International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 5 Issue: 12

ISSN: 2321-8169
204 – 208

DANES: Diet and Nutrition Expert System for Meal Management and Nutrition Counseling

Prof. Megha V. Gupta
Computer Engineering,
New Horizon Institute of
Technology and Management
NHITM, Thane, India
meghavgupta@outlook.com

Ms. Priya Bhattacharjee
B.E. Student, Computer Engineering
New Horizon Institute of
Technology and Management
NHITM, Thane, India
priyabhattacharjee31@gmail.com

Ms. NavyaKotian
B.E. Student, Computer Engineering
New Horizon Institute of
Technology and Management
NHITM, Thane, India
nsk714@gmail.com

Ms. GrishmaVipat
B.E. Student, Computer Engineering
New Horizon Institute of Technology and Management
NHITM
Thane, India
grishmavipat24@gmail.com

Abstract:"Your body is your temple"
As people across the globe are becoming more health conscious, eating more healthy food and avoiding junk food, a system that can measure calories and nutrition in every day meals can be very useful for maintaining one's health. Food calorie and nutrition measurement system is very beneficial for dieticians and patients to measure and manage their daily food intake. We also know that it's difficult to find an affordable nutritionist or a dietician across the street; therefore, we have proposed a system – DIET AND NUTRITION EXPERT SYSTEM. The proposed system is a responsive android application which contains the knowledge and data regarding the fitness of a person and nutrition content values. This application consists of the user interface which will be publicly displayed on the application i.e. the basic information regarding the fitness and nutrition values such as how to maintain good health by adapting healthy eating habits which includes the intake of calories, proteins and carbohydrates etc. in proper proportion. A dietician consults a person based on his schedule, body type, height and weight. The system too asks all this data from the user and processes it. It asks about how many hours the user works, his height, weight, age etc. The system stores and processes this data and then calculates the nutrient value needed to fill up users' needs.

Keywords: *Diet, nutrition, expert system, food, health, BMI*

*****

1. Introduction

In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning about knowledge, represented mainly as if–then rules rather than through conventional procedural code. Artificial intelligence or more specifically, expert systems, have also been developed to solve either meal planning problems or health related problems.

Expert Systems (ES) are intelligent software applications that provide advice to its users through a dialog or a conversation conducted between the user and the ES application. An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities. Expert systems provide a good platform to implement applications that can be at par with human expert. Since, these days' people tend to neglect basic health care, food consumption and overall health awareness, which serve as a motivation to create an expert system in diet and nutrition.

Nutrition is 80% of our fitness goal equation. Nowadays, human beings suffer from many health problems such as fitness problem, maintaining proper diet problem, etc. The effective personal dietary guidelines are very essential for managing our health, preventing chronic diseases and the interactive diet planning helps a user to adjust the plan in an easier way. Nutrition is getting food into the body for growth and energy, and for keeping the body healthy and living. It also includes the environmental, psychological and behavioral aspects of food and eating. The aim is to provide the end user with proper health information which is at hand's reach, easily accessible and readily available. And none other than smartphones are the way to fulfill these requirements.

Smartphones are no doubt the most widely used means of communication for its ease of use, ease of handling, and increasing capabilities. And therefore, we have proposed an Expert System application to be deployed on Android. Our project is not just another fad diet plan or a calorie counter, but a lifestyle coach to help one create daily habits by

ISSN: 2321-8169
204 – 208

teaching proper nutrition. The app delivers results through tailored, easy to follow meal plans. The app is designed to give a personalized step by step guide accommodating everyone, from beginners to advanced fitness enthusiasts. The application aims to provide customized healthy meals and diet plans, foods to eat and foods to avoid.The application is to be produced on Artificial Intelligence. The user fills the registration form and then logins to the application. After login users must fill personal information including age, weight, height, gender and activity level. For calculating BMI, age, weight, height, gender and exercise level are necessary. Based on calculated BMI (Body Mass Index) Artificial Dietician will display the calorie intake for logged user. User will be given various options and varieties in their diet chart. The daily calorie intake for each user depends upon his/her BMI, gender and eating patterns. The system provides the user to enter their daily food intake using the tracker methodology. The user will be given an analysis report at the end of their diet session.The application caters to the need of all types of users from infants to senior adults, from teenage children, special diets for pregnant women to people suffering from health problems such as Blood Pressure, Diabetes, etc.

2. Literature Survey

The existing systems help us with the basic knowledge of how to implement the Diet and Nutrition Expert System. We learn from the various elaborate explanations and intend to improvise the existing system and hence come up with our proposed system. Following are the various insights gathered from different papers which have proved helpful for our literature survey.

The eating habits of Costa Rican society are alarming. Obesity rates have increased making our country one of the most obese populations in the Latin American region. Six in ten people suffer a disproportionate increase in their weight because their poor nutritional habits (CACIA 2012). The prevalence of obesity, BMI > 30, is 59 percent (Rosero 2009). Studies estimates Costa Rica will be one of the ten most obese populations in the world by 2020 (Euromonitor 2011). This reality makes it essential to raise public awareness about the need for a much-needed dietary improvement and encourage preventive care. Many people, and particularly students, cannot afford to consult a private nutritional expert (Morales 2012). Hence, public health agencies such as the Office of Welfare and Health of the University of Costa Rica face the challenge of finding alternative methods for educating the population in incorporating healthy eating habits into their daily. One problem in healthcare is the lack of availability for frequently health monitoring. Health software offers less expensive solutions reducing the physician-patient physical relation and provides monitoring solutions. Mobile Internet and the use of Web for medicine have a strong impact on health-care models that are based on the concept of anytime and anywhere connections. Mobile software applications can help facilitate the distribution of nutritional information, learn to assess their own nutritional level, and acquire better eating habits to improve their current condition. [1]

Nutrition UCR is a prototype expert system for diagnosing, controlling, and monitoring human nutrition. The system assesses the physical characteristics of the user to determine their nutritional status and makes recommendations for reaching nutritional requirements and a balanced diet, consequently generating a knowledge database with the nutritional status and dietary habits of a university population. The system generates challenges, alerts, and constantly motivates the user to use the application and improve their nutritional habits. The expert system is implemented using the JESS, Java Expert System Shell, libraries (Laboratories 2012) and the Java programming language running as a Web Service on a Linux Web Server. The prototype calculates the BMI, Body Mass Index, as in Eq. 1 (OMS 2012), the ideal weight and physical contexture, frame size (Rivas 1991) and uses dietary information from (Bermudez 2012). [1]

Developing the expert system as a standalone application on the mobile phone has the advantage of being available at any time, and any place but it has many disadvantages. First: the application will be bounded by the hardware capabilities of the mobile device which needs special care for knowledge representation, inference engine, and interface design. Second: updating the knowledgebase or the database will require reinstalling the application on the mobile device. Third: A special version should be released for each mobile platform since standalone applications are platform-dependent.[2]

Most people gain knowledge nowadays using technology including artificial intelligence technologies. Artificial Intelligence (AI) aims to develop systems which exhibit 'intelligent' human-like behavior (Anjane, 1998; Becerra-Fernandez et al., 2004). Expert systems, a type of AI technologies, encode human expertise in specific domains by using If-Then rules, and accordingly advise and provide solutions to different problems (Becerra-Fernandez et al., 2004); the five components of expert system are user interface, working memory, knowledge base, inference engine and explanation system. [3]

Neumark-Szteiner (2009) presented five main proposals for preventing obesity and related eating disorders among girls, which include eating healthily rather than following diets, adopting a positive body image, having meals with their families instead of their friends, taking part in physical activities, and involving the families of overweight teenagers when addressing weight related problems. [4]

Usually, a dietitian evaluates a client's dietary conditions and enters those into a computer-based diet construction system. Often the diet constructed by the system requires modification by the dietitian or nutritionist to meet certain integrity constraints, such as ensuring a meat portion in each lunch, juice for breakfast, etc. or simply milk when a cereal is planned for breakfast. Since menu integrity requirements are very difficult to comprehensively formalize, the currently available diet construction systems violate at times such restrictions which are rather obvious for the human user. [5]

Over the years various mathematical models, such as linear programming, have been proposed and applied to diet construction with little success to completely automate the diet construction process

2.1. Study of Existing System

There are several nutrition expert systems reported in the literature; the first one is called "The Nutrition Diet Program" (NDP) which is developed to help the rural population who can't find dietician or the medical doctor near them. This system provides a customized diet plan for patients; the system prepares this plan based on the many details provided by the user (Ramachandran et al., 1992). [3]

Another expert system is for "Nutrition Counseling and Menu Management"; this program makes menu planning and manages the eating habit (Hong and Kim, 2005). [3]

The third one is a "Nutrition Diagnosis Expert System" that utilizes "Nutritional Care Process and Model (NCPM)", which is "defined by American Dietetic Association (ADA) in 2008 and integrate the nutrition diagnosis knowledge from dietetics professionals to establish the basics of building the rule based expert system with its knowledge base" (Chen et al., 2012, p. 2132). The system is built using Microsoft Visual Studio 2008. [3]

Kahraman and Seven (2005) presented a computer system that utilized the branch-and-bound method to minimize a diet in terms of cost, while attempting to include most of a certain individual's food preferences. [4]

Frega et al (2012) developed a program that could be used to evaluate the average dietary needs in a typical Mozambican household and present a healthy diet for such a family. Although the system provided feasible solutions regarding dietary constraints and requirements, the resulting diets were not generally very affordable. [4]

Vienna expert system for parenteral nutrition of neonates (VIE-PNN) [22] is designed to perform specific task of calculating the daily changing composition of parenteral nutrition for small new-born infants. [5]

3. DANES: Your personal nutritionist

Understanding the above-mentioned quotes on significance of human body and its health, an artificially intelligent dietician expert system is being designed to monitor/ track the user's everyday diet and suggest the end user an appropriate and healthy diet plan. It aims to provide variety of food options to the user. The application must also consider the medical history of the user (as there are many people who have various diseases due to which they are not allowed to consume some types of food items. For example, a diabetic patient is not allowed to consume food items that contain high level of sugar.) The application must be able to suggest diets as per the person(s) current situation; like, for a pregnant lady, the diet will be different than a normal lady. The system will also enable the user to keep a track of the daily activities and produce a daily, weekly and a monthly report of the user.

3.1 System Analysis

DANES is application software that provides the customer with the information of the importance of a healthy lifestyle along with a dietician recommended diet. The system, first takes in the personal details of the user such as age, weight, height, medical history etc. Based on this information, the system generates a diagnosis report which consists of the BMI along with the estimated calorie goals of the user.

The medical history of the user is considered as there are many people who have various diseases due to which they are not allowed to consume some types of food items. For example, a diabetic patient is not allowed to consume food items that contain high level of sugar. Similarly, a person suffering from blood pressure should preferably consume less amount of salt. The expert system outputs (advices) are different for people with different ages and genders. From a knowledge engineer perspective, a decision table is utilized to improve building the logic in the knowledge base of the expert system. A decision table is a good way to deal with combinations of 1371 Vision 2020: Innovation, Development Sustainability, and Economic Growth things (e.g. inputs). Decision tables provide a systematic way of stating complex business rules, which is useful for developers as well as for testers.

Different decision tables are developed depending on age groups: young children (1-3 years), 4-8 years children, adolescent male (9-13 years and 14-18 years), adolescent female (9-13 years and 14-18 years), adult males (19-30, 3150, 51-70, and >70 years), and adult females (19-30, 31-50, 51-70, and >70 years). This categorization is recommended by the Omani guide to healthier eating by the Ministry of Health (2009). It is categorized in this way

 ISSN: 2321-8169
204 – 208

because every level of age need different requirement of nutrients. It differs from young to adults and from male to female as well. The developed nutrition and diet expert system, first calculates the body mass index (BMI) based on this formula: BMI = weight/height^2. By calculating the BMI, the system concludes about the "body type": whether the person is under weight, Normal weight, overweight or obese. Consequently, based on the concluded "Body type", and other inputs, the expert system then advises the identified above nutrition and diet outputs.

$$\mathrm{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$

The user is then given a chance to select the type of diet which he may prefer i.e. vegetarian, eggetarian or non – vegetarian. System then queries the knowledge base for the diet; a suitable diet is prescribed to the user. We have also included a tracker that will take daily input from the user regarding the food that he/she has consumed today, any physical activities that are done etc. The system will keep a track of all the details entered by the user and will generate a day to day report which will help the user to know if he/she is under the estimated calorie goals or not. The entire activity of the user is tracked and analyzed over the diet tenure. After the completion of the prescribed diet, the user will have to give a feedback, this feedback will be analyzed by the analyzer and an end term report will be generated. This report will contain a detailed analysis of how much weight you have lost and the estimated weight loss. If the user decides to continue with the diet plan, then he/she will be given a strict/lenient to normal diet plan according to the feedback given to him. This rule-based system captures nutrition and diet knowledge from human expert and relevant websites and then presents it in if-then statements format, and provides solutions.

3.2 Overall System Overview

The overall system overview can be viewed as 2 modules: Management and Menu Generating. The Management Module includes two processes. First is extraction of personal information which is passed to the next process of extraction of medical and dietary information. The management module has links to the patient's database and the various documents.

The Menu Generating Module has 4 components. The first component identifies the nutrient requirements, the second component generates exchange table by foods and then by meals. The fourth component suggests menu. The extraction of medical & dietary information and the identifying nutrient requirements components are linked to the Food Composition Database. The generation of exchange table's components follow a consistent set of Rules and Constraints. The suggestion menu has links to the different databases like Case Base, Food Exchange Database and Diet Plan Menu database. All the components work together to give the required output.

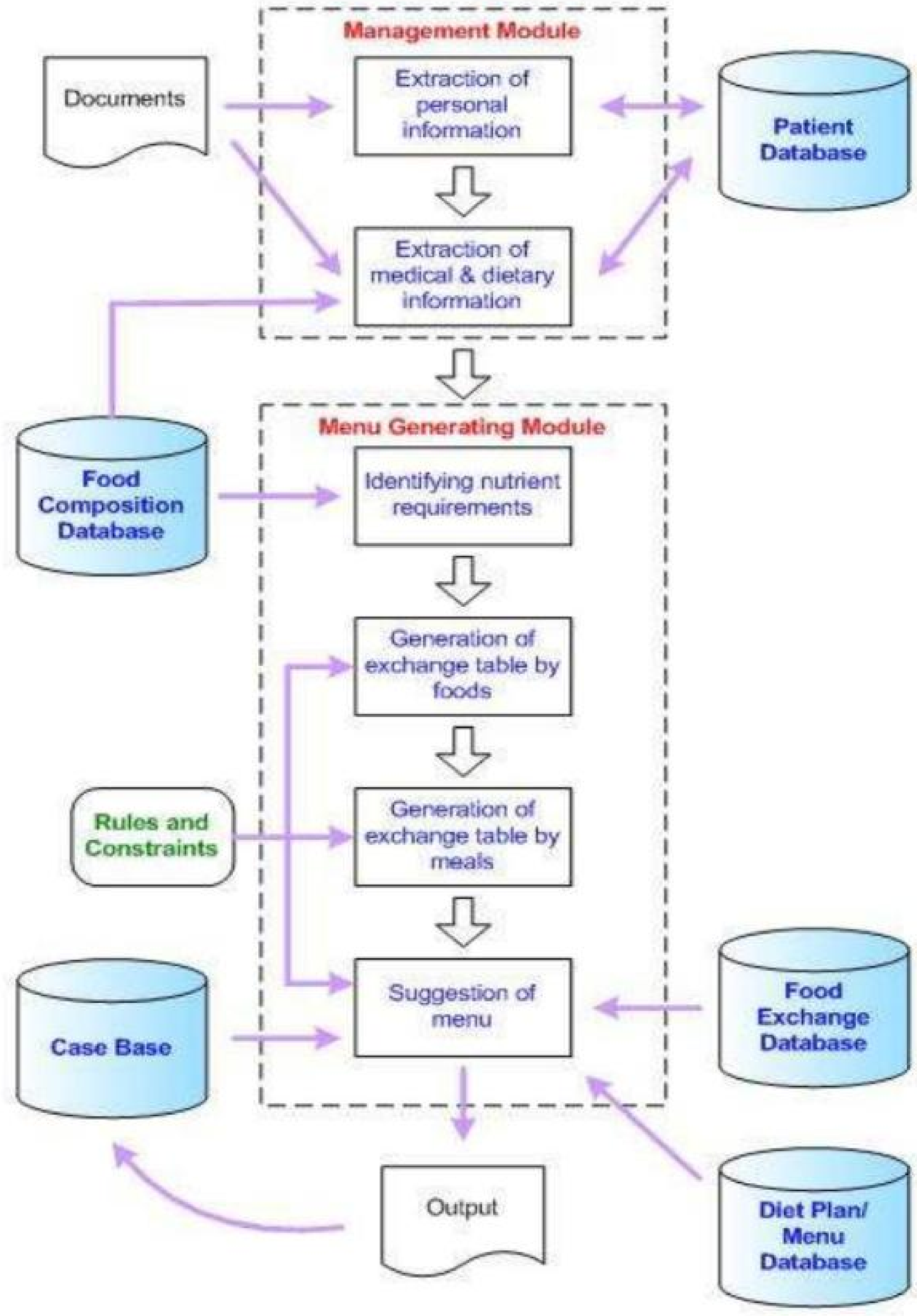

Figure 6.1: System Overview

4. Applications

The system eliminates the travelling cost in visiting a dietician and reduces the time required to get the best diet plan. As we know, dieticians are not available at every corner of the street; they are scarce and expensive. It is difficult to get access to a good dietician easily. This app overcomes that problem. It also overcomes the travelling cost to visit a dietician recurring times. It takes time for Dietician to come up with the best diet plan, this app aims to be quicker and at par with human expert. Dietitians can use this system to make sure what they recommend their patients. The system can also be utilized in gymnasium particularly for calculating the customers' calories and diet plans. Hospitals can also implement this system for recommending diets to their patients.

The diet suggested to the user will be as per his/her own BMI which ensures greater chances of reaching the goal. Individual can also use this software especially for themselves at home with step by step guidance process. This system can be very well used in medical colleges for teaching and practicing purposes so that student can learn from it. Individual can also use this software especially for themselves at home. The system can be used by people of all age categories. The application will also be useful to celebrity individuals, sports persons' and home makers to assist them in keeping track of their diet and to help them make lifestyle changes.

5. Conclusion

People these days are more concerned about their health; they are always searching ways to lead a healthy lifestyle. The use of expert systems can improve people's awareness and help them get a proper advice. Providing an expert system for Diet and Nutrition adds value to people's life especially in developing countries. The expert system will provide expertise in nutrition consulting. It will offer a wide range of advices about the quantity of various nutrients that may meet the basic needs of the body; such as proteins, vitamins, fibers, and minerals. Also, the system will help the user to decide, to increase or decrease their weight by knowing their body type. Moreover, the system will also provide the user with meal plans and the food they need to consume for their body type. In addition, the system will save time required to consult a human expert and would be easier to access the diet plan.

