Food-tracker: A food recording system based on real-time food recognition and meal intake detection

LI ZHE
44181634-6

Master of Engineering

Supervisor: Prof. Jiro Tanaka
Graduate School of Information, Production and Systems
Waseda University

September 2020

Abstract

People have started to take food images for improving their dining habits. Automated food record system can find when a person consumes food and record what kind of food a person consumes. When a person eats food, he/she needs to use their hands to grab the food and move towards to their mouth, then take a bite and chew. Currently most food record system focus on providing a more convenient user interface for user to record food manually. In this paper, we introduce Food-tracker, an automatic food recording system based on real-time food recognition and meal intake detection. Food-tracker contains an augmented glass with a camera for food recognition, an android smartwatch with a gyroscope sensor and an accelerometer sensor for wrist motion track and a throat microphone for chew event detection.

Our system consists of two main parts. One is real-time food recognition, the other is meal intake detection. For the food recognition part, we collect dataset for different kinds of food. First, we use object detection to locate different foods in a single image, then use CNN network to realize food classification. Last, we use volume estimation to help calculate the food calories. Users are required to wear the Epson glasses. The system will recognize the type of food and show the food type and calories to users in real-time through the glasses. For the meal intake detection part. There are two steps to detect the eating behavior. The first step is to determine whether the wrist moves towards the mouth through the combination of orientation sensor and gyroscope. Gyroscope is used to measure the velocity of user's wrist rotation around Y axis and orientation sensor is used to measure the degree of rotation that user's wrist makes around X axis. After the gyroscope classify some possible activities, use the orientation sensor for further filtering. The second step is to detect chew event through analysis periodicity and magnitude of the audio spectrum.

Keywords: Food recognition, Meal intake detection, wearable computing, ubiquitous computing

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Jiro TANAKA, for the continuously support on my study and research. He is the person who leads me into HCI field and helps me find out my own way in food record and diet monitoring research. During 2-year study in Waseda University, he uses his knowledge, patience, motivation and enthusiasm guides me a lot. He teaches me to read top conference papers to broaden my horizons in HCI field. He mentors me how to express my research topic understandable in seminars. He supports me by guiding me to find my interest, helping me clarify the research topic and purchasing necessary equipment for my research. He also gives me countless help during the pandemic even if we cannot meet in the campus. It is my great honor to have the opportunity to do research in IPLab and to be a student of Prof. Jiro TANAKA.

Besides, I would like to thank all members from IPLab. Thank you for your tireless efforts to help me sort out my research ideas and put forward constructive suggestions for improvement. Thank you for giving advice and encouraging me to move on. You are not only my best partner in my research, but also my best friend in my life. Thanks for your companionship and support.

Finally, I want to give my gratitude to my parents. You give me financial support and warm mental support throughout my master study life.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

The current obesity problem in the world is getting worse. Recording food to help people improve their diet is a solution. As people increasingly like to take pictures of food before eating. We think that if people can get food information (calories, etc.) in real time when they see food, it can help people better choose a healthy diet. Some current systems that help people record diets (FoodLog [1], 2009) require people to take pictures manually and upload them for recording. Research also showed that taking photos before a meal can make people's diet more healthy [2].

For many users, taking pictures before eating in a public place will be embarrassing, and the current system cannot provide users with real-time food information, and often provides food-related information after the user completes the meal. We are inspired by LifeLogging [3], a system that naturally records important data in users' lives, and we want to provide a system that automatically records users' diets. There are several issues that need to be considered to record the user's diet:

1. How to detect whether the user is eating;

2. How to identify what the user is eating;

3. How to calculate the calories of the food the user eats.

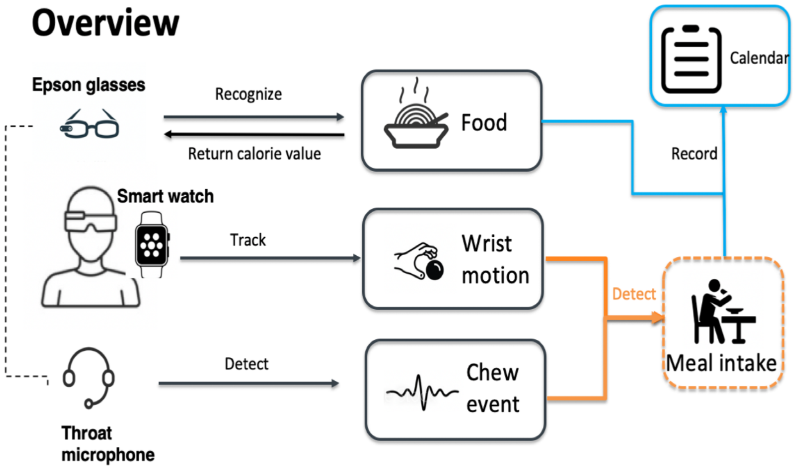


Fig. 1.1 The process of food recording

In this study, we investigated the current FoodLog [1] system and compared their advantages and disadvantages. Then proposed the concept of an automatic food recording system Food-tracker, which provides an easy and intuitive way for food recording.

The Fig. 1.1 shows the process of automatically food recording. Food-tracker provides real-time food recognition and shows the calories via smart glasses directly. This system uses wrist motion tracking and chew event detection to help detect human activity (meal intake). There are three main steps for automatically food recording:

1. Food recognition
   (a) Object detection
   (b) Food classification
   (c) Volume estimation
2. Wrist motion tracking
   (a) Gyroscope and accelerometer
   (b) Sensor fusion
3. Chew event detection

(a) Record sound signals

(b) “Chew-like” signal detection

(c) Chewing sound verification

1.2 Organization of thesis

The rest of the thesis is organized as follows: Chapter 2 will introduce the background of the thesis and also analyze how we figure out our standards and requirements. Chapter 3 will briefly talk about the research goal and also the approaches. Chapter 4 will conclude the system design part, where the design concept and ideas will be illustrated and algorithm will also be told. Chapter 5 will be the system implementation part where the detailed environment and implementation will be talked. Chapter 6 will introduce the related work. Chapter 7 will be about the preliminary evaluation, we will talk about the usability and efficiency of our system. Chapter 8, will be the conclusion and future work part, where we will conclude the previous content and discuss the future expectation.

Chapter 2

Background

2.1 Current food recording system

The problem of obesity is becoming more and more serious. In 2008, obesity-related medical expenses were 147 billion US$ [4], and the World Health Organization ranked obesity as the fifth leading cause of death in the world [5].

Measuring energy intake in daily life is important for losing weight to solve the problem of obesity. However, it is tedious and error-prone for users to calculate calories themselves. With the development of ubiquitous computing, a method that can automate food recording and calorie estimation will provide great convenience for people's daily lives. Some research focused on using food image to help people record and calculate calories [1].

Figure 2.1 shows an overview of the FoodLog system, which was developed for food recording. Firstly, users take a photo before eating food and then upload the photo to flickr through their smartphone. Then the system will automatically extract food images from flickr and estimate the food balance. Last, this system visualizes food image with the estimation for users to correct the analysis result.

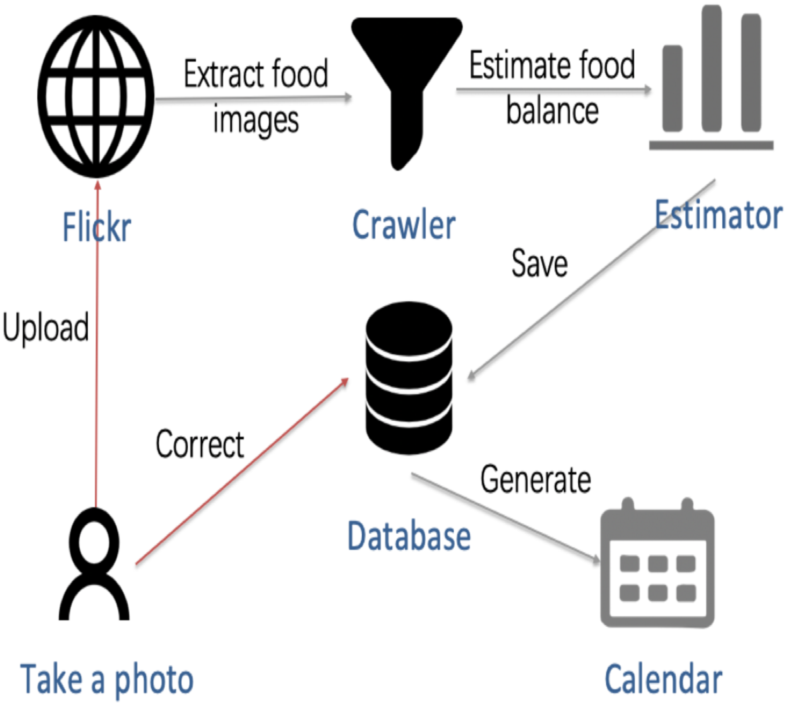


Fig. 2.1 Overview of FoodLog system

2.2 Criteria of better food recording system

When we talk about food recording system. Usually, the first thing comes to mind is filling out various forms. We need to manually fill in the name of the food, the size or weight of the food, and then the system will help us calculate the corresponding calories. Often we will fill in the corresponding information in the system after the meal and then get feedback on the food information.

A better food recording system should automatically record without requiring users to deliberately record. If we research on a system that can automatically record diet and calculate calories, we need to consider the following points:

1. How to detect if the user is eating;

2. How to identify the food the user eats;

3. How to calculate the energy of the food eaten by the user;

4. How to give user feedback on food information in real time;

5. How to automate the recording system.

In summary, current food recording system is not sufficient to meet users' requirements. An automatically food recording system is the development direction.

Chapter 3

Research Goal and Approach

3.1 Goal

The fundamental goal of this research is to provide an easy and intuitive way for food recording.

As a food recording system, we should consider recording without interfering with people's natural behavior as much as possible. At the same time, considering how to provide users with a better user experience with feedback on food information in real time.

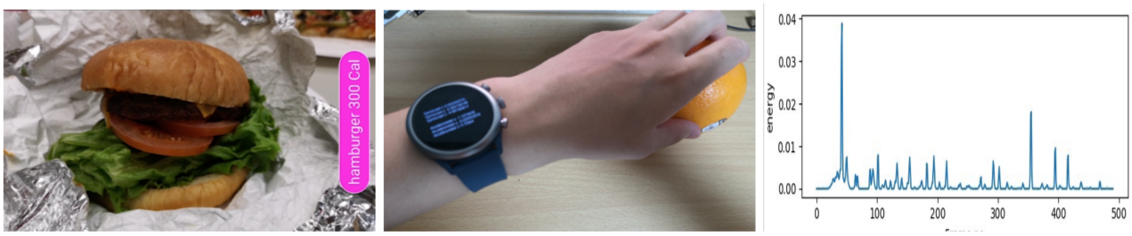
After analyzing why a good food recording system is not easy to realize, some criteria for better recording system are summarized. Therefore, in our research, we focused on the following parts.

1. Provide real-time food recognition and calories estimation and then show the feedback via smart glasses directly.

2. Use wrist motion tracking and chew event detection to help detect human activity (meal intake).

3.2 Approach

This research focused on using augmented reality glasses, smart watch and throat microphone to provide intuitive interaction for food recording system. We use the camera on the augmented reality glasses to realize real-time food recognition. Users need to wear a smart watch or smart bracelet with gyroscope sensor and acceleromter sensor to track their wrist motion. For the chew event detection part, we use throat microphone to record audio because throat microphone is a type of contact microphone so it will function well even if the background noise is high.



(a) Food recognition (b) Wrist motion track (c) Chew event detection

Fig. 3.1 Three parts for food recording system

3.3 Novelty

The novelty of this research mainly reflects in the two aspects:

1. We designed a food recording system that users can get real-time feedback through smart glasses. Compared with the feedback of food information obtained by users after eating food, our system provides real-time food information to users before they decide whether to eat.

2. We designed a meal intake detection method based on wrist motion tracking and chew event detection.

Chapter 4

Related Work

In this section, we introduce related work for human activities recognition and chewing sounds detection. Then, we review the convolutional neural network (CNN).

4.1 Human activity

Human activity recognition is a challenging problem which is still needs to be solved. As the Internet of Things (IoT) is becoming more and more popular recently. Many research work on using multiple sensors to monitor different the human activities. Especially some wearable and video sensors [6] are widely used in the filed of physical activity recognition. Research on monitoring activities of wrist, jaw and throat to detect chewing event by using some sensors[7][8].

In a variety of human activities, we pay more attention to the study which is related to eating activity.

4.1.1 Dietary monitor

Ziad Ahmad, Marc Bosch et al. [9] proposed a comprehensive meal evaluation system based on food image analysis, which used a mobile device or a smartphone. Joachim Dehais, Marios Anthimopoulos et al. [10] presented a method to detect and segment the food of the dishes that have been detected in the image. This method combines the regional growth/merging technology with CNN-based deep food boundary detection. Bedri et al. [11] [12]explored using the Outer Ear Interface (OEI) to recognize eating activities.

4.1.2 Wrist motion

Utilize the dexterous interaction of human wrist movement, we can complete the eating behavior. Hui-Shyong Yeo, Juyoung Lee et al. [13] demonstrate that wrist can provide an enables many novel but simple interactive technologies by combining the inertial measurement unit data of smart watches or smart rings.

Mark Mirtchouk, Christopher Merck et al. [14] recommend that multi-mode sensing (chewing audio plus wrist motion) can be used to more accurately classify food types because audio and exercise functions can provide supplemental information. Yujie Dong, Adam Hoover et al. [15] use a watch-like device with a microelectromechanical gyroscope to detect and record when an individual is eating food.

4.1.3 Detecting chewing sounds

Oliver Amft, Mathias Stäger et al. [16] proves that the sound from the user’s mouth can be used to detect that he/she is eating. Their work also gives a solution to identify different kinds of food by analyzing the chewing sound which is acquired through a microphone located inside the ear canal. Sebastian Päßler and Wolf-Joachim Fischer [17] evaluate seven different algorithms to detect chewing events in sound recordings. There are some efforts [18] [19] [20] [21] have focused on the neck as the sensor location to detect chew event.

4.2 Preliminary CNN

The first neural network was invented in 1949. The original neural network was inspired by people’s research on how the human brain works: a single neuron receives "signals" from other neurons and eventually produces its own output signal. But neural networks were of little use at first. There are two main things that prevent the development of deep learning, one is the lack of large data, and the other is the lack of computing power of the calculator. With the development of the big data era and the improvement of computer computing power, machine learning, especially deep learning, is now popular in the world. Convolutional neural network is a powerful artificial neural network technology. The development of convolutional neural network is to deal with some handwritten digit classification and picture classification problems [22].

