Title	Nutrition management based on food recognition and AR measurement
Sub Title	
Author	孙, 雅文(Sun, Yawen) 小木, 哲朗(Ogi, Tetsurō)
Publisher	慶應義塾大学大学院システムデザイン・マネジメント研究科
Publication year	2019
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2019年度システムエンジニアリング学 第292号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002019-0008

Master's Dissertation 2019

Nutrition Management Based on Food Recognition and AR Measurement

Yawen Sun
（Student ID Number：81734570）

Supervisor　Tetsuro Ogi

September 2019

Graduate School of System Design and Management,
Keio University
Major in System Design and Management

SUMMARY OF MASTER'S DISSERTATION

Student Identification Number	81734570	Name	Yawen Sun

Title

Nutrition Management Based on Food Recognition and AR Measurement

Abstract

With the development of medical, the average life expectancy increases gradually, but the healthy life expectancy does not increase at the same rate. Plenty of senior citizens suffered from diseases. In Japan, over 60% people dead from lifestyle diseases every year.

Health management come to be serious nowadays, which is divided into nutrition, physical activities, and mental health management. It would help people improve health states and expand healthy life expectancy. With the widespread use of smartphone, people are used to using health management applications. But its lack of nutrition management application, only applications have incomplete functions.

To do nutrition management, a new application was built to help users know the nutrition information of daily foods and drinking. This application could identify foods and drinking and recall their nutrition information from the system database, the volume of foods and drinking could be calculated, and the calories and other nutrition information can be converted.

The whole system was divided into three steps, identify, measure and calculate. Based on machine learning and AR measure, identify and measure would be achieve. And connect with database, calculation would be completed. Some participates were asked to use this application and answered the questionnaires. After using it, they think it is helpful. But this system still has some limits, for no shape foods, it is hard to estimate its nutrition information. And in future work, new models could be build to identify more food.

In conclusion, this application was developed to help people do nutrition management and improve health management.

Key Word(5 words)

Nutrition management, Machine learning, AR measure, Core ML and ARKit

List of Figure

List of Table

1. Introduction

1.1 Background

Nowadays, with the advances in science and medicine technology, the average life expectancy increases steadily. However, the healthy life expectancy does not increase at the same rate. Average life expectancy means the average of how many years people might be expected to live, and the healthy life expectancy is the average of how many years they might live in a "healthy" state. Healthy life expectancy is a key summary measure of a population's health[1].

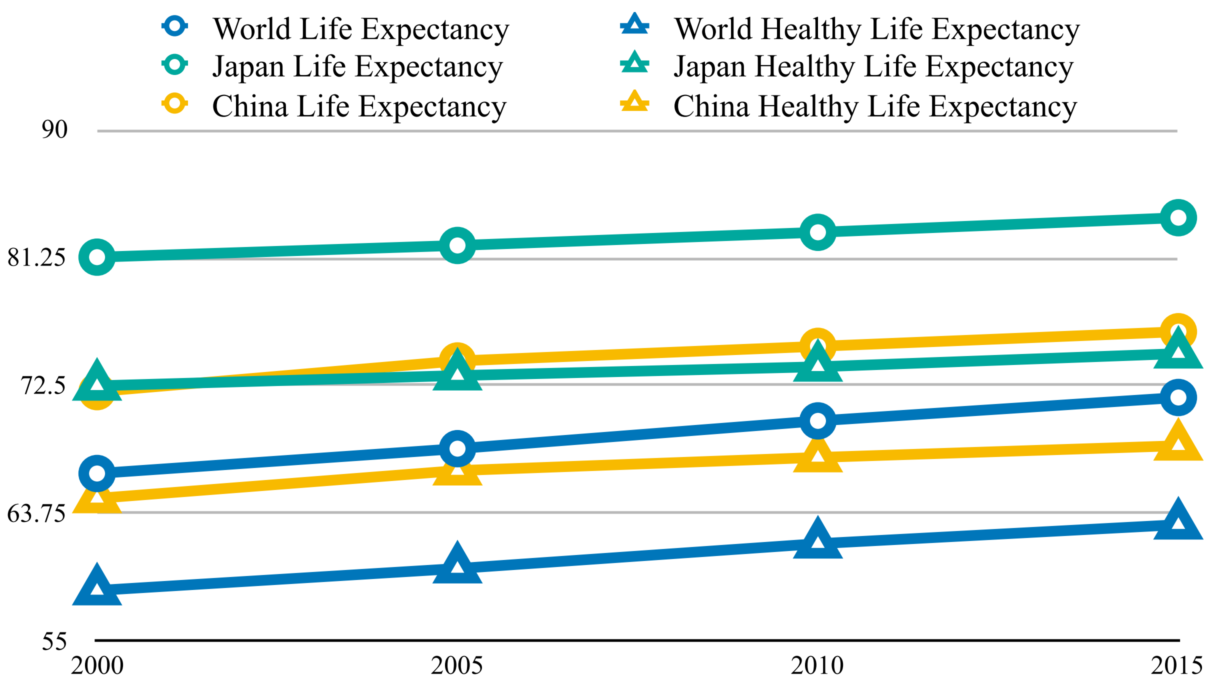

Figure 1: Average Life and Healthy Life expectancy

As figure 1 states, about 10 years of gap between the average life [2] and healthy life expectancy[3] around the world. In terms of specific countries, Japan and China, this difference still exists. In other words, plenty of people live in unhealthy states, such as senior citizens who suffered from the disease in their last few years. One reason for this phenomenon was that several people are in sub-health status with the fast pace of living rhythm. Usually, people do not mention that they are in sub-health status, but the underlying disease already exists. Along with the growth of the age, hypo-function on bodies would induce underlying these diseases.As the population ages and life expectancy increase, health management should become imperative parts as early as possible in our daily lives. Health management is a growing industry in the United States. According to the Bureau of Labor Statistics, from 2010 to 2020 the positions in health management grows to 22 percent[4].

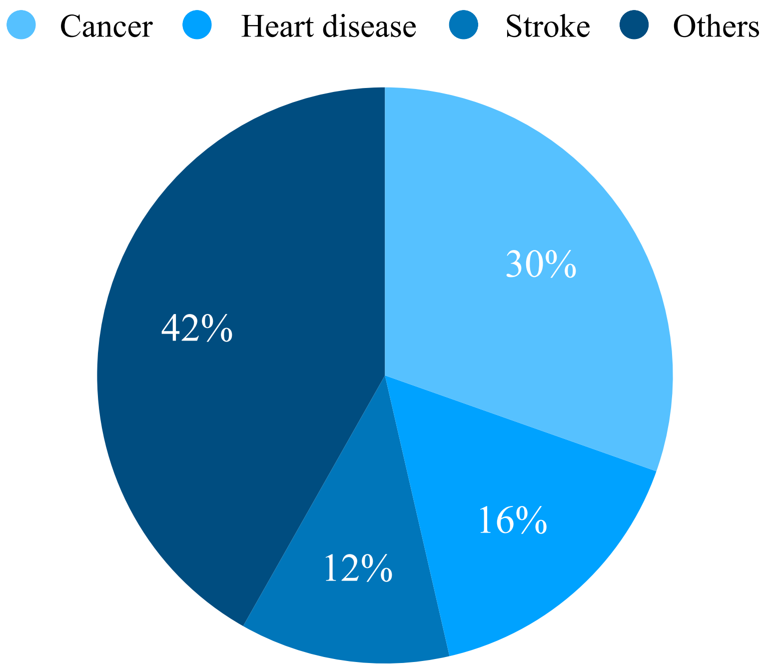

Figure 2: Japanese Death Cause

Japan, which's average life expectancy is over 80 years, is the world's leading longevity country. However, longevity does not represent that they have a long healthy life. The largest enemies of a healthy life are lifestyle diseases, which include cancer, heart disease, stroke, liver disease, kidney disease and diabetes[5]. Patients are usually suffering from these diseases for a long time. As figure 2 has shown [6], in Japan, over 60% of people die because of these three diseases, as figure 2 shows.

1.1.1 Health Management

Health management means health promotion programs, defined by World Health Organizations (WHO) which aim to engage and empower individuals and communities to choose healthy behaviors and reduce the risk of developing chronic diseases and other morbidities[7]. Health management focuses on addressing and preventing the root causes of ill health, not only treatment and cure. It would benefit and protect individual people's health and quality of life and cover a wide range of society and the environment[8].

Health Japan 21[9], a project of comprehensive implementation of national health promotion by the Japanese government. The targets of Health Japan 21 is as following, achieving extension of healthy life expectancy and reduction of health disparities, the prevention of onset and progression of life-style related diseases, maintenance and improvement of functions necessary for engaging in social life, putting in place a social environment to support and protect health and improvement of everyday habit and social environment relating to nutrition and dietary habits, phasic acuity and exercise[10]. In consequence, health management is classified into 3 aspects, diet and nutrition, physical activity and exercise, and leisure and mental health in Health Japan 21.

1.1.2 Health Management Applications

mHealth (mobile health) is a general term of using the mobile phone and other wireless technology in medical care[11], which is a component of eHealth[12]. The most frequent mission of mHealth is the use of mobile devices to educate users about preventive health services. But it is also applied for disease surveillance, treatment support, epidemic outbreak tracking, and chronic disease management.

Along with the widespread of smartphones and enhancement of health management consciousness, abundant health management applications appeared in smartphone applications stores. Health management applications offer health-related services for smartphones and tablet PCs. Users could use these applications both at home and on-the-go. Hence health applications become a part of the movement toward mobile health (mHealth) programs. Health management applications are divided into 2 types, one is designed to help users choose health behavior by offer advice about nutrition and exercise, and the others could help patients communicate with doctors and doctors to keep accurate records of patients' health status[13].

For example, the iOS system, one of the predominant smartphone systems, has its application called Health[14]. This application could help users learn about their health and. It centralizes health data from iPhone, Apple Watch and third-party applications, and it has four keys, which are activity, sleep, mindfulness, and nutrition, as figure 3 shows. Based on ResearchKit and CareKit, the user could communicate with doctors. ResearchKit is a tool to help medical research obtain health data from iPhone, users could take part and allow their data to be used in meaningful studies. And CareKit could help users track users’ symptoms and medications and share this information with the care team.

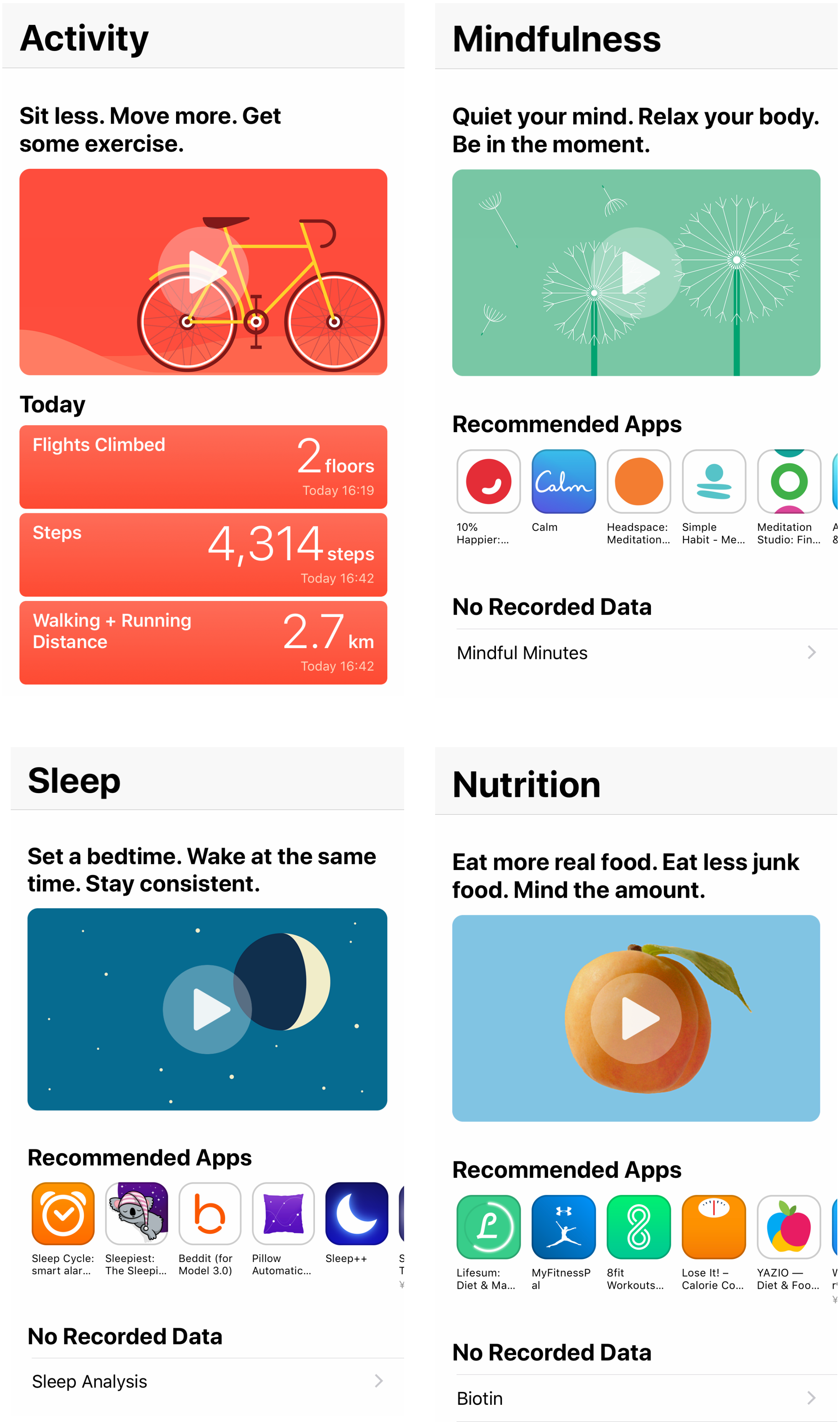

Figure 3: Apple Application Health

Health could record several physical activity data to calculate how much users exercise and users' all-day calorie burn, such as the total walking steps and distances every day. With Apple Watch, the functions were expanded, the heart rate could be measured automatically.

In terms of leisure and mental health aspects, good sleeping could help people restore and repair the body. Compare with third-party applications, which could monitor user's sleep status, these applications not only record sleeping times but also could distinguish users whether they enter deep sleeping to determine sleeping quality.

Health also could count users' carbs, calories or a host of other important nutritional metrics with third-party applications. These third-party applications could help users keep a closer eye on every meal by displaying all the nutritional data. Nutritional data covers various aspects, dietary cholesterol, dietary energy, dietary sugar, caffeine, calcium, iron, water, vitamin, etc.. So it could provide all-around nutrition management to users.

1.2 Problem

As mentioned above, people already have a realize of health management and do it with smartphone applications gradually. Considering a human being's body as a system, the applications which examine physical activities data and sleeping status record the output data from human bodies. On the contrary, the nutrition data in input data into human bodies, as figure 4 shows. This is not easy to obtain since current technology cannot monitor the digestive system at any time. Instead of monitoring the digestive system, recording daily food and drinks nutrition data is an acceptable method to do nutrition management.

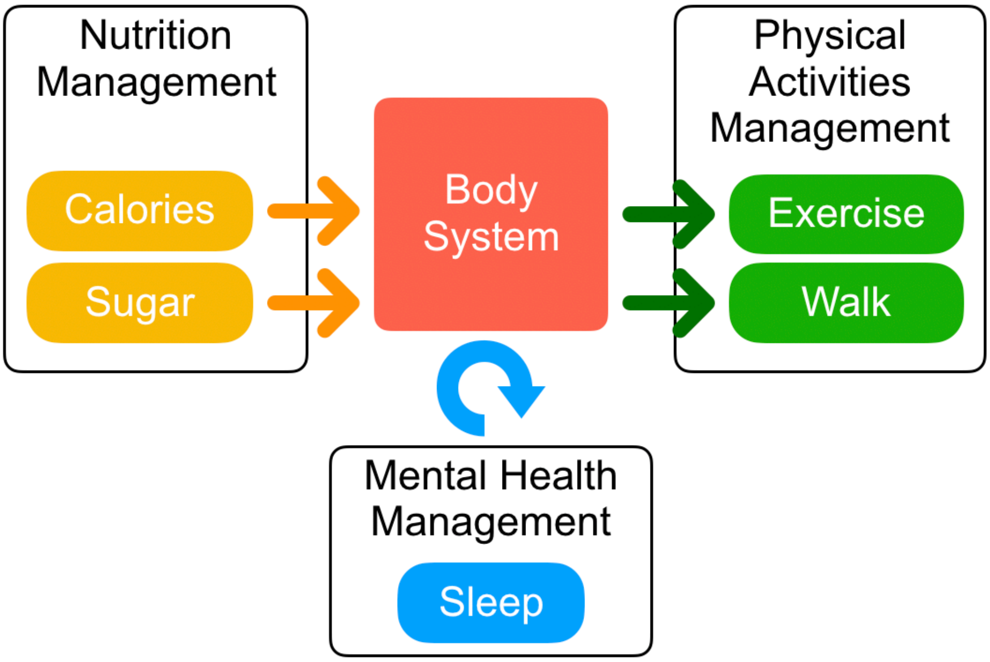

Figure 4: Body System

Though serval third-party nutrition applications are in application stores, the functions of them are single and incomplete. For example, an application called Calomeal [16]. This application, developed by Life Log Technology, Inc, aims to extend human beings' healthy lives from diet aspect. In Japan, with low birthrate and aging, labor people decline and social insurance becomes the main issue. In terms of individuals, the health problem is a serious mission. The main cause of Japanese death, which is related to three major diseases, is an unhealthy lifestyle. And the extensive factors of lifestyle-related diseases are diet, sleeping, smoking, movement, and exercise. Calomeal supports users' record life log to have a healthy life with machine learning, as figure 5 shows.

With Calomeal, users take photos for daily foods drinks firstly. The application could identify food and drinks by AI. Based on the result of AI analysis, the system would indicate nutrition information about food and drinks. As a result, the calories and nutrition information about daily meals are recorded. Besides, this application assists users to manage weight and activities. The nutrition information from this application is a reference value since the volume of foods is unknown.

Figure 5: Application Calomeal

Similar problems also happened in other related applications. Nutrition information about daily food and drink cannot be recorded accurately, and nutrition management has many imperfections.

1.3 Purpose

1.3.1 Improve Health Management

Health not only means the absence of disease, but also means satisfactory conditions of quality, physical, social, intellectual, emotional, and spiritual. Health management is a significant project for our daily life, this study intends to provide human beings with more comprehensive health management, to improve the health states and expand healthy life expectancy. As above mentioned, health management could be enhanced by several aspects, this study would focus on nutrition management, as figure 6 shows.