In conclusion, this paper illustrates the process of developing Diet and Nutrition Expert System prototype and the potential benefits of developing such system.

Acknowledgment

It is our immense pleasure to express our gratitude to Prof. Megha V. Gupta as our guide and supervisor. We express our sincere thanks to Dr. Sanjay Sharma (HOD) and all other staff members for their co-operation. We would also like to thank our Principal, Prof. P.D. Deshmukh for his guidance and providing necessary facilities during the working of this report.

References

[1] A Prototype Mobile Expert System for Nutritional Diagnosis, Christian Quesada, Marcelo Jenkins Center for ICT Research, University of Costa Rica, San Pedro, Costa Rica https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5871/6052

[2] A New Approach for Developing Diagnostic Expert Systems on Mobile Phones, Communications in Information Science and Management Engineering, Aug. 2013, Vol. 3 Iss. 8, PP. 374-384 http://www.academicpub.org/DownLoadPaper.aspx?paperid=13559

[3] Developing a Nutrition and Diet Expert System Prototype, Conference Paper, June 2013 https://www.researchgate.net/publication/244993766_Developing_a_Nutrition_and_Diet_Expert_System_Prototype

[4] A diet expert system utilizing linear programming models in a rule-based inference engine http://orlabanalytics.ca/jaor/archive/v7/n1/jaorv7n1p13.pdf

[5] Building a case-based diet recommendation system without a knowledge engineer Abdus Salam Khan* , Achim Hoffmann School of Computer Science and Engineering, The University of New South Wales, Sydney 2052, Australia Received 14 February 2002; received in revised form 12 September 2002; accepted 9 October 2002 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.3512&rep=rep1&type=pdf

IEEE TRANSACTIONS ON
SERVICES COMPUTING

A New Deep Learning-based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure

Chang Liu, Yu Cao, *Senior Member, IEEE*, Yan Luo, *Member, IEEE,* Guanling Chen, *Member, IEEE,* Vinod Vokkarane, *Senior Member, IEEE,* Yunsheng Ma, Songqing Chen, *Member, IEEE,* Peng Hou

Abstract—Literature has indicated that accurate dietary assessment is very important for assessing the effectiveness of weight loss interventions. However, most of the existing dietary assessment methods rely on memory. With the help of pervasive mobile devices and rich cloud services, it is now possible to develop new computer-aided food recognition system for accurate dietary assessment. However, enabling this future Internet of Things-based dietary assessment imposes several fundamental challenges on algorithm development and system design. In this paper, we set to address these issues from the following two aspects: (1) to develop novel deep learning-based visual food recognition algorithms to achieve the best-in-class recognition accuracy; (2) to design a food recognition system employing edge computing-based service computing paradigm to overcome some inherent problems of traditional mobile cloud computing paradigm, such as unacceptable system latency and low battery life of mobile devices. We have conducted extensive experiments with real-world data. Our results have shown that the proposed system achieved three objectives: (1) outperforming existing work in terms of food recognition accuracy; (2) reducing response time that is equivalent to the minimum of the existing approaches; and (3) lowering energy consumption which is close to the minimum of the state-of-the-art.

Index Terms— Mobile Applications, Object Recognition, Deep Learning, Edge Computing, Food Recognition

1 INTRODUCTION

In the US, more than one-third (34.9% or 78.6 millions) of adults are obese and approximately 17% (or 12.7 millions) of children and adolescents aged 2 to 19 years are obese [1]. There were more than 1.9 billion adults, 18 years and older, were overweight on earth in 2014 [2]. Documenting dietary intake accurately is crucial to help fight obesity and weight management. Unfortunately, most of the current methods for dietary assessment (for example, 24 hour dietary recall [3] and food frequency questionnaires [4]) must rely on memory to recall foods eaten.

In the last few years, we have witnessed an explosive increase of mobile and wearable computing devices (e.g., the smart watch and smart phone) in the consuming electronics market. One common characteristic of these devices is that many of them have inexpensive, unobtrusive and multimodal sensors. These sensors enable us to collect multimedia data (e.g., video and audio) in natural living environments. Due to the ubiquitous nature of mobile and wearable devices, it is now possible to use these devices to develop pervasive, automated solutions for dietary assessment [5-11]. One example of such solutions is to use mobile devices as a pervasive food journal collection tool and to employ cloud service as a data analysis platform. The combination of mobile device and cloud service could contribute to improving the accuracy of dietary assessment. As a result, in the last few years, we have seen several mobile cloud software solutions [12-14] to improve the accuracy of dietary intake estimation. One common issue among these solutions is that the users of the software must enter what they have eaten manually. To address this issue, visual-based food recognition algorithms and systems have been proposed [6-11]. A recent review by Martin et al. [15] also indicated that using digital imaging techniques for food recognition is superior to many other methods of dietary assessment techniques. Some advantages of visual-based food recognition systems include: reduced burden for users to recall the food, improved accuracy and efficiency of dietary recall.

While promising, one of the major barriers of adopting automatic dietary assessment system into practice is how to design and develop effective and efficient algorithms and system to derive the food information (e.g., food type) from food images. Considering the limited computation resources and low battery life on mobile device, it is more challenging to develop such a system within the mobile

- *Chang Liu, Yu Cao, Guanling Chen, Peng Hou are with the University of Massachusetts, Lowell, MA, 01854. E-mail: ycao@cs.uml.edu.*
- *Yan Luo and Vinod Vokkarane are with the Department of Electrical and Computer Engineering, University of Massachusetts, Lowell, MA, 01854.*
- *Yunsheng Ma is with the Department of Medicine, University of Massachusetts Medical School, Worcester, MA, 01601.*
- *Songqong Chen is with the Department of Computer Science, George Mason University, Fairfax, VA, 22030.*

******Please provide a complete mailing address for each author, as this is the address the 10 complimentary reprints of your paper will be sent***

Please note that all acknowledgments should be placed at the end of the paper, before the bibliography ***(note that corresponding authorship is not noted in affiliation box, but in acknowledgment section).***

Digital Object Identifier 10.1109/TSC.2017.2662008

cloud computing paradigm. We have carefully investigated this problem and have identified two major challenges. The first major challenge is how to design effective and efficient analytics algorithms to achieve optimal recognition accuracy. The second major challenge is how to develop a system that can minimize energy consumption and response time.

To address the first issue (recognition accuracy), we plan to develop new deep learning-based algorithms. Deep learning [16, 17] (also known as representation learning, feature learning, deep structured learning, or hierarchical learning) is a new area of machine learning research. It allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [18]. In the last five years, these techniques have improved the state-of-the-art in speech recognition, computer vision, natural language processing, and many other domains. Our extensive experiments in this paper have shown that, compared with traditional hand engineered features (e.g., SIFT [19]) and shallow learning-based classification algorithms (e.g., Support Vector Machine (SVM)), our proposed deep learning-based classification algorithms could improve the recognition accuracy substantially. We also developed other image analysis algorithms to enhance the food image quality for data analysis. All these algorithms have been integrated into an edge computing-based real-time computing system, which is discussed in the next paragraph.

To address the second issue (energy consumption and response time), we aim to design and employ a real-time food recognition system employing edge computing service paradigm. The proposed system distributes the data analytics throughout the network by splitting the food recognition task between the edge devices (close to end users) and the servers (in the cloud). Edge computing refers to the enabling technologies that allow computation to be performed at the edge of the network in a stream fashion. Edge computing is a non-trivial extension of cloud computing from the core network to the edge network [20-26]. The proposed edge computing service infrastructure is particularly useful for our application because most of the mobile devices have limited computation capacity and battery life. Hence, it is difficult for them to support computational-intensive tasks. At the same time, our proposed food image analysis algorithms usually involve heavy computation and may require much more computation resources.

In this paper, we focus on two major research efforts. The first research effort aims to develop new food recognition algorithms, including new food image recognition algorithms based on deep learning and image pre-processing and segmentation algorithms to enhance the quality of food image. The second research effort aims to design a real-time food recognition system for dietary assessment. The proposed system employs edge computing service paradigm and distributes the data analytics throughout the network. Specifically, the proposed system will split the food recognition tasks between the edge devices (which is physically close to the user) and the server (which is usually located in the remote cloud). For example, in our system, the edge devices (e.g., user's smart phone) can perform light-weight computation on food image for food recognition. Then, our system will transmit the food images (after the light-weight computation at edge device) to the server in the cloud to perform more accurate recognition tasks. By distributing the analytics throughout the network, our system can achieve significant improvement in the recognition accuracy, while minimizing the response time and energy consumption. In this project, we implemented a prototype system to verify our hypothesis and evaluate the proposed algorithms. Our prototype runs on both edge device (Xiaomi Note, running Android 6.0.1 "marshmallow") and server (an in-house GPU cluster). We also conducted extensive experiments with real-world data. The results show that our system achieves very impressive results on the following three aspects. First, to the best of our knowledge, the food recognition accuracy using our proposed approach outperformed all other reported results. Second, the response time of the proposed system is equivalent to the minimum of the existing approaches. Last but not the least, the energy consumption of the proposed system is close to the minimum of the state-of-the-art.

The rest of the paper is organized as follows. In Section 2, we introduce related work in computer-aided dietary assessment, visual-based food recognition, deep learning, and edge computing. In Section 3, we present the architecture, components, and algorithms for the proposed system based on deep learning and edge computing. In Section 4, we describe the implementation details of our system. Section 5 presents the evaluation results, which include recognition accuracy, power consumption, response time, etc. Section 6 discusses the system limitations. In Section 7, we make concluding remarks.

2 Related Work

Estimating dietary intake accurately with a high-quality food journal is crucial for managing weight loss [27]. Unfortunately, due to many technical barriers, how to improve the accuracy of dietary intake estimation is still an open question. In this paper, we aim to develop a systematic approach as a first step to address this issue. We envision that there are four most relevant research areas, listed as below.

The first related research area is to enhance the accuracy of diet assessment with computer-aided solutions. Due to the recent advances in electronics, it is now possible to develop computer-aided solutions to transform healthcare from reactive and hospital-centered to preventive, proactive, evidence-based, person-centered. Dietary assessment is one such area that has gained a lot of attentions from both academia and industry. Among thousands of existing mobile cloud health software and hardware, we have seen many of them (e.g., MyFitnessPal [12], MyNetDiary [13], and FatSecret [14]) are dedicated for improving the accuracy of dietary estimates. However, all these applications require the user to enter everything they ate manually. To address this issue, several applications have been developed to improve the level of automation. For example, a recent App entitled "Meal Snap" [28] aims to reduce human efforts by asking the user to take a picture, enter some quick information such as whether user is eating breakfast or lunch, and add a quick text annotation if the user wants to. Unfortunately, the accuracy of calorie estimation is heavily dependent on the accuracy of the manually entered text from user. Therefore, the accuracy is very unstable. Another example of such application is named "Eatly" [29]. This application requires the user to take the food image and then rates the food into one of the three categories ("very healthy", "it's O.K.", and "unhealthy"). However, the actual rating is performed manually by the community, which consists of the users of this App. In this paper, we propose new algorithms and system that can recognize the food images (caputred by the user with their mobile devices) automatically. This automation reduces the user's burden substantially.

The second related research area is to perform dietary analysis using food images and / or videos. In one paper [6], researchers proposed an approach to combine a learning method (manifold ranking-based techniques) and a statistics method (co-occurrence statistics between food items) to recognize multiple food items. In another study[7], the authors proposed a method for fast food detection by researching the relative spatial relationships of local features of the ingredients and a feature fusion technique. NIH also funded a project named "Technology Assisted Dietary Assessment (TADA)" [11]. Researchers under this project have investigated different aspects of computer-aided dietary assessment, such as food item recognition, mobile interface design, and data development for food images. They have published several papers on food image recognition [8-10]. Most of the existing visual-based food recognition algorithms employed traditional signal processing with hand-engineered features (e.g., SIFT [19], HOG [30]) and shallow machine learning algorithms (e.g., SVM). Only very recently, with the striking success of deep learning, people started to research the application of deep learning for food image recognition [31]. Deep learning has the potential to address one main issue associated with existing techniques, which is that the hand engineered features may be useful for screening a few categories of food item but are unable to generalize to other food types. The proposed approach in this paper is also based on recent advances in deep learning. Related work in deep learning is introduced in the next paragraph.

The third related field is deep learning, which is a branch of machine learning. It allows the computers to learn from experience and understand the world in terms of a hierarchy of concepts using a deep graph with multiple processing layers. Each concept is defined in terms of its relation to simpler concepts [32]. Essentially, deep learning is trying to solve the central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations [32]. It has already been proven useful in many disciplines, such as computer vision, speech recognition, natural language processing, bioinformatics, etc. There are two main classes of deep learning techniques. The first class is purely supervised learning algorithms, such as Deep Convolutional Neural Network (CNN). The second class is unsupervised and semi-supervised learning algorithms, such as Denoising Auto-encoders and Deep Boltzmann Machines. In this paper, we focus on deep Convolutional Neural Network (CNN) [33]. Our proposed approach is rooted from CNN and it belongs to the category of supervised learning algorithms. CNNs are biologically-inspired [34] (animal visual cortex) variants of Multilayer Perceptrons (MLPs). It is consisted of neurons that have learnable weights and biases. Compared with MLPs, CNN has several distinct features. First, by enforcing a local connectivity pattern between neurons of adjacent layers, CNN could exploit spatially local correlation. Second, each filter in CNN is replicated across the entire visual field, which share the same parameters (e.g., weight vector and bias). Third, the neurons are arranged in three dimensions (width, height, and depth). Furthermore, a feature map can be generated by repeated application of a function across sub-regions of the whole image. Early implementation of CNNs, such as LeNet-5 [35], has been successfully applied to hand writing digital recognition. However, due to the lack of large scale labeled data and limited computation power, CNNs failed to address more complex problems. With the help of large-scale and well-annotated dataset like ImageNet [36], new computing hardware such as graphics processing unit (GPU), and several algorithms advancements such as Dropout [37], it is now possible to train large scale CNNs for complex problems. Recently, many research, such as VGGNet [38], ZFNet[39], GoogLeNet[40], Residual Network [41], has been proposed to address the issue of limited abilities of feature representation. One common strategy is to make the network deeper and avoid saturation issues. Our proposed approach was directly inspired by CNN work from LeNet-5 [42], AlexNet [33], and GoogLeNet [40]. The LetNet-5 [42] is a 7-layer network structure with 32x32 grey-scale image as input for hand written digital recognition. It includes three convolutional layers (C1, C3 and C5), two sub-sampling layers marked as (S2, S4), one fully connected layers (F6), and one output layer. LeNet-5 generates a feature map and feed the feature map into the two fully-connected layers. After that, a 10-class output is generated. A receptive field (a.k.a. "fixed-size patch" or "kernel") is chosen during the convolutional layer to compute convolution with the same size patch in the input image. A stride is defined to make sure every pixel in the original image will be covered. The system will perform convolution operation first, followed with a sub-sampling with the feature map. The goal of sub-sampling is for dimension reduction. Then, we will move to the fully connected layers, which are used to join the multi-dimension feature maps. Finally, we will generate a ten-class output, each of which represents one digital (from zero to nine). Please note, at each layer, the parameters (e.g., weight vector and bias) are trainable. Recent progresses in CNN have focused on enhancing the object representation with more complex models. For example, AlexNet [33] is a seven-layer model which includes five convolution layers and two fully connected layers. It outperformed the state-of-the-art object recognition techniques in 2012 ImageNet [36] challenges with large margin (over 10%). Later on, we witnessed many new models with increased layers, increased layer size, more complex neurons, as well as sophisticated computation units and layer structures. Dropout and ReLU were proposed to address the issue of overfitting and saturation, respectively. Some excellent examples include VGG net [38], ZFNet[39], GoogLeNet[40], Residual Network [41]).

The last (but not the least) related research area is edge computing service infrastructure [22-26]. Under this infrastructure, part of the data processing tasks may be pushed to the edge of the network. One of the core ideas is called "Collaborative Edge" [22], which refers to the architecture that connects the edges of multiple stakeholders. These stakeholders may be geographically distributed and they may have distinct physical location and network structure. Under this infrastructure, the cloud paradigm is extended to the edge of the network. Therefore, such an edge computing service infrastructure offers a unifying paradigm for cloud-based computing and Internet of Things (IoT)-based computing. It has the potential to address the issues of delayed response time, reduced battery life, limited bandwidth, and data security and privacy. However, most of the existing use cases of edge computing-based digital health applications [43-45] are relatively simple examples with small data sets. Novel user cases and intriguing applications with more challenging tasks, such as larger data sets and sophisticated computation, are needed for evaluating the efficacy and effectiveness of edge computing in digital health. Our proposed application, which focuses on food image recognition for dietary assessment, employ very complicated computation tasks (e.g., image pre-processing, image segmentation, and deep learning) with large image data sets (in the size of GB). This application scenario provides an excellent playground to evaluate the efficacy and effectiveness of edge computing in digital health.

3 System Design

3.1 Overview

Our food recognition system employs visual sensors to capture food images as the source data. Due to the recent advances of electronics, visual sensors are now available in many Internet-of-Things(IoT) devices, such as smart phones. To simplify the design, we utilized the camera on smartphones for visual sensing. Besides the smartphone for sensing and image capturing, the recognition is done in a collaborative manner between the edge device (e.g., smartphone) and servers (e.g., servers in the cloud). As shown in **Figure *1***, our system includes end user layer (left most of **Figure 1**), edge layer (middle of **Figure 1**), and cloud layer (right most of **Figure 1**), together form a three-layer service delivery model. In our proposed system, data and computation are kept close to end users at the edge of network. Also, the end user's device can passively record the geological location. Hence, the system could provide low latency, reduced energy consumption, and location-awareness for end users. The computations are distributed throughout the network, including both the edge devices and servers in the cloud. Please note, in our system, the recognition is done in a collaborative manner.