Convolutional neural networks (CNN) is a neural network composed of multiple convolutional layers and fully connected layers. Compared with fully connected networks, convolutional neural networks use fewer parameters or weights to learn. They can automatically learn the features of the input images, audio and video, etc., and will not change the information such as the position of objects in the picture scene.

Neurons in convolutional neural networks have three dimensions: width, height, depth (depth refers to the third dimension of the activation volume, not the depth of the complete neural network, it can refer to the total number of layers in the network). Suppose the input image size is 64*64*3 (rgb), and the input neuron has a dimension of 64*64*3. A single fully connected neuron hidden in the first hidden layer in the neural network will have 64*64*3=12288 weights.

The main layers of convolutional neural networks:

1. Convolutional layer. the convolutional layer in a convolutional neural network systematically applies filters to the input image in order to create a feature map that summarizes the features of the input image. The feature map of the convolutional layer outputs the exact position of the recorded feature, which may result in the change of the feature position due to adjustments such as rotation and translation of the image. Solving this problem needs to be done in the pooling layer.

2. Rectified Linear Units Layer. After each convolutional layer, a non-linear layer or an activation layer is usually used. This layer is used to introduce nonlinearity into the system that has just completed linear operations in the convolutional layer.

3. Pooling layer. After the rectified linear units layer, you can use the pooling layer (also known as the downsampling layer) to control the number of parameters or weights, and another purpose is to control overfitting, That is, when the model is trained to be completely consistent with the training set, the characteristics of the test set cannot be well summarized.

4. Dropout Layer. This layer has a specific function in the neural network. The Dropout layer randomly sets some neurons that have been activated to 0 to "eliminate" some features. In a way, this layer makes the neural network redundant, but it is very effective in preventing overfitting.

5. Fully-Connected layer. Fully-Connected, which combines all local features into global features, used to calculate the score of the last category.

6. Softmax layer. The Softmax layer is good at determining multi-class probabilities, can normalize the prediction, and enables the network to generate the output as a probability.

Chapter 5

System Design

5.1 System Overview

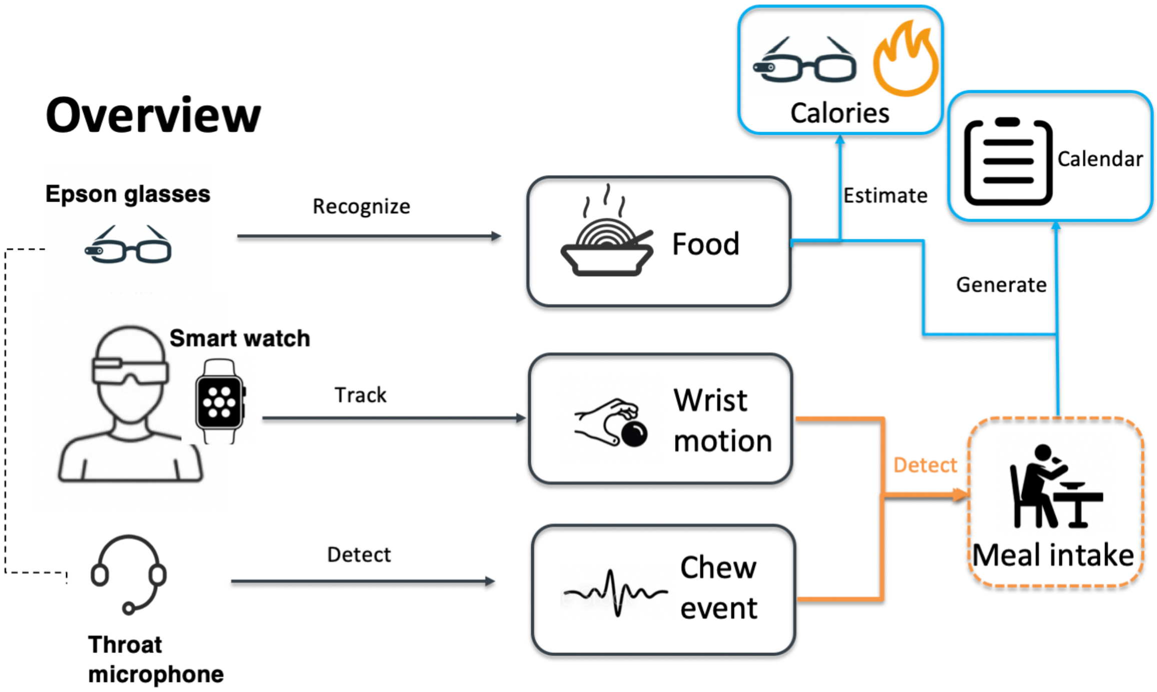


Fig. 5.1 System overview

In this chapter, we will introduce our system design and each important pieces of our approach. The Fig. 5.1 shows the overview of our system.

The Food-tracker provides the user a view of real object and virtual label. In our system, the real object can be considered as some food the user may want to eat and the virtual label contain some information of the food recognized by this system. We suppose that providing the real-time feedback on the information of the food can help user to select healthier food to improve their eating habits. Also this system can record the food eaten by user automatically for user to retrieve at any time.

In order to achieve automatically recording, our system requires a lot of information to complete this difficult task. For automatically food recording, there are two main problems that our system needs to solve.

1. How to judge whether the user took the food after the food appeared in the user's field of vision?

2. How to judge whether the user ate the food after the food appeared in the user's field of vision?

Our system gives answers by two steps. First, we utilize gyroscope and accelerometer in the smart watch to track user's wrist motion. Second, we make use of throat microphone to record the chewing audio. For most eating situations, a roll of the wrist must occur and wrist also need to move towards mouth. Thus, we can use the sensor fusion to detect whether user took the food or not. We generally need to chew and swallow in normal human eating behavior, so we utilize the audio to detect whether user chewed the food or not. With the two kinds of information and through the two steps, our system can understand whether the user took and ate the food which appeared in the user's field of vision.

After that, our system will generate a calendar to help user to retrieve what kind of food they have eaten recently.

Following subsections are details information about key parts of this research.

5.2 Food recognition

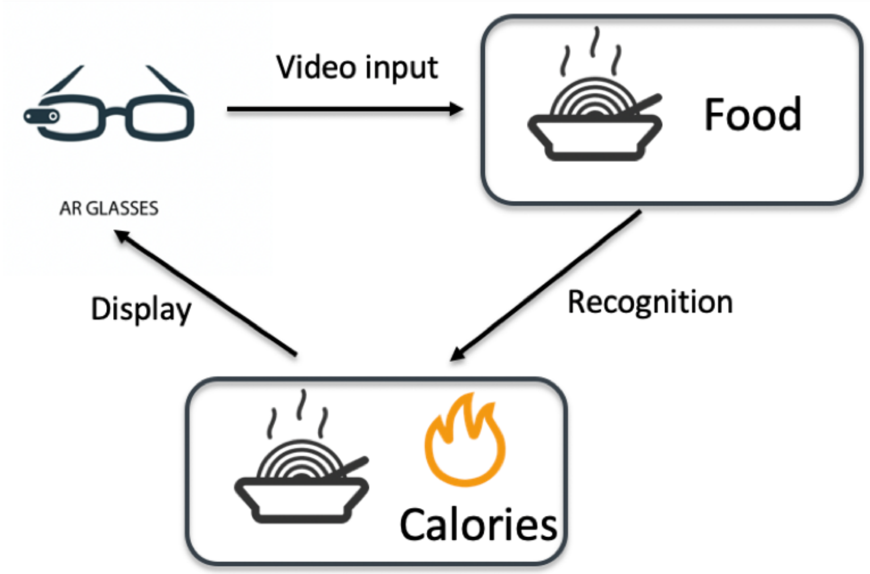


Fig. 5.2 The process of food recognition

As the Fig. 5.2 shows, users are required to wear augmented reality glasses (In our system, we use Epson glasses) to get the video input captured by the camera. The system will recognize the type of food and show the food type and calories to users in real-time via the smart glasses through three main steps.

1. Object detection

   We use object detection to locate different foods in the single image.

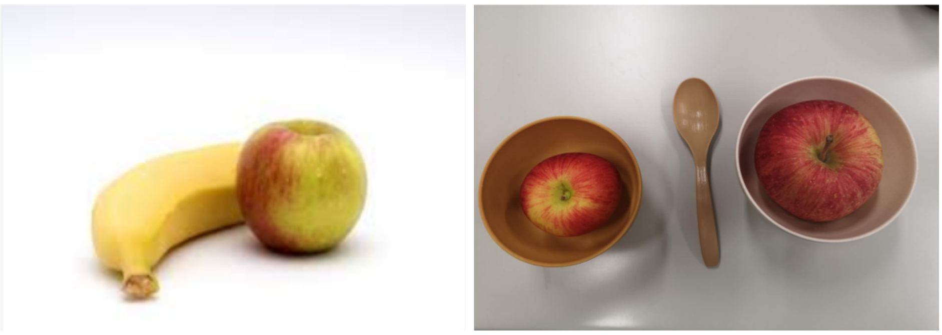
2. Food classification

   Firstly, we collect data set for different kinds of food. Then build a convolutional neural network (CNN) to train the data set and get the model for food classification.

3. Volume estimation

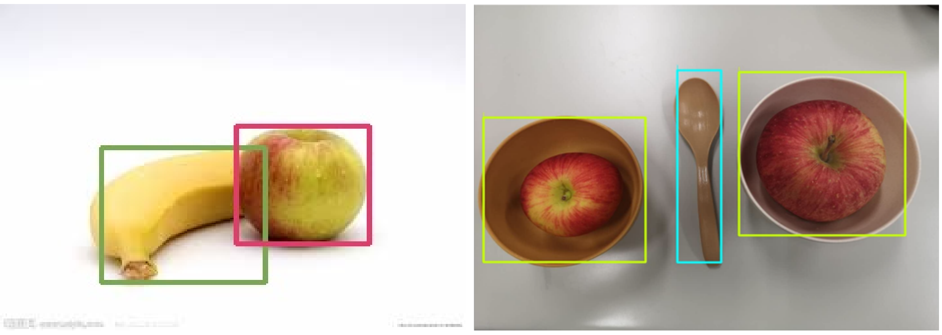
   Through volume estimation, our system can calculate the food calories more precise.

5.2.1 Object detection



(a) (b)

Fig. 5.3 Fruit image before processing



(a) (b)

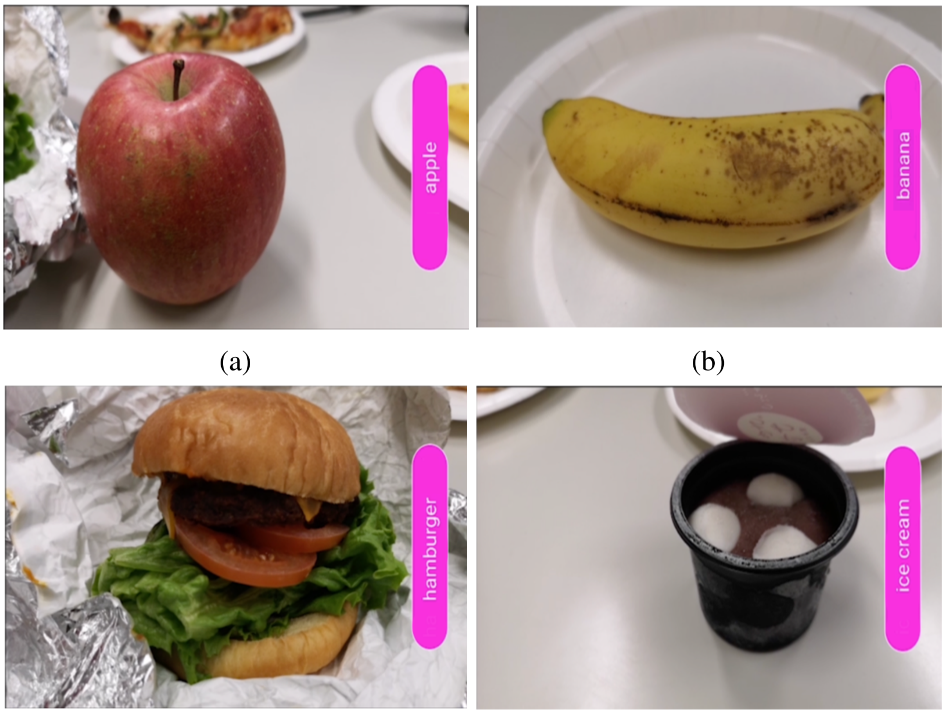
Fig. 5.4 Fruit image processed by object detection

Fig. 5.3 shows the input image and Fig. 5.4 shows the image processed by the object detection. We can see the different foods were located separately in different bounding boxes in Fig. 5.4.

We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

In this step, we distinguished different foods in a single image and this is to prepare for the subsequent classification. At as we can see the bananas in Fig. 5.4, when the foods overlap each other in the picture, the outline may not be easily recognized.

5.2.2 Food classification



(a) (b) (c) (d)

Fig. 5.5 Food classification (The right side shows the food name)

In this part, users are required to wear the Epson glasses. To identify food category, methods [23] [24] based on the pattern of meal-intake motion or sounds produced during eating or drinking. Prior research focused on using a camera to capture good view of food [25] [26] [27].Classification part will recognize the type of food and show the food type to users in real-time through the glasses, as we can see in Fig.5.5. There are mainly four steps to realize food classification.

1. Collect dataset for different kinds of food. We collect images of various foods with food name tags.

2. Build a CNN (convolutional neural network) with two convolutional layers, two pooling layers, a fully connected layer and a softmax output layer.

