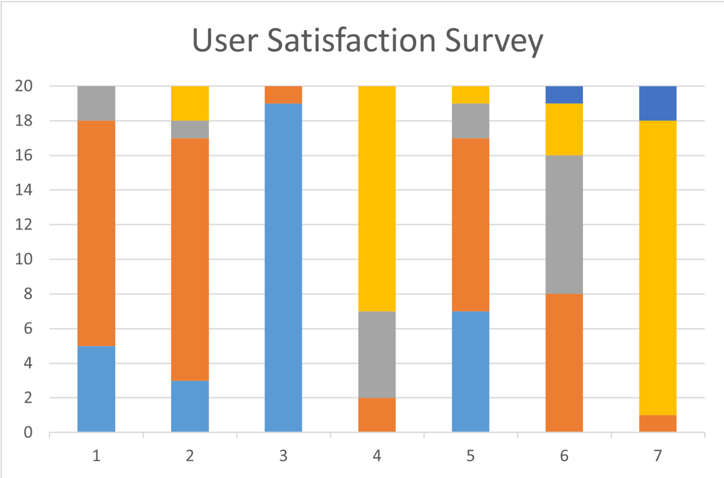


Figure 5.4: Table for Survey Results

Total=20	Survey questions	1	2	3	4	5
	Question 1: How do you like the colour scheme used in this software?	5	13	2	0	0
	Question 2: How do you like the font size?	3	14	1	2	0
	Question 3: What feature in this software do you think you would use the most?	19	1		N/A	N/A
	Question 4: Which of the following situations are you most likely to use this program?	0	2	5	13	N/A
	Question 5: How do you feel about the speed of processing?	7	10	2	1	0
	Question 6: How accurate do you think the results generated by the software are?	0	8	8	3	1
	Question 7: How would you rate the usefulness of this software?	0	1	0	17	2

Figure 5.5: Chart for Survey Results

5.2.3 Analysis

After conducting a user satisfaction survey, we have analysed the results to gain insights into our users' experience with the software. The feedback obtained from the survey has been instrumental in identifying areas of improvement, as well as validating the strengths of our software.

In summary, the user survey indicates that the food image recognition software is generally well-received, with positive feedback on its colour scheme, font size, processing speed, and usefulness. Most users (13 out of 20) rated the colour scheme positively with "Good". 5 users gave the highest score, indicating a strong preference for the colour scheme while no one rating the colour scheme negatively. The font size is very popular among users.

19 users selected "Search recipes" as the most useful feature, indicating a strong preference for this functionality. This suggests that the software can focus on developing and enhancing this feature in future updates. Aside from Question 7, where the majority of survey participants looked highly to the software's usefulness, people have varying opinions on the software's processing speed and the accuracy of its results, which could be an area for improvement.

People are most likely to use the software in situation of eating outside. Based on my conjecture, it is quite likely that trying new dishes at restaurants can inspire people to recreate similar dishes at home. By searching for recipes while eating out, they can gather ideas and learn how to make the dish themselves. This motivation for searching recipes while dining out is driven by the desire to bring the enjoyable dining experience into their home kitchens and expand their cooking repertoire.

5.2.4 Possible Improvement

It inspires me that the software could incorporate new features for users, such as automatic recipe uploading, cloud synchronization, and user-generated sharing links, to enhance user experience and functionality.

This feature would enable users to easily upload their recipes to the platform. Users could input the ingredients, steps, and other relevant information, and the software would automatically format and store the recipe.

By implementing cloud synchronization, users can access their recipes from any device with internet access. This feature ensures that users' recipes are safely stored and readily available, providing convenience and flexibility.

Allowing users to create and share custom links to their recipes encourages social interaction and recipe sharing within the software's user community. This feature can foster a sense of collaboration, helping users discover new dishes and expand their culinary knowledge.

These additional features could significantly improve the software's user experience, making it more versatile and attractive to a wider range of users. Moreover, the survey results reveal a diverse range of user preferences, suggesting that additional customization options may be beneficial. By offering more personalizing features, users would be able to tailor the software to their specific needs, further increasing satisfaction levels.