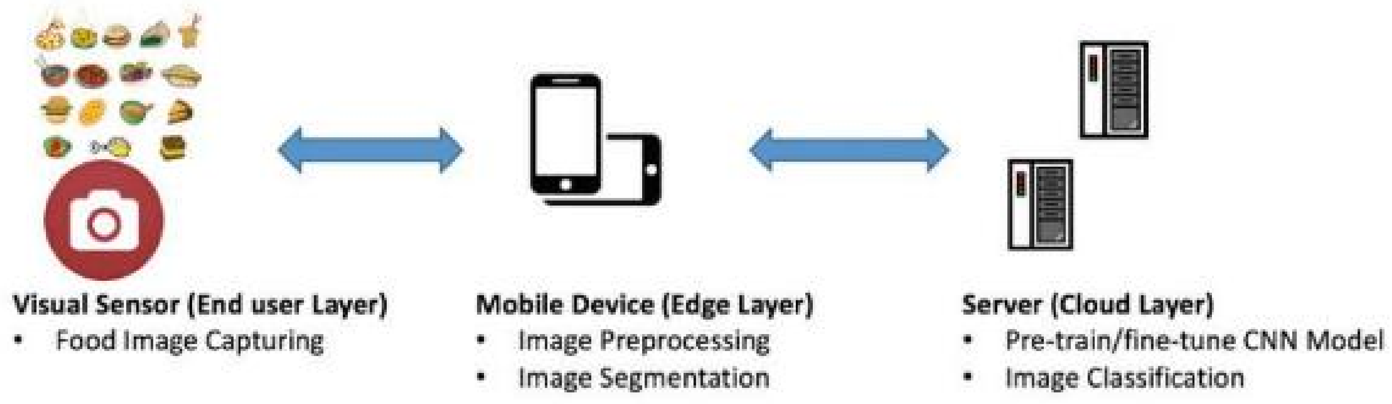

Figure 1: Overview architecture of our proposed "*Deep Food on the Edge*" system

The system design and related components are shown in **Figure 2**. As shown in this figure, our system consists of the following three major modules:

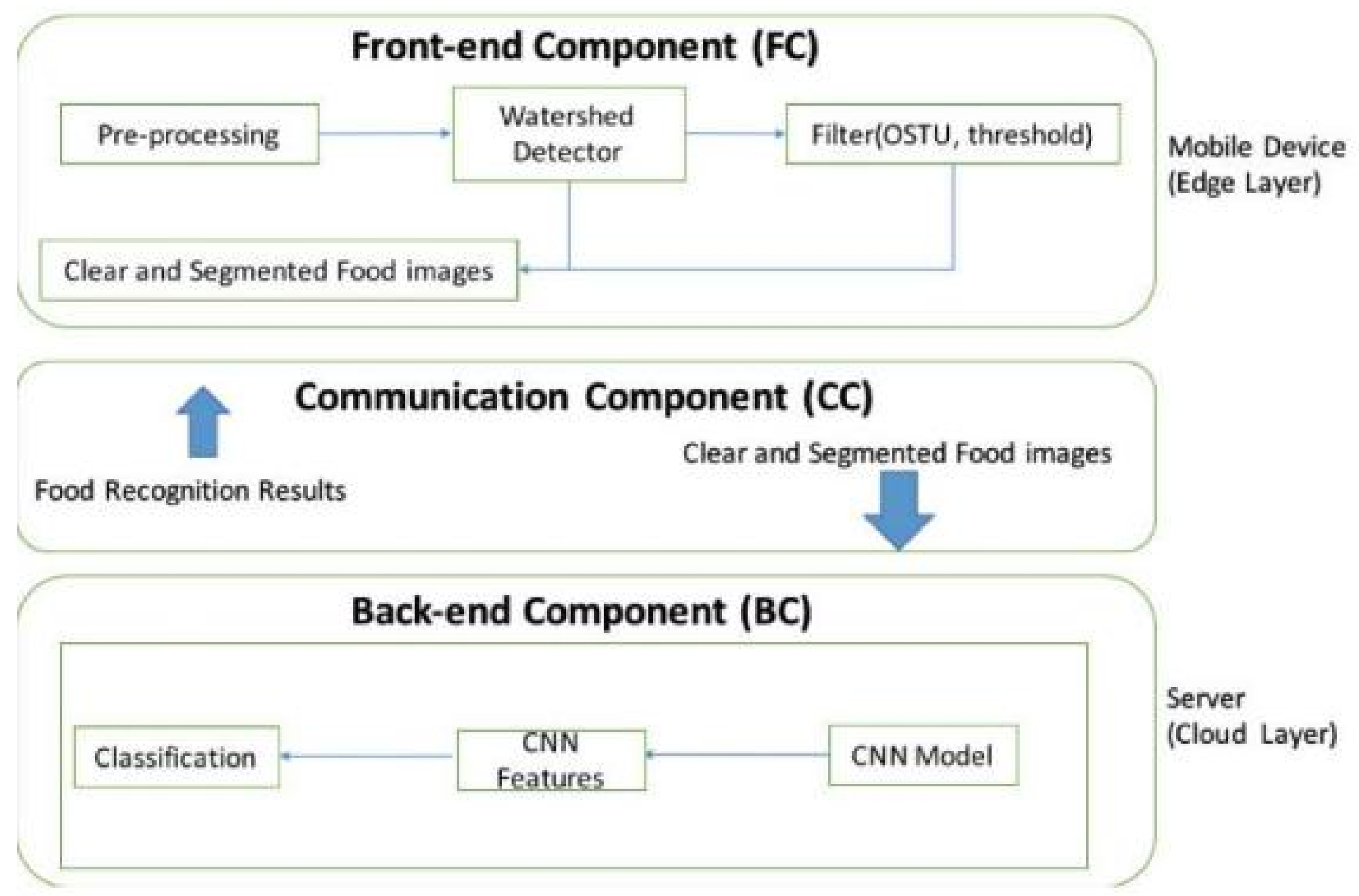

Figure 2. System design and components

Front-end Component (FC): we deploy the FC module on the edge device (smartphone). As shown in the top box in **Figure 2**, it's consisted of three submodules, which are image pre-processing (e.g., blurry image detection), watershed detectors, and the filters (OTSU or threshold)-based segmentation. After the image pre-processing module, an original clear image is generated for segmentation. Next, the watershed detector, combined with different filters (e.g., OTSU-based threshold) to segment the original image. After segmentation, we can generate the clear and segmented image. These images will be transferred to the server via the Communication Module (introduced below) for further classification.

Communication Component (CC): CC provides two channels for communication between the FC and the Back-end Component (BC), which will be introduced in more detail in the next paragraph. It transfers the image data from the FC to the BC via Input Channel, and it also passes the detection results from the BC to the FC via Output Channel.

Back-end Component (BC): the BC module runs on the cloud server, which is configured to use Caffe [46] (an open source deep learning framework) for CNN model training and testing. We use pre-trained GoogLeNet by ImageNet and fine-tune it on our food dataset (UEC-256 and Food-101). Then the trained model is deployed on the server and used for classifying the image. More specifically, the segmented image is first passed through our proposed CNN model (which is rooted from GoogLeNet model [40]), then the features are generated from the model, furthermore, a softmax classifier is used with these features to generate the probability of each category. Here we use the top-5 and top-1 probability as our prediction/classification of the food image. Our evaluation of accuracy is also based on these criteria.

3.2 Food Recognition Algorithms

In this section, we will introduce our proposed food recognition algorithms, which runs on the FC and BC. Essentially, our system is a multiple-stage food recognition system that distributes the analytics throughout the network.

3.2.1 Food Image Analysis Algorithms Running at FC

Once the food images are captured, we will conduct two types of computations at mobile device in the Front-end Component (FC) (a.k.a., Edge Layer): image pre-processing and image segmentation.

The main objective of the first computation (image pre-processing) is to identify if the image being captured is blurry or not. While many cameras on mobile devices have features such as optical zoom or auto focusing, in real-world practice, when users take the pictures of food, they may have very limited time to do so due to their busy schedule and their photo taking action may be interrupted by other matters. Hence, the chances of device shaking and other interruptions while taking pictures are high. An automatic image blurry detection algorithm running at the mobile device is needed to give a real-time alarm to reminder user to re-take the picture if the image is blurry. We define an out-of-focus image as blurry image. Our goal is to develop a light weight and effective blurry image detection algorithms running at the mobile device. In literature, image restoration has been proposed to handle blurry images. Unfortunately, these existing methods are not applicable to our case because these techniques need a reference image to compute the quality of the test image. In our applications, we may only have test images. Followed our previous research [47], we propose a simple-feature(such as "edginess" of the image) and threshold-based method to divide the images captured into two groups (i.e., the clear image group and the blurry image group). The "edginess" of the image is defined as the number of edge pixels (e.g., detected by Sobel operator) divided by the total number of image pixels. The rationale behind this method is that the percentage of edge pixels for clear image (with clear object of interests) is much higher than the percentage of edge pixels for blurry image. In our previous research [47], we also noticed that there are different patterns between the frequency spectrums of clear image and blurry image. The Fourier spectrum of a blurry image usually shows prominent components along the certain degree (e.g., 45 degree) directions that correspond to the corners of the image. This is because the blurry image usually does not contain clear object information except the four strong edges at the corners of the image running at certain degree relative to the sides. On the other hand, the clear image usually has a lot of clear edge information so that its spectrum does not show prominent components along certain degree directions because it has a wider range of bandwidth from low to high frequencies. Based on the aforementioned observation, we first employ texture analysis algorithms on the frequency spectrum image. Then we extract different types of texture features (e.g., entropy, contrast, correlation, homogeneity) from each image. Once the features are extracted, we employ different types of classifiers to classify the images into two categories (blurry image or clear image). Similar to our previous work [47], we employ a two-step K-means clustering algorithms, the details is illustrated in the **Algorithm 1**.

Data: A set of Images $Set : \{I_1, I_2, ..., I_n\}$
Result: Two clusers $Set_b : \{b_1, b_2, ..., b_p\}$, $Set_c : \{c_1, c_2, ..., c_q\}$
initialization, set i to 1;
while *i is no more than n* **do**
    Read one image I_i;
    Extract texture features T_i from frequency spectrum;
    Apply entropy feature extraction from T_i as S_1;
    Apply contrast feature extraction from T_i as S_2;
    Apply correlation feature extraction from T_i as S_3;
    Apply homogeneity feature extraction from T_i as S_4;
    Use binary classifier for S_1, S_2, S_3, S_4 separately;
    Combine classification result using majority vote;
    if *blurry* **then**
        group I_i into the Set_b;
    else
        group I_i into the Set_c;
    end
    go to next iteration;
end

Algorithm 1. Image Pre-processing in the Front-end Component(FC)

The main objective of the second computation (image segmentation) is to segment the image into two parts: foreground (which contains the actual food) and background. Based on the size of foreground, we could crop the image by removing some portion of the background that does not overlap with foreground. According to our own experiments and other people research results, when using deep learning-based model (which is the main algorithms used in server) for image analysis and object detection, if we could reduce the background information, the object detection and recognition accuracy could be improved. Inspired by this observation, we employ watershed [48] segmentation algorithm to preprocessing the image at FC. In this process, we first pre-process the image by image segmentation. Then we generate a new cropped image and send the updated image to the server in the cloud for further processing. By doing so, we can achieve the following performance improvements: (1) the volume of data transfer over the network may be reduced substantially. It also reduces the power consumption caused by network transferring; (2) The response time may be reduced by shorter transmission time, which will improve the user experiences; (3) The system uses much less network flow consumption, which is very helpful when the network connection is unreliable, or when the user is connected to the server via cellular network and/or he or she has limited data plan with the mobile device; (4) More importantly, the cropped image will eliminate the abundant information and further improve the accuracy for classification. In theory, the watershed algorithm is based on the following observations: any grayscale image can be viewed as a topographic surface, in which the high intensity indicates peaks and hills while low intensity represents valleys. The watershed algorithm starts filling every isolated valley with different colored water. When water rises, water from different valleys with different colors will start to merge. We could avoid this by building barriers in the locations where water merges. The algorithm continues to fill water and build barriers until all the peaks are under water. Finally, the barriers the system created are the segmentation result. **Algorithm 2** illustrates the details of the algorithm.

Algorithm 2 Watershed Algorithm using topographical distance

INPUT: The lower complete grey scale Image (V, E, im), which is defined from original image with a lower boundary
OUTPUT: a sequence of labels on V, representing background or foreground

WATERSHED $\leftarrow 0$ ▷ (The label of watershed for every pixel)
Init Label with a minima and $MASK$ for other pixels
U $\leftarrow \{p \in V | \exists q \in N_G(p)$: im[p] $\neq$ im[q]$\}$
while not $empty(U)$ **do**
  select point p from U with minimal grey value
  remove p from U
  for all q steeper than p **do** ▷ (pixel value is greater in the neighbour)
    if $label[q] == MASK$ **then**
      $label[q] \leftarrow label[p]$
    else
      $label[q] \leftarrow$ WATERSHED
    end if
  end for
end while ▷ (Label array represents the boundary)

Algorithm 2. Image Segmentation in the Front-end Component(FC)

3.2.2 CNN-based Food Image Analysis Algorithms Running at BC

After the image pre-processing and segmentation at FC, we will further analyze these images at BC. Our proposed approach running at BC is based on the recent advances on deep learning, which aims to learn multiple levels of representation and abstraction that help infer knowledge from data such as images, videos, audio, and text.

Our proposed approach was directly inspired by CNN work from LeNet-5 [42], AlexNet [33], and GoogLeNet [40], and it employs a new module called "Inception Module", which is motivated by recent advances named "Network-in-Network" [49]. This is also similar to the one used in GoogLeNet [40]. In this "Inception Module", an additional 1x1 convolutional layers are added to the original AlexNet [33] network architecture. This additional layer undoubtedly increases the depth of the network. However, this addition could also substantially reduce the feature map's dimension. Therefore, this module could help to remove the computation bottlenecks. Specifically, we use feature map as the input for the "Inception Module". After that, we apply multiple levels of convolutional layers and max-pooling layers. The kernel size of the convolutional layer varies from 1x1 to 3x3 and 5x5. At each layer, different outputs are generated and are concatenated to form the new feature map, which is used as input for the convolution and pooling operation for next layer. In order to perform dimension deduction, an optimized convolution is proposed based on the "Inception Module". Please note, instead of feeding the input directly into the convolutional layer, an additional convolutional layer with size 1x1 is added to reduce the input dimension. In addition, the output from the 3x3 max-pooling layer is sent into an additional convolutional layer with size 1x1. These new designs enable the new architecture to reduced dimension even the depth of the network is increased. Not surprisingly, our experiments have demonstrated that, even under constrained computational complexity, this new network structure is able to enhance the ability to capture more visual information. **Figure 3** illustrates the improved inception module. The network structure in the left (Figure a) is the original structure in regular CNN, such as AlexNet [33]. The right figure (Figure b) is the snapshot of the new network architecture with "Inception Module". As shown in Figure b, the three added 1x1 convolutional layers are annotated with dotted rectangle and green color. While the number of layers in Figure b is four (which is one layer more than the number of layers in Figure a), the total dimension of the output (at feature concatenate layer) in Figure b is still

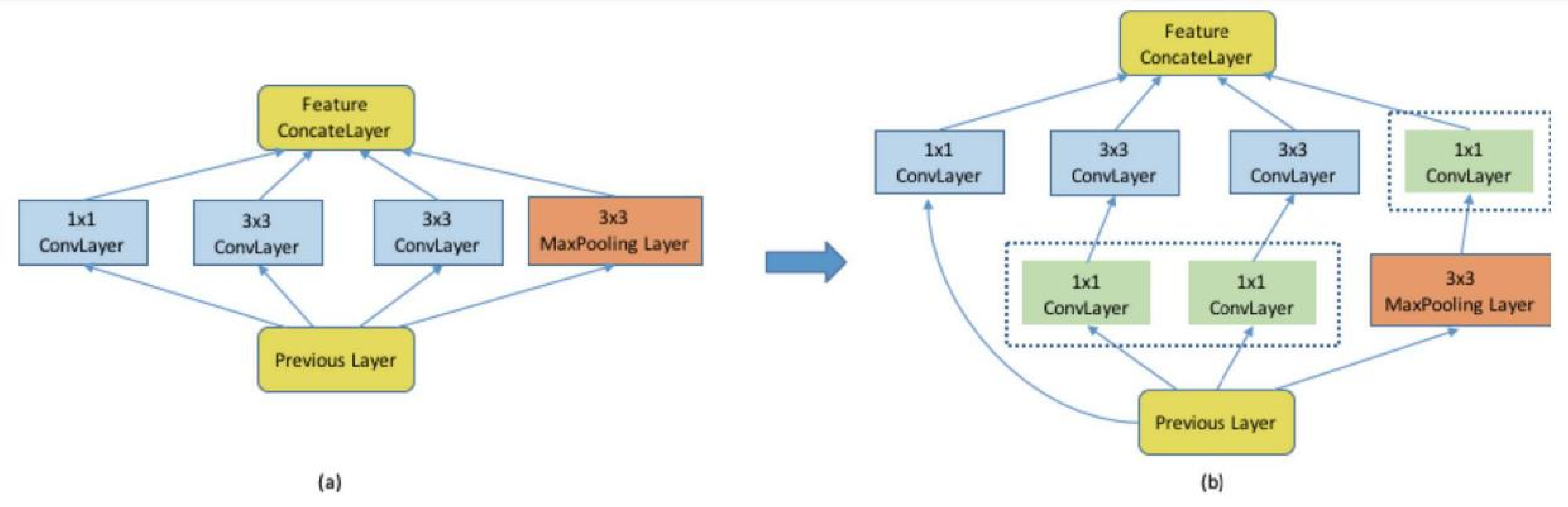

Figure 3: Illustration of the "Inception Module". Figure (a) in the left is the snapshot of the original network architecture in regular CNN, such as AlexNet. Figure (b) in the right is the snapshot of the new network architecture with "Inception Module". The figure is best viewed in color.

smaller than the output dimension of Figure a.