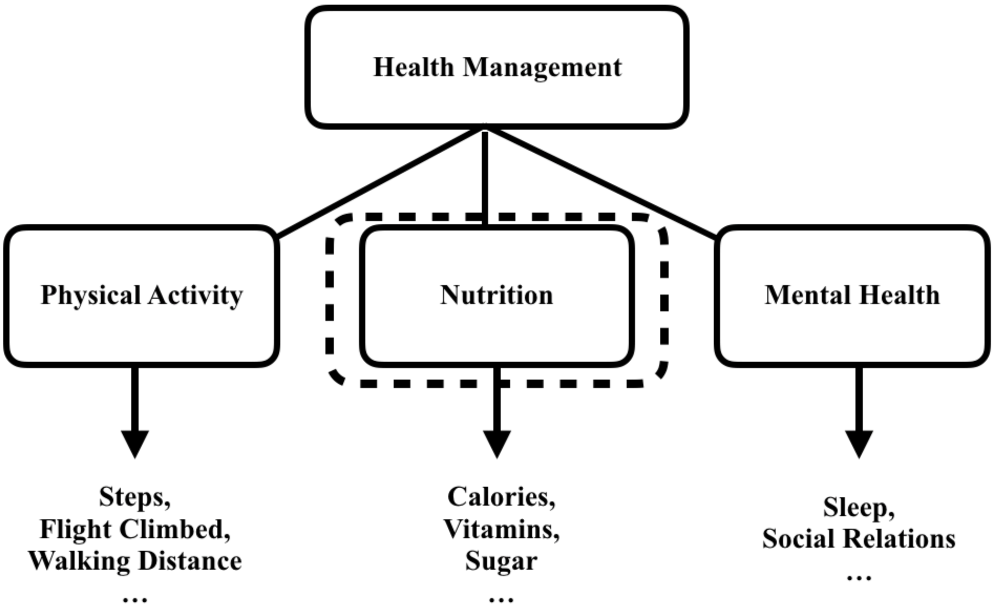

Figure 6: Health Management

Nutrition is the cornerstone of good health and the cutting edge of prevention, which provides all nutrients in both kind and amount[16]. The nutrients from the food we eat, are the most essential continuing environmental factors affecting human beings' growth, development, functional abilities, and health.

Nutrition management, which defined as assisting with or providing a balanced dietary intake of foods and fluids, is one of the best approaches to prevent and control health problems such as heart disease, high blood pressure and some types of cancers.

1.3.2 Nutrition Management Applications

As we know, to do nutrition management, recording the nutrition information of daily foods and drinking is a convenient method. To know the nutrient content of a food, you need to measure the nutrient content of each food and the number of servings of food. In the case of pizza, the same size of Hawaiian pizza and Margaret pizza have different calories. At the same time, the amount of pizza intake will also affect the amount of calories people ingest. To record this information, starting with measuring the volume of foods and drinking and then convert into weight and nutrition information would be a convenient approach.

Figure 7 shows the concept of the proposed system, to get the calories and other nutrition information, the whole system is divided into three steps. In the first step, this application could identify foods or drinks. And in the second step, the system would measure the volume of them. The method of measurement is also related to the type of food and drink. Lastly, the system calculate foods and drinking weight and nutrition data based on volume. If the nutrition data user intake every day is known, nutrition management could be done simply.

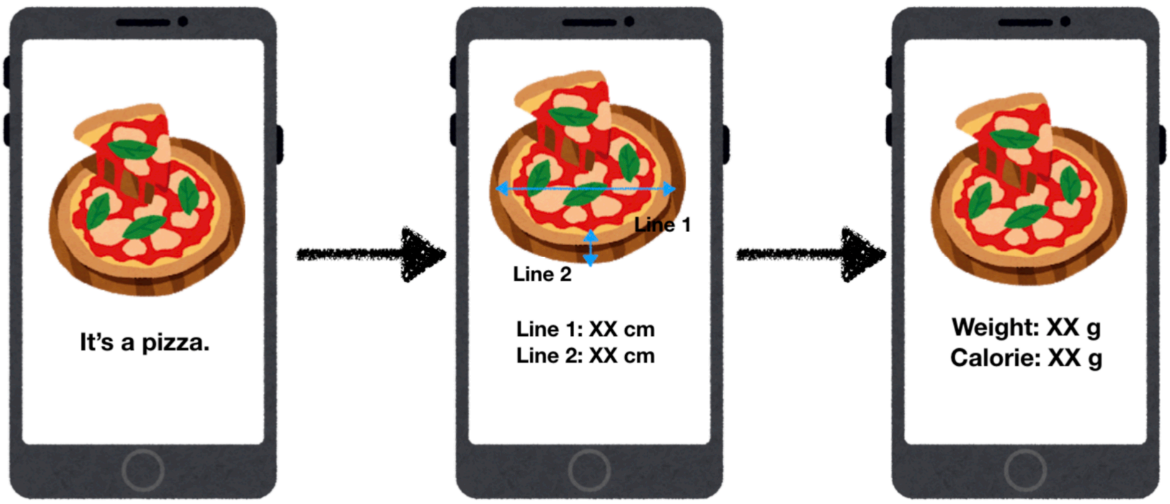

Figure 7: System Design

2. Related Projects

2.1 Related Projects

2.1.1 Food Log

Food Log[37] is a project managed by Aizawa at the University of Tokyo, which is a multimedia food-recording tool that provides a newfangled approach for recording daily food intake primarily for healthcare purposes, as figure 8 displays. It's a novel use of image-processing techniques presents significant potential for the development of new healthcare monitoring applications[17].

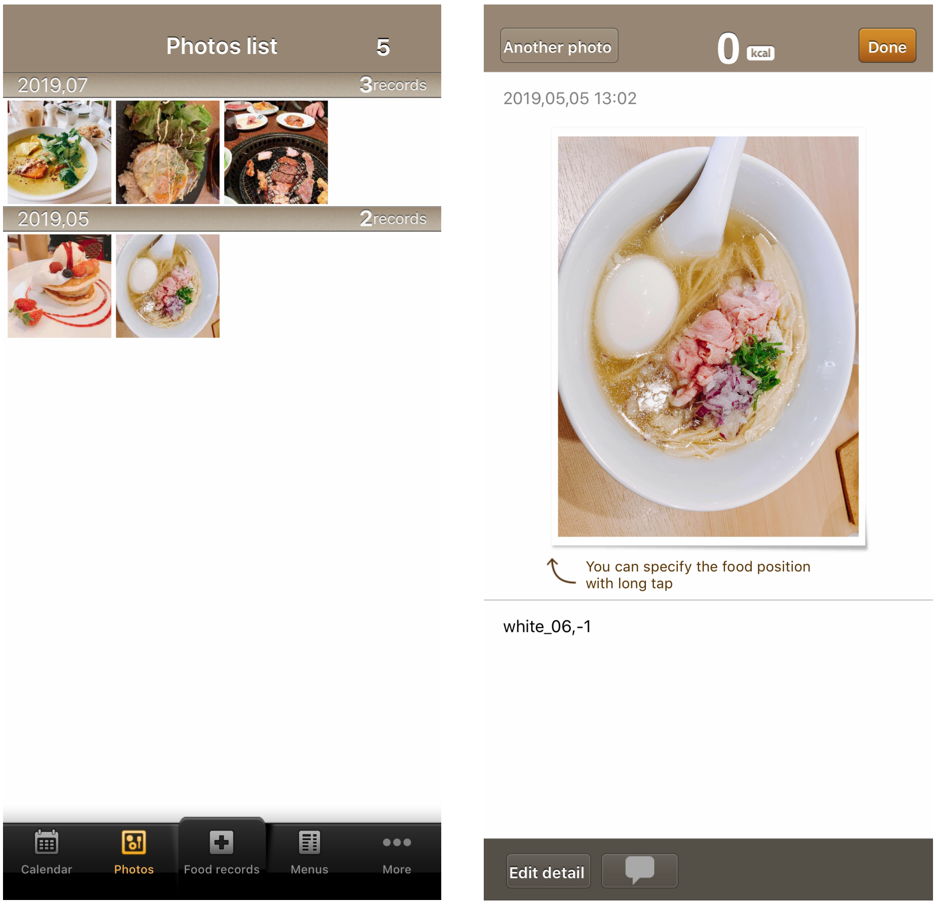

Figure 8: Application Food Log

Food is an attractive topic in the multimedia community, it is indispensable for daily lives and a significant part of fields such as healthcare, agriculture, and food marketing and production. This project focus on investigating the capture, processing, and use of multimedia data related to people's daily food intake, to the health and quality of human being lives. FoodLog, a multimedia food-recording application, provides a fresh method for recording users' daily food and drinks intake primarily for health management purposes.

2.1.2 Food Scanner

Food scanner[19], a project from Fraunhofer, is a food lab insider mobile phone. This system could analyze food and also access to quantity and nutrition facts.

The system in the smartphone will access users to choose quality food and also provide individual nutritional and fitness advice to users, as figure 9 shows. According to the volume from 3D triangulation, the system would provide real nutrition information about food. It also could connect with other applications to get the user's specific data, such as activity data. And then nutrition and consumption can be evaluated and a recommendation for the user can be provided.

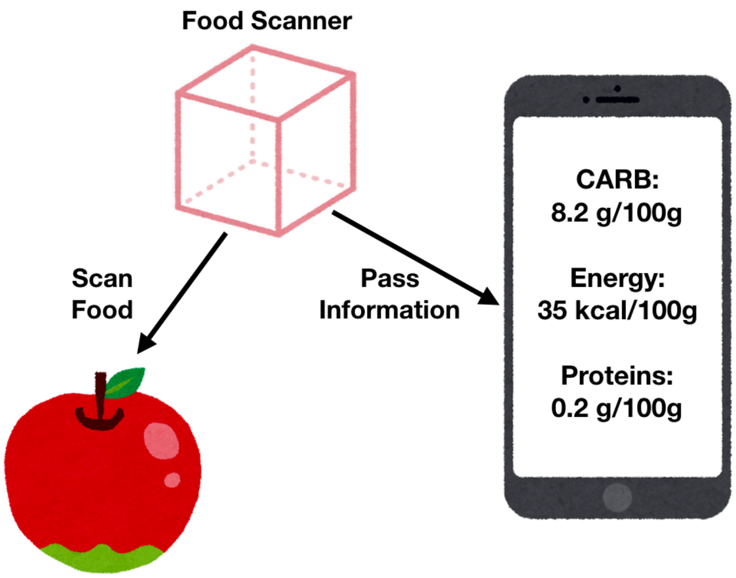

Figure 9: Food Scanner

According to the environmental organization WWF Germany[20], ten million metric tons of food, which are edible, are thrown in the garbage every year in Germany. Food scanner will support consumers and supermarket operators to determine the quality of foods. The ripeness and shelf life of foods is determined with infrared measurements and the results are passed to applications in the smartphone.

2.1.3 Open Food Facts

Open Food Facts [21] is a database of food products with ingredients, allergens, nutrition facts and all the tidbits of information users could find on product labels, as figure 10 shows. In this project, everyone could be contributors, who use the applications [22] in smartphones to scan barcodes and upload pictures of products and their labels. Meanwhile, this database is for everyone, a database is published as open data and can be reused by anyone and for any use.

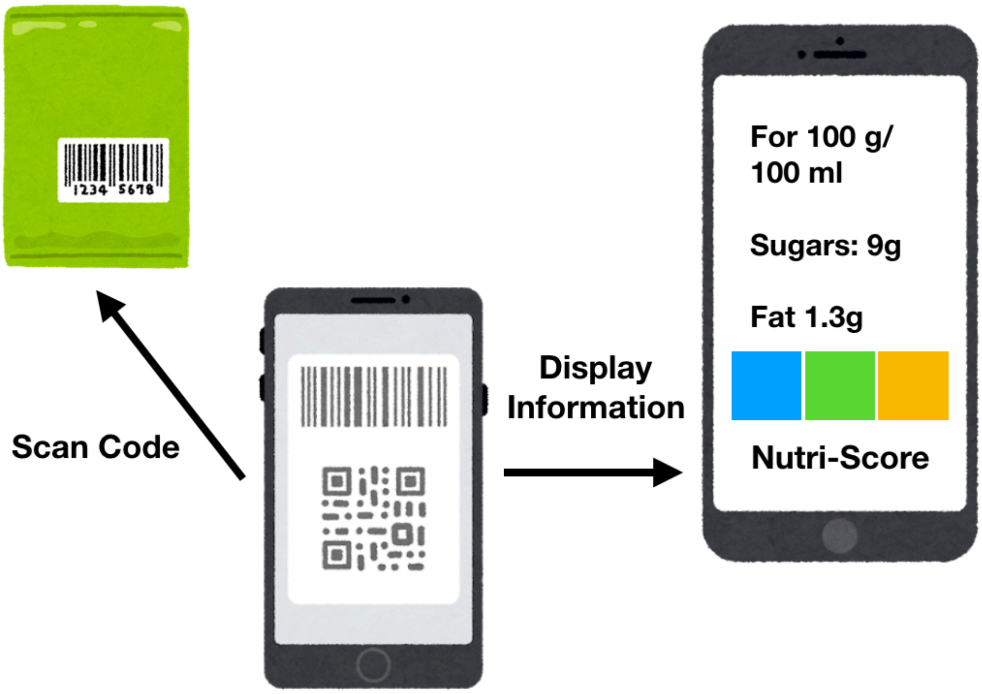

Figure 10: Open Food Facts

This application, which aims to make better food choices, could help the user understand the fine print on product labels, such as food additives E-numbers, allergens, packaging codes, and the nutritional quality of each product. It could help users find the product that matches their criteria. Users also could make comparisons from different brands' products to change for healthier cereals. This database also could be used for thought, innovation, and science.

2.2 Difference

The application in this project focus on measuring the size of food and drinking to get the volume of food and drinking. According to volume, the weight and nutrition information would be estimated. So the key to the application is using AR to measure foods and drinking.

Food Log is an application that could record photos of users' daily foods and drinks, but it does not record the nutrition data of them. But only pictures are not enough for people to do nutrition management, people need to know how much calories or other nutrition data they intake every day. From the application in this research, people could know this information to do nutrition management better.

Food scan could provide food nutrition information with 3D triangulation, but it needs additional devices. The application in this study could provide food nutrition information without any additional device. Besides, food scan is a project, which focuses on determining the quality of foods in the market, to help the consumer or the market whether the need to throw food to reduce unnecessary waste. The core purpose of a food scan is not helping people do nutrition management, providing nutrition information is just associated functions in the application.

Open Food Facts is a database of foods and drinking, which could help users know the nutrition facts of the food. The food with wrapping with labels could be provided with nutrition information easily. But for the food without wrapping and labels, such as foods in the restaurant, Open Food Facts cannot provide the nutrition information. The application in this study breaks in this limit, it could identify foods and drinking with machine learning and measure the volume of them.

3. Methodology

3.1 System

3.1.1 Functions

Nutrition management is one parts of health management, and this system has 3 main functions. The main functions of the application in this study are identifying foods or drinking and measuring their size and calculating weight and nutritions. The functions flow chart is as figure 11 shows. With Core ML, the framework of Apple for machine learning, the application could identify foods and drinking. And then the system would recall the information from the database of these foods or drinking, and based on the measured result with the framework ARKit, the system would calculate the volume of foods and drinking. Based on the volume, the weight could be estimated. And if the weigh is known, the calories and other nutrition information also could be converted with nutrition database. The applications in this study could concentrate on these three functions.

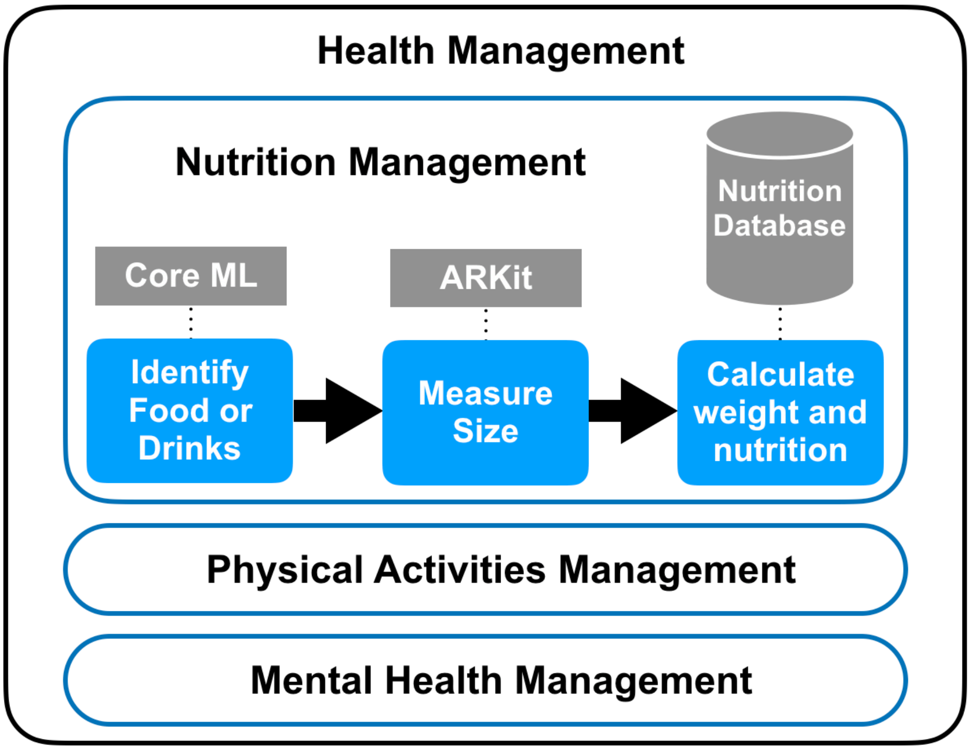

Figure 11: Functions Flow

Afterward, the application would save everyday data and cooperate with other applications, combining with information of physical activities and mental health, to get more health information from users. According to comprehensive health information from several health applications, the system could do health management more preferably and provide users more suitable healthy advice.

3.1.2 Machine Learning

3.1.2.1 Machine Learning

Machine Learning is an application of artificial intelligence (AI), lying at the intersection of computer science and statistics[42], as figure12 shows, it could improve system automatically from experience without any programming and focus on the development of a computer that can access data and use it learn for themselves[23].