However, some users have reported difficulties with navigation or clarity of instructions in the feedback. These issues highlight potential areas for improvement, and addressing them would significantly enhance the overall user experience.

In conclusion, the user satisfaction survey has provided valuable insights into how our software is being perceived and used. By addressing the identified areas of improvement and incorporating user feedback into future development, we can continue to enhance the software and deliver an even better user experience.

5.3 Evaluation Results

Table 5.2: Test Cases of Rice

Food Images	Generated Results
	**Title:** Spanish rice **Ingredients:** rice, corn, pepper, onion, oil, tomato, salt, peas **Instructions:** -Heat oil in a large skillet over medium heat. -Add onion and bell pepper and cook until tender, about 5 minutes. -Add rice and cook, stirring, until rice is lightly browned, about 2 minutes. -Add tomatoes, corn, peas, salt and pepper. -Bring to a boil, reduce heat to low, cover and simmer until rice is tender, about 20 minutes. ====================
	**Title:** Shrimp fried rice **Ingredients:** rice, shrimp, onion, pepper, oil, egg, clove, soy_sauce, salt, corn **Instructions:** -Heat oil in a large skillet over medium-high heat. -Add shrimp and cook until pink, about 2 minutes. -Add onion and garlic; cook and stir until onion is tender, about 3 minutes. -Add rice, corn, soy sauce, salt, and pepper; cook and stir until heated through, about 2 minutes. -Add eggs; cook and stir until eggs are set, about 2 minutes. ====================

Table 5.3: Test Cases of Sushi

Food Images	Generated Results
	**Title:** Salmon sushi rolls **Ingredients:** rice, nori, cheese, salmon, mayonnaise **Instructions:** -Mix the sushi rice with the sushi rice. -Spread the sushi rice on the nori seaweed. -Top with the salmon and cheese. -Roll up the sushi rice from step 1 into a log. -Cut into 4 pieces and serve.
	**Ingredients:** cucumber, carrot, nori, pepper, rice, juice, salt, seeds **Instructions:** -In a small bowl, combine the rice, lemon juice, and salt. -Stir to combine. -Cover and refrigerate until ready to use. -In a large bowl, combine the nori, cucumber, carrot, and sesame seeds. -Add the rice and toss to combine. -Spoon the rice mixture onto a nori sheet. -Roll up the sushi rice, tucking in the ends as you roll. -Cut into 4 equal pieces.

Table 5.4: Test Cases of Burger

Food Images	Generated Results
	**Title:** Hamburger buns **Ingredients:** pepper, bun, onion, beef, salt, lettuce, tomato, oil **Instructions:** -Preheat oven to 350 degrees f (175 degrees c). -Heat oil in a large skillet over medium heat. -Cook and stir onion in hot oil until tender, about 5 minutes. -Stir ground beef into onion; cook and stir until beef is crumbly, evenly browned, and no longer pink, about 10 minutes. -Drain and discard grease. -Stir tomato sauce, salt, and pepper into ground beef mixture; simmer until flavors blend, about 10 minutes. -Place hamburger buns on a baking sheet. -Spoon beef mixture into buns. -Bake in the preheated oven until buns are toasted, about 10 minutes. ====================
	**Title:** Cheeseburger **Ingredients:** cheese, bun, pepper, beef, salt, onion, garlic, lettuce, tomato **Instructions:** -In a large skillet, cook beef, onion, and garlic over medium heat until meat is no longer pink; drain. -Stir in salt and pepper. -Reduce heat; simmer, uncovered, for 10 minutes. -Place cheese on bottom half of each bun; top with meat mixture, lettuce and tomato. -Cover with top halves of buns. ====================