3. Use this neural network to train the dataset and save the output classification model.

4. Migrate the classification model to wearable device platform via TensorFlow lite.

5.2.3 Volume estimation

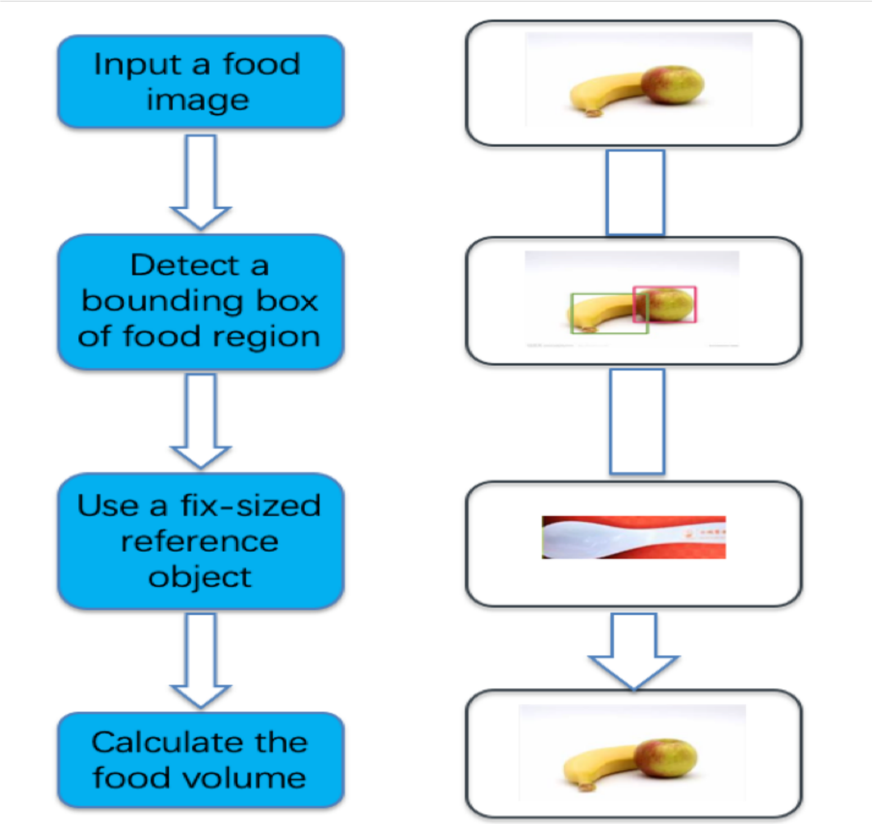

Input a food image
Detect a bounding box of food region
Use a fix-sized reference object
Calculate the food volume

Fig. 5.6 The processing steps of volume estimation

At the beginning of processing steps of volume estimation, we assume that the 3D shape of the food is related to its category, and use the estimated size of food in to top view image to calculate the volume of the food.

First, we detect the food area from the input food image. After processed by the object detection part, we can get the bounding box of the food region in the food image. Then we apply grab-cut on each bounding box to get the accurate food region.

Second, we need a fix-sized reference object appeared in the food image and we should calculate the volume of reference object in advance.

Finally, we can map the 2D size of food pixels to real 3D volume in the real world by comparing the 2D dimensions of the food bounding box and the reference object bounding box in the food image.

5.3 Wrist motion track

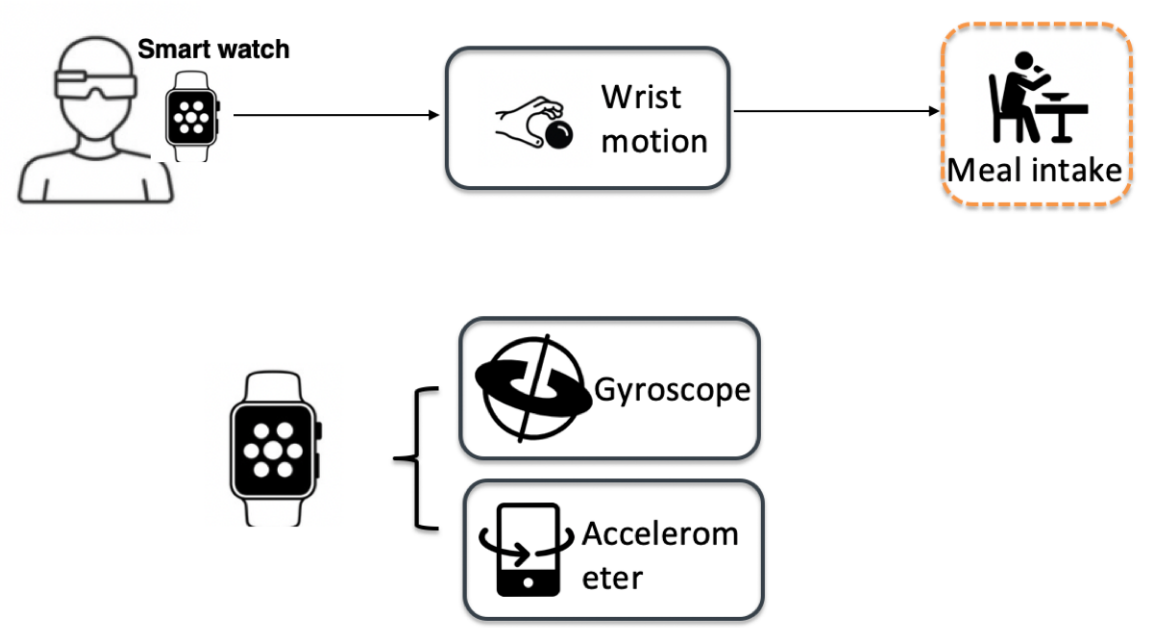


Fig. 5.7 The process of wrist motion tracking

As the Fig. 5.7 shows, users are required to wear smart watch(In our system, we use Fossil smart watch) to get the raw data captured by the gyroscope and accelerometer. Then we will process the raw data by sensor fusion to get the wrist rotation velocity and degree.

In the sensor fusion part, we remove some of the complexity of using Android orientation sensors by applying fusion sensor (Android library). We customize sensor filters and fusions for specific needs:

1. Provide sensor averaging filters in the mean, median and low-pass varieties.

2. Provide sensor fusion backed estimations of device orientation in the complementary and Kalman varieties.

3. Provide estimations of linear acceleration (linear acceleration = acceleration - gravity) in the averaging filter and sensor fusion varieties.

5.3.1 Hardware and Prototype

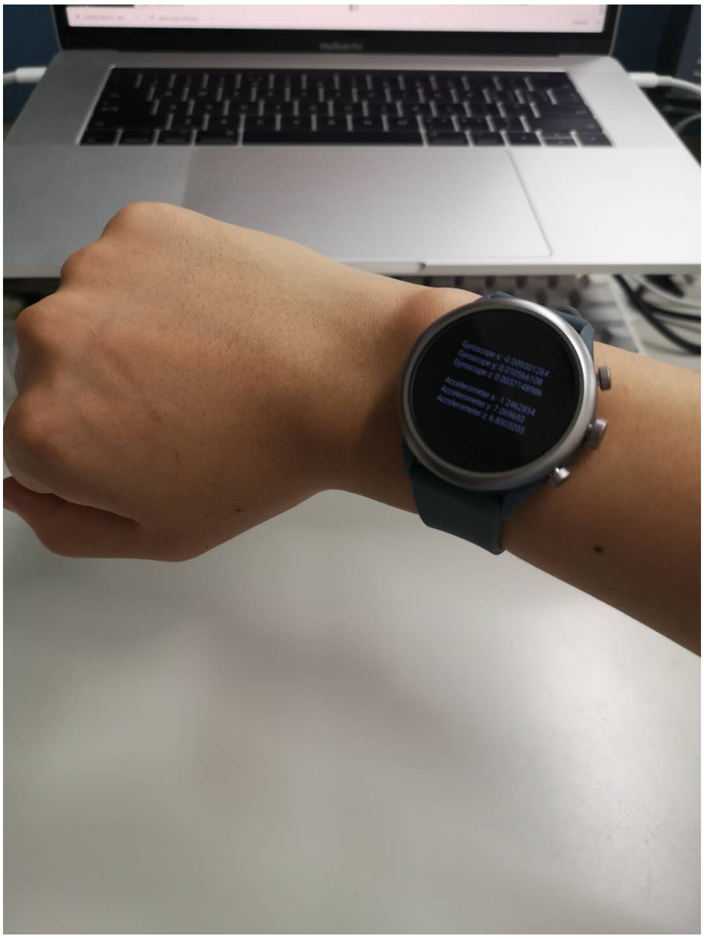


Fig. 5.8 Prototype device using Fossil watch

Fig. 5.8 shows a picture of wearing fossil smart watch. It has gyroscope sensor and accelerometer sensor. We use this smart watch to calculate an orientation heading of our wrist by sensor fusion. Then our wrist rotation velocity and degree will transfer to nearby computer and combine with the data from throat microphone to detect the meal intake activity.

5.3.2 Grab food detection

There are mainly two parts in meal-intake detection:

1. Grab food detection
2. Chew food detection

When we try to complete an eating behavior, firstly we will grab something (food) and move it towards our mouth. The wrist motion tracking is designed for grabbing food detection.

We design an experiment to study the movement of the human wrist during the eating behavior. Four kinds of scenarios are defined in our experiment:

1. User wears the fossil smart watch while eating.
2. User wears the fossil smart watch while walking.
3. User wears the fossil smart watch while typing.
4. User wears the fossil smart watch while lying down.

As we can see in Fig.5.10, we find two features after analyzing the data of wrist rotation velocity from the 30 experiments of each scenarios:

1. In 30 experiments of grabbing food, there are 26 times wrist rotation speed greater than 5 rad/s, and 29 times greater than 4.5 rad/s.
2. In other daily activities (walking, typing and sitting), the wrist rotation speed is mostly less than 4.5 rad / s.

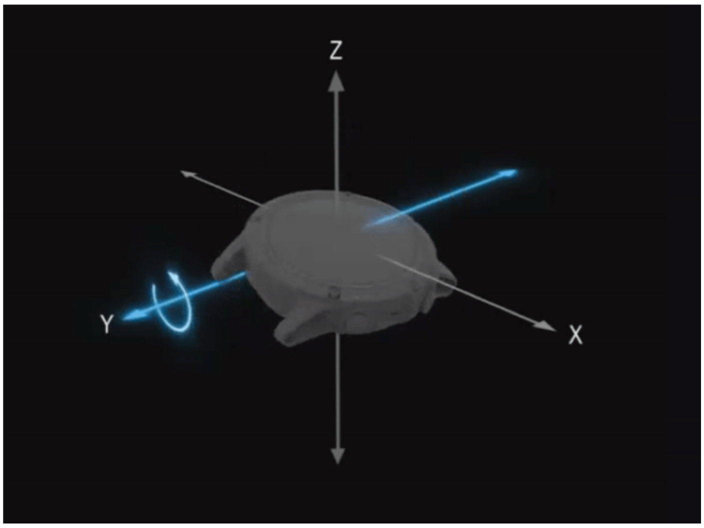
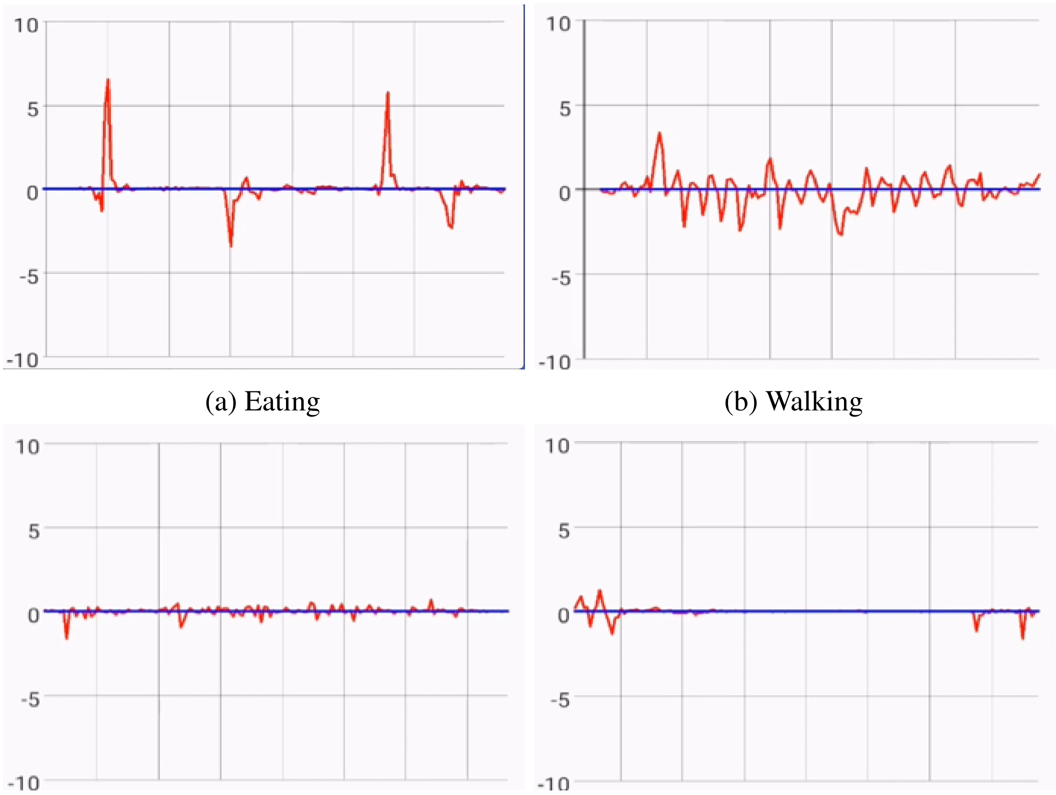


Fig. 5.9 Spatial coordinates of gyroscope and accelerometer sensor



(a) Eating

(b) Walking

(c) Typing

(d) Lying down

Fig. 5.10 Gyroscope sensor data during different activities: Rotation speed (rad/s) around X axis

So we set a threshold (wrist rotation speed should be greater than 4.5 rad/s) to classify the activities that may be a meal intake activity.