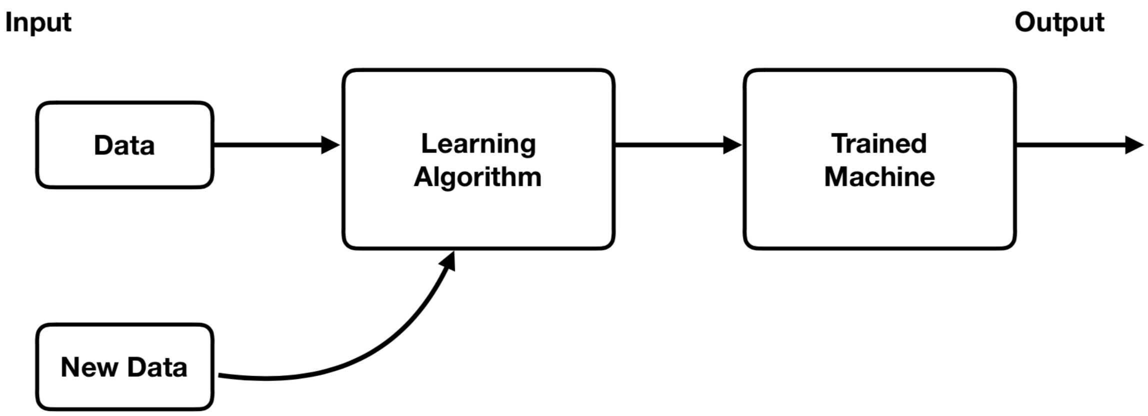

Figure 12: Machine Learning

The process of learning starts with observations or data, which is aim to look for patterns in data and make better decisions in the future based on the examples, it allows the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning can analysis of massive quantities of data. It could deliver faster, more accurate decisions to identify profitable opportunities or avoid dangerous risks, but it also needs additional time and resources to train.

Machine learning could be divided into classified as supervised, unsupervised and reinforcement[24], as figure 13 shows. For supervised learning, the data set is given and the output is also known, it was classified into regression and classification, as figure14 states. It could be applied to cancer prediction[43]. Unsupervised learning, as the name suggests, the

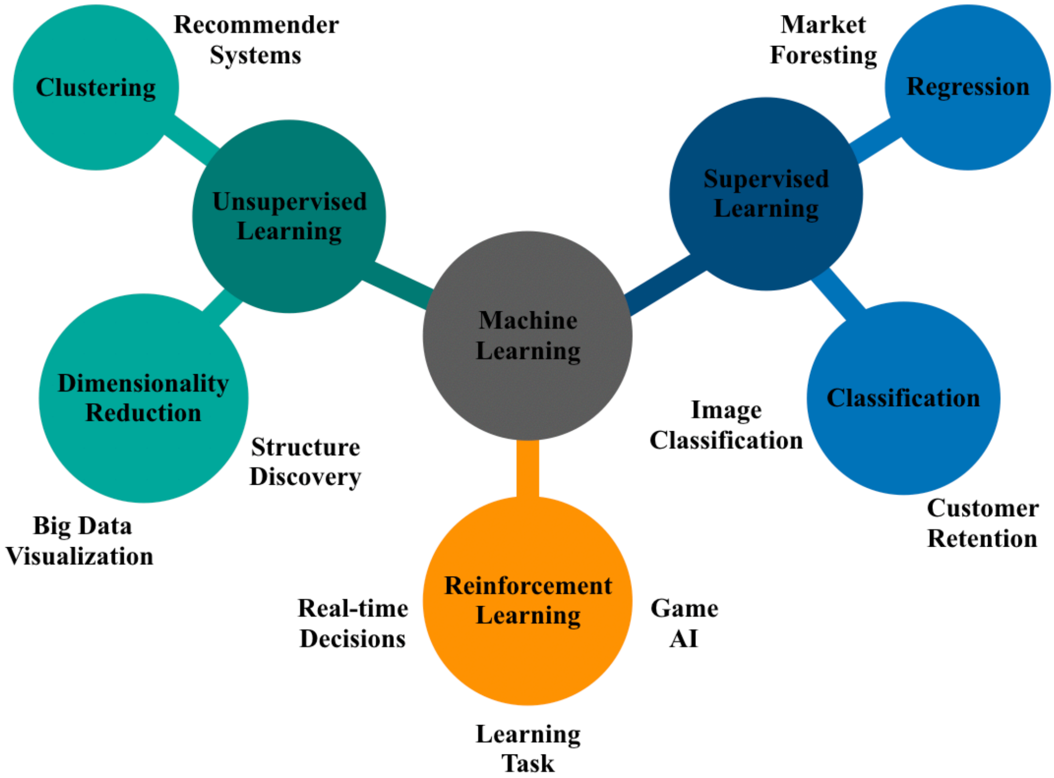

Figure 13: Machine Learning

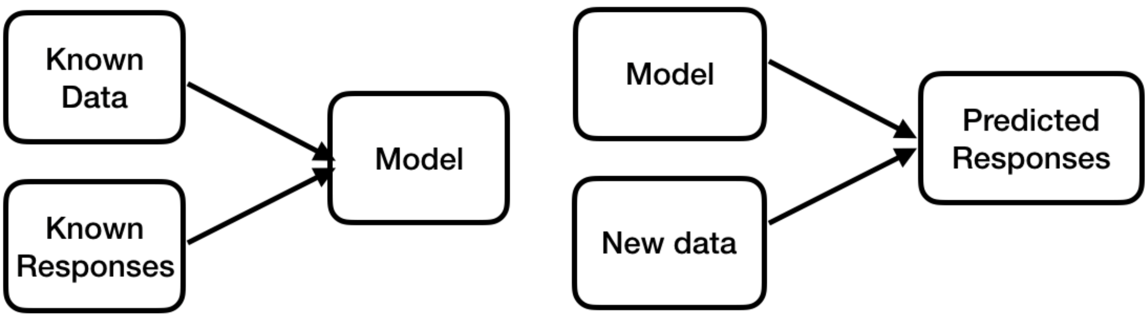

Figure 14: Supervised Learning

data set is also given, but the label of data set is unknown. Developers should find the structures from the data set. It is divided into clustering and dimensionality reduction.

Reinforcement learning means taking applicable actions to obtain maximum rewards in a particular situation. So the software and machines optimize operation in the specific situation. The difference of reinforcement learning is that it decides what to do to perform in the specific task and does not has the correct answer. But supervised learning focus on finding the correct answer.

3.1.2.2 Advantages of Machine Learning

Machine learning could identify trends and patterns easily and do not need human intervention. Since it has handling multi-dimensional and multi-vanity data and this data could be added continuously, the system also could be improved continuously. Meanwhile, it is also able to apply in applications widely[25].

Figure 15: Google Home Mini

One of the machine learning applications is virtual personal assistants, which assist in finding information, when asked over voice, such as Amazon Echo and Google Home[44], as figure 15 shows. Machine learning is a signification part of these virtual personal assistants, thee could accumulate and improve the information based on previous effects with them. According to this data, the system would provide better information which is tailored to users' preferences.

In terms of the application in this study, based on the original database, the system could identify several foods and drink. And with the mobilization of the applications, the system would collect more images of foods and drinking and extend the database at the same time. That is the system upgrade itself without manual work.

3.1.2.3 Core ML

Core ML[26] is a foundation for the domain-specific framework for machine learning from Apple, which is based on Swift. Developers could use Core Ml to integrate machine learning models into applications, and it provides a consolidated representation for all models. The application could use Core ML APIS and user data to make predictions and to train or fine-tune models on the device of users.

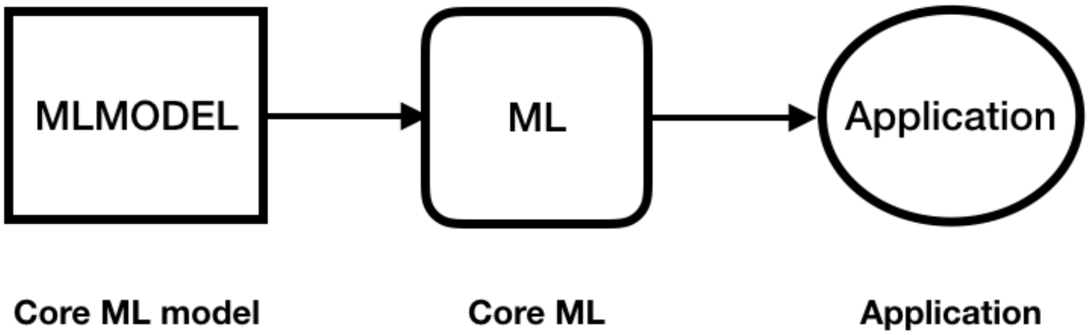

Figure 16: Core ML Model

As the figure 16 shows, the flowing is from left to right. Core ML model is a result, which was applied to a machine learning algorithm to a set of training data. Based on new input data, the model could make predictions. The model also could complete various tasks, which are difficult or impossible to write in code. For example, the model could detect specific objects from photos with its pixels.

The key to Core ML is Core ML model. Developers could download several Core ML models from Apple's official website. The models used in this research are called SqueezeNet, which is a small deep neural network architecture, it could classify the dominant object in a camera frame or image.

Furthermore, developers could build their models to match their needs with Create ML[27]. With Create ML, developers could their database for models[45]. In this process, plenty of pictures are added into the database, and the system would “remember” these objects with pictures. This process likes humans to learn new knowledge or skills, “adding pictures” is familiar with the practice process. So the database could be updated easily and quickly. With these methods, the application in this study could identify more kinds of foods, even differentiate similar foods.

Leveraging the CPU, GPU, and Neural Engine, Core ML could optimize on-device performance, and minimize the memory footprint and power consumption. A network connection is not necessary when running the model, this point would protect the privacy of the user's data and application responsive.

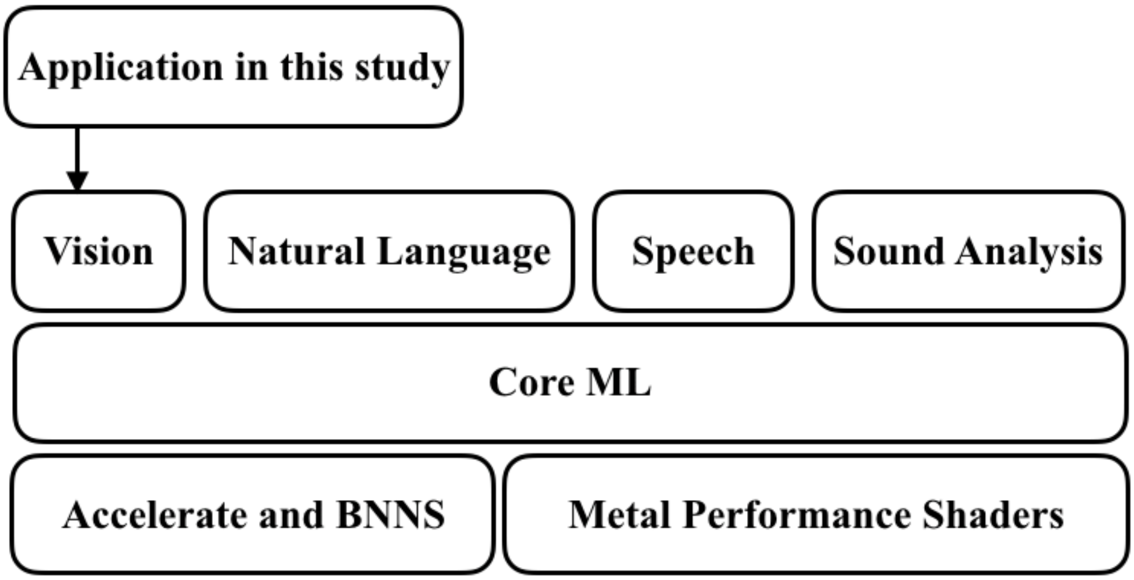

Figure 17: Core ML

Core ML builds on top of low-level primitives, such as Accelerate, BNNS, and Metal Performance Shaders, it could support Vision for analyzing images, Natural Language for processing text, Speech for converting audio to text and SoundAnalysis for identifying sounds in audio, as figure 17 shows. The application in this study uses the function of analyzing images to identify foods.

3.1.2.4 Core ML Example

Core ML could be applied into various applications, since it has well-made functions. One example of Core ML applications is using it to build an application which could recognizes hand-drawn digits[39]. The basic steps for integrating a Core ML model into iOS application with Core Ml is as figure 18.

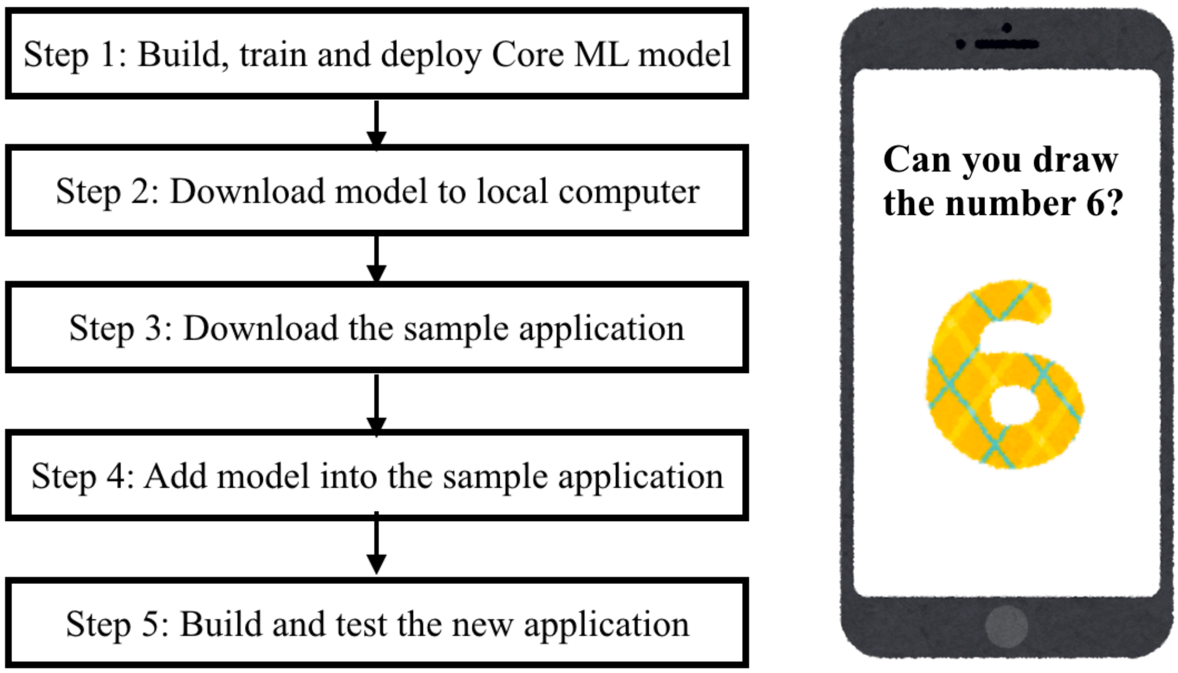

Figure 18: Application Digit

It the user draw the number in the screen by hand. As we know, the handwritten number is quite different every time, which is not that standard. But the system could recognize it and determine which is match the question. It is a basic functions of Core ML, recognizing the objects. In this process, with the handwritten number in the screen increasing,

3.1.2.5 Solution

Core ML is framework, which cannot be used directly by users, to apply Core ML and Vision in iOS device, some program process is necessary. And programming would be done in Xcode, an integrated development environment for macOS, it also includes a set of software development tools developed by Apple for developing software for Mac OS, iOS, iPadOS, watchOS, and tvOS. So the pre requisites for this process are MacOS, Xcode 9 or above and devices with iOS 11 or above.

Firstly, a Xcode project was created, Xcode has many preset. To create machine learning application, single application was selected. Meanwhile , the programming language should be swift.

To create the user interface, Xcode has storyboard to do is easily, which is a visual representation of the user interface of an iOS application, it shows screens of content and the connection between screens. Each storyboard represents a set of scenes, such as view controller and its views. And these view controllers could be connected by segue objects. So in this application storyboard including the following components, UIButtons, UIImage and UILabel. UIButtons were used to input photos, UIImages was used to display input photos and UILabel was applied to display the name of foods or drinks.

```
import UIKit
import AVFoundation
import CoreML
import Vision
```

Figure 19: Import framework

Meanwhile, in swift file, the project needs import UIKit. AVFoundation, Core ML and Vision framework would be imported in later steps, as figure 19 shows.

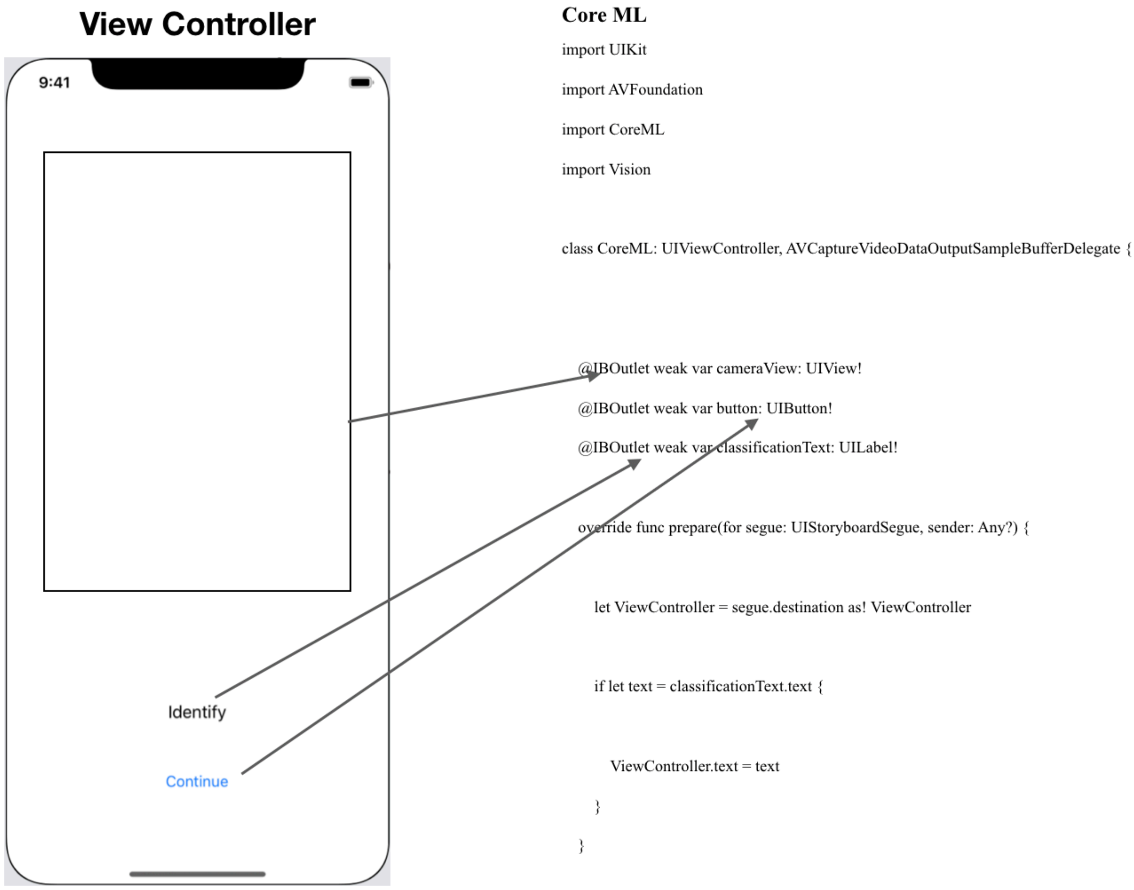

Figure 20: Component outlet

After adding components in storyboard, some components were outlet to programming editor, as figure 20 shows. IBAction in programming editor outlet for each UIButton, UILabel outlet for the label to show identify results. And UIImage outlet for the image view. Next, to use the UIImagePickerController(), class should be extended with UINavigationControllerDelegate and UIImagePickerControllerDelegate.