Table 5.5: Test Cases of Dessert

Food Images	Generated Results
	**Title:** Cinnamon coffee cake **Ingredients:** sugar, flour, egg, salt, baking_powder, butter, milk, cinnamon **Instructions:** -Preheat oven to 350 degrees f. grease and flour a 9 x 13 pan. -In a large bowl, cream butter and sugar until light and fluffy. -Add eggs one at a time, beating well after each addition. -In a separate bowl, combine flour, baking powder, salt and cinnamon. -Add flour mixture to creamed mixture alternately with milk, beginning and ending with flour mixture. -Pour batter into prepared pan. -In a small bowl, combine brown sugar and cinnamon. -Sprinkle over batter. -Bake for 45 minutes or until a toothpick inserted in the center comes out clean.
	**Title:** Strawberry jam cookies **Ingredients:** sugar, flour, butter, salt, egg, jam, water, baking_powder, milk **Instructions:** -Preheat oven to 350 degrees f. -In a large bowl, cream together the butter and sugar until light and fluffy. -Beat in the egg and milk. -In a separate bowl, combine the flour, baking powder and salt. -Gradually add the dry ingredients to the creamed mixture. -Mix until well combined. -Roll dough into 1-inch balls and place on a greased cookie sheet. -Press a strawberry into the center of each cookie. -Bake for 10-12 minutes or until lightly browned. -Remove from the oven and let cool on a wire rack. -In a small bowl, combine the jam and water. -Drizzle over the cookies.

Table 5.6: Test Cases of Meat

Food Images	Generated Results
	**Title:** Chicken in butter sauce **Ingredients:** chicken, sugar, butter **Instructions:** -Preheat oven to 350 degrees f (175 degrees c). -Place chicken in a 9x13 inch baking dish. -In a small bowl, mix together the brown sugar and butter. -Pour over chicken. -Bake in preheated oven for 1 hour, or until chicken is cooked through. ====================
	**Title:** Beef stew **Ingredients:** onion, carrot, potato, pepper, flour, oil, broth, clove, thyme, salt **Instructions:** -Heat oil in a large dutch oven over medium-high heat. -Add beef and brown on all sides. -Remove beef from pan. -Add onion, garlic, thyme, salt and pepper to pan; saute until onion is tender. -Return beef to pan. -Add broth and potatoes. -Bring to a boil. -Cover, reduce heat, and simmer for 1 1/2 hours or until beef is tender. -Stir in carrots. -Cover and cook for 15 minutes or until carrots are tender. ====================

Table 5.7: Test Cases of Shepherd's Pie

Food Images	Generated Results
	**Title:** Shepherd's pie **Ingredients:** onion, potato, pepper, beef, carrot, butter, salt, flour, oil, broth, clove **Instructions:** -Preheat oven to 350 degrees f (175 degrees c). -Heat olive oil in a large skillet over medium heat. -Cook and stir beef in hot oil until browned and crumbly, 5 to 7 minutes; drain and discard grease. -Stir onion, carrots, and garlic into beef; cook and stir until onion is softened, about 5 minutes. -Stir flour into beef mixture; cook and stir until flour is completely dissolved, about 1 minute. -Pour beef broth into beef mixture; bring to a boil, reduce heat to medium-low, and simmer until thickened, about 10 minutes. -Season beef mixture with salt and black pepper. -Spread beef mixture into a 9x13-inch baking dish. -Spread mashed potatoes over beef mixture. ====================
	**Title:** Shepherds pie **Ingredients:** potato, onion, cheese, beef, pepper, butter, milk, carrot, broth **Instructions:** -Preheat oven to 350 degrees f. -Brown ground beef in a large skillet over medium heat. -Drain fat. -Add onion and cook until tender. -Add carrots and potatoes and cook until tender. -Add broth and bring to a boil. -Reduce heat and simmer until potatoes are tender. -Mash potatoes with milk and butter. -Season with salt and pepper. -Spoon into a greased 13x9 baking dish. -Top with cheese. -Bake for 20 minutes or until cheese is melted.