The next step after forming the "Inception Module" is to employ multiple modules to form the network (similar to GoogLeNet). In this step, we will connect the two modules using one additional max pooling layer. The output from the previous module will be used as the input for the next module. Specifically, the concatenated features (output) from the previous module are fed into the newly added max pooling layer. The output from the max pooling layer is used as input for next module. **Figure 4** illustrate this architecture. This figure includes two "Inception Module", one (figure "a") is located on the top of **Figure 4** and another (figure "b") is located in the bottom of the **Figure 4**. These two components (figure "a" and figure "b") are connected via a 3x3 max-pooling layer. Essentially, the new network architecture becomes a hierarchical level step by step. In order to address the issue of increased time complexity associated with the increased network layers, we resort to the lessons learned in recent paper [50], which offer some insights for designing the network architectures by balancing factors such as depth, numbers of filters, filter sizes, etc. In this study, we design a network structure with 22 layers (similar to the one used in GoogLeNet) with different kernel size and stride. We have found that, in our study, using an input size of 224x224 with three channels (RGB), combined with "1x1", "3x3" and "5x5" convolutions, produces the best results. The 22 layers are layers with parameters. We design the pooling layer whose filter size is 5x5. The convolutional layer is 1x1 and includes 128 filters and ReLU (rectified linear activation). The dimension of the fully-connected layers is 1024. During pre-train-

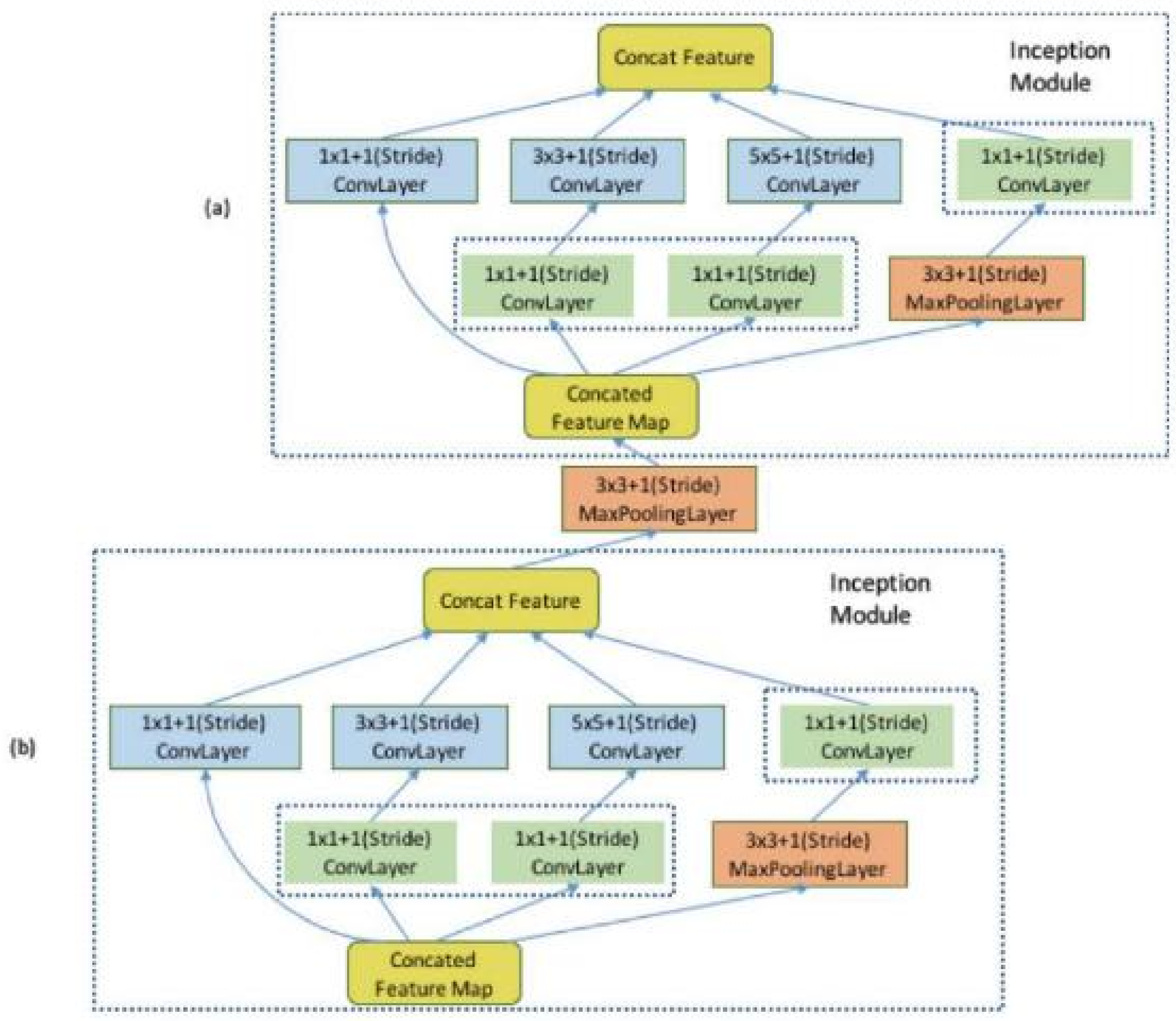

Figure 4. Illustration of module connection (best viewed in color)

ing stage, it is mapped into a 1,000-class output, similar to the ImageNet data set [36]. We use a 70% dropout rate to address the overfitting issue. Softmax is used for final classifier. Please note, based on the actual food categories, we will need to adjust the output class number during the fine-tuning stage. The proposed approach is implemented on top of open source deep learning framework Caffe [46]. CentOS 7.0 is chosen as our host system. We also use NVidia Tesla K40 GPUs for model training. The model definition is adjusted in prototxt file in Caffe. We will introduce the implementation details in Section 4.

4 System Implementation

In order to verify the efficacy and effectiveness of the proposed system, we implemented a prototype system for food recognition. Specifically, the front-end component (FC) is implemented on Android 6.0.1 (Marshmallow). The back-end component (BC) is implemented using server equipped with CentOS 7.0. The implementation of communication component (CC) includes two part. The first part is on the smartphone where we use Apache HttpClient to communicate with server. The second part is on the server we employed Django web development framework [51] and the associated RESTful web service. In this section, we present implementation details.

4.1 Implementation of Front-end Component (FC)

We develop an Android application for the front-end module. It runs on Xiaomi Note running Android 6.0.1 marshmallow. The image pre-processing algorithm, the watershed segmentation algorithm, and the threshold filter are also implemented in this application. The watershed algorithm runs on the local mobile devices and it is implemented using OpenCV [52] on Android devices. Several pre-defined markers are first constructed, the algorithm treats each pixel as a local topography, and then it fills the basins from the markers, until the basins meet on watershed lines. Here we set the markers as the local minimal of the food image, so that we can start from the bottom to fill the basins. We use OpenCV 3.10 and port the java SDK into the android studio project, which supports the OpenCV for Android SDK and also involves the image processing class.

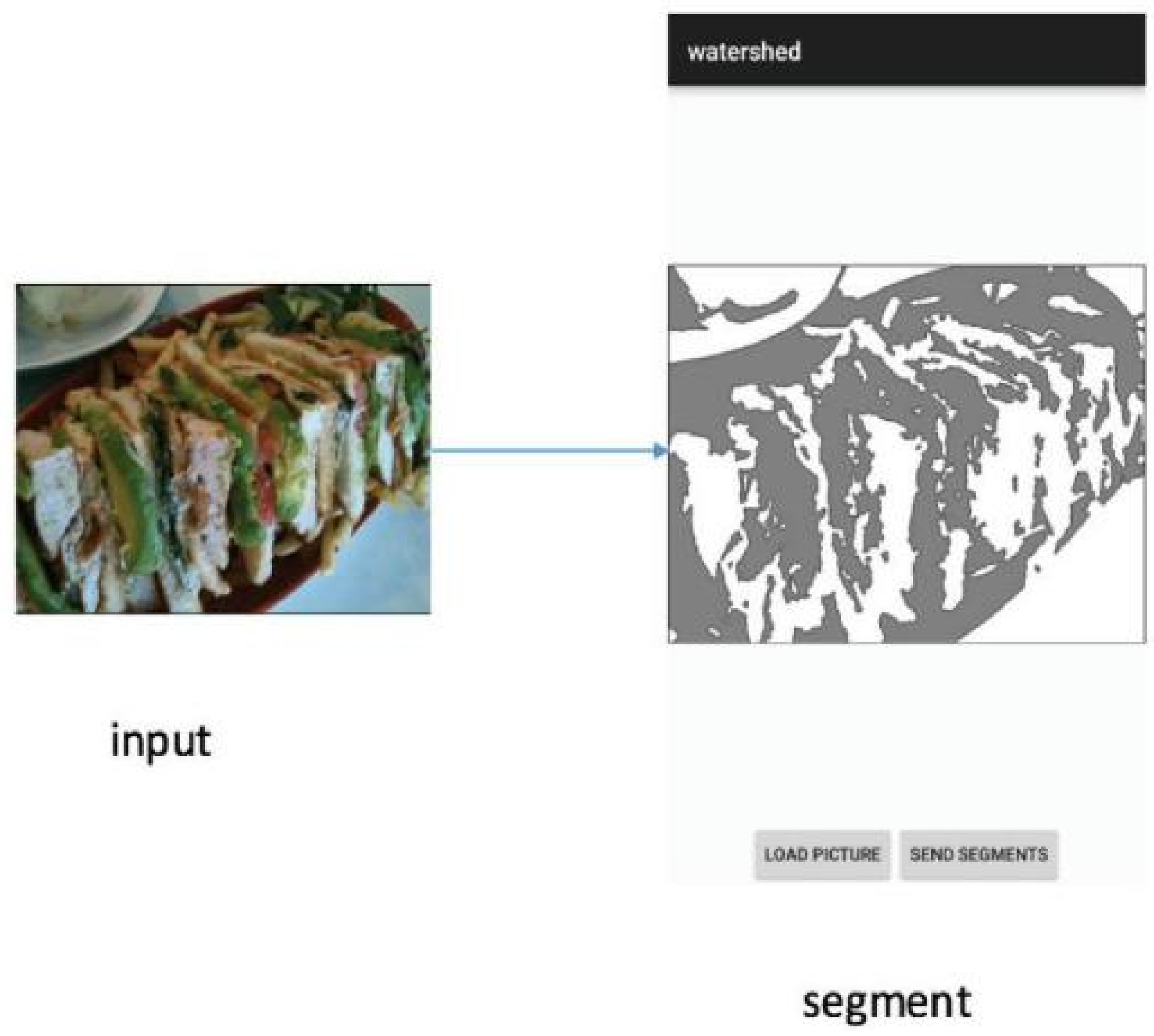

Figure 5: Screenshots showing image segmentation implementation in FC module

The App we implemented has an UI for processing and loading the image. A screenshot of the UI is shown in **Figure 5**. There is a background thread for preprocessing the image. After it finishes, the App will display the segmented image in the application's mainframe. While in the background, the thread does several tasks when preprocessing the image, that includes: (1) rescaling the image if it's exceed 1024x786, since too large image will increase the computing time and energy consumption; (2) converting the RGB image to grey level image for further image processing, the grey image is more easily computed when there're many channels and pixels; (3) creating the watershed class and watershed threshold for dividing the image into segments and non-segments; (4) saving and generating a unified image segments for future transferring.

4.2 Implementation of Communication Component (CC)

There are two implementations for communication between the Android device and cloud server. For the Android application, we use Apache HTTP Client and construct the HTTP POST request to send the segmented image into the cloud server. First, a connection bound to the server is established, and then we construct the necessary HTTP header, and fill the content with image file. Then we send the POST request to the cloud server to finish the transmission. On the cloud server, we deploy a RESTful web server using Django [53], which supports the file transferring (image, audio, video) using HTTP requests. When the server is up and deployed, it will listen to the port and save the requested file into the pre-configured destination. Our server will store all the necessary segments for the classification task using trained-well CNN models.

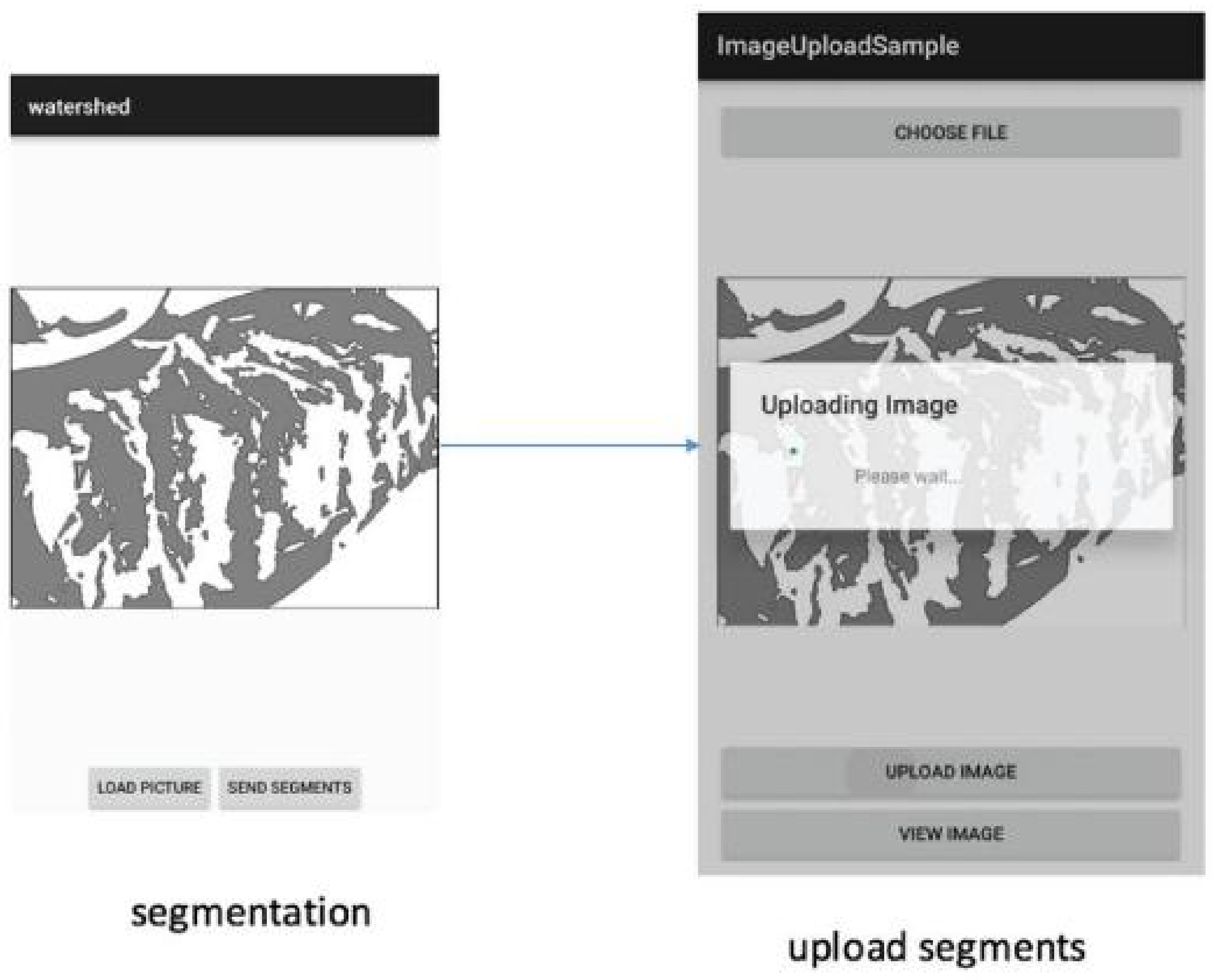

Figure 6: Screenshot showing segmented images being uploaded to the server in CM module

4.3 Implementation of Back-end Component (BC)

Our back-end system is mainly used for classification when we receive the images from the mobile device. Before testing, we used pre-trained GoogLeNet model from ImageNet, and then fine-tuned on public food data set like Food-101 and UEC-100/UEC-256. After these steps, a fine-grained model is generated which can be used for specifically food image classification. We use Caffe to train and tune the model. And our deployment of model is also based on Caffe's python interface. We first load the model into memory, when the test food image is fed into the Convolutional neural network as the input, CNN features are extracted, with max-pooling and rectified linear-unit (ReLU) layers for dimension reduction and accelerating the convergence of computing.

5 Performance Evaluation

5.1. Experiment Setup and Evaluation Data Set

In all the following experiments, we use Xiaomi Note running Android 6.0.1 "Marshmallow" as the front-end to install the FC of our system. This smartphone uses Qualcomm MSM8974AC Snapdragon 801 featuring Quad-core 2.5 GHz Krait 400 and an Adreno 330 GPU. It also has a 64 GB of internal storage and 3 GB of RAM. In the back-end, we use an in-house GPU server. This server is a SuperServer 4027GR-TR from SuperMicron. It has two Intel Xeon processor E5-2600 with 512GB RAM. This server is also equipped with four NVIDIA Tesla K40 GPU.

In order to evaluate the effectiveness and efficiency of our system, we implemented two other systems running the state-of-the-art visual-based food recognition algorithms for comparisons. The first one, entitled as C-System, employs different types of computer vision algorithms using hand engineered features (e.g., SIFT [19], SURF [54], HOG [55], Cascade [56]) running at the mobile device for food image recognition, without relying on any algorithms running in the server. These algorithms (e.g., the Cascade algorithm) have been used in many embedded computer vision systems. We also implement the second system, called D-System, for comparisons. The D-system mainly relies on using the state-of-the-art deep learning algorithms running in the server, without using any image analysis and/or pre-processing computation at mobile device. Both systems are evaluated against our proposed system, and the performance metrics we use include response time, energy consumption, and detection accuracy.

In our experiment, we use two publicly available and challenging data sets, which are UEC-256/UEC-100 [57] and Food-101 [58]. As shown in the sub-sections below, the results of our proposed approach outperformed all the existing techniques in terms of accuracy. At the same time, the response time and energy consumption of our system are close to the minimum of the existing approaches.

5.2 Experimental Results on UEC-256/UEC-100 Dataset

As we have introduced before, we employ two data sets for our experiments. We will introduce our experimental results for the first category in this section, which is UEC dataset [57]. It was first developed by DeepFoodCam project [59] and the majority of the food items in UEC dataset is Asian food. This data set includes two sub-data sets: UEC-100 and UEC-256. The first sub-data set (UEC-100) includes 100 food categories with a total of 8643 images. There are around 90 images in each category. The second sub-data set (UEC-256) includes 256 categories with a total of 28375 images. There are around 110 images in each category. The researchers have added correct annotations for each image, including food category and bounding box (used to indicate the positions of the food). We use UEC-256 as the baseline dataset since we prefer to have large scale training data. We divided the images into 5 groups (5 folds). 3 groups (out of 5 groups) were used for training and the rest of the images were used for testing.

In our experiments, the publicly available, 1000-class category from ImageNet dataset was used as the pre-trained model. This model (pre-trained model) was generated by training over 1.2 million images and testing over 100,000 images. Once we have the pre-trained model, we fine-tuned this model with the UEC-256 dataset. We fine-tuned the model with a base-learning rate of 0.01, a momentum of 0.9 and 100,000 iterations. The results are shown below in **Table 1**. If we compare the results in **Table 1** with the results in our previous publication [60], we could make two discoveries. First, our detection accuracy in this paper is slightly better. Second, the number of iterations when we reach the best performance is less than our previous paper. These two discoveries indicate that, due to the adaption of the proposed new system and algorithms, both the accuracy and the time complexity have been slightly reduced.