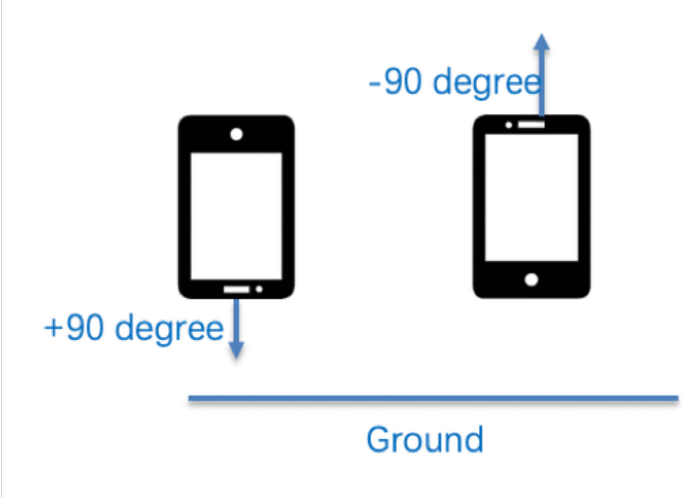
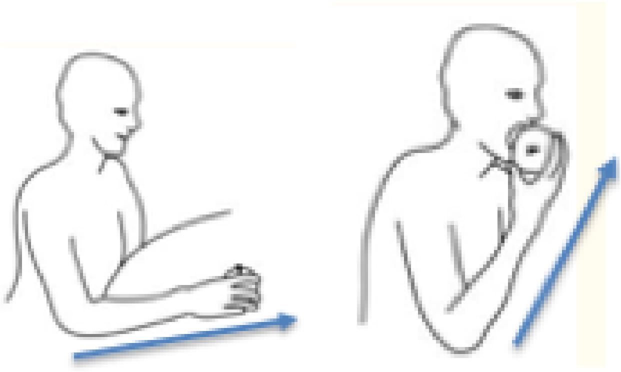


Fig. 5.11 Rotation degree calculated by sensor fusion



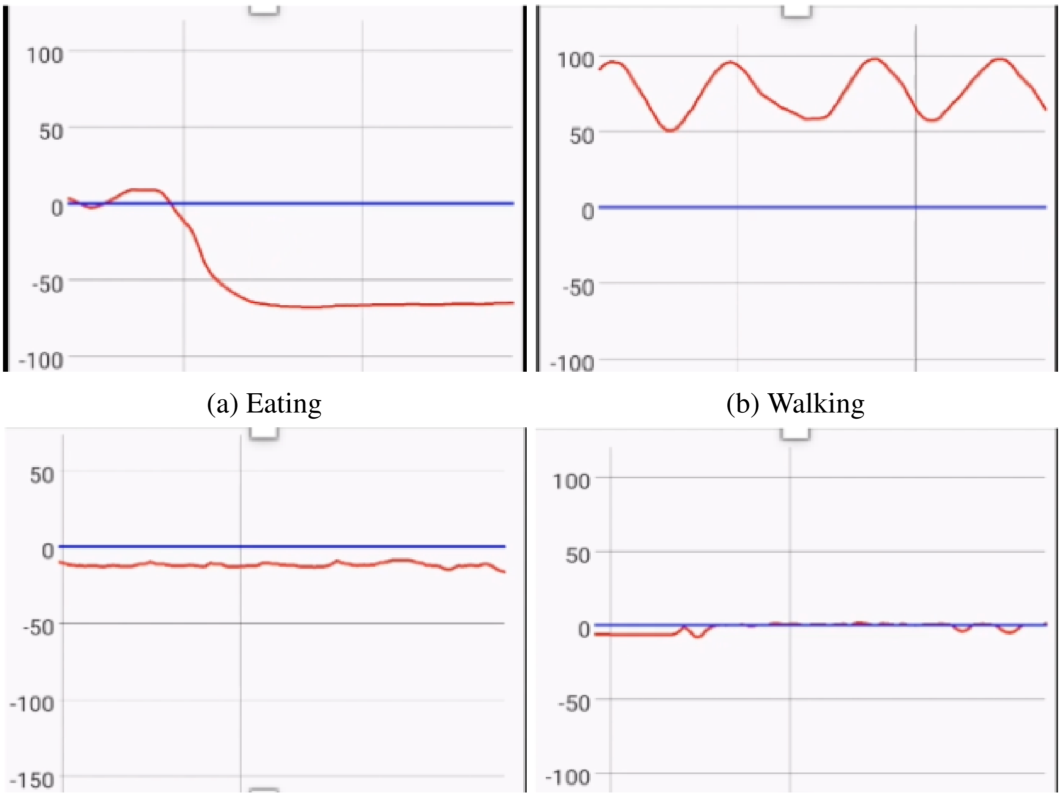
(a) Almost 0 degree (b) Almost -90 degree

Fig. 5.12 Eating apple

We can find two features after analyzing the data of wrist rotation degree from the 30 experiments of each scenarios in fig.5.11:

1. In 30 experiments of grabbing food, there are 27 times wrist rotation degree less than minus 50°.

2. In other daily activities (walking, typing and sitting), the wrist rotation degree is mostly great than minus 50°.



(a) Eating

(b) Walking

(c) Typing

(d) Lying down

Fig. 5.13 Orientation sensor data during different activities: Rotation degree around Y axis

After the gyroscope classify some possible activities, use the orientation sensor for further filtering. In this research, I set the rotation speed of wrist to meet the conditions ($V_{rotY} > 4.5$), and the degree satisfies the condition that should be less than minus 50 ($D_x < 50^\circ$) to be considered as a wrist movement of eating behaviour.

5.4 Chew event detection

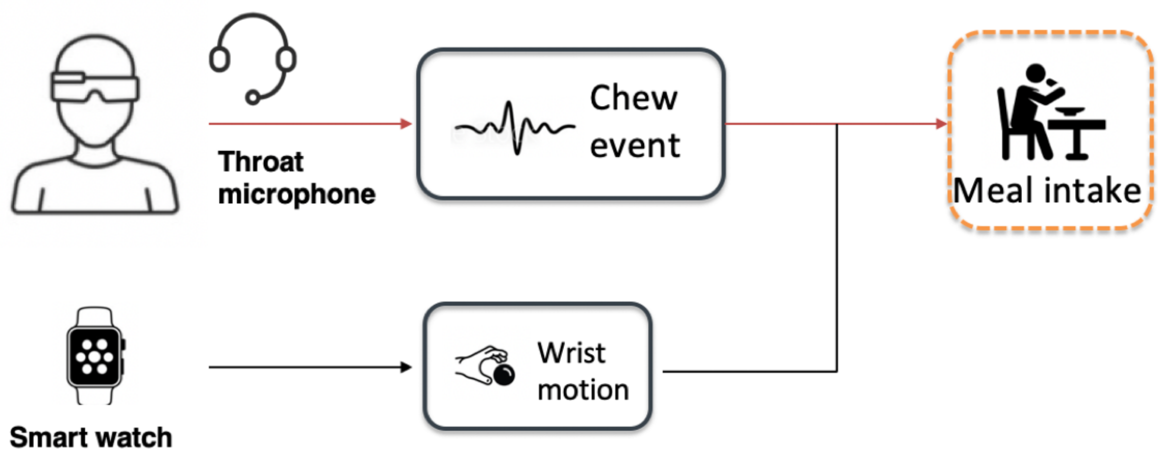


Fig. 5.14 The process of chew event detection

The process of chew event detection showed in the Fig.5.14. In our system, we record the chew sound signal with throat microphone. Then we will extract some features from the audio to classify chew event.

We analysis periodicity and magnitude of the audio spectrum. For each kinds of activities, we collect samples for feature extraction and training to realize chew event detection. The three features we extract from the audio spectrum:

1. Energy

   The sum of squares of the signal values, normalized by the respective frame length.

2. Spectral Flux

   The squared difference between the normalized magnitudes of the spectra of the two successive frames.

3. Zero Crossing Rate

   The rate of sign-changes of the signal during the duration of a particular frame.

5.4.1 Hardware and prototype

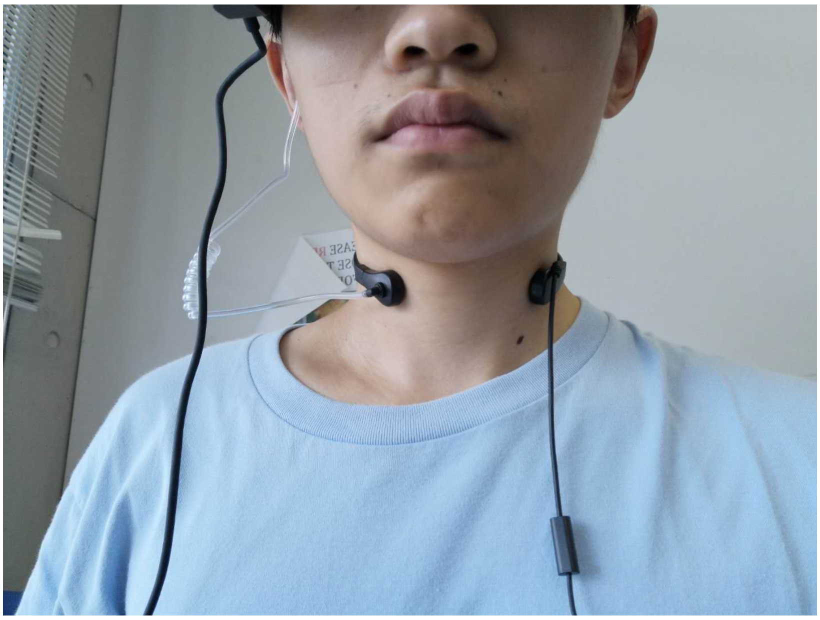


Fig. 5.15 Prototype device using throat microphone

Fig.5.15 shows a picture of wearing throat microphone connected to head mounted display MOVERIO BT-300. We use throat microphone to detect the chew event. It is a type of piezoelectric microphone (contact microphone) and is convenient to wear around the neck. Compared to other type of microphone, such as normal dynamic microphone, it can eliminate the background noise when the user is eating in a noisy ambient sound background.

5.4.2 Meal intake detection

We find that when the user grabs something to eat, there will be a necessary wrist rotation occurs and the user should put that into mouth. For most eating situations, the wrist must be rolled regardless of the type of food or liquid.

Our algorithm for detecting meal intake based on this motion pattern can be implemented as follows:

```
Let EVENT = 0
Loop
```

```
Let $V_t$ = measured rotation.velocity at time t
Let $D_{t+1}$ = measured rotation.degree at time t+1
If $V_t$ > threshold.velocity and EVENT = 0
  EVENT = 1
If $D_{t+1}$ < threshold.degree and EVENT = 1
  EVENT = 2
If EVENT = 2 and ChewEvent.detected
  Meal intake detected
```

We define a wrist motion pattern and chew event pattern for meal intake detection. First, the velocity of wrist rotation must surpass a preset threshold of velocity; second, the degree of wrist rotation must be great than a preset threshold of degree. The third event is detecting the chewing sound signal.

The variable EVENT iterated through the event of the cycle of detecting human activities. The preset threshold we mentioned in the wrist motion tracking section.

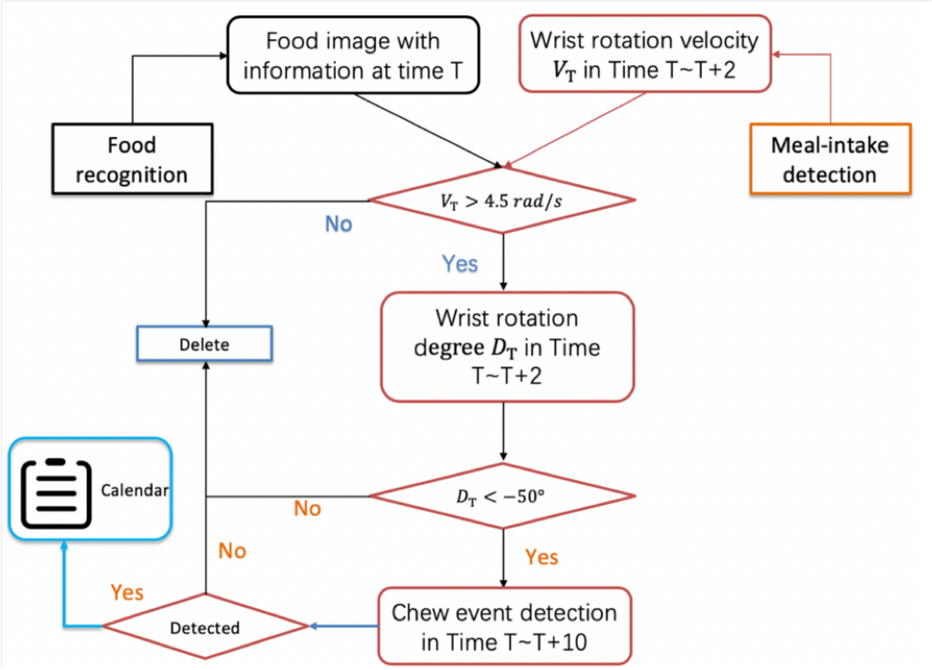

Food image with information at time T
Wrist rotation velocity V_T in Time T~T+2
Food recognition
Meal-intake detection
$V_T > 4.5\ rad/s$
No
Yes
Delete
Wrist rotation degree D_T in Time T~T+2
Calendar
$D_T < -50°$
No
No
Yes
Yes
Detected
Chew event detection in Time T~T+10

Fig. 5.16 Process of generating Calendar

Chapter 6

System Implementation

6.1 Hardware Setup

We divided this system into two parts: the wearable Unit and server. To achieve this system, we need:

1. An augmented reality glass with a camera;
2. An android platform smartwatch with gyroscope sensor and accelerometer sensor;
3. A contact microphone that can eliminate background noise;
4. A computer as a server to store the component data.

Fig. 6.1 Required Hardware

Therefore, we choose MOVERIO BT-300. It is a see-through type head-mounted display with a built-in camera. It has a controller with a trackpad and some keys to operate the android operating system.

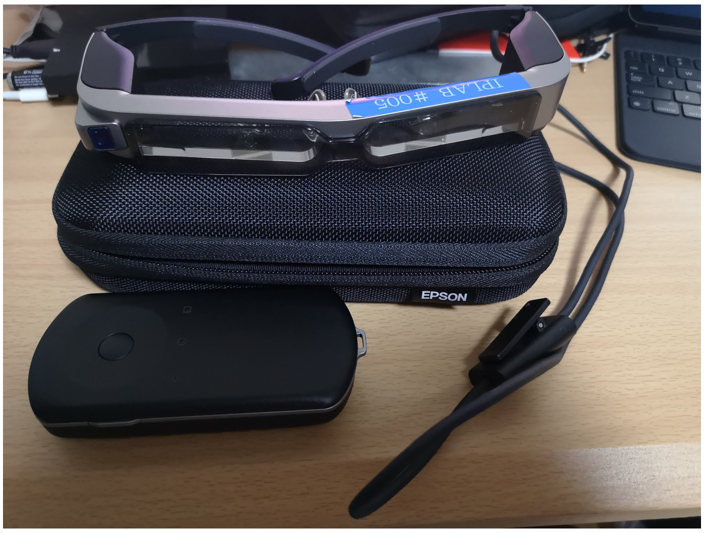


Fig. 6.2 MOVERIO BT-300

To make meal intake detection possible, we use a fossil smartwatch with gyroscope sensor and accelerometer sensor to track our wrist motion. Because in most eating situations, a roll of wrist must occur and wrist also moves towards mouth.