Table 5.8: Test Cases of Sides

Food Images	Generated Results
	**Title:** Fried potato wedges **Ingredients:** oil, salt, potato, pepper, flour, lemon, egg **Instructions:** -Peel potatoes and cut into wedges. -In a large bowl, combine flour, salt and pepper. -In a separate bowl, beat egg and add to flour mixture. -Dip potato wedges into egg mixture, then into flour mixture. -Heat oil to 375 degrees. -Fry wedges until golden brown, about 5 minutes. -Drain on paper towels. -Serve hot.
	**Title:** Lemon chicken **Ingredients:** oil, pepper, salt, chicken, lemon, flour, juice **Instructions:** -In a large bowl, combine the flour, salt and pepper. -Add the chicken and toss to coat. -In a large skillet, heat the oil over moderately high heat. -Add the chicken and cook, turning once, until golden, about 6 minutes. -Transfer the chicken to a plate. -Add the lemon juice and zest to the skillet and cook, scraping up any browned bits, until the sauce is thickened, about 2 minutes. -Return the chicken to the skillet and toss to coat. -Serve with lemon wedges. ====================

Table 5.9: Test Special cases

Food Images	Generated Results
	**RECIPE 1** **Title:** Coffee creamer **Ingredients:** milk, sugar, water, coffee, extract **Instructions:** -Combine all ing. -Whisk until creamer and sugar are disolved. -Store in fridge.
	**RECIPE 1** **Title:** Vodka cocktail **Ingredients:** juice, vodka **Instructions:** -Pour vodka and lemon juice into a shaker over ice. -Cover, and shake until the outside of the shaker has frosted. -Strain into a chilled martini glass to serve.

Chapter 6

Summary and Reflections

During the first semester of my dissertation, my focus was on researching various food image processing methods. My ideas evolved from simple classification algorithms to LSTM models, and eventually to the current state-of-the-art model developed by the MIT research group. Their Recipe 1M+ database and pre-trained model provided a significant advantage in terms of time-saving and accuracy.

In the second semester, I shifted my focus to software development and implemented the inverse cooking algorithm into the back end of the software. I also designed a user-friendly interface and conducted evaluations to collect user feedback. Comparing the current software against the previous milestones, it has become more intelligent and multi-functional. While it may not always produce correct answers, the average precision of the results exceeds 70%, indicating that it performs well in most scenarios.

Overall, the goal of this dissertation was to evaluate different food image recognition methods and develop a software solution that implements an optimized algorithm. Combining specific image recognition algorithms with natural language processing and a graphical user interface is a significant innovation in this field.

6.1 Project management

Working on this dissertation was a much larger undertaking than I initially imagined it would be. Finishing the project to an acceptable standard required an average of approximately 6 hours a day, which has been consistently upheld since February. While the time plan in the proposal for this project estimated development would be finished by the end of March, the actual development did not finish until the 6^{th} of April. This shows the time plan in the proposal to be flawed, as it did not adequately account for some special issues that arose during the project. In the first term, due to the significant time commitment required for the TOEFL test, remote research in a Purdue University research group, as well as applying for the graduate program, I had limited available time to devote to my dissertation project. As a result, progress was slower than anticipated.

Thanks to the waterfall methodology and agile development, it was helpful to make the whole project on track. With a Gantt chart, it makes possible for me to identify issues and adjust my time schedule rapidly.