This application needs to access to iPhone camera and photo for predictions, the key in privacy should be added, Camera Usage Description and Value. The initial setup was completed by this step.

And then adding Core ML model to the project, Apple has some open source model could be download in office website. After downloading, the model was dragged and dropped into the Xcode project structure. The input and output parameters and other information of model should be checked. The information of model showed the input the model takes in and the output the model returns. In this project, it takes in a 299x299 images, and would return with the most like category, and the probability of each category.

Clicking the arrow in front of the model label in the Model Class section to open the model class, Xcode would generate for Core ML model automagically. To get the expected prediction as output from the model, the expected input should be given. Importing Core ML and the Vision framework into ViewController.Swift file to convert the image.

To using Core ML, some code are necessary. A generate method to use Core ML model is prediction(image:), it is used to predict the object in the given image. With pixelBuffer variable, the size of image could be resized. Once the prediction is return, which is String type. The classifier label would be updated to set its text to what it has been recognized.

The machine learning parts process was as above. The project could identify the foods and drinks, and pass these information to the next view controller, ARKit, with segue.

3.1.3 Argument Reality

3.1.3.1 Argument Reality

Argument reality (AR) means the integration of digital information with the user's environment in real-time[46]. Different from virtual reality (VR), which creates an artificial environment, AR adds new information into the existing environment.

The main approaches to enable AR works are simultaneous localization and mapping (SLAM), recognition based and location-based[28]. SLAM is the most effective approach to present virtual objects in real-world objects, it localizes sensors concerning their surroundings, and at the same time map the structure of the environment. SLAM is aim to solve simultaneous localization and mapping problem and now every augmented reality development kit could provide SLAM functionality. Recognition based means using a camera to recognize visual markers or objects to present a cover when the marker is observed by the

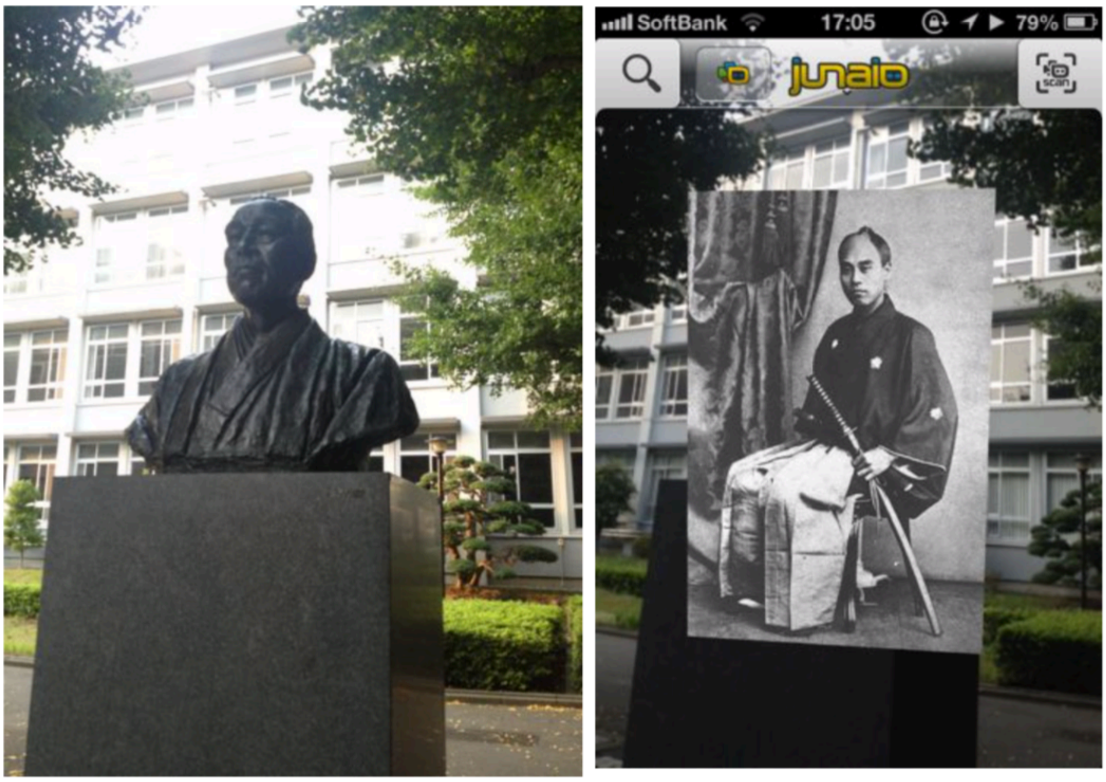

Figure 21: Marker-based AR

device, as figure 21 shows. Location-based, different from recognition base, is based on GPS digital compass, velocity meter, or accelerometer to provide location information and input augmented reality visualizations. It is also called marker-less augmented reality, as figure 22 shows.

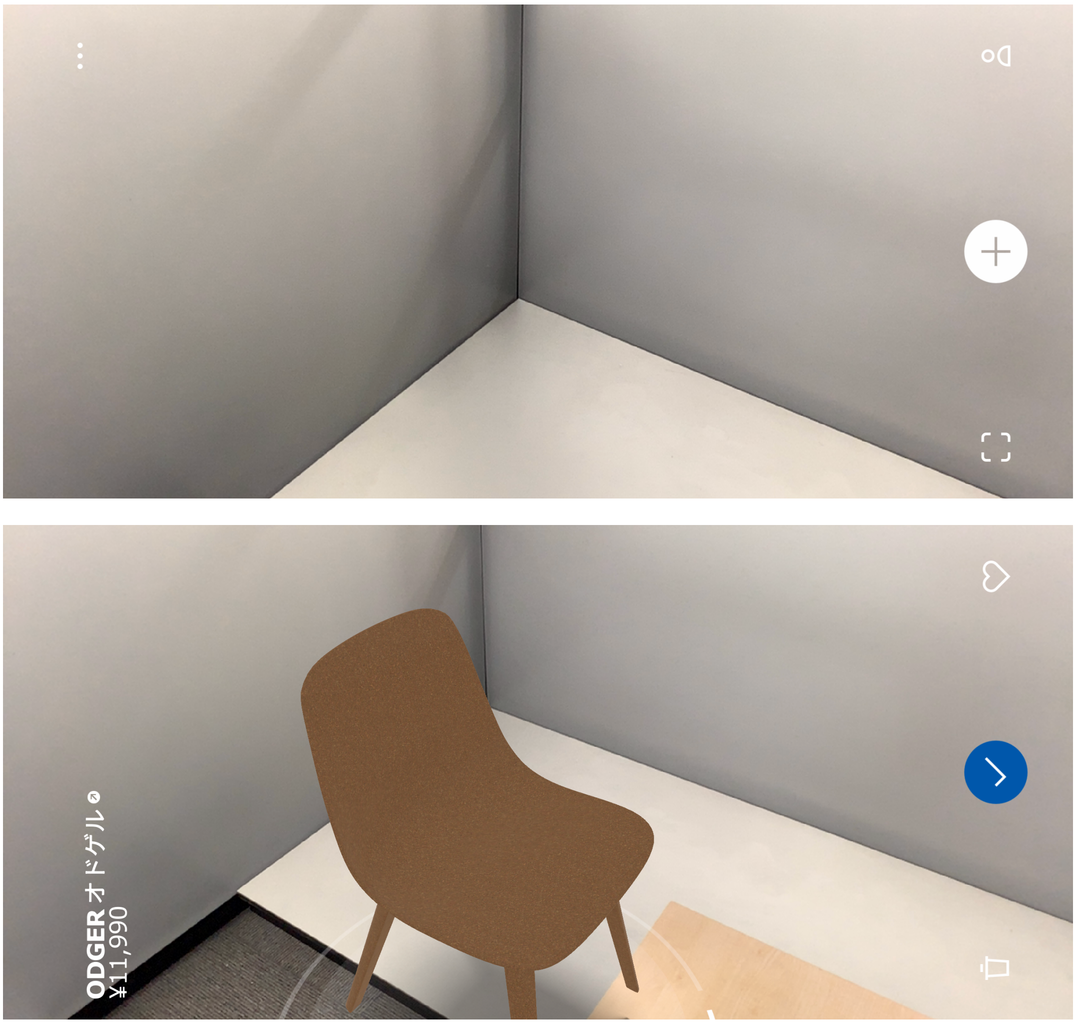

Figure 22: Marker-less AR

As above figure 19[41] and 20 show, Marker-based AR recognizes a simple marker and displays the real things in the screens. But for marker-less AR, it could recognizes object without any markers and meanwhile add some virtual information into real environment.

In AR applications, a live, onscreen view of the physical world is presented by the camera. And on this view, three-dimensional virtual objects are superimposed to create the illusion that they exist veritably.

3.1.3.2 ARKit

ARKit[29] is another framework of Apple to produce augmented reality experiences in applications by integrating iOS device camera and motion tracking functions. ARKit could blend virtual objects with the real world seamlessly to deliver impressively and engage experience.

Creating and tracking a correspondence between the real-world space the users inhabits and a virtual space where users can model visual content is the basic requirement for AR and the defining feature of ARKit[38]. To create this relationship, ARKit uses a technology called Visual-Inertial Odometry.

ARKit combines the benefit of device motion tracking, camera scene capture, advanced scene processing, and display conveniences, which could simplify the processor building an AR experience. In WWDC 2019, ARKit 3 was released[40], now it has the following functions, people occlusion, motion capture, simultaneous front and back camera, multiple face tracking, collaborative sessions. Among these functions, People occlusion, a new function of AR, means AR content could pass behind and in front of people in the real world, it would make AR experiences more immersive and enable green screen-style effects in almost any environment at the same time. Motion capture means the system could understand human body position and movement, it could track joints and bones. So it would provide a new AR experiences involved with people and objects at the same time. ARKit 3 was also reformed in other aspects, it could detect up to 100 images at a time, in complex environment it could detect 3D-object more robust. For facial recognition, it would use front and back camera simultaneously, which allow users interact with AR content in new ways, with new iPhone device it can track up to three faces at once.

ARKit could track motion fast and stable since VIO blends CoreMotion data and camera sensor data. These two inputs allows the system know how device move more accurate without any need for additional calibration. It also makes object look like being kept in actual environment, rather than hovering over it. ARKit also makes full use of the camera sensor to decide the amount of light present in a scene and apply the idealize amount of lighting to virtual objects. ARKit has high-performance hardware and allows impressive levels of detail and visual fidelity since it was supported by Scenekit, Metal and third-party tools[30].

3.1.3.3 AR Measure

AR measurement is one function of AR, which could gauge the size of real-world objects in real-time. The principle of AR measure is that the captured frames of the camera are processed by the program, and computer vision tries to identify the track some relevant points in the images, figure 23 shows the flow charts. The tracked points are called anchors[31], as figure 24 shows. These anchors were identified, and as device moves, the perspective of the camera is also moved. The position of anchors is updated, as figure 25 shows. With an amount of processing, the distance could be estimated.

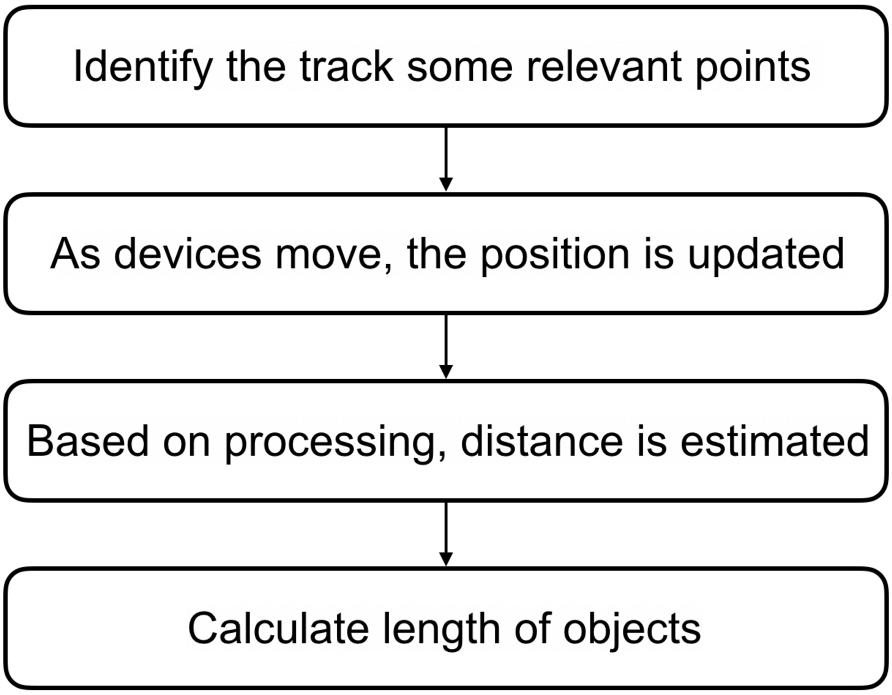

Figure 23: AR measure principle

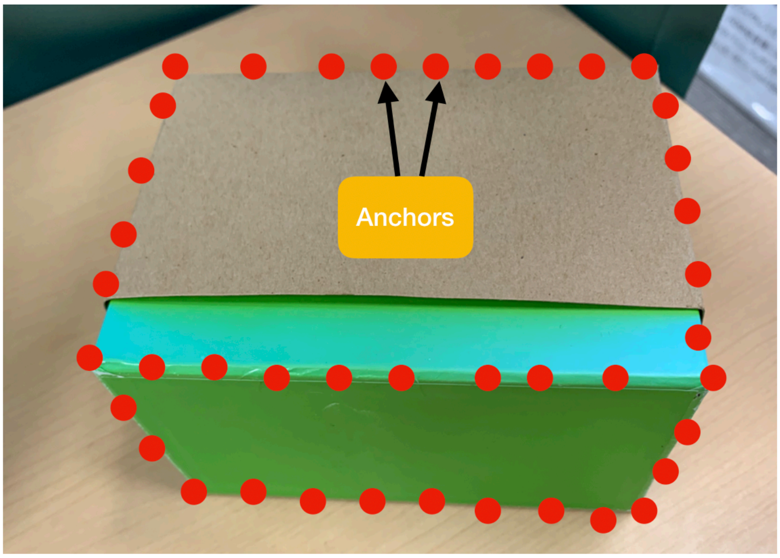

Figure 24: Anchors

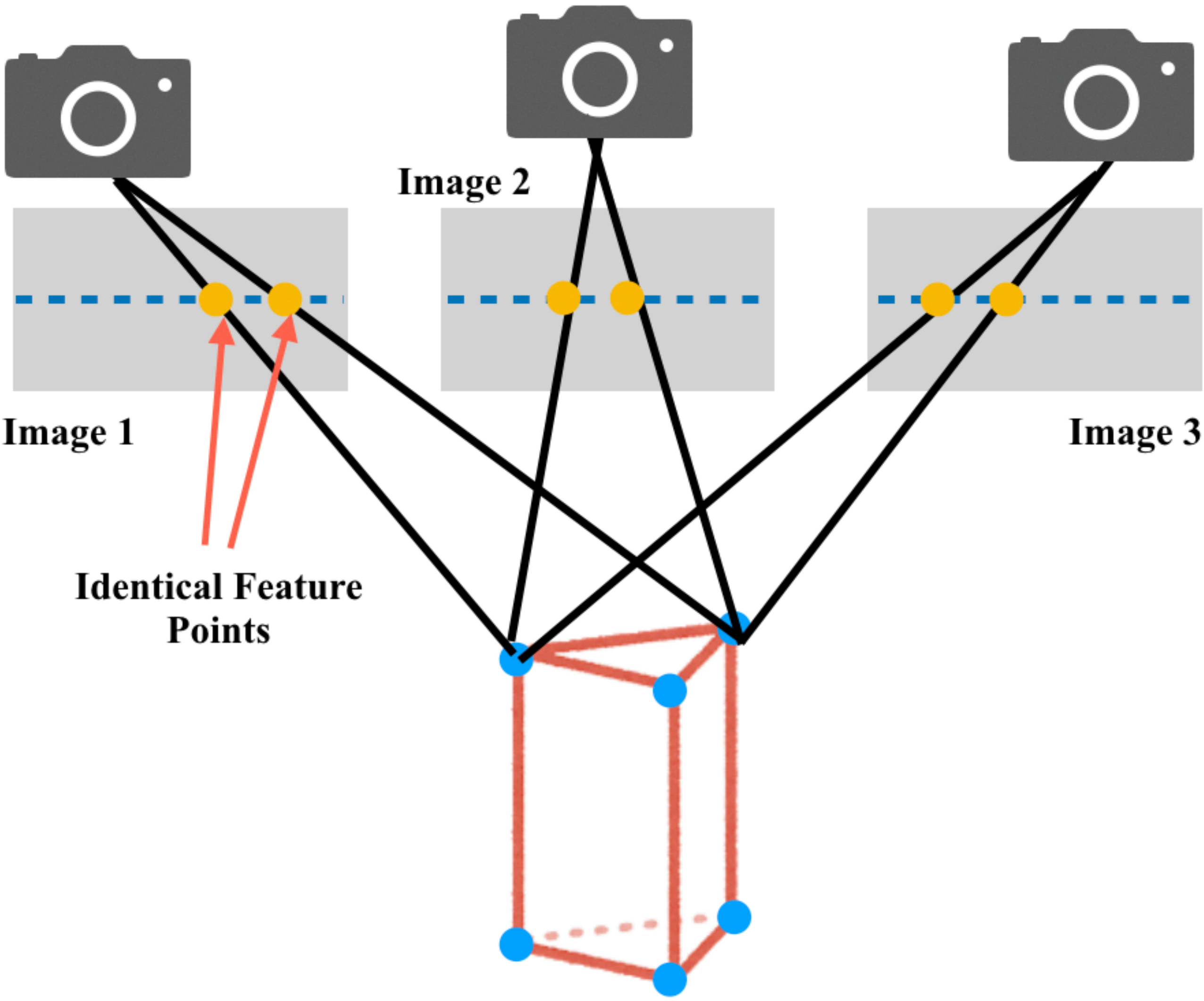

Figure 25: AR measure Principle

3.1.3.3 ARKit Example

Apple released an application called Measure[32] in September 2018, which could gauge the size of real-world objects quickly, detect the dimensions of rectangular objects automatically with the iOS device camera, and also save measure result to a photo, as figure 26 shows.