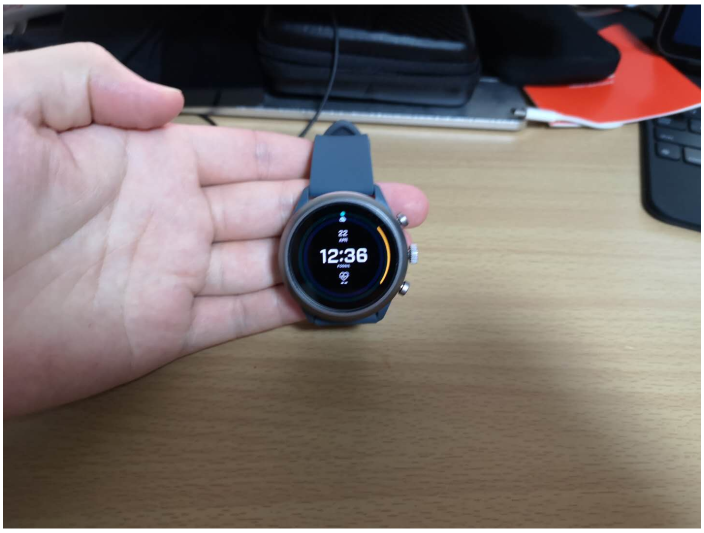


Fig. 6.3 Fossil smart watch

In order to make sure the user have eaten the food detect by our glasses, we utilize the throat microphone to record the chew event. Throat microphone is a type of contact microphone and is designed to be worn around the neck. Other types of microphone do not function well if background noise is high.

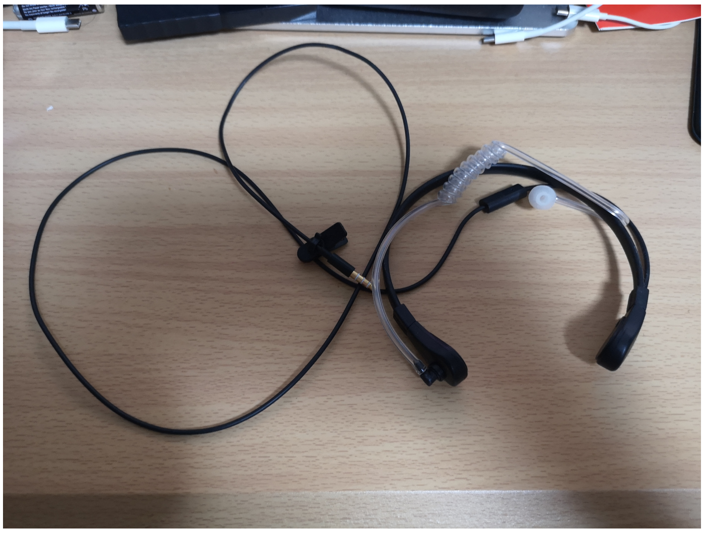


Fig. 6.4 Throat microphone

So as to store and process the metadata from the gyroscope sensor and accelerometer sensor, we use a laptop as a server.

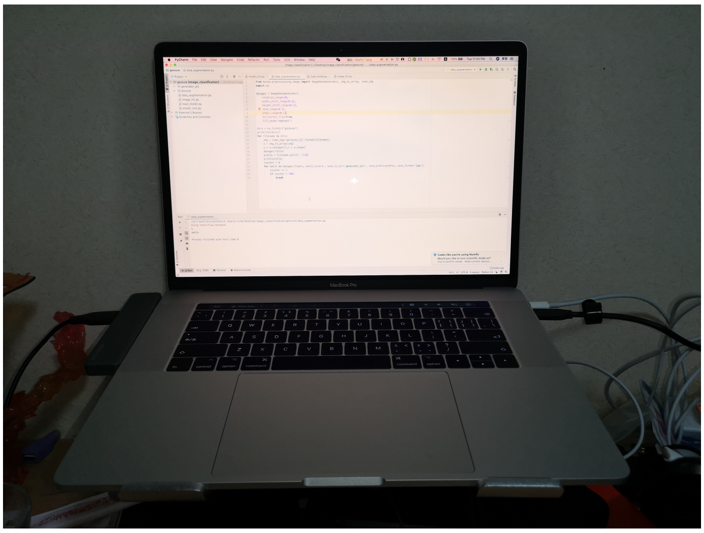


Fig. 6.5 Laptop

Method	Model specifications	Main Unit
Food recognition	MOVERIO BT-300	Camera and display
Wrist motion tracking	Fossil smart watch	Gyroscope and accelerometer
Chew event detection	Throat microphone	Microphone

Table 6.1 Hardware Setup

6.2 Software Environment

The other technical supports are:

1. Tensorflow, an end-to-end open source platform for machine learning. We use it for building and training the machine learning model.
2. Tensorflow Lite, an open source deep learning framework for on-device inference. It helps us to migrate the model to wearable device platform (Android operating system).

To process the metadata and develop the user interface, I use Pycharm professional 2019, and Android Studio 3.

6.3 Food image classification

Some traditional image process techniques have been researched before deep learning. Bosch [28]proposed an approach to food identification using several features based on local and global measures and a “voting” based late decision. Technology Assisted Dietary Assessment (TADA) [29] project researched on some method [30] [31] [32] to realize food recognition, calories estimation and signal processing techniques to assess dietary intake.

Most recent works focus on convolutional neural network for food image classification [33] [34].

6.3.1 Data set

Use Google Dataset Search and other public data sets to search for data set, and get suitable data set on the Kaggle website. For the shortcomings of the data set, we buy some foods and take photos to complete the collection of data sets.

Due to different factors such as the size of the open source data set on the Internet and the data set collected by ourselves, the data set needs to be processed. And then we get the data set that can be used for machine learning training phase. Because the number of data sets we collected is limited, so the data set needs to be augmented. We enhance the data set through processing the images by rotation and stretching. Deep learning emphasizes the ability to automatically learn features from data. Without enough training samples, it is difficult to achieve the desired results.

Fig. 6.6 A single picture of apple

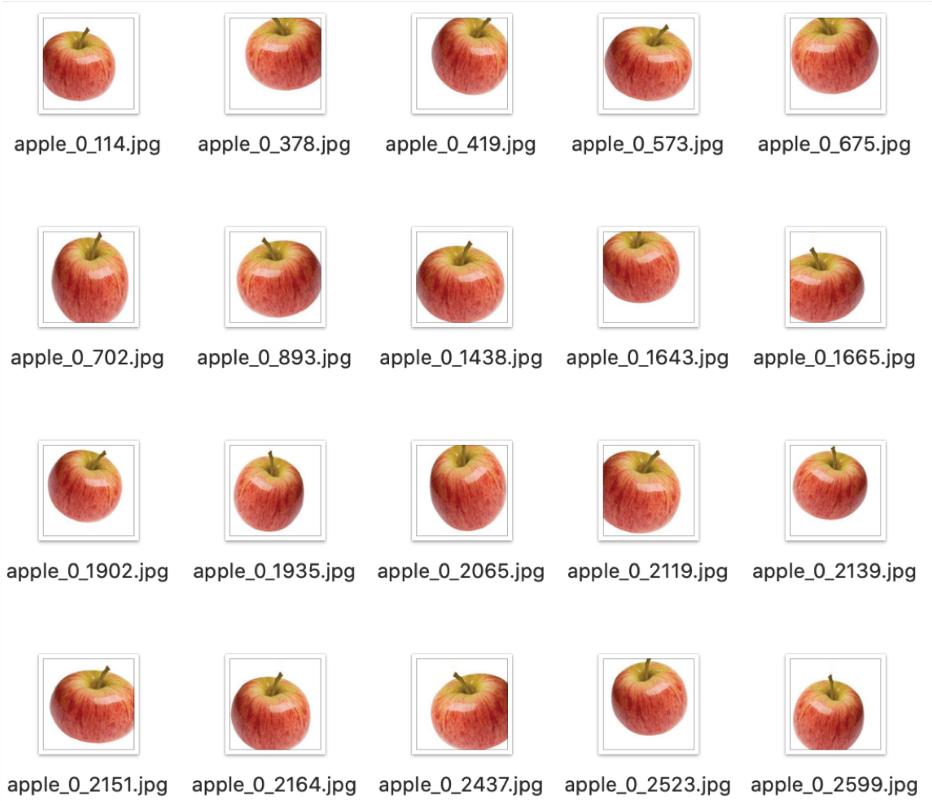


Fig. 6.7 After data enhancement

As we can see in .In order to obtain different training data as much as possible, we use the Keras module to randomly transform the data set for data enhancement. By setting the angle of the randomly selected picture, specifying the degree of random movement in the horizontal and vertical directions, the degree of shear transformation, random enlargement, and then horizontally flipping the picture and filling adjacent pixels, the single picture is enhanced to 100 different pictures.

6.3.2 Data format

The H5 file is the fifth generation version of the Hierarchical Data Format, which is a file format and library file used to store scientific data. This file format was developed by the US Super computing and Application Center to store and organize large-scale data. H5 files have excellent characteristics in terms of memory footprint, compression, and access speed. HDF5 simplifies the file structure into two main object types:

1. Data set, multi-dimensional arrays of the same type of data.

2. Group, a container structure that can contain data sets and other groups.

The resources inside the HDF5 file are accessed through POSIX-like syntax. Metadata is defined by the user and attached to groups and data sets in the form of named attributes. More complex storage forms such as images and tables can be constructed using data sets, groups, and attributes. HDF5 also includes an improved type system and data space objects to represent the selection of data areas.

The implementation steps are:

1. Convert the images in the data set into RGB matrix.

2. Save the data set (matrix) and the label of each picture as a .h5 file.

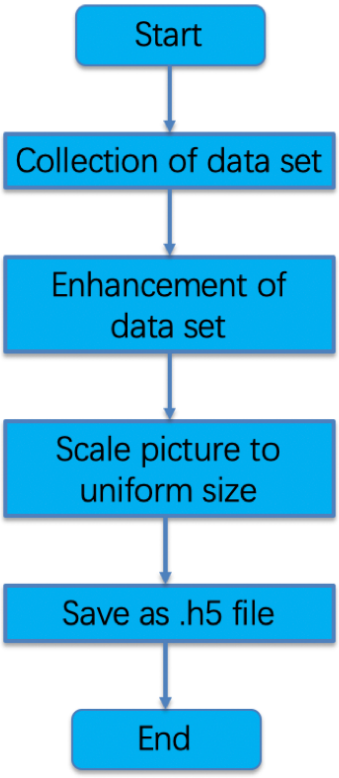


Fig. 6.8 Flow chart of preparing data set

6.3.3 Choose optimizer

There are many optimizers for selection in TensorFlow: Optimizer, GradientDescent, AdadeltaOptimizer, and AdamOptimizer. Keras also provides SGD and some other optimizers. Choose an suitable optimizer is also critical to the experimental results. There are the most common optimizers:

1. Batch gradient descent (BGD)

   Gradient update rule: Each iteration requires all training samples. The advantage is that under ideal conditions, the global optimal can be obtained after enough iterations, which is suitable for small sample data sets. However, the calculation speed will be very slow for large data sets, which is not suitable for large data sets.

2. Stochastic gradient descent (SGD)

   Compared with BGD's calculation of all data sets at once, SGD performs gradient updates for each sample every time it is updated. SGD only performs one update at a time, there is no redundancy, and the speed is faster. You can add samples. The disadvantage is that the update is more frequent, and the cost function will have serious shocks.

3. Mini-batch gradient descent (MBGD)

   MBGD uses a part of samples to update the parameters each time, which can make the convergence more stable when reducing the parameter update variance. The disadvantage is that it does not guarantee good convergence. If the selection of learning rate is too small, the convergence rate will be very slow. If it is too large, the loss function will oscillate or even deviate at the minimum value.

4. Adam

   This algorithm is another method to calculate the adaptive learning rate of each parameter. The Adam algorithm uses the mean of the gradient and the uncentralized variance of the gradient for comprehensive consideration, and finally calculates the appropriate update step size. TensorFlow.train.AdamOptimizer, which is provided by TensorFlow, can control the learning speed and after the offset correction, the learning rate of each iteration has a certain range which makes the parameters relatively stable.

Overall, Adam is the best choice because the data is relatively sparse, which is more suitable for this research method.

6.3.4 Loss function

The loss function represents the loss value by calculating the difference between the predicted value and the true value. When training the convolutional neural networks, we try different parameter sets to make the loss function converge faster and let the loss function reach the minimum value. By visualizing the loss function, we can visually observe the convergence of the loss function from the chart. There are some common loss functions:

1. Mean square error cost function

   Minimize the square of the difference between the target value and the predicted value.

2. Customized loss function

   Customize the loss function according to the demand, for example: define the relationship between the prediction accuracy and the predicted amount, give weight to the predicted amount and the predicted result, and evaluate the loss function.

3. Cross entropy cost function

   $$C = -\frac{1}{n}\sum_{x}[y\ln a + (1-y)\ln(1-a)]$$

   In the equation, a is the input of the neuron. n is the total number of training data, and y is the predicted value for the training data.

4. L1 norm and L2 norm loss functions

   The two decisions in machine learning are: loss function of L1 norm and L2 norm, L1 regularization and L2 regularization.

The L1 norm loss function is also called the minimum absolute value deviation (LAD) and the minimum absolute value error (LAE). In general, it is to minimize the sum of the absolute difference between the target value (Y_i) and the estimated value ($f(x_i)$):

$$S = \sum_{i=1}^{n} |Y_i - f(x_i)|$$

The L2 norm loss function is also known as the least square error (LSE). In general, it is to minimize the sum of squares (S) of the difference between the target value (Y_i) and the estimated value ($f(x_i)$):

$$S = \sum_{i=1}^{n} (Y_i - f(x_i))^2$$

Intuitively, because the L2 norm squares the error, the model's error will be larger than the L1 norm, so the model will be more sensitive to the sample, and the model needs to be adjusted to minimize the error. If the sample is an outlier, the model needs to be adjusted to fit the single outlier.

In deep learning, regularization is an important technique to prevent overfitting. Mathematically speaking, it will add a regular term to prevent the coefficients from being fitted too well to overfitting. The only difference between L1 and L2 is that L2 is the sum of squares of weights, and L1 is the sum of weights. as follows:

(a) L1 regularization of the least square loss function

$$w^* = \arg min_w \sum_j (t(x_j) - \sum_i w_i h_i(x_j))^2 + \lambda \sum_{i=1}^{k} |w_i|$$

(b) L2 regularization of the least square loss function

$$w^* = \arg min_w \sum_j (t(x_j) - \sum_i w_i h_i(x_j))^2 + \lambda \sum_{i=1}^{k} (w_i)^2$$

Built-in feature selection is a useful property that the L1 norm is often mentioned, while the L2 norm does not. This is a result of the L1 norm, which tends to produce sparse coefficients. Suppose the model has 100 coefficients, but only 10 of them are non-zero, which actually means "the remaining 90 coefficients are all useless when predicting the target value." The L2 norm produces non-sparse coefficients, so it does not possess this property.