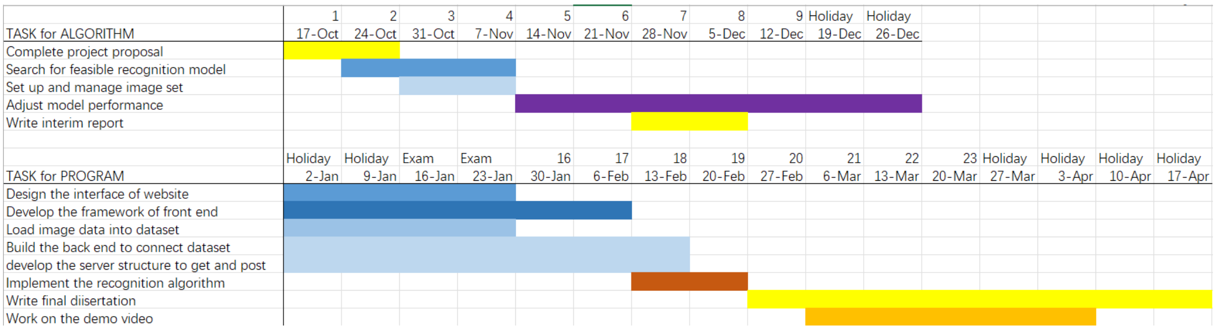


Figure 6.1: The Time Schedule in Proposal

Figure 6.1 shows the initial plan for this project. Due to the delay in evaluating the performance of the model and constructing the back-end for connecting the dataset with Spring Boot, the project timeline has been adjusted, and all software development tasks have been postponed. Consequently, implementing user login functionality has been skipped, and the development of the server structure has been rescheduled to occur earlier in the timeline to catch up with the plan, as Figure 6.2 shows.

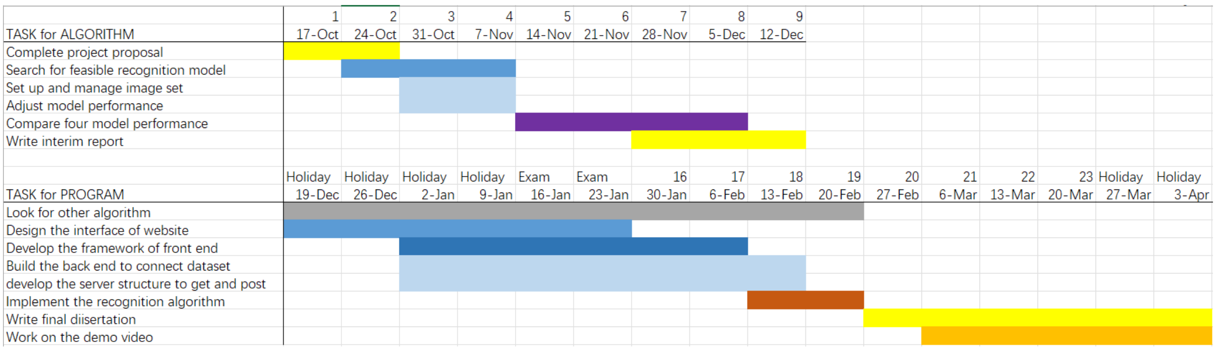


Figure 6.2: The Time Schedule in Interim

After confirming the assignment deadlines and timetable for the second term, I have updated my plan accordingly. I have allotted less time for the development of the software interface. However, I have scheduled two spare weeks before the submission deadline to allow for unexpected delays. Despite these setbacks, I am happy to report that the project is progressing well and is on track to meet my goals.

Additionally, the weekly meetings with my supervisor, Dr Andrew French, were quite beneficial. For the most part, these meetings are a way for me to provide status updates, based on my own view of the project. During the half an hour meeting period, the questions put forward allowed me to have a better understanding of the structure of the whole software, more potential improvement of software features, and to concentrate on the greater context of the project. This is particularly valuable, as I sometimes focused too much on the developing back end, leading to unnecessarily ignoring to realise the function in the interface.

6.2 Contributions and reflections

During the development of the food image recognition software, several LSEPI issues were taken into consideration. First and foremost, intellectual property rights were addressed, as the software involves the creation of source code and documents. Open source licenses were also considered while I was developing based on the project of Recipe 1M+, I have cited the reference in where I mention it, and I attached much importance to the distribution and promotion of the "Food Master" software.

Research ethics were also taken into account, particularly when it comes to the use of data subjects. Fortunately, it was not necessary to collect any human participants' data, but only some users' feedback towards the software performance. In this case, I did not take into consideration data protection legislation. Before the survey and data collection, I have already submitted all the related documents including the consent form, research ethics committee, data management plan, and info sheet and prepared the survey template to the supervisor. The project followed the University of Nottingham research ethics requirements, and the work was designed to safeguard participants and data subjects. Informed consent and voluntary participation were also ensured throughout the development of the software.