On the subject of the application in this research, AR measure could help users get the volume of foods and drinks in real-time.

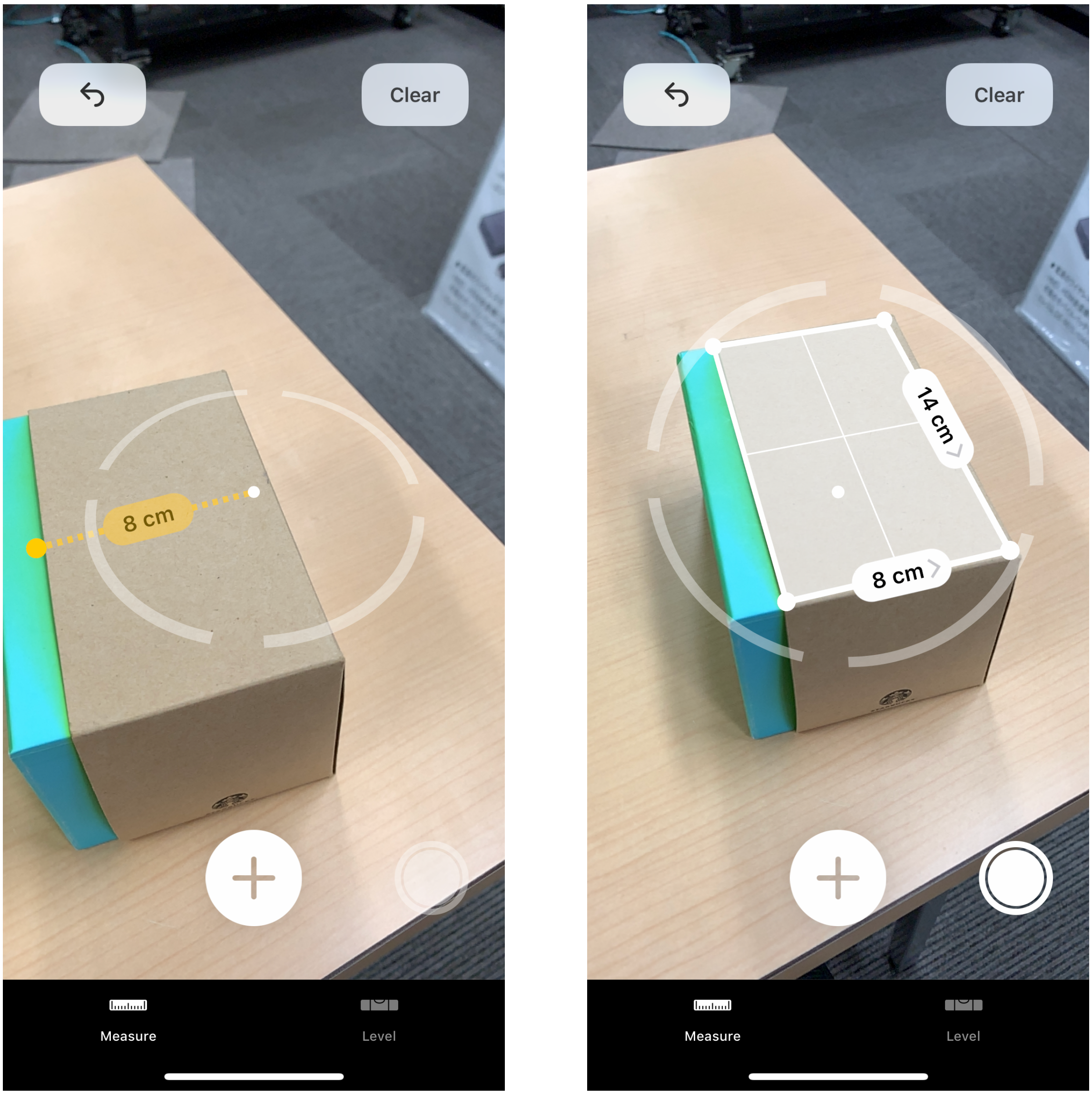

Figure 26: Application Measure

3.1.3.4 Solution

An ARKit project was created to measure the size of foods and drinking. When creating a new project, this time using the "Augmented Reality App" template. Except ARKit, this project would also use SceneKit for its 3D and 2D feature.

Camera Usage Description in privacy still need to be added to this project, since augmented reality requires the use of a camera.

A new swift file was created, in the measure view controller file, UIKit, SceneKit and ARKit were imported with the ARSCNViewDelegate, which provide SceneKit content corresponding to ARAnchor objects tracked by the view's AR session and manage the view's automatic updating of such content.

In storyboard, add a new view controller, and ARKit SceneView object. To set up the SceneView, add the IBOutlet to the SceneView object in the storyboard file, create a set scene function and set up the ARSCNViewDelegate. And then set up the ARSession, create a world-tracking session with horizontal plane detection and implement the ARSCNViewDelegate function. Then create a sphere.

```
func distance(from vector: SCNVector3) -> Float {
    let distanceX = self.x - vector.x
    let distanceY = self.y - vector.y
    let distanceZ = self.z - vector.z

    return sqrtf((distanceX * distanceX) + (distanceY * distanceY) + (distanceZ * distanceZ))
```

Figure 27: Calculate distance

To measure the distance between two points, an extension to SCNVector3 was added for calculating the distance between 2 vectors, as figure 27 shows. Calculate the distance between two points and draw a line between them, add the extension to eh SCNNode with functions to achieve it. And the UI to display the distance and a point to recognize the center of the screen.

A new SCNNode class was created, the distance would be presented as a node in the SceneView. And adding the TextNode to the SceneView. And the measure result would be get. For example, figure 28 shows the diameter of the pizza.

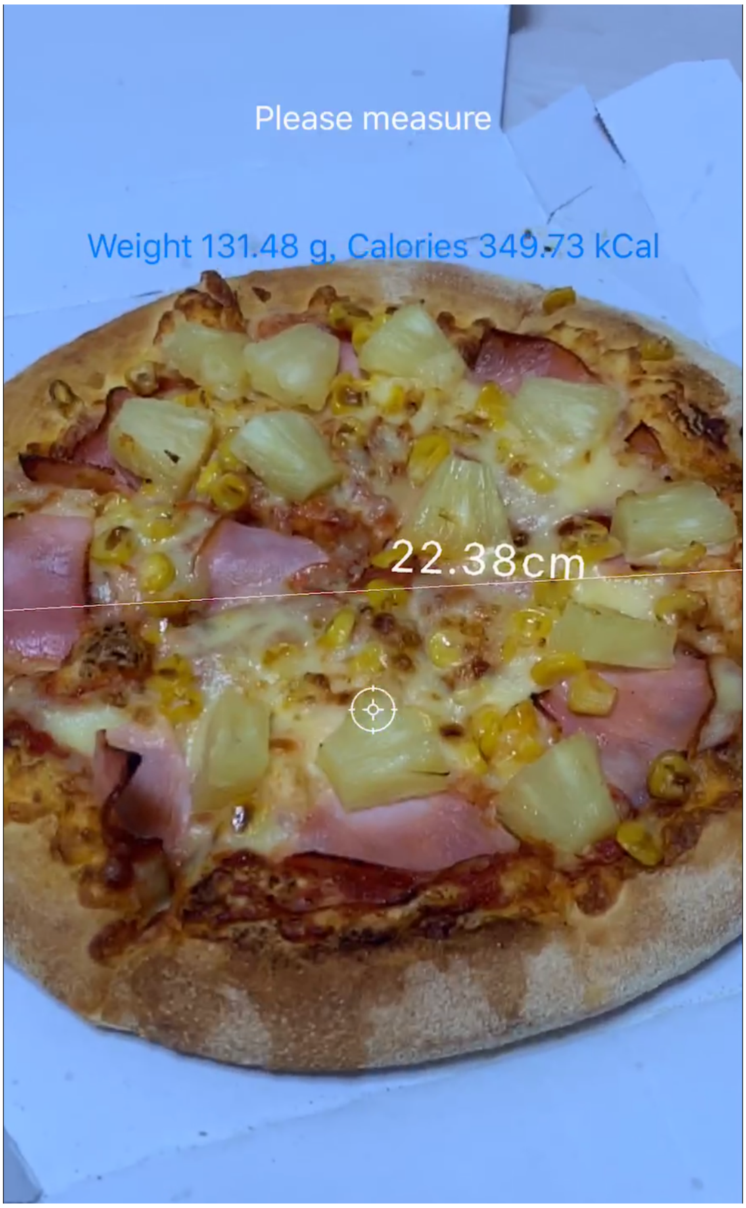

Figure 28: Measure pizza diameter

3.1.4 Segue

Segue[33], which is a UIKit prepares for and performs the visual transition between two view controllers[34], defines the flow of the application's interface. Segue defines a transition between two view controllers in the application storyboard file. The starting point of Segue could be a button, table row, or gesture recognizer that initiates the segue, and the endpoint would be the view controller should be displayed, as figure 29 shows.

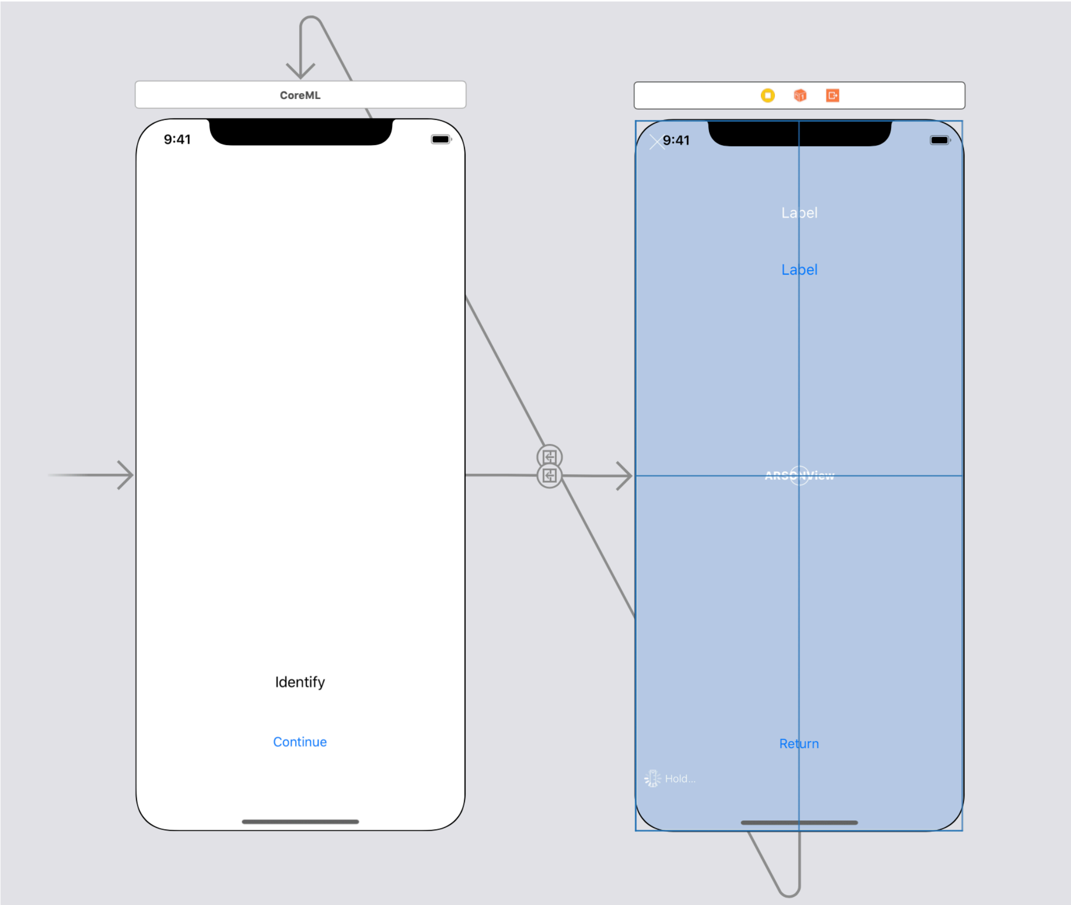

Figure 29: Using Segue to connect to Storyboards

Segue has four different types, show, show detail, present modally and presents as Popover. In this application, the type is shown, which displays the new content using the showViewController:sender: method of the target view controller. For most view controllers, this segue presents the new content modally over the source view controller.

In terms of this application, with segue, Core ML and ARKit, which are in two different view controller, could be combined in one application. These two view controllers were created independently before. And then open ARKit view controller as initial, and add Core ML view controller into storyboard and two new buttons into each view controllers as connect buttons. When the object was identified by Core ML, segue passes this information to ARKit to her system to calculate the calorie and nutrition data. So after identifying, users need to click up the connect button, the screen would jump to measure screen, then users could measure the size of food.

3.2 System Structure

Considering the application as a whole system, and its elements could be divided into two types, software and hardwares. For hardwares, it has iPhone and camera on it. For software, it has two frameworks, Core ML and ARKit. These elements are independent objects, but they have some connections for each other to contribute the application.

The system structure of this application is simple, base on hardware iPhone and its camera, camera is one parts of iPhone. Swift is the programming language for this application, which is the basic of software part. Core ML and ARKit are two frameworks of Swift, one is for identify and the other is for measurement. To achieve the functions of these two frameworks, the camera would be used. Core ML and ARKit are two independent frameworks, this application uses Segue to combine these two functions, which is a UIKit of Apple. With these connection, the system could implement integral functions, as figure 30 shows.

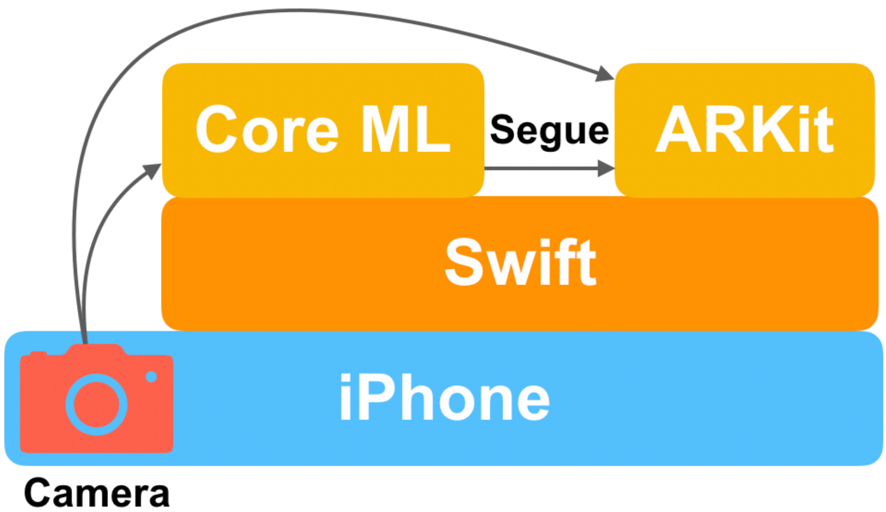

Figure 30: System Structure

4. Verification & Validation

4.1 Verification & Validation

Verification, a process to proof of compliance with specifications, determines whether or not a product meets the requirements or specifications. In this system, identity and measure both should be verified.

Validation, a process to proof user satisfaction, means the assessment of whether the planned or delivered system fulfills the sponsor's operational need in the realistic environment.

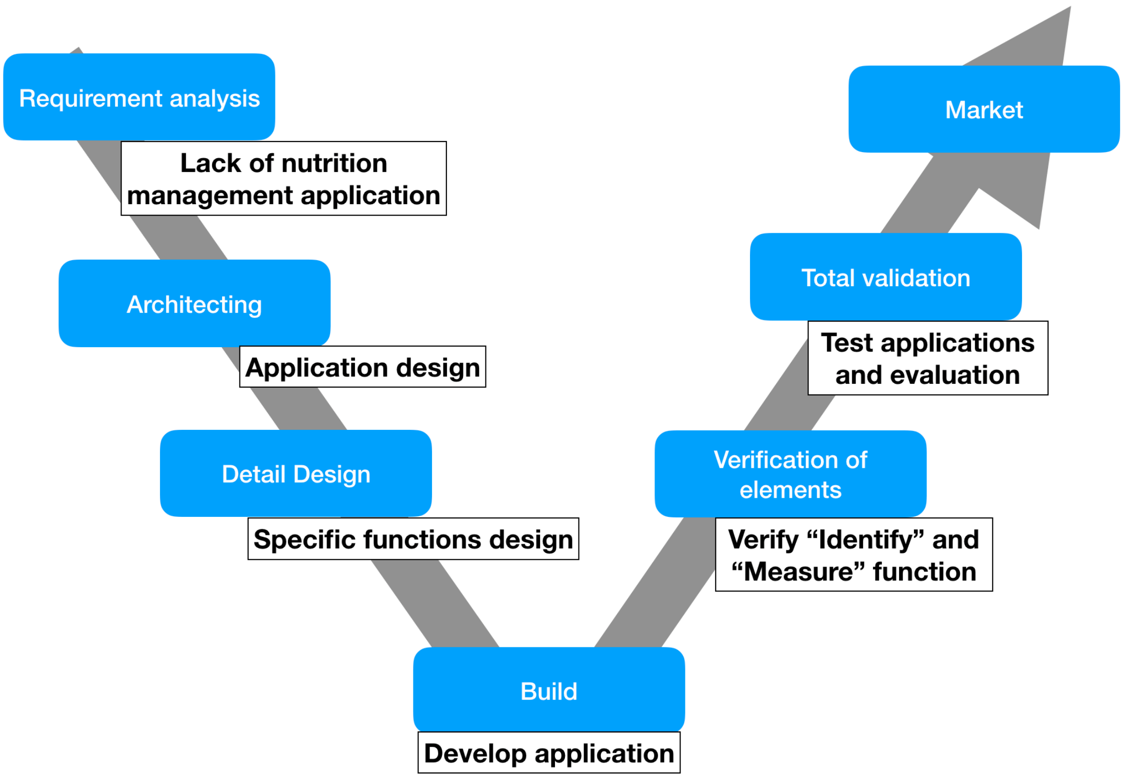

Figure 31: System Design

Verification and validation support the development process, ensure the developer provers credible results and satires the customers. The main techniques of verification and validation are inspection, analysis, demonstration and test, and to collect data for verification and validation, questionnaire, interview and observation are common methods.