Sparseness means that only a few items in a matrix (or vector) are non-zero. The L1 norm has properties: it produces many 0 or very small coefficients and a few large coefficients.

Computational efficiency. The L1 norm does not have an analytical solution, but the L2 norm does. This allows the L2 norm to be calculated efficiently. However, the solution of the L1 norm is sparse, which allows it to use sparse algorithms to make the calculation more efficient.

Using the cross-entropy cost function and L2 norm regularization can ensure higher computational efficiency and faster learning speed.

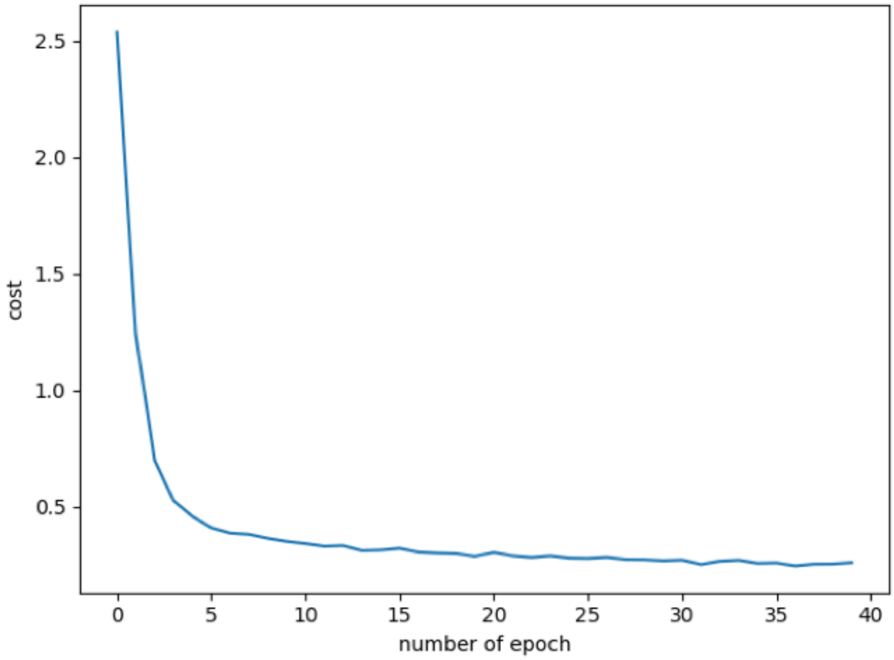


Fig. 6.9 Visualization of loss function

6.4 Graphical User Interface For Food-tracker

6.4.1 Calendar

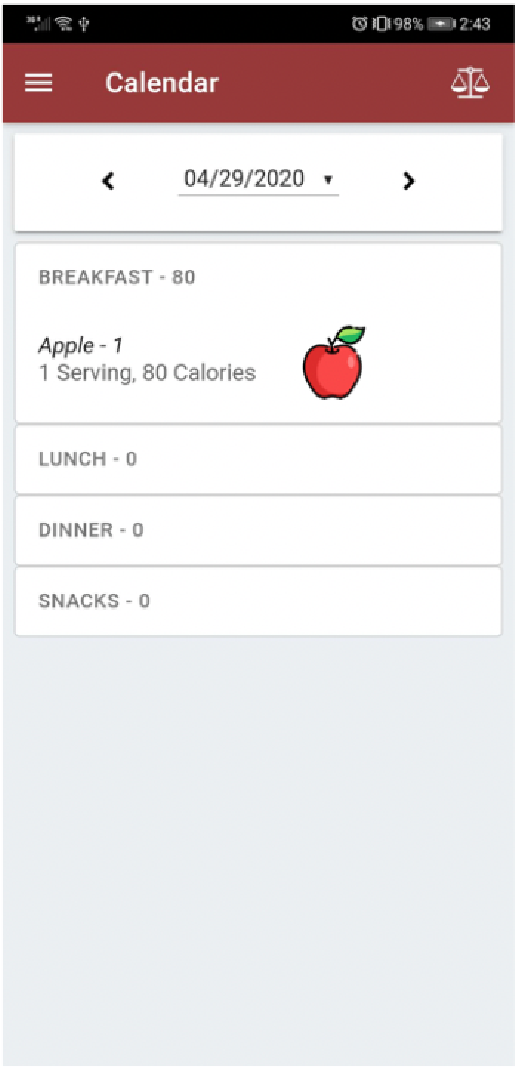
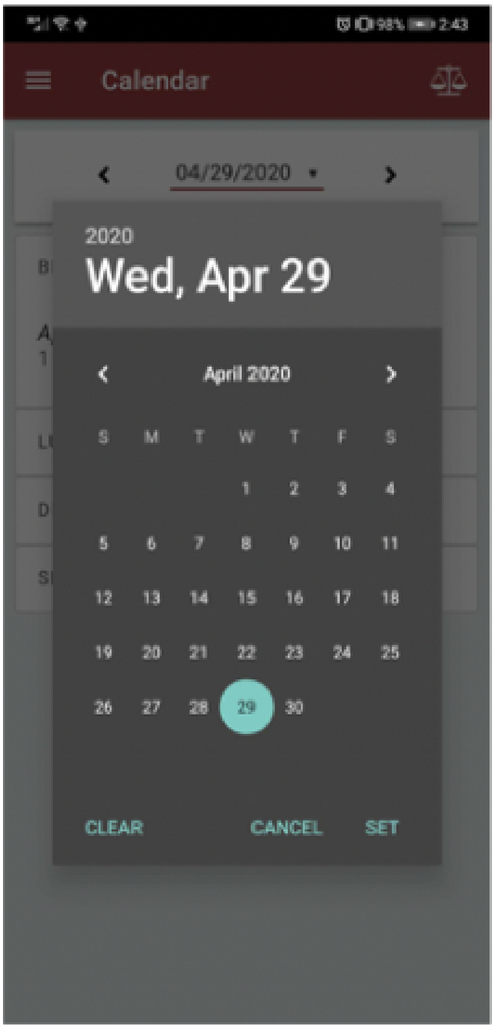


Fig. 6.10 Calendar interface for reviewing the account of food

When the user uses our system to review the diet record, the user needs to know the time of meal, the amount of food they have eaten and the calories of the food. As we can see in Fig.6.10. In order to facilitate user review, our system record the food user eat at different times every day. In this calendar interface, we divide the day into four parts:

1. Breakfast: Food consumed by users between 6:00am and 9:00am.
2. Lunch: Food consumed by users between 11:00am and 13:00pm.
3. Dinner: Food consumed by users between 17:00pm and 20:00pm.
4. Snacks: Food consumed by users at other times.

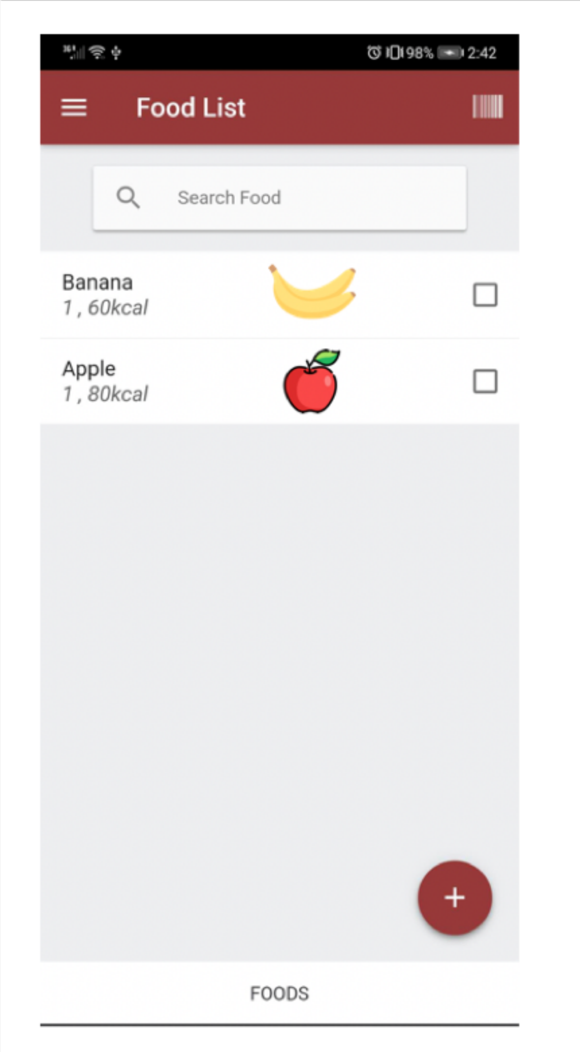


S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Fig. 6.11 Expanding the calendar when selecting a date

User can get an expanded calendar by clicking the time bar displayed on the top of the interface when the user wants to view the diet record of a previous day, as we can see the interface in the Fig.6.11.

6.4.2 Food list



Banana
1 , 60kcal □

Apple
1 , 80kcal □

Fig. 6.12 Food list interface for creating local database of foods

The user can create a local database of foods here. This interface is mainly composed of three parts:

1. Search bar

When the user wants to use our system to search for a specific delicious food that has been eaten one day before, or wants to modify the information of a specific food. The user can quickly search by inputting the name of the food.

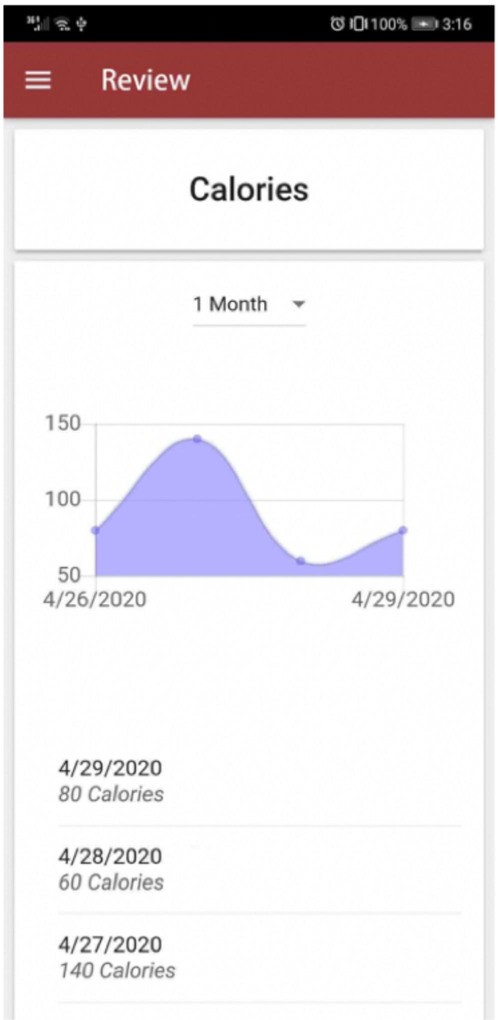
2. Display of food information

   This part displays the information of food that has been automatically recognized and recorded by our system or manually added by the user. The information concludes the name of the food and the calories of the food.

3. Manually add button

   If for some special food that cannot be automatically recognized by the system, the user can also add the food to the local database by clicking the manual add button.

6.4.3 Review



4/29/2020
80 Calories

4/28/2020
60 Calories

4/27/2020
140 Calories

Fig. 6.13 Review interface for viewing past calories intake

For a good recording system, statistical functions are essential. This interface is mainly used to help users organize their diet data about the calorie intake. Users can choose different time periods (one year, one month, one week, etc.) to view the calorie intake curve, which may motivate users to develop better eating habits.

6.5 Sensor fusion

6.5.1 Averaging Filters

We implements the three most common smoothing filters, namely low-pass filter, mean filter and median filter. The user can configure all filters according to the time constant (in seconds). The larger the time constant, the smoother the signal. However, the delay also increases with the time constant. Since the filter coefficients are in the time domain, the difference in sensor output frequency has little effect on the performance of the filter. Regardless of the sensor frequency, the performance of these filters should be about the same on all devices.

1. Low-Pass Filter

   The coefficient alpha can be adjusted based on the sampling period of the sensor to produce the required time constant on which the filter will act. It takes the simple form of output [0] = alpha * output [0] + (1-alpha) * input [0]. Alpha is defined as alpha = timeConstant / (timeConstant + dt), where the time constant is the length of the signal that the filter should act on, and dt is the sampling period of the sensor (1/frequency). The calculation efficiency is relative to the mean or median filter (constant time and linear time).

2. Mean Filter

   A mean filter is designed to make the data points based on a time constant in units of seconds more smoothly. The meal filter will average the samples that appear within the time period defined by the time constant. The average number of samples is called the filter window. This method allows a filter window to be defined over a period of time instead of a fixed number of samples.

3. Median Filter A median filter functions similar to mean filter. The median filter will take the median of the samples that occur over a period defined by the time constant. The average number of samples is called the filter window. This method allows a filter window to be defined over a period of time instead of a fixed number of samples.

6.5.2 Orientation Sensor Fusions

The gyroscope is the basic direction sensor. However, due to rounding errors and other factors, the gyroscope tends to drift. Most gyroscopes work by measuring very small vibrations in the rotation of the earth, which means they really do not like external vibrations. Due to drift and external vibration, the gyroscope must be compensated with a second estimate of device orientation from the acceleration sensor and magnetic sensor. The acceleration sensor provides pitch and roll estimates, while the magnetic sensor provides azimuth.

Among the two fusion sensors returned by the gyroscope, acceleration and magnetic sensors, the first is based on a complementary filter supported by quaternion, and the second is based on a Kalman filter supported by quaternion. Both fusions use acceleration sensors, magnetic sensors and gyro sensors to estimate the orientation of the device relative to world space coordinates.

Quaternion provides an angular axis solution for rotation so that rotation will not affected by many singularities in the rotation matrix, including gimbal locking. Quaternions can also be scaled and applied to complementary filters. Although the quaternion complementary filter may also be the most difficult to implement, it may be the most durable and accurate filter among the filters.

1. Quaternions complementary Filter

   The complementary filter is a frequency domain filter. In the strictest sense, the definition of complementary filter refers to the use of two or more transfer functions, which are mathematical complements of each other. Therefore, if the data from one sensor is calculated by G(s), the data from the other sensor is calculated by I-G(s), and the sum of the transfer functions is I, which is the identity matrix. In fact, it looks almost the same as a low-pass filter, but uses two different sets of sensor measurements to produce a value that can be viewed as a weighted estimate.

   Complementary filters are used to fuse the two estimated values (gyroscope and acceleration/magnetic field, respectively). It takes the form of gyroscope [0] = alpha * gyroscope [0] + (1- alpha) * acceleration/magnetism [0]. Alpha is defined as alpha = timeConstant / (timeConstant + dt), where the time constant is the signal length that the filter should act on, and dt is the sampling period of the sensor (1/frequency).