Table 1: Comparison of accuracy on UEC-256 at different iterations using UEC-256

# of Iterations	Top-1 accuracy	Top-5 accuracy
4,000	46.0%	77.5%
24,000	51.0%	78.8%
56,000	51.3%	79.6%
84,000	53.3%	80.7%
92,000	**54.5%**	**81.8%**

We also compared our results with both the C-System and the D-System. As we introduced before, the D-system is employing different sophisticated deep learning-based food image recognition algorithms, including the algorithms from the DeepFoodCam papers [57, 59]. To make a fair comparison, we used the same dataset as original papers, which is UEC-100, as well as the same strategy of dividing image dataset, the result is shown in the **Table 2**. Please note, there are five "C-system" in this table because we tried different types of computer vision algorithms using hand engineered features. Each sub-category of "C-system" (the first five rows in **Table 2**) represents one type of hand engineered feature. From this table, we can tell that our proposed method outperformed all existing methods using the same dataset:

Table 2: Comparison of accuracy between our proposed approach and existing approaches using the same data set (UEC-100)

Method	Top-1	Top-5
C-System (SURF-BoF+ColorHistogram)	42.0%	68.3%
C-System (HOG Patch-FV+Color Patch-FV)	49.7%	77.6%
C-System (HOG Patch-FV+Color Patch-FV(flip))	51.9%	79.2%
C-System (MKL)	51.6%	76.8%
C-System (Extended HOG Patch-FV+Color Patch-FV(flip))	59.6%	82.9%
D-System (DeepFoodCam(ft))	72.26%	92.0%
Proposed Approach in this paper	**77.5%**	**95.2%**

Table 3 shows the corresponding energy consumption of the three systems upon each food image. This table shows that the energy consumption of our system is very close to the energy consumption of the both C-system and D-system. Please note, in **Table 3**, we computed the energy consumption for both image analysis (on mobile device) and the image transferring (from the mobile device to the server). However, we did not compute the energy consumption if the computation is performed at the server in the cloud. Therefore, the D-system's energy consumption for image analysis is zero because D-system does not include any computation on mobile device. On the other hand, the energy consumption for image transferring for C-system is zero. Because in C-system, there is no need for data uploading since all the recognition tasks have been done on the mobile device.

Table 3: Energy consumption (Joule) from different systems

Method	Energy Consumption (Joule) Per Image for Image Analysis	Energy Consumption (Joule) Per Image for Image Transferring
C-System	1.01	0
D-System	0	0.98
Proposed Approach in this paper	**0.51**	**0.57**

As of the computation and response time, let's first discuss the computing time. Indeed, our algorithms is based on deep learning and training a large deep learning model requires a large amount of time. For example, on a NVidia Tesla K40 GPU, it takes 2 to 3 seconds per image for forward-backward pass using our proposed architecture. Since large dataset like ImageNet and Microsoft COCO [61] contains so many images, it may not be wise to train the model from scratch. One practical strategy is to use the pre-trained model in model zoo from existing implementation (e.g., Caffe [46]), which is public for all researchers. In our own experiment, the training time is largely impacted by the computation capacity of the server (e.g., the types of CPU and GPU), how large the image candidate is, how many iterations we choose, and what value we choose for learning rate, etc. According to the rough estimation, if we use the pre-trained GoogLeNet model, then fine-tune on the UEC-100, UEC-256, Food-101 dataset, it roughly takes 2 to 3 days nonstop for a server equipped with Nvidia K40 GPU to train the model. Once the model is trained, we can directly apply the model for classifying the image. On average, it takes 50 seconds for recognition for one image. Therefore, the average response time (the time duration between capturing the image and getting the food recognition results) is 1 minute per image for our proposed approach, which include time for image pre-processing on mobile device, the time to uploading the image to server, and the time for recognition in the server. As a comparison, the response time for C-system is usually around 35 to 55 seconds (depends on what hand engineered features we use). For example, the average computation time for a SIFT-like feature extraction and analysis algorithm on a mobile device (Xiaomi Note) is 50 seconds. On the other hand, in the D-system, the response time (the time duration between capturing the image and getting the food recognition results) is 70 seconds per image in our experiments. This is mainly because in D-system, the image being processed is the raw image without pre-processing. Hence, we could conclude that the response time of our proposed approach is very close to the minimal response time of existing approach.

5.2 Experimental Results on Food-101

In addition to the first data set (UEC data set), we use the second data set, Food-101 data set [58], in our experiment. This dataset includes a total of 101 categories. For each food category, there are around 1000 images. We used around three-quarters (75%) of these images for training and the rest of the images are used for testing. Altogether, there are over 100,000 images in this data set. One thing about this data set is that this data set does not provide any bounding box information (which can be used to indicate the food location in the image). Instead, this data set offers food type information for each image. Different from the UEC data set, most of the images in this data set are popular western food images.

For this data set, we used a similar implementation as the one used in Section 4.1. The parameters were adjusted to fit for this new data set. We used a base learning rate of 0.01, a momentum of 0.9. Similar to the methods we used in Section 4.1, we fine-tuned the model on Food-101 dataset. **Table 4** below shows the accuracy (both top-1 accuracy and top-5 accuracy are listed as below). Again, if we compare the results in **Table 4** with the results in our previous publication [60], we can find that, due to the new system and algorithms in this paper, both the accuracy and the time complexity have been slightly reduced.

Table 4: Comparison of accuracy on Food-101 at different iteration

# of Iterations	Top-1 accuracy	Top-5 accuracy
5,000	65.6%	88.7%
10,000	70.7%	91.2%
20,000	73.4%	92.6%
60,000	**77.0%**	**94.0%**

We also compared our experimental results with the results of both the C-System and the D-System using the same data set (Food-101 datasets). As shown in **Table 5**, our proposed method is better than all existing work using the same dataset and division.

Table 5: Comparison of accuracy using different method on Food-101

Method	top-1	top-5
C-System (RFDC-based Approach from Lukas et.al[58])	50.76%	NA
D-System (CNN-based Approach from Lukas et.al[58])	56.40%	NA
Proposed Approach in this paper	**77.0%**	**94%**

From the above table, we can see that pre-trained model with domain specific fine-tuning can boost the classification accuracy significantly. And fine-tuning strategy improves the accuracy comparing with non-fine-tuning method. The "NA" value in the "top-5" column means "not available", as we used the original experiment data from their paper[58], and they don't provide the top-5 result in it.

As of the energy consumption and response time, we have similar results reported as our previous data set (UEC-256/UEC-100), as introduced in the last paragraph of Section 5.1. Due to the space limitation, we did not report the exact numbers here.

5.3 The Employment of Bounding Box

As shown in both Section 5.1 and Section 5.2, the detection accuracy of our proposed approach is better than all existing approaches. We believe that one of the reasons we could achieve such performance boost is because in our proposed approach, image pre-processing and image segmentation are performed at the mobile device before analyzing these images in the server. To verify this hypothesis, we conducted a simple experiment. Our goal is to demonstrate that even very simple pre-processing can help improve the recognition performance. For example, we can use a simple bounding-box strategy to reduce the image size without analyzing the image content fully.

Specifically, we first employed the bounding box to crop the raw image. After this processing, only the food image part is remained for training and testing. Then, we performed similar experiment on UEC-256 dataset.

Table 6: Comparison of accuracy of proposed approach using bounding box on UEC-256

Method	top-1	top-5
Proposed approach (no bounding box)	53.7%	80.7%
Proposed approach (with bounding box)	**63.6%**	**87.0%**

We also conduct the experiment on UEC-100, as follows:

Table 7: Comparison of accuracy of proposed approach using bounding box on UEC-100

Method	top-1	top-5
Proposed approach (no bounding box)	54.8%	81.4%
Proposed approach (with bounding box)	**76.3%**	**94.6%**

As we can see from the two tables (**Table 6** and **Table 7**), the employment of bounding box could boost the classification accuracy substantially. A simple explanation for this is that the abundant information in the raw image is removed after the images were cropped using bounding-box. Therefore, a more accurate and clear image candidate for training can be generated. Please note, these results are valid only if we assume the majority of food image have the foreground centered on the image. Using this simple cropping-based approach will not work well if the food is scattered on different parts of the image. In this case, our proposed approach, which conducts image pre-processing and image segmentation based on the image content, is certainly necessary to improve the recognition accuracy.

6 DISCUSSIONS

Our findings indicated that our system achieves very high detection accuracy, as shown in previous sections. However, the response time, while very close the minimal of existing systems, is still around 5% slower than the best performer. While this is not surprising since deep learning-based algorithms are usually very time-consuming, we believe that more research should be devoted to further improving the speed. In particularly, we plan to investigate new deep learning algorithms that can be executed in mobile devices. There are some recent papers that have started to explore this area with some preliminary results [62, 63], which further motivate us to pursue this route in the future. While pushing the deep learning-based computation further to the edge device sounds like a good idea in the initial look, we will have to consider the energy consumption if we execute the deep learning algorithms at the edge device. We believe much more research is needed in the area of distributed deep learning-based analytics in the era of edge computing.

7 CONLUSION

In this paper, we aimed to develop a practical deep learning based food recognition system for dietary assessment within the edge computing service infrastructure. The key technique innovation in this paper includes: the new deep learning-based food image recognition algorithms and the proposed real-time food recognition system employing edge computing service paradigm. Our experimental results on two challenging data sets using our proposed approach have demonstrated that our system has achieved the three major objectives: (1) it outperforms the results from all existing approaches in terms of recognition accuracy; (2) it develops a real-time system whose response time is close to the minimal of existing techniques; and (3) it saves the energy by keep the energy consumption equivalent to the minimum of the existing approaches. In the future, we plan to continue improving performance of the algorithms (in terms of detection accuracy) and system (in terms of response time and energy consumption). We also plan to integrate our system into a real-world mobile devices and edge/cloud computing-based system to enhance the accuracy of current measurements of dietary caloric intake estimate. As our research is related to the biomedical field, much larger data sets are needed to provide convincing evidence to verify the efficacy and effectiveness of our proposed system. Backed by several major federal grants from NSF and NIH, we are in the process of collaborating with UMass Medical School and the University of Tennessee, College of Medicine to deploy our system in the real-world clinical practice.

ACKNOWLEDGMENT

The paper described is supported in partial by National Science Foundation (NSF) of the United States (Award No. 1547428, 1541434, 1440737, and 1229213). Points of view or opinions in this document are those of the authors and do not represent the official position or policies of the U.S. NSF.

REFERENCES

[1] C. L. Ogden, M. D. Carroll, B. K. Kit, and K. M. Flegal, "Prevalence of childhood and adult obesity in the United States, 2011-2012," *Jama*, vol. 311, pp. 806-814, 2014.

[2] (2016). *World Health Organization: fact sheets on obesity and overweight, Available at* http://www.who.int/mediacentre/factsheets/fs311/en/.

[3] G. H. Beaton, J. Milner, P. Corey, V. McGuire, M. Cousins, E. Stewart, *et al.*, "Sources of variance in 24-hour dietary recall data: implications for nutrition study design and interpretation," *American Journal of Clinical Nutrition*, vol. 32, 1979.

[4] J. Cade, R. Thompson, V. Burley, and D. Warm, "Development, validation and utilisation of food-frequency questionnaires–a review," *Public health nutrition*, vol. 5, pp. 567-587, 2002.

[5] R. Steele, "An overview of the state of the art of automated capture of dietary intake information," *Critical Reviews in Food Science and Nutrition*, 2013.

[6] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 2017-2020.

[7] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2249-2256.

[8] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, *et al.*, "The use of mobile devices in aiding dietary assessment and evaluation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, pp. 756-766, 2010.

[9] B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. M. Zhu, M. Bosch, E. J. Delp, *et al.*, "Novel technologies for assessing dietary intake: evaluating the usability of a mobile telephone food record among adults and adolescents," *Journal of medical Internet research*, vol. 14, 2012.

[10] C. Xu, Y. He, N. Khannan, A. Parra, C. Boushey, and E. Delp, "Image-based food volume estimation," in *Proceedings of the 5th international workshop on Multimedia for cooking & eating activities*, 2013, pp. 75-80.

[11] *TADA: Technology Assisted Dietary Assessment at Purdue University, West Lafayette, Indiana, USA, available at* http://www.tadaproject.org/.

[12] *MyFitnessPal.com: Free Calorie Counter, Diet & Exercise Tracker, available at* http://www.myfitnesspal.com/.

[13] *MyNetDiary: the easiest and smartest free calorie counter and free food diary for iPhone, iPad, Android, and BlackBerry applications, available at* http://www.mynetdiary.com/.

[14] *FatSecret: All Things Food and Diet, available at* http://www.fatsecret.com/.

[15] C. K. Martin, T. Nicklas, B. Gunturk, J. B. Correa, H. R. Allen, and C. Champagne, "Measuring food intake with digital

photography," *J Hum Nutr Diet*, vol. 27 Suppl 1, pp. 72-81, Jan 2014.

[16] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, pp. 1-127, 2009.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, pp. 1527-1554, 2006.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, pp. 91-110, November 2004.

[20] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, pp. 14-23, 2009.

[21] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and Applications," in *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*, 2015, pp. 73-78.

[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, 2016.

[23] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," 2014, pp. 1-9.

[24] W. Shi and S. Dustdar, "The Promise of Edge Computing," *IEEE Computer Magazine*, vol. 29, pp. 78-81, 2016.

[25] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, 2015, pp. 37-42.

[26] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, *et al.*, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 37-42, 2015.

[27] R. R. Wing and S. Phelan, "Long-term weight loss maintenance," *The American journal of clinical nutrition*, vol. 82, pp. 222S-225S, 2005.

[28] *Meal Snap: Magical Meal Logging for iPhone, available at* http://mealsnap.com/.

[29] *Eatly: Eat Smart (Snap a photo of your meal and get health ratings), available at* https://itunes.apple.com/us/app/eatly-eat-smart-snap-photo/id661113749.

[30] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005, pp. 886-893.

[31] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, *et al.*, "Im2Calories: towards an automated mobile vision food diary," 2015, pp. 1233-1241.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*: MIT Press, 2016.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012, p. 4.

[34] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, pp. 106-154, 1962.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248-255.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer vision–ECCV 2014*, ed: Springer, 2014, pp. 818-833.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, "Going Deeper With Convolutions," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* vol. 86, pp. 2278-2324, 1998.

[43] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare," *Sensors & Transducers*, vol. 185, p. 121, 2015.

[44] Y. Cao, S. Chen, P. Hou, and D. Buhl-Brown, "FAST: A Fog Computing Assisted Distributed Analytics System to Monitor Fall for Stroke Mitigation," in *Prof. of 10th IEEE International Conference on Networking, Architecture, and Storage (NAS 2015) (Best Paper Award)*, Boston, MA, U.S.A, 2015.

[45] Y. Cao, C. Liu, B. Liu, M. J. Brunette, N. Zhang, T. Sun, *et al.*, "Improving Tuberculosis Diagnostics using Deep Learning and Mobile Health Technologies among Resource-poor and Marginalized Communities," in *IEEE Conference on Connected Health: Applications, Systems and Engineering Technologies*, 2016.

[46] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675-678.

[47] J. Oh, S. Hwang, Y. Cao, W. Tavanapong, D. Liu, J. Wong, *et al.*, "Measuring objective quality of colonoscopy," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 2190-2196, September 2009.

[48] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*: Prentice Hall, 2008.

[49] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[50] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353-5360.

[51] A. Holovaty and J. Kaplan-Moss, *The definitive guide to Django: Web development done right*: Apress, 2009.

[52] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*: " O'Reilly Media, Inc.", 2008.

[53] *Django: The Web framework for perfectionists with deadlines, available at* https://www.djangoproject.com/.

[54] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for

human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on,* 2005, pp. 886-893.

[56] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai Marriott, Hawaii, USA, 2001, pp. 511-518.

[57] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications,* pp. 1-25, 2015.

[58] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Computer Vision–ECCV 2014,* ed: Springer, 2014, pp. 446-461.

[59] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 589-593.

[60] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "DeepFood: Deep Learning-based Food Image Recognition for Computer-aided Dietary Assessment," in *Prof. of The 14th International Conference on Smart homes and Health telematics (ICOST 2016)*, Wuhan, China, 2016.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014,* ed: Springer, 2014, pp. 740-755.

[62] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized Convolutional Neural Networks for Mobile Devices," *arXiv preprint arXiv:1512.06473,* 2015.

[63] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, *et al.*, "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," *arXiv preprint arXiv:1512.01274,* 2015.

applied sciences

Article

Precision Nutrient Management Using Arti cial Intelligence Based on Digital Data Collection Framework

Hsiu-An Lee [1], Tzu-Ting Huang [2], Lo-Hsien Yen [2], Pin-Hua Wu [3], Kuan-Wen Chen [1], Hsin-Hua Kung [1], Chen-Yi Liu [2,*] and Chien-Yeh Hsu [2,4,*]

1 National Health Research Institutes-The National Institute of Cancer Research, Tainan 704, Taiwan; hsiuanlee@nhri.edu.tw (H.-A.L.); hank9134277698@nhri.edu.tw (K.-W.C.); hsinhuakung@nhri.edu.tw (H.-H.K.)
2 Department of Information Management, National Taipei University of Nursing and Health Sciences, Taipei 112, Taiwan; 102216003@ntunhs.edu.tw (T.-T.H.); 102216013@ntunhs.edu.tw (L.-H.Y.)
3 College of Health Technology, National Taipei University of Nursing and Health Sciences, Taipei 112, Taiwan; 102016002@ntunhs.edu.tw
4 Master Program in Global Health and Development, College of Public Health, Taipei Medical University, Taipei 110, Taiwan
* Correspondence: 102216016@ntunhs.edu.tw (C.-Y.L.); cyhsu@ntunhs.edu.tw (C.-Y.H.)

Abstract: (1) Background: Nutritional intake is fundamental to human growth and health, and the intake of different types of nutrients and micronutrients can affect health. The content of the diet affects the occurrence of disease, with the incidence of many diseases increasing each year while the age group at which they occur is gradually decreasing. (2) Methods: An arti cial intelligence model for precision nutritional analysis allows the user to enter the name and serving size of a dish to assess a total of 24 nutrients. A total of two AI models, including semantic and nutritional analysis models, were integrated into the Precision Nutritional Analysis. A total of ve different algorithms were used to identify the most similar recipes and to determine differences in text using cosine similarity. (3) Results: This study developed two models to form a precision nutrient analysis model. The 2013 2016 Taiwan National Nutrition Health Status Change Survey (NNHS) was used for model veri cation. The model's accuracy was determined by comparing the results of the model with the NNHS. The results show that the AI model has very little error and can signi cantly improve the ef ciency of the analysis. (4) Conclusions: This study proposed an Intelligence Precision Nutrient Analysis Model based on a digital data collection framework, where the nutrient intake was analyzed by entering dietary recall data. The AI model can be used as a reference for nutrition surveys and personal nutrition analysis.