Other legislation that may have affected the software, such as the Equality Act (2010), was also taken into consideration. The broader ethical and social considerations were also addressed, particularly in terms of who will benefit from the software and who may be affected by it. The software was designed to be accessible and beneficial to all, and the potential unintended consequences and impact on societal inequalities were taken into account.

Overall, the project considered seriously various LSEPI issues, ensuring that the software was developed ethically and responsibly. The project team reflected on the potential implications of the software and made sure that it aligned with the values and principles of responsible innovation.

In terms of personal reflection, I have found this project to be both challenging and rewarding. I have learned a great deal about food image processing, front-end and back-end development, and the ethical considerations surrounding software development. I have also gained valuable experience in project management, communication, and problem-solving.

Looking at the bigger picture, I recognize the potential impact of this software on the food industry and on people's lives. It can provide a convenient and efficient way for people to discover new recipes and experiment with different types of cuisine. However, it is important to consider the potential consequences of relying too heavily on technology for food-related decisions and the need for maintaining a balance between technology and traditional cooking methods.

In conclusion, I am proud of my contributions to this project and believe that it has the potential to make a positive impact on society.

Bibliography

[1] Terasoluna server framework for java (5.x) development guideline. Accessed: 20-04-2023.

[2] Healthy living: Healthy 365, Accessed Year. Accessed on: 22-04-2023.

[3] Dji action 2 dual-screen combo - dji mobile online store (netherlands), n.d. Accessed: October 24, 2022.

[4] Bagadiya, J. 500+ social media statistics you must know in 2023, 2023. SocialPilot.

[5] Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 - mining discriminative components with random forests. In *Proceedings of the European Conference on Computer Vision* (2014), pp. 446–461.

[6] Chen, C.-F. R., Fan, Q., and Panda, R. Crossvit: Cross-attention multiscale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 357–366.

[7] Chen, Z., and Tao, Y. Food safety inspection using "from presence to classification" object-detection model. *Pattern Recognition 34*, 12 (2001), 2331–2338.

[8] Dhruv, P., and Naskar, S. Image classification using convolutional neural network (cnn) and recurrent neural network (rnn): a review. *Machine Learning and Information Processing: Proceedings of ICMLIP 2019* (2020), 367–381.

[9] Face, H. Image classification, n.d. Accessed: 16-04-2023.

[10] Hajibabaee, P., Malekzadeh, M., Heidari, M., Zad, S., Uzuner, O., and Jones, J. H. An empirical study of the graphsage and word2vec algorithms for graph multiclass classification. In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (2021), IEEE, pp. 0515–0522.

[11] Harris, M. Comment on vayda's review of good to eat: Riddles of food and culture. *Human Ecology 15*, 4 (1987), 511–517.

[12] Hou, S., Feng, Y., and Wang, Z. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *Proceedings of the International Conference on Computer Vision* (2017), pp. 541–549.

[13] Karamti, H., Tmar, M., Visani, M., Urruty, T., and Gargouri, F. Vector space model adaptation and pseudo relevance feedback for content-based image retrieval. *Multimedia Tools and Applications 77* (2018), 5475–5501.

[14] KHANNA, S. K. Food and culture: A reader, by carole counihan and penny van esterik. *Ecology of Food and Nutrition 48*, 2 (2009), 157–159.

[15] KITAMURA, K., YAMASAKI, T., AND AIZAWA, K. Food log by analyzing food images. In *Proceedings of the 16th International Conference on Multimedia* (2008), pp. 999–1000.

[16] LAHITANI, A. R., PERMANASARI, A. E., AND SETIAWAN, N. A. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management* (2016), IEEE, pp. 1–6.

[17] MANE, A., BAWANKAR, M., GORADIA, M., AND CHAUDHARI, S. Image to recipe and nutritional value generator using deep learning. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2022*. Springer, 2022, pp. 731–748.

[18] MARIN, J., BISWAS, A., OFLI, F., HYNES, N., SALVADOR, A., AYTAR, Y., WEBER, I., AND TORRALBA, A. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 43*, 1 (2021), 187–203.

[19] MIN, W., JIANG, S., LIU, L., RUI, Y., AND JAIN, R. C. A survey on food computing. *ACM Computing Surveys 52*, 5 (2019), 1–36.