2. Quaternion Kalman Filter

   Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, which contains noise (random changes) and other inaccuracies, and generates unknown variables The estimates are often more accurate than those based on only one measurement. Kalman filter performs a recursive operation on the noisy input data stream to produce a statistically optimal estimate of the basic system state. Much like the complementary filter, the Kalman filter requires two sets of estimates, which we obtained from the gyroscope and acceleration/magnetic sensor.

6.5.3 Linear Acceleration

We design different linear acceleration filter. Linear acceleration is defined as linearAcceleration = (acceleration-gravity). The acceleration sensor cannot determine the difference between gravity/tilt and true linear acceleration. There is an independent method, a low-pass filter and many methods based on sensor fusion.

1. Android Linear Acceleration

   Android provides its own linear acceleration implementation through Sensor.Type_LINEAR_ACCELE In most cases, the device must have a gyroscope to support this sensor type. However, some devices may implement Sensor.TYPE_LINEAR_ACCELERATION without a gyroscope, probably through a low-pass filter. In our research, we found that Sensor.TYPE_LINEAR_ACCELERATION works well in short-term linear acceleration, but it does not work well in the long-term (more than a few seconds).

2. Low-Pass Linear Acceleration

The simplest linear acceleration filter is based on a low-pass filter. The advantage is that no other sensors are needed to estimate the linear acceleration, and the calculation efficiency is high. The low-pass filter is implemented in such a way that only long-term (low-frequency) signals (i.e, gravity) are allowed to pass through. All short-term (high-frequency) content will be filtered out. Then subtract the estimated gravity value from the current acceleration sensor measurement to provide an estimated linear acceleration. The low-pass filter is an IIR unipolar implementation. The coefficient (alpha) can be adjusted based on the sampling period of the sensor to produce the required time constant on which the filter will act. It uses a simple form of gravity [0] = alpha *gravity [0] + (1-alpha) * acceleration [0]. Alpha is defined as alpha = timeConstant / (timeConstant + dt), where the time constant is the signal length that the filter should act on, and dt is the sampling period of the sensor (1/frequency). The linear acceleration can then be calculated as linearAcceleration = (acceleration-gravity). Assuming that the acceleration sensor is installed in a relatively fixed position and the period of linear acceleration is relatively short, this implementation can work well.

3. IMU Sensor Fusion Linear Acceleration

   Calculating the gravity components of a normalized orientation is trivial, so we can use the IMU orientation fusions to provide an estimation of linear acceleration that is far more customizable than what Android provides alone.

6.5.4 Sensor Offset Calibration

Our system contains an algorithm that can compensate for the deformation of hard and soft iron distortions in the magnetic field, which can also be used to correct static offsets in acceleration sensors. This algorithm fits the points on the ellipsoid to the polynomial expression:

$$Ax^2+By^2+Cz^2+2Dxy+2Exz+2Fyz+2Gx+2Hy+2Iz=1$$

The polynomial expression is then solved and the center and radius of the ellipse are determined.

6.6 Audio analysis

There are mainly three steps in audio analysis for chew event detection:

1. Chewing sound signals recording
2. "Chew-like" signal detection
3. Chewing sound verification

6.6.1 Chewing sound signal recording

Throat microphone is a type of piezoelectric microphone. We use throat microphone in our system to eliminate the background noise as much as possible. When we use the normal dynamic microphone to test the chewing sound signals recording, we find that:

1. Under quiet ambient sound, the record of chewing sounds are similar with throat microphone.
2. Under a noisy ambient sound background, the record of chewing sounds is more affected compared with throat microphone.

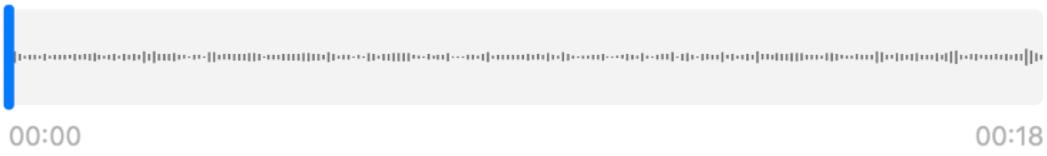


Fig. 6.14 Headphone recording when background is playing music

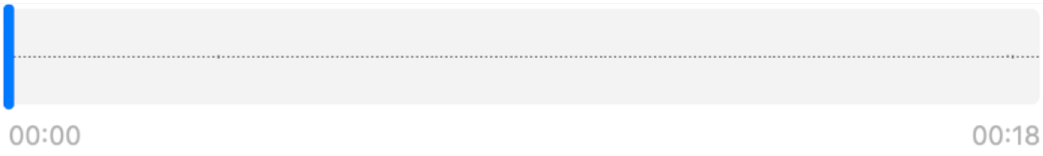


Fig. 6.15 Throat microphone records when background is playing music

6.6.2 Energy

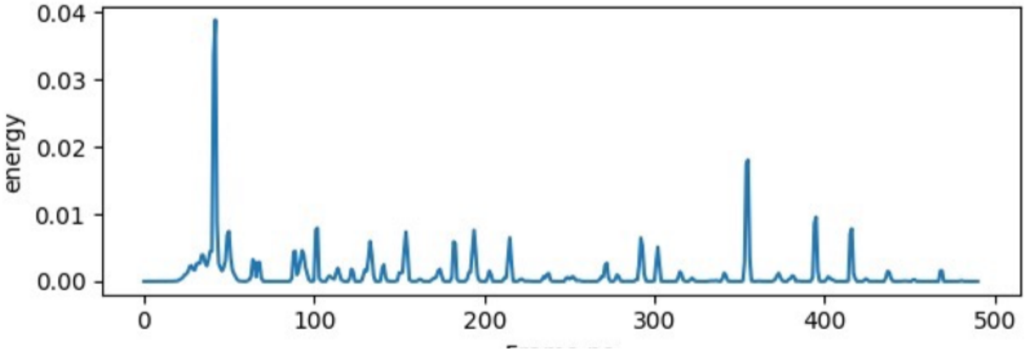


Fig. 6.16 Log energy of each frame during eating activity

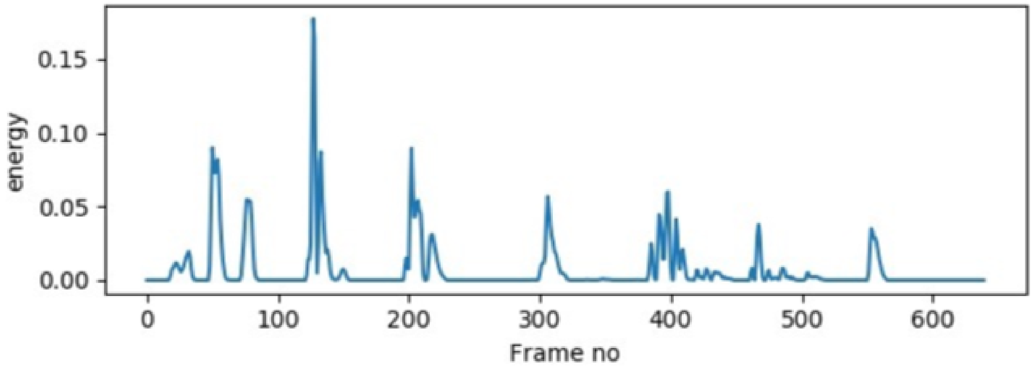


Fig. 6.17 Log energy of each frame during speaking activity

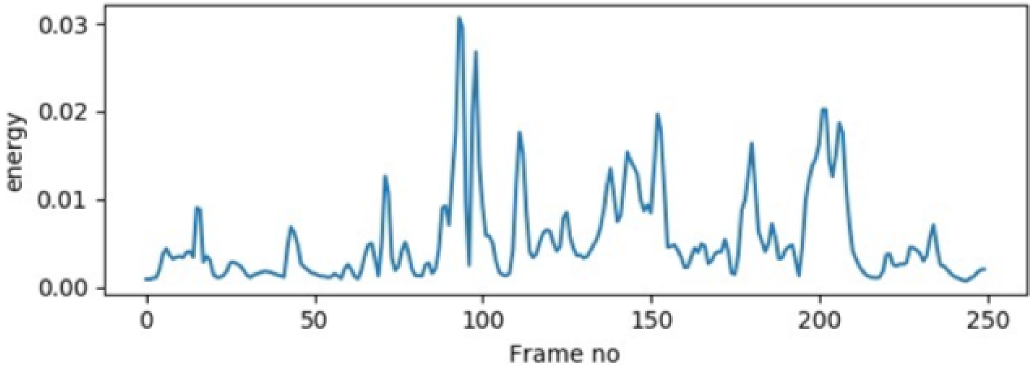


Fig. 6.18 Log energy of each frame during singing activity

6.6.3 HTTP Server

We designed a HTTP server for our system to transmit the data captured by sensors.

When the server receives a request from the AR glasses or other sensors, it converts the request into a JSON object. Then a Kafka Producer is called to send the string out for processing.

During the processing phase, the server calls a Consumer who starts monitoring message of the output topic. Each received message will be parsed, and once the sensor ID in a message is the same as the current sensor, server will return it to AR glasses or the Calendar.

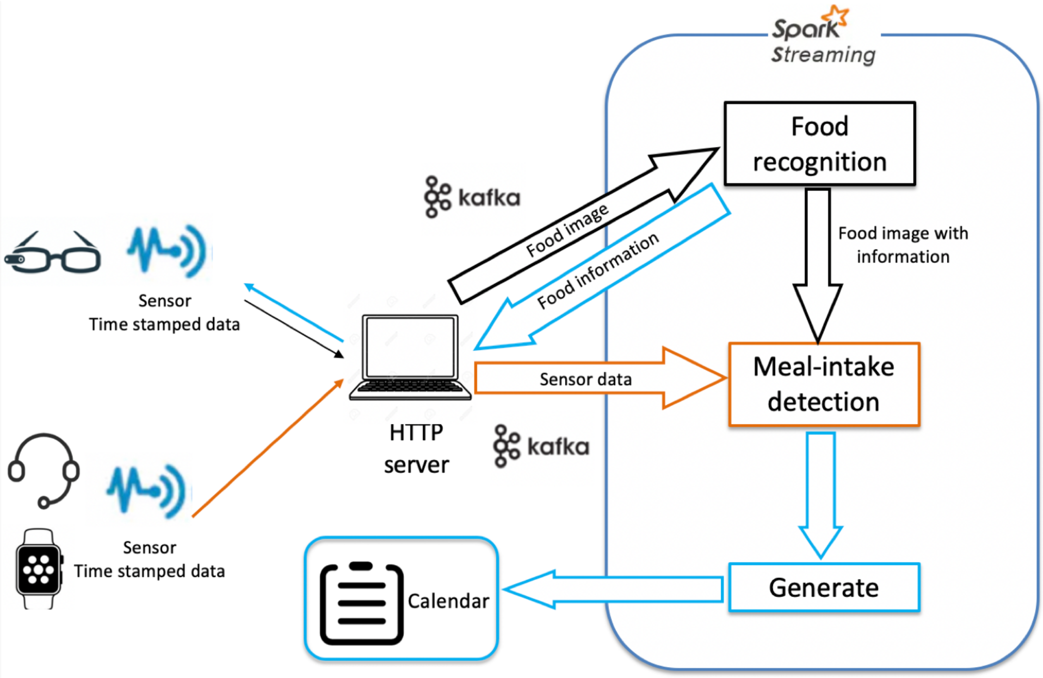


Fig. 6.19 Data transmission

Chapter 7

Preliminary Evaluation

In this section, we introduce our preliminary user research and result in analysis. We asked our participants to accomplish several tasks with our system in order to verify whether our system can provide better experiences and better efficiency when compared with current food recording system.

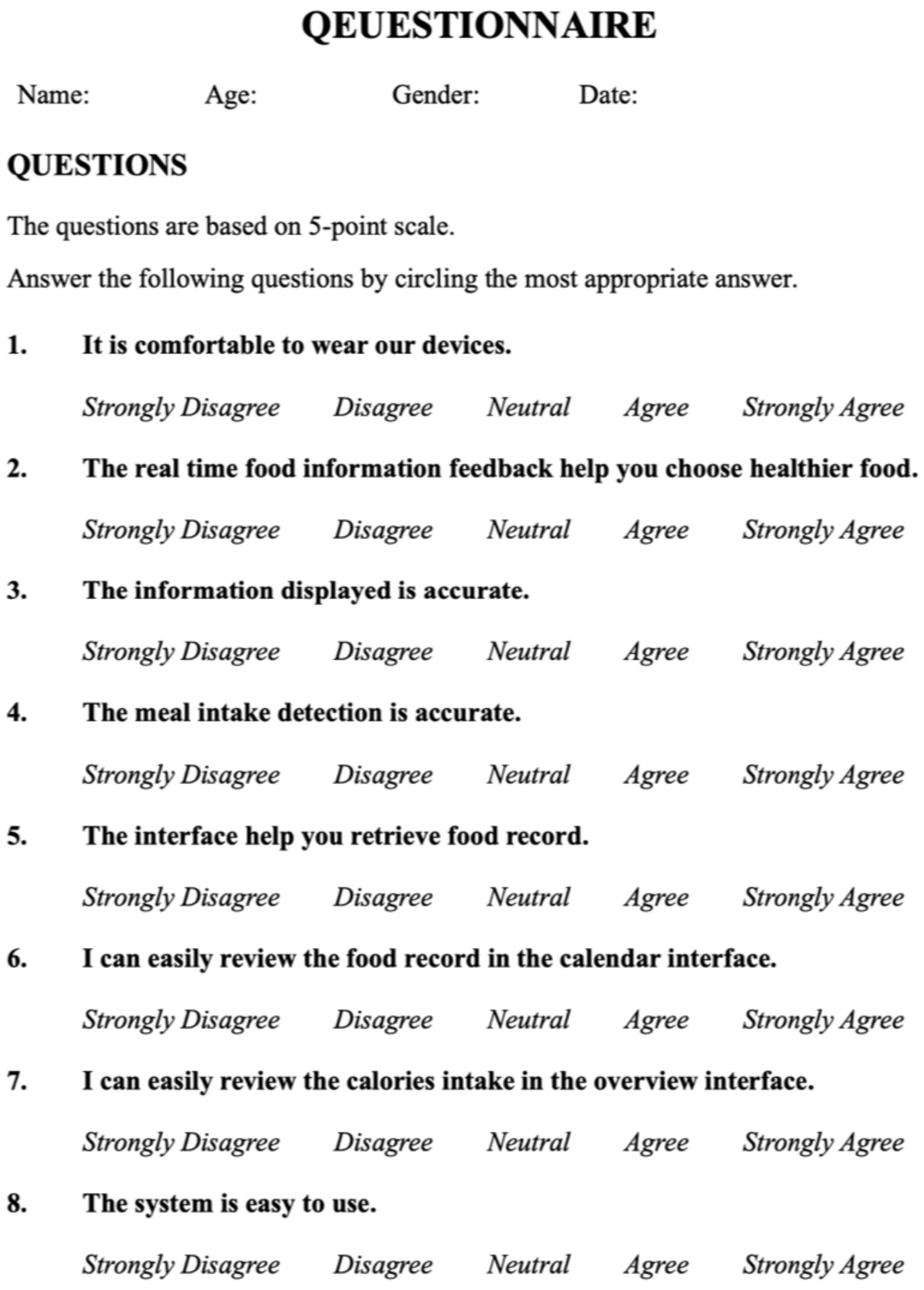
7.1 Participants

We invited 10 participants (4 females and 6 males), ranging from 22 to 25 year of age. All participants have basic computer skills. Six of them had experience with head-mounted display.