Keywords: nutrition survey; precision diet analysis; medical intelligence

check for updates

Citation: Lee, H.-A.; Huang, T.-T.; Yen, L.-H.; Wu, P.-H.; Chen, K.-W.; Kung, H.-H.; Liu, C.-Y.; Hsu, C.-Y. Precision Nutrient Management Using Arti cial Intelligence Based on Digital Data Collection Framework. Appl. Sci. 2022, 12, 4167. https://doi.org/10.3390/app12094167

Academic Editors: Neil Yuwen Yen, Chen-Kun Tsung, Vinod Kumar Verma and Chao-Tung Yang

Received: 17 February 2022
Accepted: 19 April 2022
Published: 20 April 2022

1. Introduction

Nutritional intake is the basis for human growth and health, and the intake of different types of nutrients and micronutrients can affect health. Most diseases are inextricably linked to diet. Diabetes, cardiovascular diseases (hypertension, hyperlipidemia), gout, peptic ulcers, and gastroenteritis are all diet-related diseases that are increasing in prevalence every year, while the age group of those suffering from these diseases is gradually decreasing. The development of the Internet has made it possible to conduct online nutrition surveys through large-scale food and nutrition databases linked to automated dietary records, and there are now a growing number of software, platforms, and applications for nutrition surveys [1].

The most common technologies used for dietary recording are web-based or online tools, mobile apps, camera-based image analysis tools, wearable sensors, etc., while traditional methods rely on the use of Food Frequency Questionnaires (FFQs) or 24 h dietary recording methods. However, past techniques have suffered from a lack of accuracy in recording, as recall methods may not accurately record the food consumed or have dif culty estimating portion sizes or limited food ingredient lists [2].

The coding and translation of food records from nutrition surveys into nutrient analyses are labor-intensive and time-consuming, meaning that it is more dif cult to collect detailed information regarding food intake in large scale population studies. Such studies rely on answers to food frequency questionnaires, and the accuracy of this data is dependent on the expertise of the interviewer compared to other self-reported measures [3,4].

Innovative technological tools have evolved with the development of various IT technologies, including natural language analysis of text, speech analysis, and image processing. The popularity of smartphones, tablets, and computers has increased the acceptance of using IT for nutritional intake assessments [5 8].

This study develops an arti cial intelligence model for a precision nutrient analysis, which allows users to enter the name of a dish and serving size to assess a total of 24 nutrients. The recipes can be modi ed by the user, which allows the model to be used in all countries and all contexts, thus improving interoperability and accuracy of the analysis.

2. Related Works

The Food Record, the 24 h dietary recall (24HR), and the Food Frequency Questionnaires (FFQs) are three common methods of collecting nutritional data. The Food Record is a comprehensive record of all foods, beverages, and nutritional supplements consumed by the respondent over a speci ed period of time. Usually, 3 4 days of intake are recorded, as the quality (accuracy) of the record is reduced due to the burden of recording too many days. Ideally, dietary intake should be weighed and measured; however, most respondents only recorded pre and postestimates of intake, which would lead to differences in weight judgments [9].

The 24HR method assesses the nutritional intake of a respondent over the past 24 h. Ideally, the survey collects information on nutritional intake over multiple 24 h periods on nonconsecutive, random dates. The 24HR method is usually conducted by a dedicated interviewer by telephone or in person [9]. Some 24HR surveys can also be self-recorded or collected online (e.g., Automated Self-Administered 24 h dietary recall and ASA24 [10]). The differences between the ASA24 and 24HR methods primarily reduce interviewer burden and interview costs and allow respondents to answer questions at their own pace; however, this method may not be suitable for all study populations.

The use of exploratory questions in the 24HR recall method facilitates easy response and has been shown to improve the accuracy of data collection [11]. The survey includes how the food was prepared, what was added after preparation (seasonings, creams, and spices), and when the meal was served [9]. The FFQ assesses general nutritional intake over a speci c period of time, usually a longer period, and asks how often a person consumes food. The FFQ method is a more cost-effective alternative to the 24HR method because respondents can complete the survey themselves, and it can be used for large sample studies [12].

There are several types of systematic measures of self-reported dietary information; for example, based on general perceptions, most respondents tend to report foods that are perceived as healthy and to report less on less healthy foods. However, differences in susceptibility to this tendency between groups of respondents can lead to additional personal bias. Differences in the ability to self-assess and recall portions can also lead to individual subjective differences. This systematic error is unpredictable, but studies suggest that it may be related to factors such as age and gender [13]. While each person uses different strategies to recall portion sizes, including taking photographs and using measurement aids to estimate (e.g., food models) [14,15], research shows that training can lead to a more accurate assessment of food portions [16,17]. In addition, researchers or the methods used to collect dietary data may also be biased [18].

Finally, the accuracy of the conversion of nutrient totals from nutrition dietary records depends on the accuracy and availability of the food ingredient database for conversion to calories and nutrients. In summary, both types of errors reduce the judgement of the relationship between diet and health, as well as the accuracy of the statistical analysis. However, while there may be some slight deviations in the database of the relationships tested [19], when the results of signi cant analyses are properly evaluated, valid conclusions can be drawn.

3. Materials and Methods

This study developed an AI model based on semantic text to analyze the nutritional ingredients of a nutrient, and a digital data semantic analysis model was designed to determine the names and servings of the dishes consumed. The AI model is based on the ingredients of common Taiwanese recipes and automatically calculates the nutrient intake. The model structure consists of a digital data semantic analysis model, an AI precision nutrient analysis model, a database of 1590 recipes, and 7869 ingredients from common Taiwanese recipe databases, and the model structure is shown in Figure 1. The nutrition information of the ingredients was obtained from the public data of the Health Promotion Administration, Ministry of Health and Welfare Taiwan (HPA, MoHW).

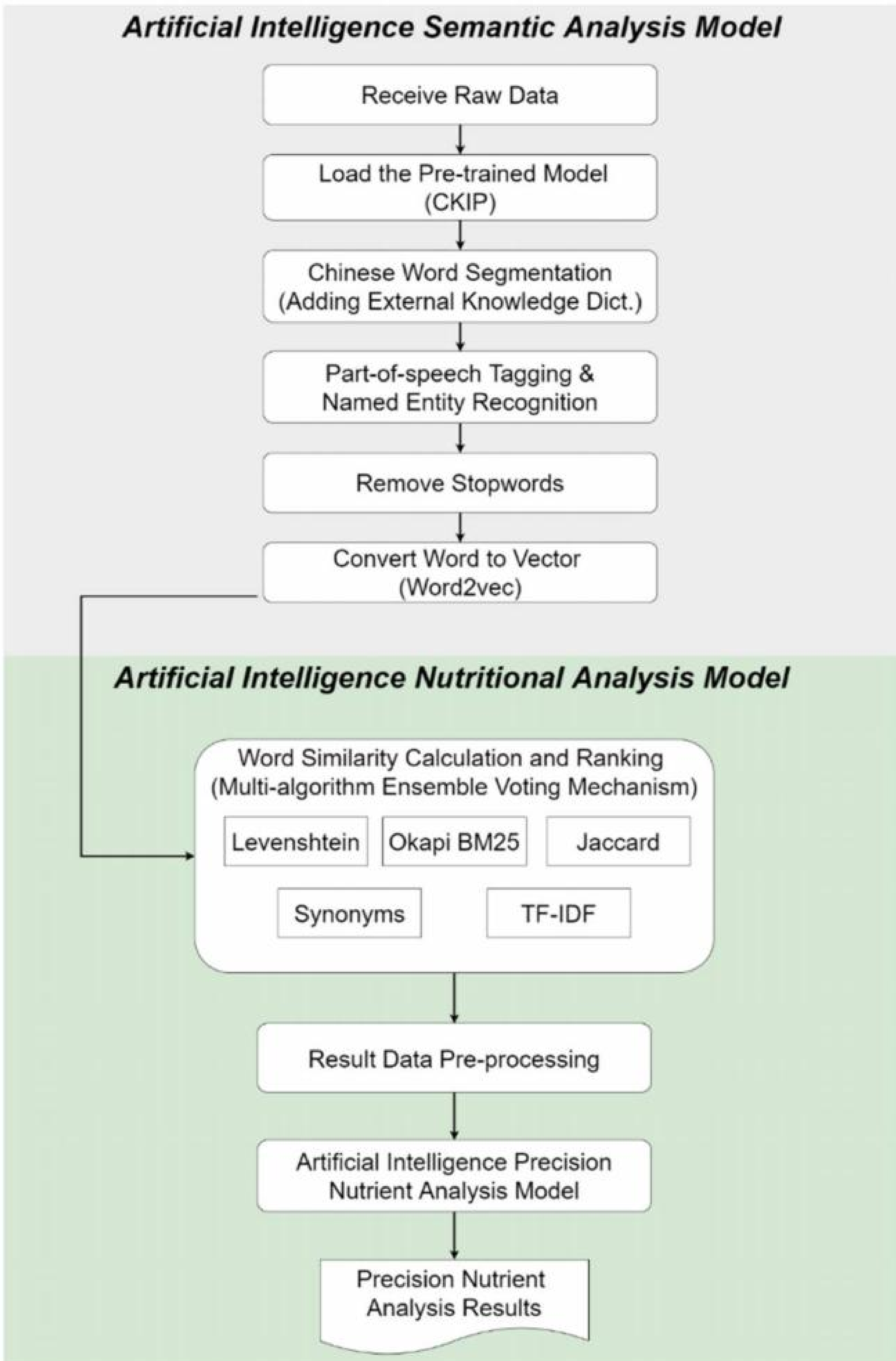

Figure 1. Model Structure.

3.1. Artificial Intelligence Semantic Analysis Model

Data were intercepted and annotated after data entry, and a CKIP pretraining model was used to interpret Chinese words. After completion, lexical annotation and entity identification were performed. Finally, the nouns (dish names) were converted into vector structures using word2vec, which is an application of Natural Language Processing proposed by Tomas Mikolov et al. at Google in 2013 and is one of the most significant advances in the field of machine learning in recent years. Word2vec is an application framework that learns large amounts of textual data and transforms words into mathematical vectors to discriminate their semantic meanings by embedding words into a two-dimensional space in order that words with similar semantic meanings can be closer together.

This study used the continuous bag-of-words (CBOW) method, which aims to determine the lexical properties of the input words using a whole paragraph of context and to determine the relationship between similar words by concatenating them. As similar words are clustered together, the direction of the vector corresponds to the relative relationship.

3.2. Artificial Intelligence Nutritional Analysis Model

The Nutritional Analysis Model is divided into three steps. Step 1 conducts artificial intelligence analysis to determine the most similar recipes. Due to the multicharacter nature of Chinese, single algorithm of semantic analysis may not be precise enough. Therefore, a variety of algorithms were used for the analysis. The AI model is composed of five different algorithms, including 1. Okapi BM25, 2. TF-IDF, 3. Levenshtein, 4. Jaccard, and 5. Synonyms. The algorithm also uses cosine similarity to determine differences in text and then compares it with a database to obtain food information and portion sizes for recipes and ingredient judgement. Step 2 is to determine the best solution by the common voting mechanism. Step 3 is nutritional ingredient calculation.

3.2.1. Step 1. Artificial Intelligence Analysis

(1) Okapi Best Matching (Okapi BM25)

This algorithm was proposed by Stephen E. Robertson, Karen Spärck Jones, and other scholars in 1970 [20–22]. As a probabilistic search framework, BM25 is still widely regarded as one of the most advanced ranking algorithms. BM25 is a bag-of-words model, which ranks a set of documents based on their similarity to each other and obtains a set of scores that can be compared with each other.

The BM25 similarity formula is shown in Equation (1).

$$\mathrm{score}(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \left[\frac{f(q_i,D)(k_1+1)}{f(q_i,D) + k_1\left(1 - b + b\frac{|D|}{avgdl}\right)} \right] \quad (1)$$

Equation (1) BM25 Similarity Formula

$f(q_i,D)$: Frequency of the term q_i in Document D_0
$|D|$: Length of Document D (in words).
K_1: The terminology described above is saturated with parameters.
b: The length normalization parameters, as described above.
avgdl: Average document length in document collection.
IDF: Frequency of inverse text files.
$n(q_i)$: Number of documents containing q_i.
n: Total number of text files in the collection.

(2) Term Frequency–Inverse Document Frequency (TF–IDF)

This algorithm is a weighting technique widely used in information retrieval and text mining, and the combination of TF and IDF was first discussed by Karen Spärck Jones [23]. The TF–IDF was used to assess the importance of a word in a document, which increased positively with the number of times the word appears in the document but decreased inversely with the frequency of its occurrence. The TF-IDF formula is shown in Equation (2), while the Inverse Document Frequency IDF Formula is shown in Equation (3).

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2)$$

Equation (2) TF-IDF formula

Molecular formula: $n_{i,j}$ denotes the number of occurrences of the word in document d_j.
Denominator: The sum of all occurrences of the word in document d_j.

$$idf_i = \lg \frac{|D|}{|\{j : t_i \in d_j\}|} \quad (3)$$

Equation (3) Inverse Document Frequency IDF Formula

Molecular formula: total number of documents.
Denominator: the number of documents containing the term.
The result of the calculation is obtained by quoting the logarithm of the number of documents with a base of 10.

(3) Levenshtein

The Russian scientist Vladimir Levenshtein first proposed this algorithm in 1965 [24]. The basic form of Levenshtein is carried out using a regressive algorithm, where a threshold can be set as an upper limit for the number of steps to be moved. The Levenshtein distance formula is shown in Equation (4).

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (4)$$

Equation (4) Levenshtein Distance Formula

(4) Jaccard

The intersection and union of the two samples can be used to derive the Jaccard similarity coefficient and Jaccard distance for different applications [25]. Jaccard's coefficient gives the degree of similarity and the ratio between the size of the intersection of two sets and the size of the union in a finite set of samples. The Jaccard index formula is shown in Equation (5).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (5)$$

Equation (5) Jaccard Index Formula

(5) Synonyms

Synonyms is an open-source package for natural language tasks in Python and maintained by Chatopera. It provides a variety of NLP tasks, such as text alignment, recommendation algorithms, similarity calculation, semantic shifting, keyword extraction, concept extraction, automatic summarization, and search engines with a multisource lexical database for predata use. Regarding the word vector conversion task, the suite uses Google's gensim suite with a word2vec model for conversion and the vector distance of words with a smooth gradient descent algorithm for approximation [26].

3.2.2. Step 2. Common Voting Mechanism

In this study, the same approximation task was assigned to the abovementioned five different algorithms, and after obtaining the best dish selection results for each algorithm, the highest vote was tallied as the best solution by pooling. The confidence scores of the algorithms were not equally comparable among the different algorithms (Levenshtein distance does not have a con dence score, but a minimum step), as the meanings of the con dence scores of the algorithms are limited to intragroup comparisons. For this reason, instead of using the average of the sum of similar scores for the same project, the highest score of each algorithm was used for vote recognition, and in the nal vote counting process, the votes for each algorithm were equal, which rendered it a fair majority vote decision.

3.2.3. Step 3. Nutritional Ingredient Analysis

The recipe data were obtained through a fuzzy analysis of the arti cial intelligence model, and the nutritional ingredient analysis automatically determined all the ingredients in the dish. Finally, this study consolidated all the nutrients by means of portion calculation to complete the nutrient analysis. The dietary information conversion process is shown in Figure 2.

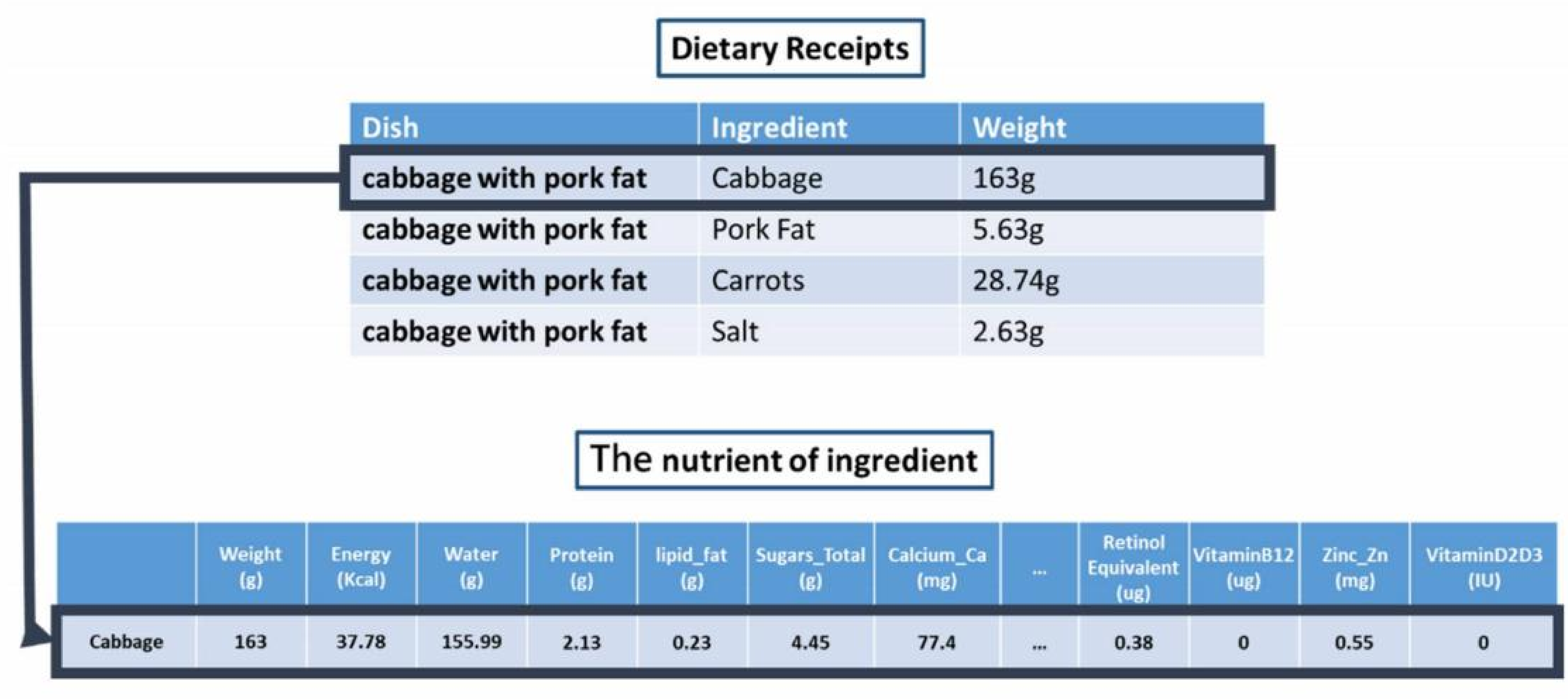

Figure 2. Nutrition Survey Dietary Information Conversion Process.