In this system, verification means verifying the functions of identifying and measure, and validation means this application whether meets users' needs, which could help users get the nutrition information quickly and convenient, as figure 31 shows.

4.1.1 Identity

Using this application to identify several kinds of objects, such as bottled water and pizza. The system could identify them correctly, and it also indicates how much percent it matches to result. Figure 32 shows the results, the result of pizza is 91.49%, the accuracy of which is quite high. The result of measurement of a bottle of water is 77.22%, which is also very accurate. Most food objects could be identified and the probability could be calculated scientifically. As the number and types of objects increase, the ability of recognizing the objects will be stronger due to the machine learning function.

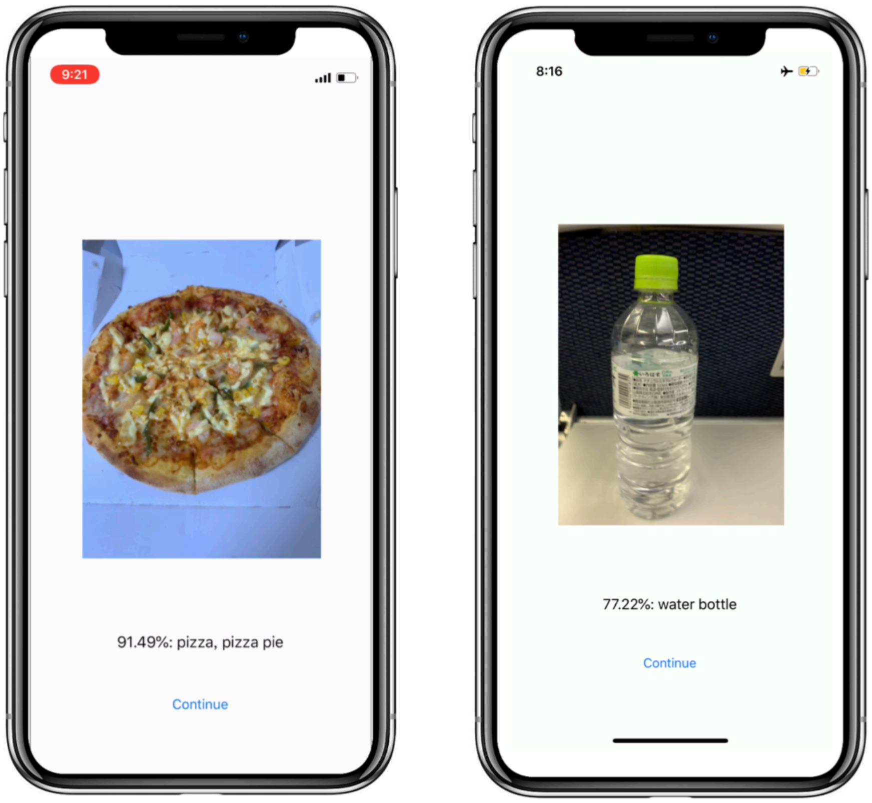

Figure 32: Identity

4.1.2 Measure

Verifying the measure function means verifying the results from the AR measurement part whether close to real value. So comparing the measured result from the real rule and AR measure system is a convenient approach. Table 1 shows three objects are used to evaluate the system: bottle water, pizza and doughnut. The size and shape of them are quite different.

		Bottle Water	Pizza	Doughnut
Real ruler		20.14 cm	23.00 cm	7.9 cm
AR measure	**First time**	19.13 cm	22.25 cm	7.4 cm
	Second time	21.46 cm	21.95 cm	8.2 cm
	Third time	19.27 cm	21.75 cm	7.2 cm
	Average	19.95 cm	21.98 cm	7.6 cm
	Standard deviation	1.31	0.25	0.53
	Error rate	1%	4%	4%

Table 1: Measure Results

From the value of average and standard deviation in the table, we could find that despite the results from the AR measure system are different every time, but they are stable around the real size. Compared with the results from the real rule, the error rate is under 5%, which means the result from the AR measure system is reliable. Just providing the weight of the food is not enough for users to know the actual calories of the food. The calories of the food is provided by the time the weight or size is measured. In the experiment, the size, weight and calories of a pizza are measured and showed to the user. By comparing with the reference calories, we found that the calorie calculated by the system is very close to the true value. Calculating food calories through the system is reliable.

4.1.3 Evaluation

In this research evaluation, 20 participants took part in the survey, the gender ratio of the experimental participants is reasonable. Before the questionnaires, the tutorial of how to use this application was shown to participants, and then the participants were asked to do it by themselves. All the answers collected by the participants was based on the experience of using the application.

Here are some analysis results from the evaluation.

With the development of the smartphone, participants started to use smartphone applications to manage their health. Among these applications, they are willing to use applications to record their activities, as figure 48 shows, such as everyday steps. They often use motion detection software to record their movements, such as exercise time and heart rate. Besides, some participants have used software to detect sleep. In conversations with female participants, we also learned that some women use software to record their physiological cycles, such as menstrual time, ovulation, etc. This type of health software is also very popular.

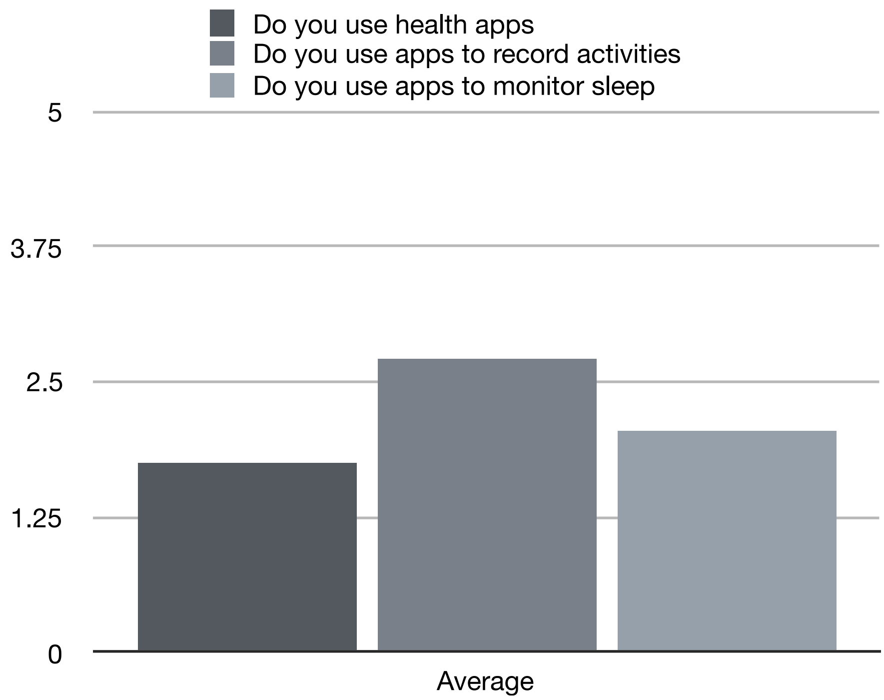

Figure 33: Health applications

From the analysis results, figure 34, it is obvious that participants care about foods or drink calories but they seldom do the calculation for calories. There may be many reasons for this phenomenon, and we learned a few main reasons for the participants. Some participants reported that it is very difficult to calculate calories for food because there are different kinds of foods. Some participants mentioned that since no convenience tools could be used to get calories, it is not easy to get nutrition information for food and drinking. From the gap between the two fold lines, we can also think that for people who pay attention to calories, it is necessary for them to know the clear and accurate calorie information.

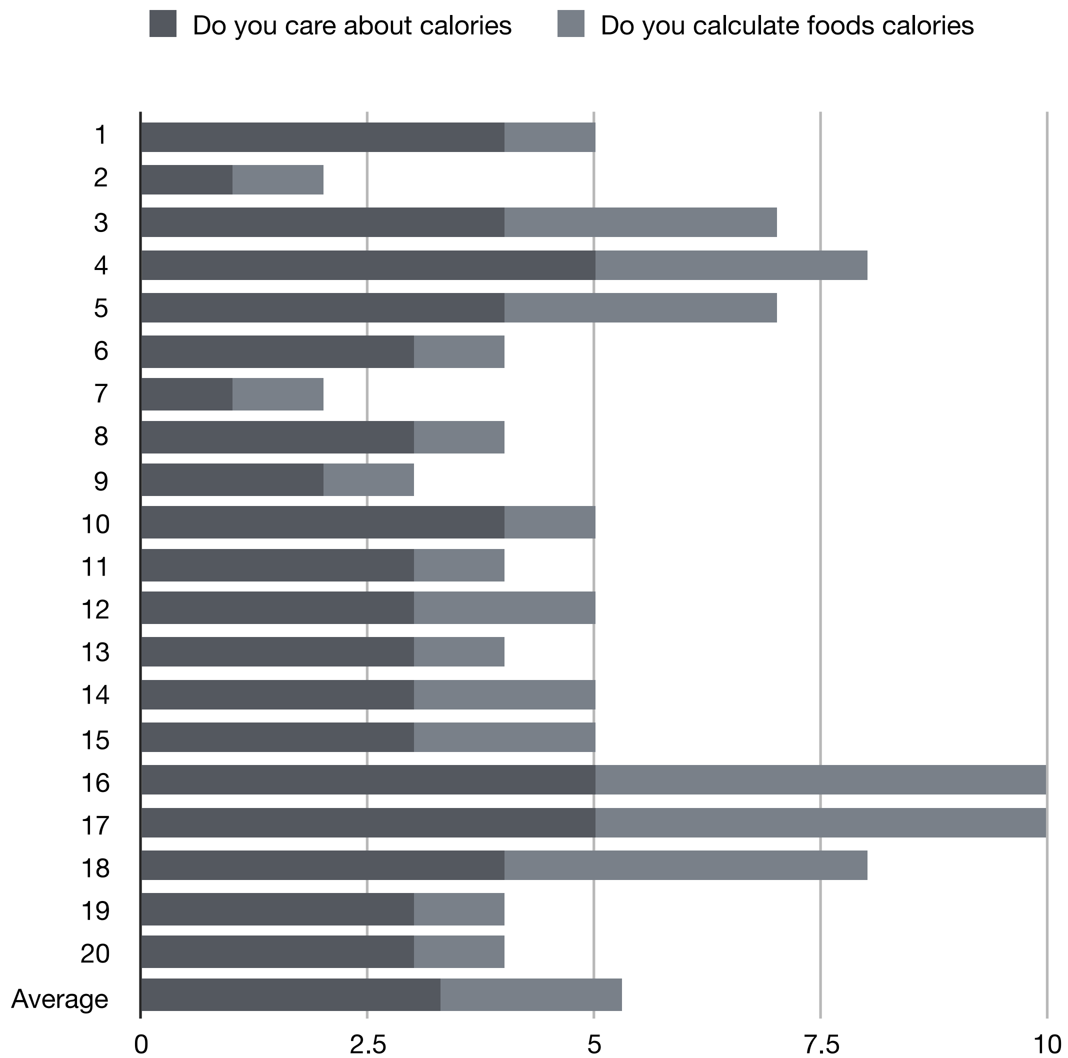

Figure 34: Calories

Besides, most of the participants know what is machine learning and AR. However, they seldom use machine learning and AR-related applications. Also, there is a possibility that they do not know some of the applications they are using are based on machine learning and AR technology, as figure 35 shows.

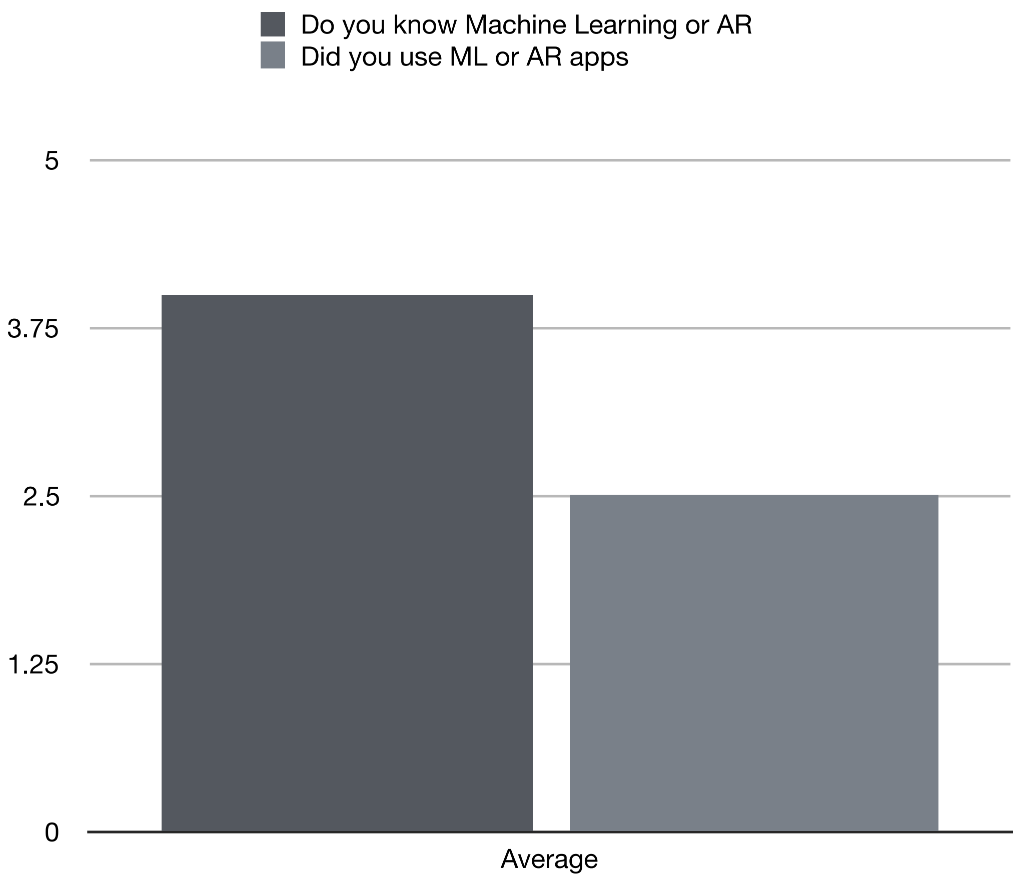

Figure 35: Machine Learning and AR

After trying this application, participants think this application is convenience and the information in this application is helpful for them to do nutrition and health management. Many participants said that if the software is developed successfully, they are willing to download this software.

Figure 36 shows that the conclusion could be drawn from the survey that the participants concern about the calories of food in daily life and want to know the number of calories they consume every day. Since calorie calculations are not easy, they are usually calculated by the calorie content and the amount of food consumed per meal. The measurement of food weight and size is not very easy in daily life, so many people are not willing to calculate by themselves. A software that can simply show calories is very useful.

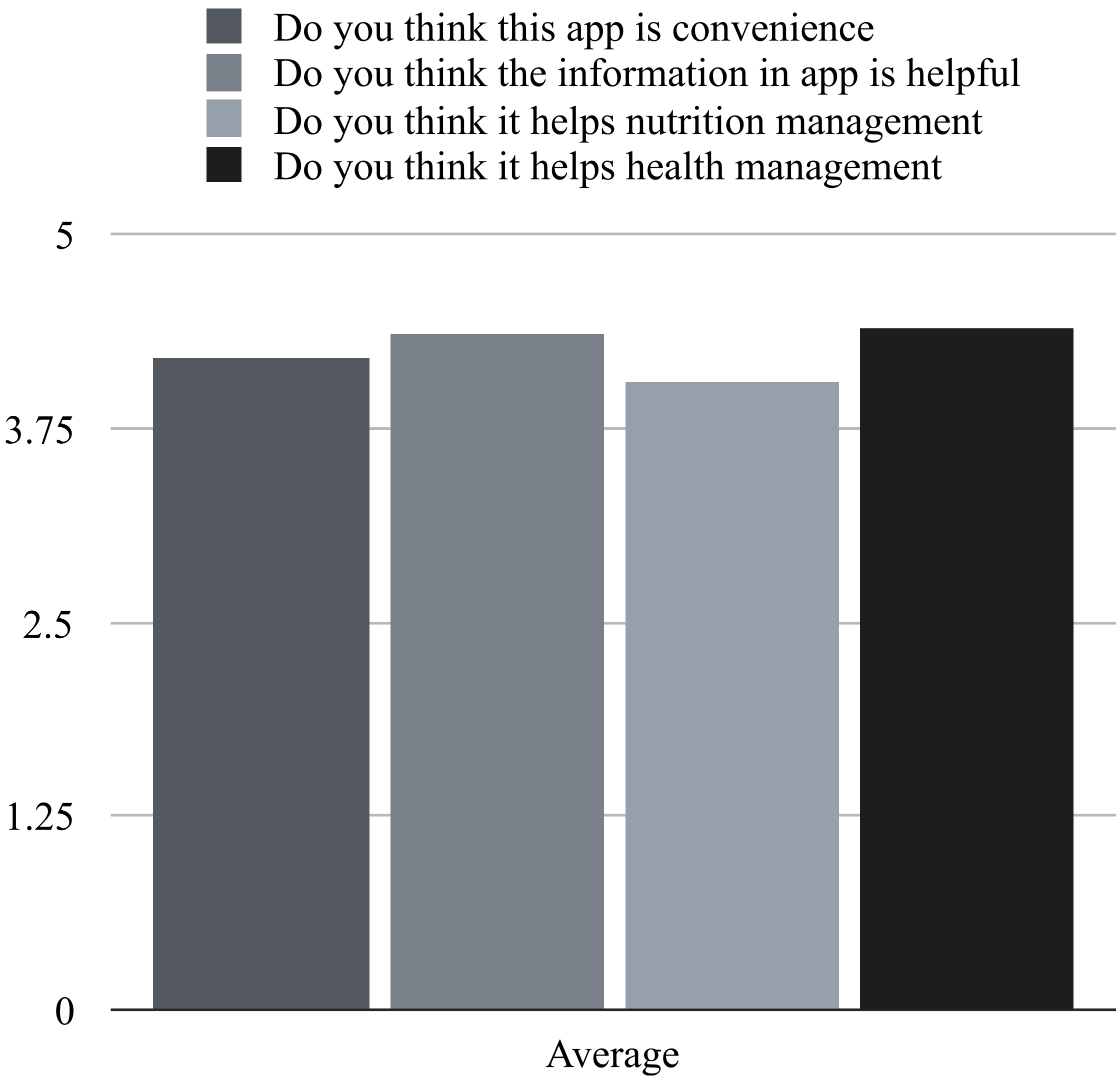

Figure 36: Application

5. Application

5.1 Application

Except for its functions, the system also could be applied to other health management projects. For instance, there is one project called Health Log, which is focused on pushing messages to users to keep health. The application is related to a database, which includes user health condition, time, location and steps count, etc.. And according to this information, it could push personalized messages to users, as figure 37 and 38 shows.