7.2 Method

All participants are given a brief introduction of our system. Before each study, we introduced the basic operations of MOVERIO BT-300 to the participants. After the participants became familiar with the device, we asked them to perform our task. Also, participants were asked to wear fossil smart watch and the throat microphone for meal intake detection.

Our take is about asking our participants to use our system and choose one food to eat from several food choices. During the experiment, participant can talk with others and do some other human activities. After that, the participants will be asked to fill in a questionnaire as shown in figure 7.1. The questionnaire has following questions and these questions use the 5-point Likert scale.



QEUESTIONNAIRE

Name: Age: Gender: Date:

QUESTIONS

The questions are based on 5-point scale.

Answer the following questions by circling the most appropriate answer.

1. **It is comfortable to wear our devices.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

2. **The real time food information feedback help you choose healthier food.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

3. **The information displayed is accurate.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

4. **The meal intake detection is accurate.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

5. **The interface help you retrieve food record.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

6. **I can easily review the food record in the calendar interface.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

7. **I can easily review the calories intake in the overview interface.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

8. **The system is easy to use.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

Fig. 7.1 Questionnaire

7.3 Result

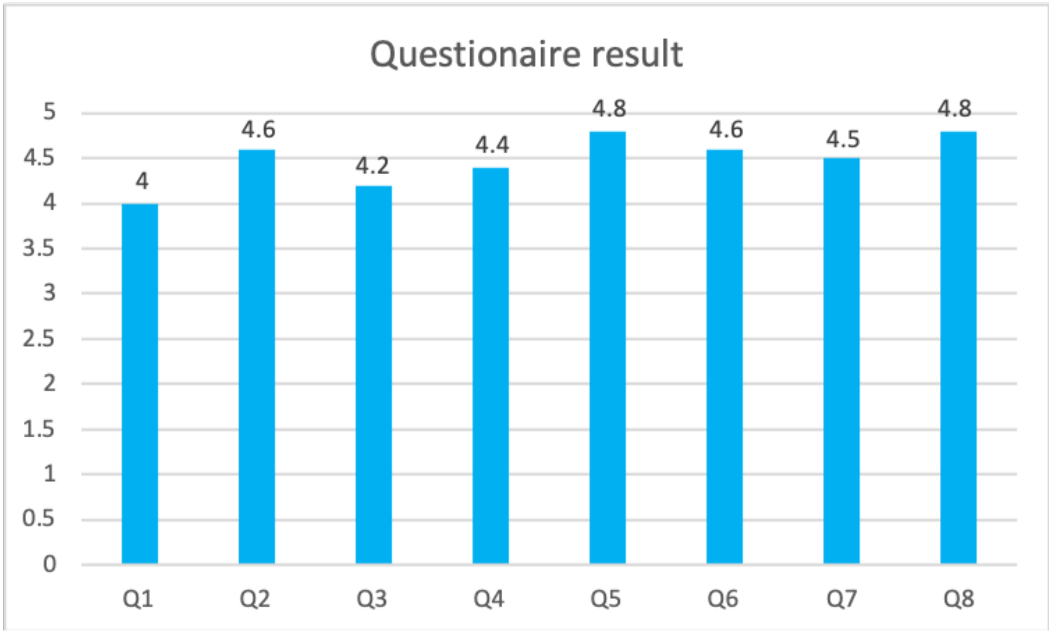


Fig. 7.2 Questionnaire result

Question 1 is used to assess whether our wearable devices will have a greater impact on the daily behavior of users. Question 1 has an average score of 4. The results prove that our wearable devices have little effect on the daily behavior of users, and are convenient to wear. Question 2 is used to determine whether our real-time feedback information can help users better choose a healthy diet. The average score of question 2 is 4.6. The results show that real-time food information feedback can help users choose a healthier diet when they choose food.

Questions 3, 4, and 8 are used to test the ease of use and usability of the system. The average scores for questions 3, 4 and 8 are 4.2, 4.4 and 4.8. The results show that our system's food recognition, real-time information feedback, and meal-intake detection realized by wrist motion tracking and chew event detection are feasible. As the system is automated, users give a higher score for ease of use.

Questions 5, 6, and 7 mainly evaluate whether the food recording interface can help the user to review the previous eating behavior. For this part, the user's main concern is whether it is easy to interact with the system interface. Questions 5, 6 and 7 have average scores of 4.8, 4.6 and 4.5. The results prove that the interactive interface of the system is easy to operate. Users can easily view the food records and calorie intake over a period of time through the interactive interface. One participant believes that when using a system that requires manual recording of eating behaviors, he often forgets to input manually after eating, and also forgets some food intake after a period of time. This problem can be solved by our automated recording system.

Generally speaking, all participants evaluate our system higher than traditional systems. This may indicate that our system design is reasonable and practical. It shows that our system can establish an automatic food tracking system for human meal-intake activities, so that users can intuitively obtain real-time food information feedback when choosing foods and retrieve food records after meal-intake.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In terms of design, our research compared current food recording system's advantages and disadvantages, then proposed the automatically food recording system utilizing some wearable devices (Head-mounted display, smart watch and throat microphone).

Technically, we designed a method that can provide a real time food information visual feedback to help user to choose a healthier diet. Other than that, we created two sensor fusion filters that reduce the drift and external vibrations. First is based on a complementary filter, and the second is based on a Kalman filter.

For the system users, we provide several simple interfaces for user to retrieve their food records. Users are required to wear a head-mounted display (Epson glass), a smart watch (Fossil smart watch) and a throat microphone.

Overall, we think we have successfully accomplished these following requirements.

1. Real time food recognition;

2. Meal intake detection;

3. Hands-free;

4. Automatically food recording system.

8.2 Future Work

Although we have proposed a prototype of automatically food recording system, there are still some limitations and future possibilities to improve its efficiency. In this system, we require the users to wear several wearable devices which will affect the user's daily life to a certain extent. Later, these devices can be replaced with a more advanced device such as Microsoft HoloLens 2 to facilitate users to use our system.

For the calorie calculation part, we cannot yet analyze the various nutrient components of food to estimate calories. A feasible method is to cooperate with some restaurants to obtain accurate calculation of our system by obtaining various nutrients calculated by the restaurants for the foods they make. For example, in cooperation with McDonald's, we can use AR glasses to identify the type of food ordered by users and then display the food information to the users based on the nutritional content of different hamburgers provided by McDonald's. The volume estimation part can be used to identify the size of the user's order of small fries and large fries or small coke and big coke.

References

[1] Keigo Kitamura, Toshihiko Yamasaki, and Kiyoharu Aizawa. Foodlog: Capture, analysis and retrieval of personal food images via web. In *Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*, CEA '09, page 23–30, New York, NY, USA, 2009. Association for Computing Machinery.

[2] Lydia Zepeda and David Deal. Think before you eat: photographic food diaries as intervention tools to change dietary decision making and attitudes. *International Journal of Consumer Strudies*, 32(6):692–698, 2008.

[3] Aiden Doherty, Alan Smeaton, and Cathal Gurrin. Lifelogging: Personal big data. *Found. Trends Inf. Retr.*, 8(1):1–125, June 2014.

[4] Centers for Disease Control and Prevention. Adult obesity facts. 2014. Available online. http://www.cdc.gov/obesity/data/adult.html (Accessed May 15, 2020).

[5] World Health Organization. Obesity and overweight. 2012. Available online. http://www.who.int/mediacentre/factsheets/fs311/en (Accessed May 15, 2020).

[6] Lara Oscar and Labrador Miguel. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, 2013.

[7] Temiloluwa Prioleau, Elliot Moore II, and Maysam Ghovanloo. Unobtrusive and wearable systems for automatic dietary monitoring. *IEEE Transactions on Biomedical Engineering*, 64(9):2075–2089, 2017.

[8] Giovanni Schiboni and Oliver Amft. Automatic dietary monitoring using wearable accessories. In *Seamless healthcare monitoring*, pages 369–412. Springer, 2018.

[9] Ziad Ahmad, Marc Bosch, Nitin Khanna, and Deborah Kerr. A mobile food record for integrated dietary assessment. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, MADiMa '16, page 53–62, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Joachim Dehais, Marios Anthimopoulos, and Stavroula Mougiakakou. Food image segmentation for dietary assessment. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, MADiMa '16, page 23–28, New York, NY, USA, 2016. Association for Computing Machinery.

[11] Abdelkareem Bedri, Apoorva Verlekar, Edison Thomaz, Valerie Avva, and Thad Starner. Detecting mastication: A wearable approach. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 247–250, 2015.

[12] Abdelkareem Bedri, Apoorva Verlekar, Edison Thomaz, Valerie Avva, and Thad Starner. A wearable system for detecting eating activities with proximity sensors in the outer ear. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 91–92, 2015.

[13] Hui-Shyong Yeo, Juyoung Lee, Hyung-il Kim, Aakar Gupta, Andrea Bianchi, Daniel Vogel, Hideki Koike, Woontack Woo, and Aaron Quigley. Wrist: Watch-ring interaction and sensing technique for wrist gestures and macro-micro pointing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[14] Mark Mirtchouk, Christopher Merck, and Samantha Kleinberg. Automated estimation of food type and amount consumed from body-worn audio and motion sensors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, page 451–462, New York, NY, USA, 2016. Association for Computing Machinery.

[15] Yujie Dong, Adam Hoover, Jenna Scisco, and Eric Muth. A new method for measuring meal intake in humans via automated wrist motion tracking. pages 37(3):205–215. Appl Psychophysiol Biofeedback, 2012.

[16] Oliver Amft, Mathias Stäger, Paul Lukowicz, and Gerhard Tröster. Analysis of chewing sounds for dietary monitoring. In *Proceedings of the 7th International Conference on Ubiquitous Computing*, UbiComp'05, page 56–72, Berlin, Heidelberg, 2005. Springer-Verlag.

[17] Sebastian Päßler and Wolf-Joachim Fischer. Evaluation of algorithms for chew event detection. In *Proceedings of the 7th International Conference on Body Area Networks*, BodyNets '12, page 20–26, Brussels, BEL, 2012. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[18] Shengjie Bi, Tao Wang, Nicole Tobias, Josephine Nordrum, Shang Wang, George Halvorsen, Sougata Sen, Ronald Peterson, Kofi Odame, Kelly Caine, et al. Auracle: Detecting eating episodes with an ear-mounted sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–27, 2018.

[19] Temiloluwa Olubanjo and Maysam Ghovanloo. Real-time swallowing detection based on tracheal acoustics. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4384–4388. IEEE, 2014.

[20] Tauhidur Rahman, Alexander Travis Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. Bodybeat: a mobile system for sensing non-speech body sounds. In *MobiSys*, volume 14, pages 2594368–2594386. Citeseer, 2014.

[21] Koji Yatani and Khai Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 341–350, 2012.

[22] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Yin Bi, Mingsong Lv, Chen Song, Wenyao Xu, Nan Guan, and Wang Yi. Autodietary: A wearable acoustic sensor system for food intake recognition in daily life. *IEEE Sensors Journal*, 16(3):806–816, 2015.

[24] Mark Mirtchouk, Drew Lustig, Alexandra Smith, Ivan Ching, Min Zheng, and Samantha Kleinberg. Recognizing eating from body-worn sensors: Combining free-living and laboratory data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–20, 2017.

[25] Jindong Liu, Edward Johns, Louis Atallah, Claire Pettitt, Benny Lo, Gary Frost, and Guang-Zhong Yang. An intelligent food-intake monitoring system using wearable sensors. In *2012 ninth international conference on wearable and implantable body sensor networks*, pages 154–160. IEEE, 2012.

[26] Sougata Sen, Vigneshwaran Subbaraju, Archan Misra, Rajesh Balan, and Youngki Lee. Annapurna: building a real-world smartwatch-based automated food journal. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–6. IEEE, 2018.

[27] Edison Thomaz, Aman Parnami, Irfan Essa, and Gregory Abowd. Feasibility of identifying eating moments from first-person images leveraging human computation. In *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference*, pages 26–33, 2013.

[28] Marc Bosch, Fengqing Zhu, Nitin Khanna, Carol Boushey, and Edward Delp. Combining global and local features for food identification in dietary assessment. In *2011 18th IEEE International Conference on Image Processing*, pages 1789–1792. IEEE, 2011.

[29] Nitin Khanna, Carol Boushey, Deborah Kerr, Martin Okos, David Ebert, and Edward Delp. An overview of the technology assisted dietary assessment project at purdue university. In *2010 IEEE International Symposium on Multimedia*, pages 290–295. IEEE, 2010.

[30] Fengqing Zhu, Marc Bosch, Insoo Woo, SungYe Kim, Carol Boushey, David Ebert, and Edward Delp. The use of mobile devices in aiding dietary assessment and evaluation. *IEEE journal of selected topics in signal processing*, 4(4):756–766, 2010.

[31] Bethany Daugherty, TusaRebecca Schap, Reynolette Ettienne-Gittens, Fengqing Zhu, Marc Bosch, Edward Delp, David Ebert, Deborah Kerr, and Carol Boushey. Novel technologies for assessing dietary intake: evaluating the usability of a mobile telephone food record among adults and adolescents. *Journal of medical Internet research*, 14(2):e58, 2012.

[32] Chang Xu, Ye He, Nitin Khannan, Albert Parra, Carol Boushey, and Edward Delp. Image-based food volume estimation. In *Proceedings of the 5th international workshop on Multimedia for cooking & eating activities*, pages 75–80, 2013.

[33] Keiji Yanai and Yoshiyuki Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[34] Yoshiyuki Kawano and Keiji Yanai. Food image recognition with deep convolutional features. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 589–593, 2014.