4. Results

This study developed two models to form a precision nutrient analysis model. The rst model is a Digitized Data Semantic Analysis Model for dish analysis and portion size determination. The second model is a Nutrient Analysis Model that uses ve different algorithms to nd precision recipes, which conducts analyses of dish ingredients and nutrients using a common voting process, and the nal outputs from both models calculate the intake of 24 common nutrients. The operational framework of the model is illustrated below. The recipe database contains 1590 recipes and nutrient information for 7869 ingredients. The model operating framework is shown in Figure 3.

4.1. Operation Example

An example of a dietary recall record for precise nutritional analysis is as follows:

1. Input the dietary record to the model. Dietary Record: Today I had a plate of cabbage with pork fat and a bowl of bamboo shoots and pork ribs soup.
2. The names of the dishes and the portion sizes were analyzed by the Semantic Analysis Model. Nd is de ned as time; Nf is de ned as a quantity, and Na is de ned as a common noun. Segmentation of record:

    [{ label : Today , Pos : Nd }, { label : plate , Pos : Nf }, { label : cabbage , Pos : Na }, { label : pork fat , Pos : Na }, { label : bowl , Pos : Nf }, { label : bamboo shoots , Pos : Na }, { label : pork ribs , Pos : Na }, { label : soup , Pos : Na }]

In this study, the plate of the dish represents 200 g, and the soup bowl represents 200 g.

3. Nutrition intake calculation by the Precision Nutrient Analysis Model.
   (1) In Step 1: Each dish was separated into its ingredients according to the recipes. 200 g of cabbage with pork fat: Cabbage: 163 g, Pork Fat: 5.63 g, Carrots: 28.74 g, and Salt: 2.63 g. 200 g of bamboo shoots and pork ribs soup: Water: 50.13 g, White pepper: 0.63 g, Bamboo shoot: 73.43 g, Pork chops: 75.19 g, and Salt: 0.63 g.
   (2) In Step 2: 24 nutrients were calculated for each ingredient, and the precision nutrient analysis results were calculated based on the sum of all nutrients.

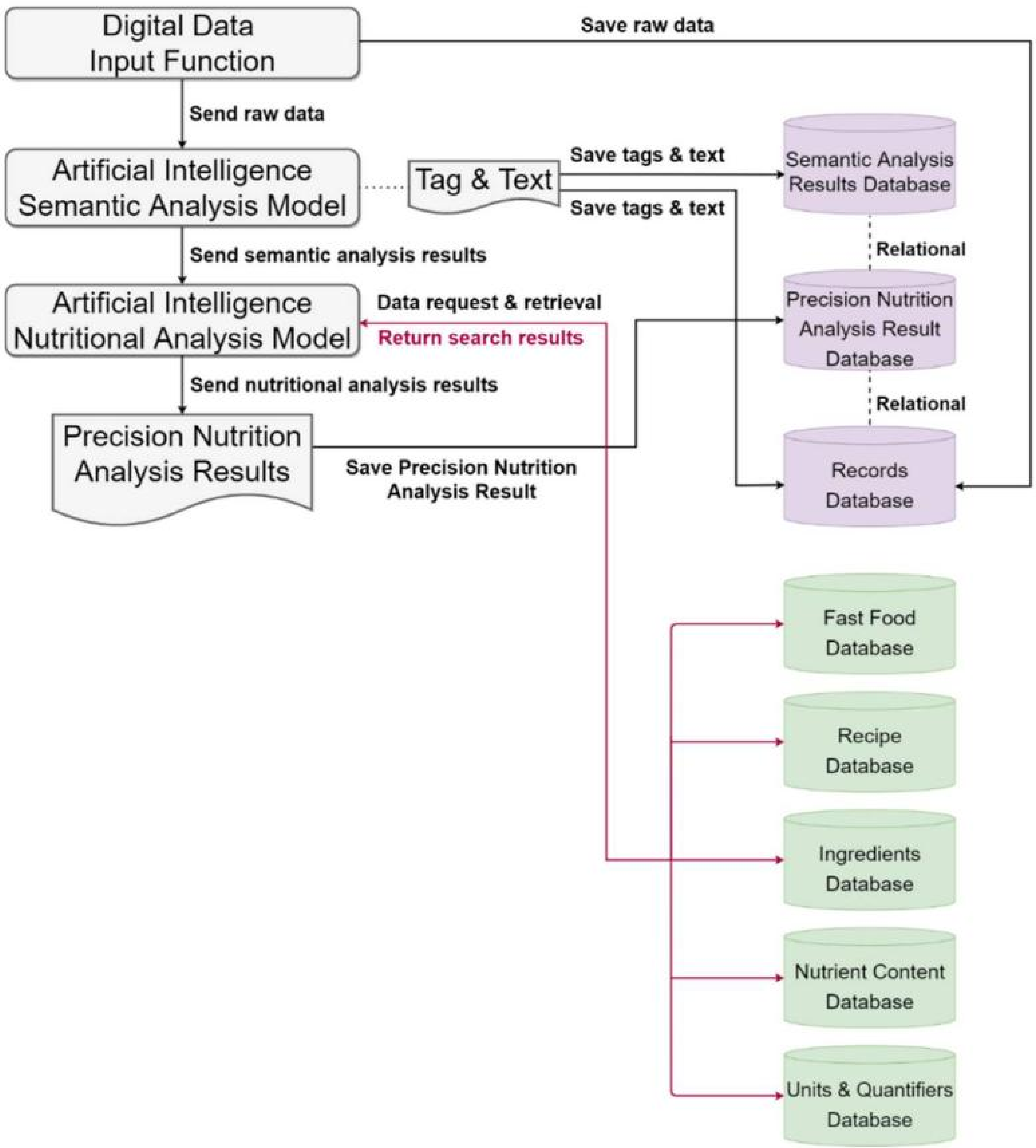

Figure 3. Model Operating Framework.

4.2. Model Accuracy Veri cation

The accuracy of the model was analyzed using data from the Nutrition Survey. In this study, the 2013 2016 National Nutrition Health Status Change Survey (NNHS) was used for analysis. The NNHS was initiated by the HPA MoHW and conducted in a four-year cycle and considered county and city distribution, as well as seasonal effects. The collected data were used as a reference for the formulation of national nutrition and health-related policies in Taiwan.

The aim of the survey is to understand the nutrition, health, diet, and lifestyle of the Taiwanese people and their relevance, in order to establish a long-term, stable, and nationally representative nutrition and health surveillance mechanism. The results can be used as a basis for government policies regarding diet and nutrition and health promotion and disease prevention and can help improve the health status of the population and prevent possible future health problems.

The NNHS uses a multistage strati ed cluster sampling design, with the sample group being the entire age cohort, excluding pregnant and breastfeeding women, people without self-awareness, and institutional care residents, and the overall sample is representative of the Taiwanese population. The nutrition data were stored in a 24 h dietary memory record and analyzed by a professional nutritionist.

4.2.1. Data Resource

The 2013 2016 National Survey of Changes in Nutritional Health Status was used to validate the accuracy of the model. The data contain a 24-h dietary recall nutrient intake sum analysis le and a 24-h dietary recall food weight and nutrient ingredient le with the information of 24 nutrients, including Energy, Water, Protein, Lipid Fat, Sugars Total, Calcium Ca, Phosphorus P, Iron Fe, VitaminB1, VitaminB2, Nicotinic, Vitamin C, Saturated Fat, Cholesterol, Vitamin E alpha TE, Sodium Na, VitaminB6, Magnesium, Dietary Fiber, Potassium K, Equivalent, VitaminB12, Zinc Zn, and VitaminD2D3.

24 h dietary recall nutrient intake sum analysis le (sum_nutrients_24hH total 2602 data entries. This le includes the data of the total nutrient intake in a single 24 h dietary recall survey.

24 h dietary recall food weight and nutrient ingredient le (food_wt_and_nutrients) totaling 113,824 data entries. This le includes the data of the sum of nutrients for each individual dish, food, health product, etc.

4.2.2. Validation Process

(1) Inputting data from the 102 105 National Survey of Changes in Nutritional Health Status into a digitized data semantic analysis model;
(2) Model analysis of dishes, portion sizes, and the ingredients in the dishes;
(3) Analysis of nutrient intake using the AI Precision Nutrient Analysis Model;
(4) Analyze the results against the 24-h dietary recall nutrient intake sum analysis le and the 24-h dietary recall food weight and nutrient ingredient le ;
(5) Compare the accuracy of the model.

4.3. Analysis Result

The results of the nutrition survey team analysis (from the 24 h dietary recall nutrient intake sum analysis le) were used as the gold standard, while the results of this study model analysis were used as the control group for the nutrient difference ratio analysis. The discrepancy comparison tables of the NNHS analysis with the results of this study in the 24 h dietary recall nutrient intake sum analysis are shown in Tables 1 3. A total of 2602 data entries were analyzed for total nutrient intake, with 24 different nutrients analyzed for each data item. The differences between the results of this study and the results of the nutrition survey are shown in Tables 1 3. While 13 nutrients had a total of more than 95% (2472 data entries) of the data with an intake error of <5%, 3 nutrients had a total of 90 94% of the data with an intake error of <5%; 5 nutrients had a total of 89.99 80% of the data with an intake error of <5%; Vitamin E alpha TE had a total of more than 95% (2472 data entries) with an intake error of <10%; Sugars Total and VitaminD2D3 had a 70% data error <10%.

The results of the nutrition survey team analysis (from 24 h dietary recall food weight and nutrient ingredient le) were used as the gold standard, while the results of this study model analysis were used as the control group for a nutrient difference ratio analysis. The discrepancy comparison table of the NNHS analysis with the results of this study in the 24 h dietary recall food weight and nutrient ingredients are shown in Tables 4 6.

A total of 113,824 data entries were analyzed for food weight and nutrient ingredients, with 24 different nutrients analyzed for each data item. The differences between the results of this study and the results of the nutrition survey are shown in Tables 4 6.

Table 1. Discrepancy comparison table of NNHS analysis with the results of this study in the sum_nutrients_24h (1).

Error Range	Energy (kcal)		Water (g)		Protein (g)		Lipid Fat (g)		Sugars Total (g)		Calcium Ca (mg)		Phosphorus P (mg)		Iron Fe (mg)	
<1%	2187	84.1%	2503	96.2%	2284	87.8%	1902	73.1%	352	13.5%	1168	44.9%	2087	80.2%	1401	53.8%
1% and <2%	297	11.4%	68	2.6%	228	8.8%	319	12.3%	436	16.8%	601	23.1%	306	11.8%	734	28.2%
2% and <3%	45	1.7%	24	0.9%	43	1.7%	136	5.2%	270	10.4%	298	11.5%	82	3.2%	199	7.6%
3% and <4%	22	0.8%	5	0.2%	12	0.5%	84	3.2%	186	7.1%	194	7.5%	44	1.7%	116	4.5%
4% and <5%	9	0.3%	1	0.0%	10	0.4%	42	1.6%	142	5.5%	109	4.2%	21	0.8%	39	1.5%
5% and <6%	8	0.3%	1	0.0%	8	0.3%	29	1.1%	91	3.5%	68	2.6%	15	0.6%	41	1.6%
6 10%	27	1.0%	0	0.0%	12	0.5%	56	2.2%	316	12.1%	122	4.7%	32	1.2%	51	2.0%
11 15%	6	0.2%	0	0.0%	2	0.1%	13	0.5%	172	6.6%	22	0.8%	8	0.3%	7	0.3%
16 20%	1	0.0%	0	0.0%	2	0.1%	6	0.2%	136	5.2%	11	0.4%	3	0.1%	7	0.3%
21 30%	0	0.0%	0	0.0%	0	0.0%	8	0.3%	170	6.5%	2	0.1%	2	0.1%	7	0.3%
31 40%	0	0.0%	0	0.0%	1	0.0%	6	0.2%	91	3.5%	5	0.2%	2	0.1%	0	0.0%
41 60%	0	0.0%	0	0.0%	0	0.0%	1	0.0%	121	4.7%	0	0.0%	0	0.0%	0	0.0%
61 80%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	49	1.9%	2	0.1%	0	0.0%	0	0.0%
81 100%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	32	1.2%	0	0.0%	0	0.0%	0	0.0%
>100%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

Table 2. Discrepancy comparison table of NNHS analysis with the results of this study in the sum_nutrients_24h (2).

Error Range	VitaminB1 (mg)		VitaminB2 (mg)		Nicotinic (mg)		Vitamin C (mg)		Saturated Fat (g)		Cholesterol (mg)		Vitamin E Alpha TE (mg)		Sodium Na (mg)	
<1%	1990	76.5%	1601	61.5%	1811	69.6%	1652	63.5%	1440	55.3%	1375	52.8%	921	35.4%	1836	70.6%
1% and <2%	345	13.3%	644	24.8%	416	16.0%	234	9.0%	480	18.4%	586	22.5%	398	15.3%	296	11.4%
2% and <3%	100	3.8%	135	5.2%	146	5.6%	139	5.3%	193	7.4%	542	20.8%	307	11.8%	106	4.1%
3% and <4%	59	2.3%	81	3.1%	76	2.9%	81	3.1%	112	4.3%	20	0.8%	240	9.2%	87	3.3%
4% and <5%	35	1.3%	38	1.5%	45	1.7%	60	2.3%	76	2.9%	12	0.5%	176	6.8%	60	2.3%
5% and <6%	17	0.7%	17	0.7%	23	0.9%	37	1.4%	69	2.7%	8	0.3%	128	4.9%	55	2.1%
6 10%	40	1.5%	46	1.8%	57	2.2%	104	4.0%	137	5.3%	21	0.8%	288	11.1%	93	3.6%
11 15%	7	0.3%	23	0.9%	20	0.8%	50	1.9%	49	1.9%	12	0.5%	82	3.2%	37	1.4%
16 20%	4	0.2%	6	0.2%	5	0.2%	35	1.3%	11	0.4%	9	0.3%	31	1.2%	12	0.5%
21 30%	4	0.2%	9	0.3%	1	0.0%	52	2.0%	21	0.8%	3	0.1%	18	0.7%	8	0.3%
31 40%	1	0.0%	2	0.1%	0	0.0%	27	1.0%	11	0.4%	1	0.0%	9	0.3%	8	0.3%
41 60%	0	0.0%	0	0.0%	1	0.0%	40	1.5%	2	0.1%	1	0.0%	4	0.2%	1	0.0%
61 80%	0	0.0%	0	0.0%	1	0.0%	27	1.0%	1	0.0%	1	0.0%	0	0.0%	1	0.0%
81 100%	0	0.0%	0	0.0%	0	0.0%	11	0.4%	0	0.0%	5	0.2%	0	0.0%	2	0.1%
>100%	0	0.0%	0	0.0%	0	0.0%	53	2.0%	0	0.0%	6	0.2%	0	0.0%	0	0.0%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

Table 3. Discrepancy comparison table of NNHS analysis with the results of this study in the sum_nutrients_24h (3).

Error Range	VitaminB6 (mg)		Magnesium (mg)		Dietary Fiber (g)		Potassium K (mg)		Retinol Equivalent (ug)		VitaminB12 (ug)		Zinc Zn (mg)		VitaminD2D3 (IU)	
<1%	1511	58.1%	1793	68.9%	1518	58.3%	1773	68.1%	1234	47.4%	1388	53.3%	1579	60.7%	723	68.9%
1% and <2%	518	19.9%	529	20.3%	327	12.6%	487	18.7%	475	18.3%	350	13.5%	626	24.1%	209	20.3%
2% and <3%	180	6.9%	153	5.9%	196	7.5%	155	6.0%	241	9.3%	255	9.8%	206	7.9%	162	5.9%
3% and <4%	123	4.7%	34	1.3%	95	3.7%	76	2.9%	148	5.7%	165	6.3%	75	2.9%	143	1.3%
4% and <5%	60	2.3%	22	0.8%	61	2.3%	43	1.7%	98	3.8%	95	3.7%	27	1.0%	120	0.8%
5% and <6%	48	1.8%	18	0.7%	53	2.0%	20	0.8%	61	2.3%	68	2.6%	18	0.7%	86	0.7%
6 10%	95	3.7%	22	0.8%	187	7.2%	35	1.3%	140	5.4%	181	7.0%	35	1.3%	339	0.8%
11 15%	28	1.1%	14	0.5%	60	2.3%	8	0.3%	98	3.8%	53	2.0%	18	0.7%	250	0.5%
16 20%	11	0.4%	5	0.2%	36	1.4%	5	0.2%	39	1.5%	12	0.5%	8	0.3%	75	0.2%
21 30%	16	0.6%	3	0.1%	46	1.8%	0	0.0%	48	1.8%	13	0.5%	0	0.0%	81	0.1%
31 40%	8	0.3%	4	0.2%	10	0.4%	0	0.0%	9	0.3%	5	0.2%	1	0.0%	54	0.2%
41 60%	3	0.1%	5	0.2%	9	0.3%	0	0.0%	5	0.2%	4	0.2%	1	0.0%	58	0.2%
61 80%	0	0.0%	0	0.0%	3	0.1%	0	0.0%	4	0.2%	2	0.1%	4	0.2%	40	0.0%
81 100%	1	0.0%	0	0.0%	1	0.0%	0	0.0%	0	0.0%	5	0.2%	0	0.0%	31	0.0%
>100%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	2	0.1%	6	0.2%	4	0.2%	231	0.0%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

Table 4. Discrepancy comparison table of NNHS analysis with the results of this study in the food_wt_and_nutrients (1).