Connecting with these applications, the nutrition information could be added into the database. If users do not intake apposite nutrition, it could remind users to eat more or less. For example, for people with high blood pressure, lack of exercise, and low-temperature sensitivity, calorie intake standards are different. For people who are under-exercised, a smaller amount of calories will not cause fat accumulation. In addition, the software can push different messages to the user by reading the calorie value of the user's daily intake of food. If the user eats a lot of high-calorie food today but has fewer steps, the software can push messages to the user to take more steps or exercise.

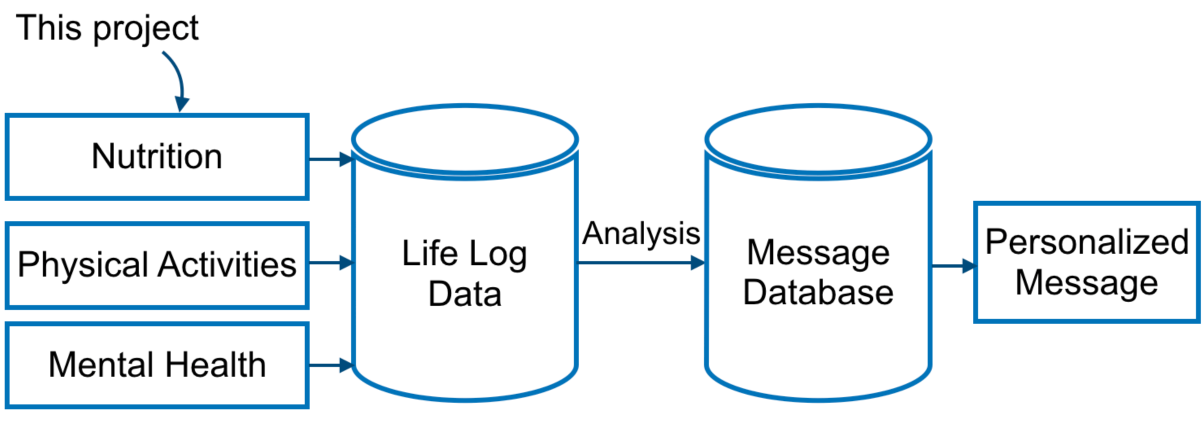

Figure 37: Health Log System

At the same time, because the Health Log application divides users into groups with different characteristics, the software in this article can recommend foods suitable for their physical condition to different users. For example, for people with high blood pressure, excessive exposure to high-calorie foods may be detrimental to good health, and the software will recommend some healthy foods that are good for fruits and vegetables and cardiovascular and cardiovascular health. Or if the user wants to take some high-calorie foods, the software can strictly control the amount of food.

Combining with other health data, the system could judge users' health status and push better messages to users.

Figure 38: Push message

6. Observation

6.1 Discussion

6.1.1. Background Problem

As background part mentioned, healthy life expectancy problem became serious in modern society, plenty of people suffer from disease when they are old. Meanwhile lifestyle diseases, cancer, heart disease and stroke, become the main reason for Japanese death causes. So a healthy good lifestyle become significant for people's life.

This application would help users know the calories and other nutrition information of food to do nutrition management. Nutrition management is one of imperative parts of health management. If health management is proved, in the other words a health lifestyle is adopted, the health state would also improved. So healthy life expectancy would increase and lifestyle disease would reduce as also, the problems mentioned in the background part would be solved.

6.1.2 Unique Feature

The application could identify food and drinking and calculate the calories and nutrition information at real time, which is convenience for user to do nutrition management. Compare with other nutrition management application, it provide a reference value to users, which is more closed to the real value, since the calories and nutrition value is based on the measure results from AR measure.

Many measure applications are in application market, but most of them provide the directly measure results. But this application do some convert for measure results connected with nutrition database. It presents users a comprehensive and favorable information to users. It is also the innovation of this application.

Besides, this application combines two frontier technology, machine learning and AR, and make full use of their function and advantages. Usually they are used in dependent application for single function. Actually these two technology have some common points, for example, in this application they both use some image-process functions. So once the application obtain the authority of camera and photos, the application could collect the information of foods and drinking quickly.

6.1.3 Application Insufficient

However, by now this application still has many defects. For example, in this application, machine learning and AR measure are in two screens. So after identify, the interfaces need jump to the next one, which is not succinct. Ideally, machine learning and AR measure are in one screen, after identify, the system could get into measure process automatically.

Additionally, the project does not have its own Core ML models and comprehensive database. It means that the application only can identify and measure several kinds of foods and drinks. Building a specific machine learning model and database would enhancement utility.

6.1.4 System Limit

Sine in this system the nutrition information are from volume, and the volume are from AR measure. Once the food or drinks is irregular, such as Mapo Tofu, which means the shape is not easy to measure and calculate, the system could not provide nutrition information. To solve this problem, 3D scanner would be applied.

6.2 Future work

Since the measured result from AR is a rough value, the calories and nutrition value from this application is just reference value. The reference value should close to real value as close as possible. One of the methods, which improves accuracy, is that making the system identify more kinds of foods.

Pizza	Serving	Calories
Mushroom Pizza	1 slice (288g)	611 cal
Beef Pizza	1 slice (235g)	714 cal
Vegetable Pizza	1 slice (156g)	399 cal
Hawaiian Pizza	1 slice (134g)	154 cal

Table 2: Pizza nutrition fact

Identifying more kinds of foods not only means identify more foods, but also could classify similar foods. For example, pizza has amounts of kinds, different kinds have different calories and nutrition value. Table 2 shows the nutrition fact of different pizza[35], it is obvious that vegetable pizza has fewer calories than meat series pizza.

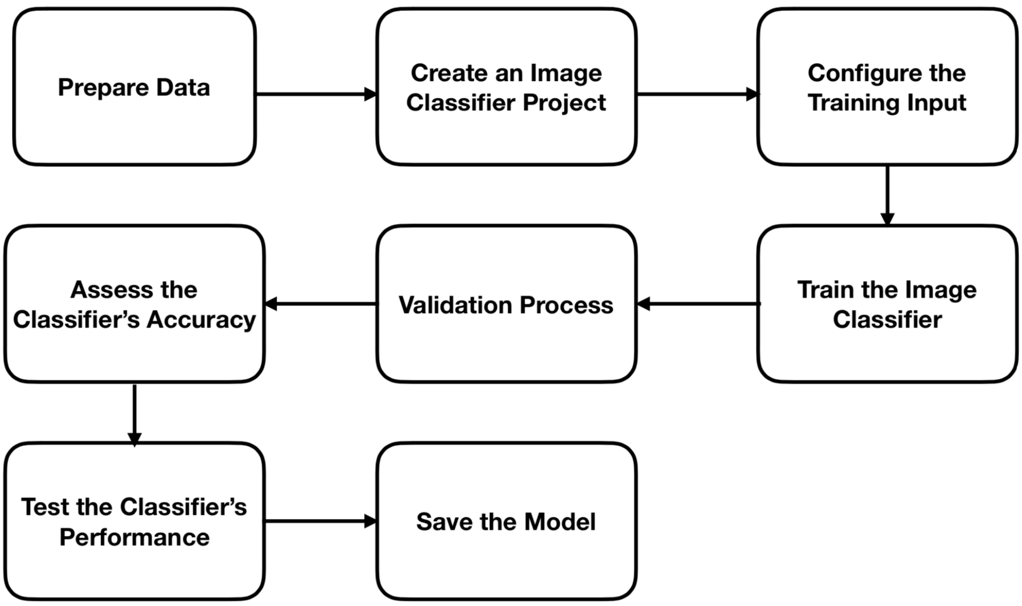

Figure 39: Train Model

To classify these pizzas and get more accurate results, new Core ML models could be trained, which has a larger database of food and drinks. The process of training Core ML models is as following flow chart, as figure 39[36] shows.

Preparing data is the first step of training, which means to prepare plenty of images of foods and drinks, as figure 40 shows. And then creating a training data set from 80% of food images for each label, and reaming 20% of images for testing. During this process, any given image should appear in only one data set or the other. Then adding images into Training Data and Testing Data folders, and creating labels. In each labels, at least 10 images for the training set, the more the better. But the number of images for each label should be balanced. The format and size of images are no limit, but angles and lighting conditions of images should be different. Preparing data is the most significant step for training models.

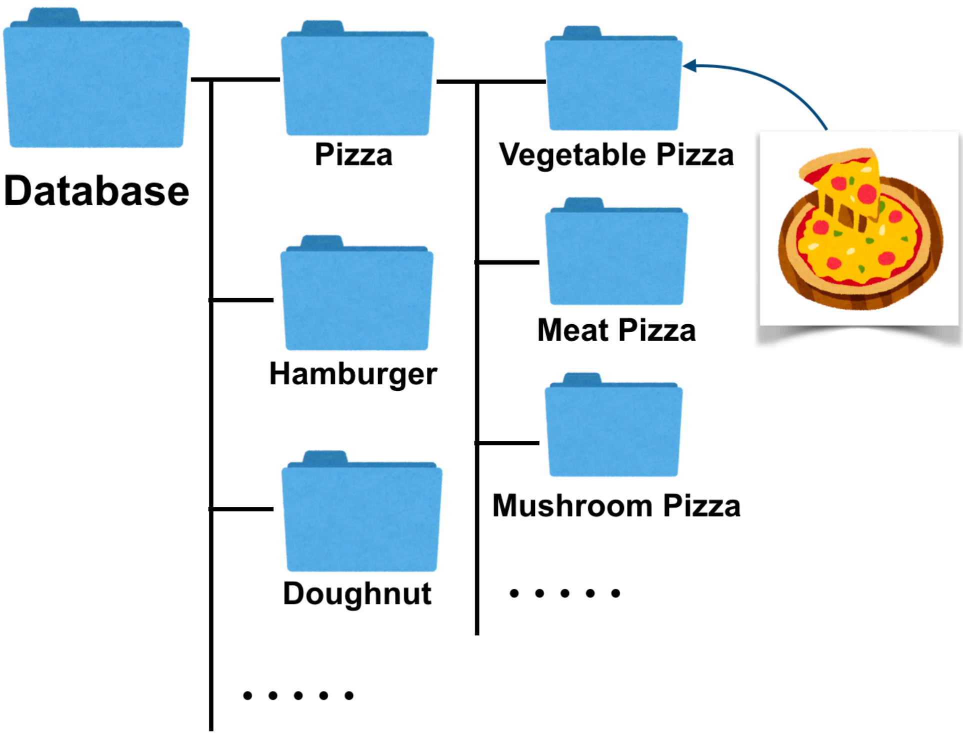

Figure 40: Add database

Creating an image classifier project means using Create ML to create an image classification project in Xcode and then drag the Training Data folder and Testing Data folder onto the project, and then staring the training process. During the training process, the model would learn from the training set iteratively and checks accuracy with the validations set, the data of these two settings is random. So the results of each time training are different. After training, the system would report training and validation accuracy scores.

In the testing process, the system would use the image from the Testing Data folder, which has never seen before. The system would test all of the images and present the overall tasing accuracy. If the evaluation performance is not good enough, training data should be changed. If the model performs well enough, the model could be saved and used into applications.

7. Conclusion

7.1 Conclusion

A new application was builded to support users do nutrition management and improve health management. This application could identify foods and drinking, and measure the volume of foods and drinking to obtain the calories and other nutrition information of them.

In order to accomplish the functions of identify and measure, Apple’s frameworks Core ML and ARKit were applied into the system. Core ML is a framework for machine learning in the iOS devices. Based on Core ML models, the system could classify various objects with camera in iPhone or photos. The models of Core ML are able to train to classify more objects. ARKit is a framework for AR, measure is one of significant AR functions, which could measure the volume of foods and drinking.

Based on Core ML and ARKit, the volume of foods and drinking is known, the weight, calories and other nutrition information could be estimated. With calories and nutrition information, users would able do nutrition management conveniently. And combine with other health management application, users could get more applicable nutrition management suggestions to do health management. As a result, the healthy life expectancy and lifestyle disease problem would be solved.

Furthermore, to obtain more accurate result of calories and nutrition information, the models of machine learning could be trained by developers themselves to meet specific requirements. The new models would identify more categories and kinds of foods and drinks to get more accurate nutrition value.

Acknowledgements

First of all, thanks to Professor Tetsuro Ogi, who leaded me to do this research in these two years. I would like to express the deepest appreciation to him, . Without his guidance and persistent help this research, it would not possible to complete.

Thanks to Professor Naohiko Kohtake, who gave me helpful advices for my research, particularly verification and validation of the application.

Thanks to Professor Tetsuya Toma, who supported in health management project.

Thanks to my parents, who sponsored me to finish my master studies and gave me a chance to study in Japan and experience different culture.

Thanks to all lab members, who encouraged me and shared research experience with me. This experience assisted me continue my research.

Thanks to all participants, who tried to use the application and answered the questionnaire to help me finish the evaluation parts of the application.

July 2019

Yawen Sun

Publication List

The following is a list of publications containing the work presented in this thesis as well as some addition results.

1. *AR Measurement Technology Applied to Nutritional Management, Asian Conference on Design and Digital Engineering*, Yawen Sun, Tetsuro Ogi, 2018 November.
2. *Nutrition Management Based on Deep Learning and AR Measurement technology*, Asian Conference on Design and Digital Engineering, Yawen Sun, Tetsuro Ogi, 2019 July.

Reference

1. “Healthy Life Expectancy: Key Points.” *ScotPHO*, www.scotpho.org.uk/population-dynamics/healthy-life-expectancy/key-points/.

2. “Life Expectancy.” *World Health Organization*, World Health Organization, 6 July 2018, www.who.int/gho/mortality_burden_disease/life_tables/situation_trends/en/.

3. “GHO | By Category | Healthy Life Expectancy (HALE) - Data by Country.” *World Health Organization*, World Health Organization, apps.who.int/gho/data/view.main.HALEXv?lang=en.

4. “What Is Healthcare Management?” *Healthcare Administration Degree Programs*, www.healthcare-administration-degree.net/faq/healthcare-management/.

5. “60％以上の人が生活習慣病で死亡している！.” *HelC*＋（ヘルシー），www.health.ne.jp/library/detail?slug=hcl_3000_w3000680&doorSlug=life-style.

6. *Hoken*, www.hoken-erabi.net/seihoshohin/kazunasemi/insu/medical/76.html.

7. “Defining Health Promotion and Disease Prevention - RHIhub Toolkit.” *Defining Health Promotion and Disease Prevention - RHIhub Toolkit*, www.ruralhealthinfo.org/toolkits/health-promotion/1/definition.

8. “What Is Health Promotion?” *World Health Organization*, World Health Organization, 3 Aug. 2016, www.who.int/features/qa/health-promotion/en/.

9. *Health Japan 21*, www.nibiohn.go.jp/eiken/kenkounippon21/en/.

10. “List of Targets.” *Health Japan 21 (the Second Term) | Health Japan 21*, www.nibiohn.go.jp/eiken/kenkounippon21/en/kenkounippon21/mokuhyou.html.

11. “What Is MHealth (Mobile Health)? - Definition from WhatIs.com.” *SearchHealthIT*, searchhealthit.techtarget.com/definition/mHealth.

12. MHealth: New Horizons for Health through Mobile Technologies. World Health Organization, 2011.

13. “What Is Health Apps? - Definition from WhatIs.com.” *SearchHealthIT*, searchhealthit.techtarget.com/definition/health-apps.

14. “IOS - Health.” *Apple (Malaysia)*, www.apple.com/my/ios/health/.

15. “ライフログテクノロジー株式会社 健康寿命を太く、長くする.” ライフログテクノロジー株式会社, calomeal.com/.

16. Krehl, W A. “The Role of Nutrition in Maintaining Health and Preventing Disease.” *Health Values*, U.S. National Library of Medicine, 1983, www.ncbi.nlm.nih.gov/pubmed/10260843.

17. “Pizza Calories.” *CATEGORIES*, www.calories.info/food/pizza.

18. “Food Lab inside Your Mobile Phone.” *Fraunhofer Food Chain Management Alliance*, www.fcm.fraunhofer.de/en/beispiele11/food-lab-inside-your-mobile-phone.html.

19. “Food Lab inside Your Mobile Phone.” *Fraunhofer Food Chain Management Alliance*, www.fcm.fraunhofer.de/en/beispiele11/food-lab-inside-your-mobile-phone.html.

20. “Pocket-Size Food Scanner.” *Fraunhofer*, www.fraunhofer.de/en/press/research-news/2019/january/pocket-size-food-scanner.html.

21. *Open Food Facts*, world.openfoodfacts.org/discover.

22. Gigandet, Stephane. “Open Food Facts.” *App Store*, 3 Jan. 2013, apps.apple.com/nz/app/open-food-facts/id588797948.

23. “What Is Machine Learning? A Definition.” *Expert System*, 5 Oct. 2017, www.expertsystem.com/machine-learning-definition/.

24. /@rajesh_khadka. “Machine Learning Types #2.” *Medium*, Towards Data Science, 8 Sept. 2017, towardsdatascience.com/machine-learning-types-2-c1291d4f04b1.

25. “Advantages and Disadvantages of Machine Learning Language.” *DataFlair*, 1 Mar. 2019, data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/.

26. “Core ML.” *Core ML | Apple Developer Documentation*, developer.apple.com/documentation/coreml.

27. “Create ML.” *Create ML | Apple Developer Documentation*, developer.apple.com/documentation/createml.

28. Sales. “Know the Augmented Reality Technology: How Does AR Work?” *Cloud Computing*, www.newgenapps.com/blog/augmented-reality-technology-how-ar-works.

29. Apple Inc. “Augmented Reality.” *Apple Developer*, developer.apple.com/augmented-reality/.

30. Ashwini, Amit. “What Are the Advantages of Apple AR Kit ?” *Digital Product Insights*, Digital Product Insights, 26 June 2017, www.cognitiveclouds.com/insights/what-is-apple-ar-kit-and-what-does-it-do/.

31. “Nate Kaiser.” *Nate Kaiser::Portfolio Website*, njkaiser.github.io/portfolio/compviz.html.