[20] MITTAL, A., MOORTHY, A. K., AND BOVIK, A. C. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing 21*, 12 (2012), 4695–4708.

[21] MOROL, M. K., ROKON, M. S. J., HASAN, I. B., SAIF, A., KHAN, R. H., AND DAS, S. S. Food recipe recommendation based on ingredients detection using deep learning. In *Proceedings of the 2nd International Conference on Computing Advancements* (2022), pp. 191–198.

[22] NESTLE, M., WING, R., BIRCH, L., ET AL. Behavioral and social influences on food choice. *Nutrition Reviews 56*, 5 (2009), 50–64.

[23] PARRISH, E., AND GOKSEL, A. K. Pictorial pattern recognition applied to fruit harvesting. *Transactions of the ASAE 20*, 5 (1977), 822–827.

[24] PAULY, D. A simple method for estimating the food consumption of fish populations from growth data and food conversion experiments. *National Oceanic and Atmospheric Administration 84*, 4 (1986), 827–840.

[25] QIU, J., LO, F. P. W., SUN, Y., WANG, S., AND LO, B. Mining discriminative food regions for accurate food recognition. In *Proceedings of the British Machine Vision Conference* (2019), p. 158.

[26] SAHOO, D., HAO, W., KE, S., ET AL. Foodai: Food image recognition via deep learning for smart food logging. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 2260–2268.

[27] Salvador, A., Drozdzal, M., Giró-i Nieto, X., and Romero, A. Inverse cooking: Recipe generation from food images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10453–10462.

[28] Salvador, A., Hynes, N., Aytar, Y., et al. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 3020–3028.

[29] Sørensen, L. B., Møller, P., Flint, A., et al. Effect of sensory perception of foods on appetite and food intake: a review of studies on humans. *International Journal of Obesity 27*, 10 (2003), 1152–1166.

[30] Wang, H., Sahoo, D., Liu, C., et al. Learning cross-modal embeddings with adversarial networks for cooking recipes and food images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 11572–11581.

[31] Wang, L., Zhang, Y., and Feng, J. On the euclidean distance of images. *IEEE transactions on pattern analysis and machine intelligence 27*, 8 (2005), 1334–1339.

[32] Wankhede, K., Wukkadada, B., and Nadar, V. Just walk-out technology and its challenges: A case of amazon go. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)* (2018), IEEE, pp. 254–257.

[33] Yu, Y., Si, X., Hu, C., et al. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation 31*, 7 (2019), 1235–1270.

[34] Zhang, X., Zhou, F., Lin, Y., et al. Embedding label structures for fine-grained feature representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2016), pp. 1114–1123.

[35] Zhou, F., and Lin, Y. Fine-grained image classification by exploring bipartite-graph labels. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2016), pp. 1124–1133.

Appendix A

User Evaluation Questionnaire

Question 1: How do you like the colour scheme used in this software?
1. Excellent 2. Good 3. Neutral 4. Not so good 5. Awful

Question 1*: What another colour would you like?

____

Question 2: How do you like the font size?
1. Excellent 2. Good 3. Neutral 4. Too large 5. Too small

Question 3: What feature in this software do you think you would use the most?
1. Search recipes 2. Recognise recipe 3. Recognise ingredients 4. Other

Question 3*: Why do you think this function the most useful?

____

Question 4: Which of the following situations are you most likely to use this program?

1. Shopping in a supermarket.
2. Looking for food in the fridge.
3. Cooking in the kitchen.
4. Eating outside.
5. Other: __________________________

Question 5: How do you feel about the speed of processing?
1. Excellent 2. Good 3. Neutral 4. Not so good 5. Too slow

Question 6: How accurate do you think the results generated by the software are?
1. Over 80% 2. 80%-60% 3. 60%-40% 4. 40%-20% 5. Awful

Question 7: How would you rate the usefulness of this software?
1. 1 2. 2 3. 3 4. 4 5. 5

Question More feedback*: What else do you want to say about this software?