Error Range	Energy (kcal)		Water (g)		Protein (g)		Lipid Fat (g)		Sugars Total (g)		Calcium Ca (mg)		Phosphorus P (mg)		Iron Fe (mg)	
<1%	97,435	85.6%	99,390	87.3%	97,211	85.4%	102,695	90.2%	95,184	83.6%	97,059	85.3%	100,940	88.7%	99,540	87.5%
1% and <2%	6263	5.5%	2592	2.3%	4949	4.3%	1878	1.6%	3485	3.1%	661	0.6%	666	0.6%	514	0.5%
2% and <3%	1177	1.0%	1357	1.2%	570	0.5%	139	0.1%	75	0.1%	685	0.6%	306	0.3%	130	0.1%
3% and <4%	595	0.5%	250	0.2%	1114	1.0%	67	0.1%	265	0.2%	146	0.1%	101	0.1%	88	0.1%
4% and <5%	153	0.1%	67	0.1%	514	0.5%	338	0.3%	370	0.3%	915	0.8%	663	0.6%	1880	1.7%
5% and <6%	472	0.4%	774	0.7%	114	0.1%	77	0.1%	311	0.3%	84	0.1%	502	0.4%	625	0.5%
6 10%	2231	2.0%	1028	0.9%	991	0.9%	1096	1.0%	1756	1.5%	2583	2.3%	4554	4.0%	3749	3.3%
11 15%	575	0.5%	2868	2.5%	3500	3.1%	850	0.7%	258	0.2%	878	0.8%	399	0.4%	791	0.7%
16 20%	790	0.7%	606	0.5%	122	0.1%	518	0.5%	285	0.3%	4737	4.2%	184	0.2%	333	0.3%
21 30%	2952	2.6%	1205	1.1%	323	0.3%	552	0.5%	493	0.4%	2020	1.8%	666	0.6%	547	0.5%
31 40%	383	0.3%	1691	1.5%	1596	1.4%	2303	2.0%	598	0.5%	268	0.2%	343	0.3%	2657	2.3%
41 60%	80	0.1%	1281	1.1%	96	0.1%	637	0.6%	2803	2.5%	760	0.7%	504	0.4%	1566	1.4%
61 80%	189	0.2%	182	0.2%	231	0.2%	167	0.1%	449	0.4%	432	0.4%	460	0.4%	309	0.3%
81 100%	330	0.3%	362	0.3%	2142	1.9%	1979	1.7%	3087	2.7%	2191	1.9%	2051	1.8%	378	0.3%
>100%	199	0.2%	171	0.2%	351	0.3%	528	0.5%	4405	3.9%	405	0.4%	1485	1.3%	717	0.6%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

Table 5. Discrepancy comparison table of NNHS analysis with the results of this study in the food_wt_and_nutrients (2).

Error Range	VitaminB1 (mg)		VitaminB2 (mg)		Nicotinic (mg)		Vitamin C (mg)		Saturated Fat (g)		Cholesterol (mg)		Vitamin E Alpha TE (mg)		Sodium Na (mg)	
<1%	103,564	91.0%	103,347	90.8%	103,396	90.8%	102,357	89.9%	99,471	87.4%	110,481	97.1%	100,867	88.6%	99,264	87.2%
1% and <2%	1899	1.7%	1869	1.6%	360	0.3%	1721	1.5%	782	0.7%	27	0.0%	150	0.1%	3296	2.9%
2% and <3%	301	0.3%	55	0.0%	1075	0.9%	237	0.2%	280	0.2%	1717	1.5%	2102	1.8%	227	0.2%
3% and <4%	376	0.3%	388	0.3%	197	0.2%	69	0.1%	1228	1.1%	5	0.0%	284	0.2%	171	0.2%
4% and <5%	82	0.1%	284	0.2%	40	0.0%	98	0.1%	172	0.2%		0.0%	31	0.0%	138	0.1%
5% and <6%	98	0.1%	257	0.2%	175	0.2%	62	0.1%	151	0.1%	10	0.0%	119	0.1%	240	0.2%
6 10%	2417	2.1%	3394	3.0%	2291	2.0%	3267	2.9%	1875	1.6%	18	0.0%	2710	2.4%	4153	3.6%
11 15%	1331	1.2%	578	0.5%	609	0.5%	308	0.3%	782	0.7%	17	0.0%	129	0.1%	503	0.4%
16 20%	216	0.2%	691	0.6%	877	0.8%	337	0.3%	1750	1.5%	28	0.0%	175	0.2%	396	0.3%
21 30%	825	0.7%	187	0.2%	2583	2.3%	621	0.5%	259	0.2%	29	0.0%	2448	2.2%	975	0.9%
31 40%	445	0.4%	400	0.4%	392	0.3%	499	0.4%	2312	2.0%	40	0.0%	203	0.2%	1087	1.0%
41 60%	143	0.1%	422	0.4%	139	0.1%	242	0.2%	484	0.4%	25	0.0%	1142	1.0%	639	0.6%
61 80%	186	0.2%	95	0.1%	77	0.1%	392	0.3%	411	0.4%	27	0.0%	335	0.3%	231	0.2%
81 100%	1657	1.5%	1478	1.3%	1395	1.2%	2022	1.8%	2819	2.5%	1201	1.1%	2182	1.9%	2305	2.0%
>100%	284	0.2%	379	0.3%	218	0.2%	1592	1.4%	1048	0.9%	199	0.2%	947	0.8%	199	0.2%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

Table 6. Discrepancy comparison table of NNHS analysis with the results of this study in the food_wt_and_nutrients (3).

Error Range	VitaminB6 (mg)		Magnesium (mg)		Dietary Fiber (g)		Potassium K (mg)		Retinol Equivalent (ug)		VitaminB12 (ug)		Zinc Zn (mg)		VitaminD2D3 (IU)	
<1%	99,965	87.8%	93,126	81.8%	105,576	92.8%	102,151	89.7%	101,920	89.5%	109,024	95.8%	97,681	85.8%	110,264	96.9%
1% and <2%	294	0.3%	2576	2.3%	1991	1.7%	452	0.4%	121	0.1%	9	0.0%	151	0.1%	346	0.3%
2% and <3%	2042	1.8%	2091	1.8%	295	0.3%	734	0.6%	91	0.1%	1	0.0%	964	0.8%	3	0.0%
3% and <4%	295	0.3%	160	0.1%	458	0.4%	162	0.1%	59	0.1%	4	0.0%	1998	1.8%	3	0.0%
4% and <5%	27	0.0%	190	0.2%	128	0.1%	56	0.0%	5	0.0%	2	0.0%	424	0.4%	120	0.1%
5% and <6%	46	0.0%	702	0.6%	308	0.3%	1118	1.0%	24	0.0%	12	0.0%	95	0.1%	86	0.1%
6 10%	2877	2.5%	2045	1.8%	1154	1.0%	1391	1.2%	3627	3.2%	38	0.0%	2568	2.3%	14	0.0%
11 15%	1679	1.5%	1406	1.2%	296	0.3%	131	0.1%	118	0.1%	1700	1.5%	1084	1.0%	1707	1.5%
16 20%	576	0.5%	282	0.2%	258	0.2%	396	0.3%	175	0.2%	31	0.0%	2991	2.6%	10	0.0%
21 30%	1016	0.9%	625	0.5%	512	0.4%	475	0.4%	174	0.2%	34	0.0%	1764	1.5%	29	0.0%
31 40%	205	0.2%	300	0.3%	63	0.1%	239	0.2%	1046	0.9%	11	0.0%	324	0.3%	5	0.0%
41 60%	781	0.7%	597	0.5%	446	0.4%	1962	1.7%	407	0.4%	171	0.2%	643	0.6%	12	0.0%
61 80%	574	0.5%	1594	1.4%	204	0.2%	136	0.1%	141	0.1%	27	0.0%	100	0.1%	11	0.0%
81 100%	1896	1.7%	7775	6.8%	1878	1.6%	3980	3.5%	5097	4.5%	2215	1.9%	2532	2.2%	1368	1.2%
>100%	1551	1.4%	355	0.3%	257	0.2%	441	0.4%	819	0.7%	545	0.5%	505	0.4%	52	0.0%

Note: Each number indicates how much of the data are actually within that margin of error. The percentages indicate the percentage of the total data.

While 3 nutrients had a total of more than 95% of the data with an intake error of <2%, 9 nutrients had a total of 90 94% of the data with an intake error of <2%; 12 nutrients had a total of 89.99 80% of the data with an intake error of <2%.

5. Discussion

Each 24 h dietary recall nutrition survey in this study took approximately 40 min. The volume and complexity of the survey data and the variation in the ability to self-assess and recall portions can lead to individual subjective differences [13]. Similarly, the researchers or the methods used to collect dietary data may be biased [18].

Therefore, this study balanced the accuracy of nutrient intake analysis by compensating for errors through fuzzy analysis and arti cial intelligence. Conventional FFQs are primarily designed to assess total nutrient intake or changes in intake over time [27 29]; however, the FFQ limits the range of foods that can be investigated as it combines food and beverages thus determining the exact amount of nutrients is less precise than other more detailed methods. It is also not possible to accurately measure absolute intakes of different food components. Moreover, FFQs require literacy and the physical ability to complete the questionnaire, and the FFQ survey can be burdensome for subjects and dif cult or confusing to complete due to poor descriptions or dif cult-to-understand questions. The most commonly used methods in nutrition research are the Diet Record, 24HR, and FFQ.

The Food Record is also used as the gold standard in validation studies [30]. Given the contingent nature of the respondents' food choices, a variety of food and beverage combinations [31] and nutrient supplementation [32] are the best methods to investigate. In order to reduce the burden on surveyors, the arti cial intelligence model in this study has proven to be a feasible strategy for large-scale nutritional surveys after data discrepancy comparisons.

When comparing the difference between our model and the data analyzed in the actual nutrition survey, it was found that the results of the 24-h dietary recall food weight and nutrient ingredient method were highly accurate, with less than 2% discrepancy in analysis for almost all nutrients. This result shows that the nutrients of the ingredient data in our model are correct. In the 24 h dietary recall nutrient intake sum analysis, the model was used to conduct an arti cial intelligence analysis of the dishes, meaning it conducted an automated analysis of the components and servings to estimate nutrient intake. The results show a margin of error of less than 10% thus con rming the high accuracy of the model in this study.

6. Conclusions

This study proposed an Intelligence Precision Nutrient Analysis Model based on a digital data collection framework, where the nutrient intake was analyzed by entering dietary recall data. The AI Precision Nutrient Analysis Model was used to analyze the ingredients of the dishes and calculate nutrient intake by automatically analyzing the dishes, and portion sizes were analyzed using a digital data semantic analysis model. The results of this study show very little difference in nutrient intake between the model and the NNHS analysis and are highly accurate; therefore, the AI model can be used as a reference for nutrition surveys and personal nutrition analysis. In terms of data access, as there is not yet a complete set of publicly available data on food nutrient ingredients; more complete data and references on micro-nutrients should be available in the future. On the other hand, the scope of recipes should be expanded.

Author Contributions: The work presented in this paper was carried out in collaboration among all authors. H.-A.L. and C.-Y.L. formed the conception and study design. K.-W.C. and C.-Y.L. carried out the data analysis; H.-A.L. performed the literature review; C.-Y.L., T.-T.H., L.-H.Y., P.-H.W., and H.-H.K. performed the model development; H.-A.L., P.-H.W., and K.-W.C. drafted the manuscript, and C.-Y.H. made signi cant revisions and supplied valuable improvement suggestions. All authors have read and agreed to the published version of the manuscript.

Funding: This project received funding from the Ministry of Science and Technology, Taiwan, under the project no. 110-2221-E-227 -003 -MY3 and Ministry of Education, Taiwan, under the project no. 107EH12-402-022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Con icts of Interest: The authors declare no con ict of interest.

References

1. Cade, J.E. Measuring diet in the 21st century: Use of new technologies. Proc. Nutr. Soc. 2017, 76, 276 282. [CrossRef] [PubMed]
2. Cade, J.E.; Consortium, O.B.O.T.D.; Warthon-Medina, M.; Albar, S.; Alwan, N.A.; Ness, A.; Roe, M.; Wark, P.A.; Greathead, K.; Burley, V.J.; et al. DIET@NET: Best Practice Guidelines for dietary assessment in health research. BMC Med. 2017, 15, 1 15. [CrossRef] [PubMed]
3. Freedman, L.S.; Schatzkin, A.; Midthune, D.; Kipnis, V. Dealing with Dietary Measurement Error in Nutritional Cohort Studies. JNCI 2011, 103, 1086 1092. [CrossRef] [PubMed]
4. Freedman, L.S.; Potischman, N.; Kipnis, V.; Midthune, U.; Schatzkin, A.; Thompson, F.; Troiano, R.P.; Prentice, R.; Patterson, R.; Carroll, R.; et al. A comparison of two dietary instruments for evaluating the fat breast cancer relationship. Int. J. Epidemiol. 2006, 35, 1011 1021. [CrossRef]
5. Carter, M.C.; Burley, V.; Nykjaer, C.; Cade, J. Adherence to a Smartphone Application for Weight Loss Compared to Website and Paper Diary: Pilot Randomized Controlled Trial. J. Med. Internet Res. 2013, 15, e32. [CrossRef]
6. Timon, C.M.; Blain, R.J.; McNulty, B.; Kehoe, L.; Evans, K.; Walton, J.; Flynn, A.; Gibney, E.R. The Development, Validation, and User Evaluation of Foodbook24: A Web-Based Dietary Assessment Tool Developed for the Irish Adult Population. J. Med. Internet Res. 2017, 19, e158. [CrossRef]
7. Boushey, C.J.; Harray, A.J.; Kerr, D.A.; Schap, T.; Paterson, S.; A ague, T.; Ruiz, M.B.; Ahmad, Z.; Delp, E.J.; Harnack, L.; et al. How Willing Are Adolescents to Record Their Dietary Intake? The Mobile Food Record. JMIR MHealth UHealth 2015, 3, e47. [CrossRef]
8. Thompson, F.E.; Dixit-Joshi, S.; Potischman, N.; Dodd, K.W.; Kirkpatrick, S.I.; Kushi, L.H.; Alexander, G.L.; Coleman, L.A.; Zimmerman, T.P.; Sundaram, M.E.; et al. Comparison of Interviewer-Administered and Automated Self-Administered 24-Hour Dietary Recalls in 3 Diverse Integrated Health Systems. Am. J. Epidemiol. 2015, 181, 970 978. [CrossRef]
9. Thompson, F.E.; Byers, T. Dietary assessment resource manual. J. Nutr. 1994, 124, s2245 s2317. [CrossRef]
10. Subar, A.F.; Kirkpatrick, S.I.; Mittl, B.; Zimmerman, T.P.; Thompson, F.E.; Bingley, C.; Willis, G.; Islam, N.G.; Baranowski, T.; McNutt, S.; et al. The automated self-administered 24-hour dietary recall (ASA24): A resource for researchers, clinicians, and educators from the national cancer institute. J. Acad. Nutr. Diet. 2012, 112, 1134 1137. [CrossRef]
11. Campbell, V.A.; Dodds, M.L. Collecting Dietary Information from Groups of Older People. Limitations of the 24-Hr. Recall. J. Am. Diet. Assoc. 1967, 51, 29 33. [CrossRef] [PubMed]
12. Heady, J.A. Diets of Bank Clerks Development of a Method of Classifying the Diets of Individuals for Use in Epidemiological Studies. J. R. Stat. Soc. Ser. A 1961, 124, 336. [CrossRef]
13. Dwyer, J.T.; Gardner, J.; Halvorsen, K.; Krall, E.A.; Cohen, A.; Valadian, I. MEMORY OF FOOD INTAKE IN THE DISTANT PAST. Am. J. Epidemiol. 1989, 130, 1033 1046. [CrossRef] [PubMed]
14. Iv, C.; Edgar; Godwin, S.L.; Vecchio, F.A. Cognitive strategies for reporting portion sizes using dietary recall procedures. J. Am. Diet. Assoc. 2000, 100, 891 897.
15. Guthrie, H.A. Selection and quanti cation of typical food portions by young adults. J. Am. Diet. Assoc. 1984, 84, 1440 1444. [CrossRef]
16. Bolland, J.E.; Yuhas, J.A.; Bolland, T.W. Estimation of food portion sizes: Effectiveness of training. J. Am. Diet. Assoc. 1988, 88, 817 821. [CrossRef]
17. Howat, P.M.; Mohan, R.; Champagne, C.; Monlezun, C.; Wozniak, P.; Bray, G. Validity and reliability of reported dietary intake data. J. Am. Diet. Assoc. 1994, 94, 169 173. [CrossRef]
18. Gibson, R.S. Principles of Nutritional Assessment; Oxford University Press: Oxford, UK, 2005.
19. Kirkpatrick, S.I.; Dodd, K.W.; Tooze, J.; Bailey, R.L.; Freedman, L.; Midthune, D. Measurement Error Webinar Series. Risk Factor Monitoring and Methods, National Cancer Institute, National Institutes of Health. 2012; Bethesda: Bethesda, MD, USA, 2012.
20. Gatford, S.E.R.; Walker, S.; Jones, S. Micheline Hancock-Beaulieu & Mike. In Proceedings of the Third Text REtrieval Conference (TREC 1994), Gaithersburg, MD, USA, 2 4 November 1994; Available online: https://trec.nist.gov/pubs/trec3/t3_proceedings.html (accessed on 1 January 2021).
21. Jones, K.S.; Walker, S.; Robertson, S. A probabilistic model of information retrieval: Development and comparative experiments: Part 2. Inf. Process. Manag. 2000, 36, 809 840. [CrossRef]

22. Jones, K.S.; Walker, S.; Robertson, S. A probabilistic model of information retrieval: Development and comparative experiments: Part 1. Inf. Process. Manag. 2000, 36, 779 808. [CrossRef]
23. Jones, K.S. Index term weighting. Inf. Storage Retr. 1973, 9, 619 633. [CrossRef]
24. Levenshtein, V.I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Sov. Phys. Dokl. 1966, 10, 707.
25. Michael, L.; Winter, D. Distance between Sets. Nature 1971, 234, 34 35.
26. Chatopera. Synonyms. Available online: https://github.com/chatopera/Synonyms#references (accessed on 20 January 2021).
27. Tucker, K.L.; Chen, H.; Hannan, M.T.; Cupples, L.A.; Wilson, P.W.F.; Felson, D.; Kiel, D. Bone mineral density and dietary patterns in older adults: The Framingham Osteoporosis Study. Am. J. Clin. Nutr. 2002, 76, 245 252. [CrossRef] [PubMed]
28. Wirfält, A.E.; Jeffery, R.W. Using Cluster Analysis to Examine Dietary Patterns: Nutrient Intakes, Gender, and Weight Status Differ Across Food Pattern Clusters. J. Am. Diet. Assoc. 1997, 97, 272 279. [CrossRef]
29. Haveman-Nies, A.; Tucker, K.; de Groot, L.; Wilson, P.; van Staveren, W. Evaluation of dietary quality in relationship to nutritional and lifestyle factors in elderly people of the US Framingham Heart Study and the European SENECA study. Eur. J. Clin. Nutr. 2001, 55, 870 880. [CrossRef]
30. Walter, W. Nutritional Epidemiology; Oxford University Press: Oxford, UK, 2012; Volume 40.
31. Freedman, L.S.; Midthune, D.; Arab, L.; Prentice, R.L.; Subar, A.F.; Willett, W.; Neuhouser, M.L.; Tinker, L.F.; Kipnis, V. Combining a Food Frequency Questionnaire With 24-Hour Recalls to Increase the Precision of Estimation of Usual Dietary Intakes Evidence from the Validation Studies Pooling Project. Am. J. Epidemiol. 2018, 187, 2227 2232. [CrossRef]
32. Nicastro, H.L.; Bailey, R.; Dodd, K.W. Using 2 Assessment Methods May Better Describe Dietary Supplement Intakes in the United States. J. Nutr. 2015, 145, 1630 1634. [CrossRef]