32. “Use the Measure App on Your IPhone, IPad, or IPod Touch.” *Apple Support*, 28 May 2019, support.apple.com/en-my/HT208924.

33. *View Controller Programming Guide for IOS: Using Segues*, 16 Sept. 2015, developer.apple.com/library/archive/featuredarticles/ViewControllerPGforiPhoneOS/UsingSegues.html.

34. "UIStoryboardSegue." *UIStoryboardSegue - UIKit | Apple Developer Documentation*, developer.apple.com/documentation/uikit/uistoryboardsegue.

35. "Pizza Calories." *CATEGORIES*, www.calories.info/food/pizza.

36. "Creating an Image Classifier Model." *Creating an Image Classifier Model | Apple Developer Documentation*, developer.apple.com/documentation/createml/creating_an_image_classifier_model.

37. Aizawa, Kiyoharu, and Makoto Ogawa. "FoodLog: Multimedia Tool for Healthcare Applications." *IEEE MultiMedia*, vol. 22, no. 2, 2015, pp. 4–8., doi:10.1109/mmul.2015.39.

38. "Understanding World Tracking." *Understanding World Tracking | Apple Developer Documentation*, developer.apple.com/documentation/arkit/understanding_world_tracking.

39. "Core ML Tutorial: Build an IOS App That Recognizes Hand-Drawn Digits." *Core ML - IBM Watson*, 23 Feb. 2019, dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ml-core-ml-mnist-tutorial.html.

40. Apple Inc. "Introducing ARKit 3 - WWDC 2019 - Videos." *Apple Developer*, developer.apple.com/videos/play/wwdc2019/604/.

41. Sekiguchi, Kentaro. "Museum Exhibition Linking with the Real World by Using Augmented Reality Technology." *Keio Gijuku Daigaku Daigakuin Shisutemu Dezain Manejimento Kenkyuka*, Mar. 2014, pp. 101.

42. Jordan, M. I., and T. M. Mitchell. “Machine Learning: Trends, Perspectives, and Prospects.” *Science*, vol. 349, no. 6245, 2015, pp. 255–260., doi:10.1126/science.aaa8415.

43. Kourou, Konstantina, and Themis P Exachors. “Machine Learning Applications in Cancer Prognosis and Prediction.” *Computational and Structural Biotechnology Journal*, vol. 13, 2015, pp. 8–17., doi:10.1016/j.csbj.2014.11.005.

44. Li, Bo, et al. “Acoustic Modeling for Google Home.” *Interspeech 2017*, 2017, doi: 10.21437/interspeech.2017-234.

45. Thakkar, Mohit. “Custom Core ML Models Using Create ML.” *Beginning Machine Learning in IOS*, 2019, pp. 95–138., doi:10.1007/978-1-4842-4297-1_4.

46. Wu, Hsin-Kai, et al. “Current Status, Opportunities and Challenges of Augmented Reality in Education.” *Computers & Education*, vol. 62, 2013, pp. 41–49., doi:10.1016/j.compedu.2012.10.024.

Appendix

Codes for Application

Core ML

```
import UIKit

import AVFoundation

import CoreML

import Vision


class CoreML: UIViewController, AVCaptureVideoDataOutputSampleBufferDelegate {



    @IBOutlet weak var cameraView: UIView!

    @IBOutlet weak var button: UIButton!

    @IBOutlet weak var classificationText: UILabel!


    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {


        let ViewController = segue.destination as! ViewController


        if let text = classificationText.text {


            ViewController.text = text

        }

    }
```

```
    private var requests = [VNRequest]()
    private lazy var cameraLayer: AVCaptureVideoPreviewLayer =
AVCaptureVideoPreviewLayer(session: self.captureSession)
    private lazy var captureSession: AVCaptureSession = {
        let session = AVCaptureSession()
        session.sessionPreset = AVCaptureSession.Preset.photo
        guard
            let backCamera = AVCaptureDevice.default(for: .video),
            let input = try? AVCaptureDeviceInput(device: backCamera)
            else { return session }
        session.addInput(input)
        return session
    }()

    override func viewDidLoad() {
        super.viewDidLoad()
        self.cameraView?.layer.addSublayer(self.cameraLayer)
        let videoOutput = AVCaptureVideoDataOutput()
        videoOutput.setSampleBufferDelegate(self, queue: DispatchQueue(label: "MyQueue"))
        self.captureSession.addOutput(videoOutput)
        self.captureSession.startRunning()
        setupVision()
    }
```

```
    override func viewDidLayoutSubviews() {
        super.viewDidLayoutSubviews()
        self.cameraLayer.frame = self.cameraView?.bounds ?? .zero
    }

    func setupVision() {
        guard let visionModel = try? VNCoreMLModel(for: Inceptionv3().model)
            else { fatalError("Can't load VisionML model") }
        let classificationRequest = VNCoreMLRequest(model: visionModel,
completionHandler: handleClassifications)
        classificationRequest.imageCropAndScaleOption = .centerCrop
        self.requests = [classificationRequest]
    }

    func handleClassifications(request: VNRequest, error: Error?) {
        guard let observations = request.results
            else { print("no results: \(error!)"); return }
        let classifications = observations[0...4]
            .compactMap({ $0 as? VNClassificationObservation })
            .filter({ $0.confidence > 0.3 })
            .sorted(by: { $0.confidence > $1.confidence })
            .map {
                (prediction: VNClassificationObservation) -> String in
                return "\(round(prediction.confidence * 100 * 100)/100)%: \(prediction.identifier)"
        }
```

```
        DispatchQueue.main.async {
            print(classifications.joined(separator: "###"))
            self.classificationText.text = classifications.joined(separator: "\n")
        }
    }

    func captureOutput(_ output: AVCaptureOutput, didOutput sampleBuffer:
CMSampleBuffer, from connection: AVCaptureConnection) {
        guard let pixelBuffer = CMSampleBufferGetImageBuffer(sampleBuffer) else {
            return
        }
        var requestOptions:[VNImageOption : Any] = [:]
        if let cameraIntrinsicData = CMGetAttachment(sampleBuffer,
kCMSampleBufferAttachmentKey_CameraIntrinsicMatrix, nil) {
            requestOptions = [.cameraIntrinsics:cameraIntrinsicData]
        }
        let imageRequestHandler = VNImageRequestHandler(cvPixelBuffer: pixelBuffer,
orientation: CGImagePropertyOrientation(rawValue: 1)!, options: requestOptions)
        do {
            try imageRequestHandler.perform(self.requests)
        } catch {
            print(error)
        }
    }
```

```
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

}
```

ViewController

```
import UIKit

import SceneKit

import ARKit


final class ViewController: UIViewController {

    @IBOutlet weak var sceneView: ARSCNView!

    @IBOutlet weak var targetImageView: UIImageView!

    @IBOutlet weak var loadingView: UIActivityIndicatorView!

    @IBOutlet weak var messageLabel: UILabel!

    @IBOutlet weak var meterImageView: UIImageView!

    @IBOutlet weak var resetImageView: UIImageView!

    @IBOutlet weak var resetButton: UIButton!


    @IBOutlet weak var newlabel2: UILabel!




    fileprivate lazy var session = ARSession()

    fileprivate lazy var sessionConfiguration = ARWorldTrackingConfiguration()

    fileprivate lazy var isMeasuring = false;

    fileprivate lazy var vectorZero = SCNVector3()

    fileprivate lazy var startValue = SCNVector3()

    fileprivate lazy var endValue = SCNVector3()

    fileprivate lazy var lines: [Line] = []

    fileprivate var currentLine: Line?
```

```
fileprivate lazy var unit: DistanceUnit = .centimeter

var text:String?


@IBOutlet weak var label: UILabel!

override func viewDidLoad() {

    let str = text

    if str!.contains("pizza"){

        label.text = "Please measure"
    }

    else {
        label.text = "No information"
    }


    super.viewDidLoad()
    setupScene()
}
```

```
override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)
    UIApplication.shared.isIdleTimerDisabled = true
}

override func viewWillDisappear(_ animated: Bool) {
    super.viewWillDisappear(animated)
    session.pause()
}

override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
    resetValues()
    isMeasuring = true
    targetImageView.image = UIImage(named: "targetGreen")
}

override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {
    isMeasuring = false
    targetImageView.image = UIImage(named: "targetWhite")
    if let line = currentLine {
        lines.append(line)
        currentLine = nil
        resetButton.isHidden = false
        resetImageView.isHidden = false
    }
```

```
    }

    override var prefersStatusBarHidden: Bool {
        return true
    }
}

extension ViewController: ARSCNViewDelegate {
    func renderer(_ renderer: SCNSceneRenderer, updateAtTime time: TimeInterval) {
        DispatchQueue.main.async { [weak self] in
            self?.detectObjects()
        }
    }

    func session(_ session: ARSession, didFailWithError error: Error) {
        messageLabel.text = "Error occurred"
    }

    func sessionWasInterrupted(_ session: ARSession) {
        messageLabel.text = "Interrupted"
    }

    func sessionInterruptionEnded(_ session: ARSession) {
        messageLabel.text = "Interruption ended"
```

```
    }
}

extension ViewController {
    @IBAction func meterButtonTapped(button: UIButton) {
        let alertVC = UIAlertController(title: "Settings", message: "Please select distance unit
options", preferredStyle: .actionSheet)
        alertVC.addAction(UIAlertAction(title: DistanceUnit.centimeter.title, style: .default)
{ [weak self] _ in
            self?.unit = .centimeter
        })
        alertVC.addAction(UIAlertAction(title: DistanceUnit.inch.title, style: .default) { [weak
self] _ in
            self?.unit = .inch
        })
        alertVC.addAction(UIAlertAction(title: DistanceUnit.meter.title, style: .default) { [weak
self] _ in
            self?.unit = .meter
        })
        alertVC.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: nil))
        present(alertVC, animated: true, completion: nil)
    }

    @IBAction func resetButtonTapped(button: UIButton) {
```

```
        resetButton.isHidden = true
        resetImageView.isHidden = true
        for line in lines {
            line.removeFromParentNode()
        }
        lines.removeAll()
    }
}

extension ViewController {
    fileprivate func setupScene() {
        targetImageView.isHidden = true
        sceneView.delegate = self
        sceneView.session = session
        loadingView.startAnimating()
        meterImageView.isHidden = true
        messageLabel.text = "Detecting the world…"
        resetButton.isHidden = true
        resetImageView.isHidden = true
        session.run(sessionConfiguration, options: [.resetTracking, .removeExistingAnchors])
        resetValues()
    }

    fileprivate func resetValues() {
        isMeasuring = false
```

```
        startValue = SCNVector3()
        endValue =  SCNVector3()
    }

    fileprivate func detectObjects() {
        guard let worldPosition = sceneView.realWorldVector(screenPosition: view.center) else
{ return }
        targetImageView.isHidden = false
        meterImageView.isHidden = false
        if lines.isEmpty {
            messageLabel.text = "Hold screen & move your phone…"
        }
        loadingView.stopAnimating()
        if isMeasuring {
            if startValue == vectorZero {
                startValue = worldPosition
                currentLine = Line(sceneView: sceneView, startVector: startValue, unit: unit)
            }
            endValue = worldPosition
            currentLine?.update(to: endValue)
            messageLabel.text = currentLine?.distance(to: endValue) ?? "Calculating…"

            let newstring = messageLabel.text
            let newnum = NSString (string : newstring!).doubleValue
            let newweight1 = (newnum / 2) * (newnum / 2) * 1 * 1.05
```

```
            let newweight2 = newweight1 * 2.66

            let newweight3 = String(format: "%.2f", newweight1)

            let newweight4 = String(format: "%.2f", newweight2)

            newlabel2.text = "Weight \(newweight3) g, Calories \(newweight4) kCal" as String


        }

    }

}
```

Line

```
import SceneKit

import ARKit


enum DistanceUnit {

    case centimeter

    case inch

    case meter


    var fator: Float {

        switch self {

        case .centimeter:

            return 100.0

        case .inch:

            return 39.3700787

        case .meter:

            return 1.0

        }

    }


    var unit: String {

        switch self {

        case .centimeter:

            return "cm"

        case .inch:
```

```
            return "inch"
        case .meter:
            return "m"
        }
    }

    var title: String {
        switch self {
        case .centimeter:
            return "Centimeter"
        case .inch:
            return "Inch"
        case .meter:
            return "Meter"
        }
    }
}

final class Line {
    fileprivate var color: UIColor = .white

    fileprivate var startNode: SCNNode!
    fileprivate var endNode: SCNNode!
    fileprivate var text: SCNText!
    fileprivate var textNode: SCNNode!
```

```
fileprivate var lineNode: SCNNode?


fileprivate let sceneView: ARSCNView!
fileprivate let startVector: SCNVector3!
fileprivate let unit: DistanceUnit!


init(sceneView: ARSCNView, startVector: SCNVector3, unit: DistanceUnit) {
    self.sceneView = sceneView
    self.startVector = startVector
    self.unit = unit


    let dot = SCNSphere(radius: 0.5)
    dot.firstMaterial?.diffuse.contents = color
    dot.firstMaterial?.lightingModel = .constant
    dot.firstMaterial?.isDoubleSided = true
    startNode = SCNNode(geometry: dot)
    startNode.scale = SCNVector3(1/500.0, 1/500.0, 1/500.0)
    startNode.position = startVector
    sceneView.scene.rootNode.addChildNode(startNode)


    endNode = SCNNode(geometry: dot)
    endNode.scale = SCNVector3(1/500.0, 1/500.0, 1/500.0)


    text = SCNText(string: "", extrusionDepth: 0.1)
    text.font = .systemFont(ofSize: 5)
```

```
        text.firstMaterial?.diffuse.contents = color
        text.alignmentMode  = kCAAlignmentCenter
        text.truncationMode = kCATruncationMiddle
        text.firstMaterial?.isDoubleSided = true

        let textWrapperNode = SCNNode(geometry: text)
        textWrapperNode.eulerAngles = SCNVector3Make(0, .pi, 0)
        textWrapperNode.scale = SCNVector3(1/500.0, 1/500.0, 1/500.0)

        textNode = SCNNode()
        textNode.addChildNode(textWrapperNode)
        let constraint = SCNLookAtConstraint(target: sceneView.pointOfView)
        constraint.isGimbalLockEnabled = true
        textNode.constraints = [constraint]
        sceneView.scene.rootNode.addChildNode(textNode)
    }

    func update(to vector: SCNVector3) {
        lineNode?.removeFromParentNode()
        lineNode = startVector.line(to: vector, color: color)
        sceneView.scene.rootNode.addChildNode(lineNode!)

        text.string = distance(to: vector)
        textNode.position = SCNVector3((startVector.x+vector.x)/2.0, (startVector.y+vector.y)/
2.0, (startVector.z+vector.z)/2.0)
```

```
        endNode.position = vector
        if endNode.parent == nil {
            sceneView?.scene.rootNode.addChildNode(endNode)
        }
    }

    func distance(to vector: SCNVector3) -> String {
        return String(format: "%.2f%@", startVector.distance(from: vector) * unit.fator,
unit.unit)
    }

    func removeFromParentNode() {
        startNode.removeFromParentNode()
        lineNode?.removeFromParentNode()
        endNode.removeFromParentNode()
        textNode.removeFromParentNode()
    }
}
```

ARSCNView

```
import SceneKit

import ARKit


extension ARSCNView {

    func realWorldVector(screenPosition: CGPoint) -> SCNVector3? {

        let results = self.hitTest(screenPosition, types: [.featurePoint])

        guard let result = results.first else { return nil }

        return SCNVector3.positionFromTransform(result.worldTransform)

    }

}
```

SCNVector3

```
import ARKit


extension SCNVector3 {
   static func positionFromTransform(_ transform: matrix_float4x4) -> SCNVector3 {
      return SCNVector3Make(transform.columns.3.x, transform.columns.3.y,
transform.columns.3.z)
   }

   func distance(from vector: SCNVector3) -> Float {
      let distanceX = self.x - vector.x
      let distanceY = self.y - vector.y
      let distanceZ = self.z - vector.z

      return sqrtf((distanceX * distanceX) + (distanceY * distanceY) + (distanceZ *
distanceZ))
   }

   func line(to vector: SCNVector3, color: UIColor = .white) -> SCNNode {
      let indices: [Int32] = [0, 1]
      let source = SCNGeometrySource(vertices: [self, vector])
      let element = SCNGeometryElement(indices: indices, primitiveType: .line)
      let geometry = SCNGeometry(sources: [source], elements: [element])
      geometry.firstMaterial?.diffuse.contents = color
      let node = SCNNode(geometry: geometry)
```

```
        return node
    }
}
extension SCNVector3: Equatable {
    public static func ==(lhs: SCNVector3, rhs: SCNVector3) -> Bool {
        return (lhs.x == rhs.x) && (lhs.y == rhs.y) && (lhs.z == rhs.z)
    }
}
```

AppDelegate

```
import UIKit


@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {


    var window: UIWindow?
  func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {


        return true
    }


    func applicationWillResignActive(_ application: UIApplication) {


    }
```

```
    func applicationDidEnterBackground(_ application: UIApplication) {

    }

    func applicationWillEnterForeground(_ application: UIApplication) {

    }

    func applicationDidBecomeActive(_ application: UIApplication) {
    }
    func applicationWillTerminate(_ application: UIApplication) {
    }
}
```

Questionnaire

1	**Gender**	Female	Male			
2	**Age**	Under 20	20-29	30-39	40-49	Over 50
3	**Do you care about food calories**	Never	Seldom	Sometimes	Often	Always
4	**Do you prefer health food in meal**	Never	Seldom	Sometimes	Often	Always
5	**How often do you eat fast food**	Never	Seldom	Sometimes	Often	Always
6	**Do you calculate food calories**	Never	Seldom	Sometimes	Often	Always
7	**Do you use health applications**	Never	Seldom	Sometimes	Often	Always
8	**Do you use apps record activities**	Never	Seldom	Sometimes	Often	Always
9	**Do you use apps monitor sleeping**	Never	Seldom	Sometimes	Often	Always
10	**Do you know machine learning and AR**	Strongly Disagree	Disagree	Sometimes	Agree	Strongly Agree
11	**Did you use machine learning or AR apps**	Never	Seldom	Sometimes	Often	Always
12	**Do you think this app is connivence**	Strongly Disagree	Disagree	Neither	Agree	Strongly Agree
13	**Do you think the information in apps is helpful**	Strongly Disagree	Disagree	Neither	Agree	Strongly Agree
14	**Do you think this app help you do nutrition management**	Strongly Disagree	Disagree	Neither	Agree	Strongly Agree
15	**Do you think this app help you do health management**	Strongly Disagree	Disagree	Neither	Agree	Strongly Agree