

Master's Thesis  
Academic Year 2017

Make a Face : User-defined Face Gesture  
Interactions Using Webcams

Keio University  
Graduate School of Media Design

Yenchin Lai

A Master's Thesis  
submitted to Keio University Graduate School of Media Design  
in partial fulfillment of the requirements for the degree of  
MASTER of Media Design

Yenchin Lai

Thesis Committee:

Associate Professor Kai Kunze	(Supervisor)
Professor Keiko Okawa	(Co-supervisor)
Associate Professor Nanako Ishido	(Member)

Abstract of Master's Thesis of Academic Year 2017

## Make a Face : User-defined Face Gesture Interactions Using Webcams

Category: Science / Engineering

### Summary

With the advance of science and technology, facial recognition technology has become wildly used in our everyday computer interactions. While we can find this technique in entertaining and socializing use cases in phone applications or social network, it is, in fact, helpful and valuable to be implemented on the purpose of recognizing facial explicit input in our daily interactions.

In this thesis, we present a face gesture interaction based on the user-defined gesture study we conducted and implementation of the system on computers using webcams as input. Three computer applications are discussed in the study. They are e-mail checking, media playback, and a document reader. The study indicates that gestures made with the eye region are the most used ones because they are easier to perform and more socially acceptable. Through the detection accuracy test conducted on the implemented system, it shows that the accuracy achieves over 80% in both media playback and reader application.

### Keywords:

Human-computer Interaction, Face Gesture, Facial Recognition, Hands-free Input, User-defined Gesture, Elicitation Study

Keio University Graduate School of Media Design

Yenchin Lai

# Acknowledgements

I would first like to thank my supervisor, Associate Professor Kai Kunze, who has guided me to this research field and always given me feedback with a playful and unique insight. I would sincerely like to thank him for his support for this thesis.

I would also like to thank Professor Keiko Okawa for giving advice of thesis structuring and writing. And, I would like to thank Associate Professor Nanako Ishido for her guidance about this thesis.

Next, I would like to thank Project Assistant Professor Tilman Dingler at Osaka Prefecture University and Assistant Professor Liwei Chan at National Chiao Tung University of Taiwan for giving me advice on this research topic and the studies.

I would also like to thank Benjamin Tag, Yun Suen Pai, Katsutoshi Masai, and Junichi Shimizu for discussion on this thesis when I ran into a trouble spot. Also, I would like to thank all Geist members and all participants in my studies for their helping with this research.

Finally, I would like to thank my supportive family and friends. Without their beliefs in me, I would not have the courage to confront all difficulties during the time I was writing this thesis.

# Table of Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contribution . . . . .	3
<b>2 Related Works</b>	<b>4</b>
2.1 Human Face . . . . .	4
2.1.1 Facial Action Coding System . . . . .	4
2.1.2 Facial Recognition . . . . .	5
2.2 Gesture Interaction . . . . .	6
2.3 User-defined Gesture Study . . . . .	8
Notes . . . . .	9
<b>3 Make a Face: System</b>	<b>10</b>
3.1 Approach . . . . .	10
3.1.1 User-defined Gesture . . . . .	10
3.1.2 Webcam Based Facial Recognition . . . . .	11
3.2 System Overview . . . . .	11
3.3 Application Scenarios . . . . .	13
3.3.1 Hands-free Use Case . . . . .	13
3.3.2 Subtle Interaction . . . . .	14
3.4 Experimental Design . . . . .	14
3.4.1 Pilot Study: Focus Group . . . . .	14
3.4.2 Defining Gesture . . . . .	17
3.4.3 Defined Gesture Set . . . . .	21
3.5 Face Gesture Analysis . . . . .	23
3.5.1 Taxonomy of Face Gestures . . . . .	23
3.5.2 Emotions in Face Gesture Mapping . . . . .	27
3.5.3 Mapping Assessment . . . . .	27
Notes . . . . .	29

## TABLE OF CONTENTS

---

<b>4</b>	<b>Implementation</b>	<b>30</b>
4.1	Prototypes . . . . .	30
4.1.1	First Prototype . . . . .	30
4.1.2	Defined Gesture Recognition . . . . .	32
4.1.3	Communication . . . . .	32
4.2	Algorithm . . . . .	34
4.2.1	Face Detection and Processing . . . . .	34
4.2.2	Face Gesture Recognition . . . . .	34
	Notes . . . . .	36
<b>5</b>	<b>Evaluation</b>	<b>37</b>
5.1	Detection Accuracy . . . . .	37
5.2	Usability Study . . . . .	38
5.2.1	Study Design . . . . .	39
5.2.2	Results . . . . .	40
	Notes . . . . .	41
<b>6</b>	<b>Discussion</b>	<b>42</b>
6.1	Limitation of the Implementation . . . . .	42
6.1.1	Detection of Both Eye Regions . . . . .	42
6.1.2	Natural Face and Continuous Detection . . . . .	43
6.2	Gestures and Daily Usages . . . . .	43
6.2.1	Limitation of the Gestures . . . . .	44
6.2.2	Alternative Use Cases . . . . .	45
<b>7</b>	<b>Conclusion</b>	<b>46</b>
	<b>References</b>	<b>48</b>

# List of Figures

2.1	Examples of Action Units in Facial Action Coding System [9] . .	5
2.2	EarFieldSensing proposed by Matthies et al. [25] . . . . .	7
2.3	FaceSwitch [31] . . . . .	7
3.1	Controlled commands and the mapped gestures in "Make a Face" system. . . . .	12
3.2	Ideas developed by participants. . . . .	16
3.3	GUIs of 3 applications presented in the experiment. . . . .	18
3.4	Example of questionnaire. . . . .	19
3.5	Participants made their own gestures for each command in the experiment. . . . .	20
3.6	User-defined face gesture set of 3 applications. . . . .	22
3.7	Facial Parts . . . . .	24
3.8	Taxonomy breakdown . . . . .	24
3.9	The distribution of the 84 named emotion coordinates: the numbers of the coordinates are marked with colors. . . . .	28
3.10	Emotion distribution of defined gestures. . . . .	28
4.1	Face region processing in the first prototype. . . . .	31
4.2	Four gestures recognized in the first prototype. . . . .	31
4.3	Image processing. . . . .	33
4.4	68 face landmarks. . . . .	35
4.5	Face Alignment. . . . .	35
5.1	Wizard of Oz experiment setup. . . . .	39

# List of Tables

3.1	Categories and examples of ideas including face interaction application scenarios. . . . .	15
3.2	Selected Applications and Commands. . . . .	17
3.3	User-defined face gesture set . . . . .	21
3.4	Taxonomy of Face Gestures . . . . .	25
3.5	Grouped Gestures . . . . .	26
3.6	The mean of self-assessment scores on each gesture mapping. . .	29
5.1	Confusion Matrix . . . . .	37
5.2	Detection Accuracy . . . . .	38
5.3	Applications and Tasks . . . . .	40
5.4	Usability Score . . . . .	41



# Chapter 1

## Introduction

In our daily life, we take a lot of information in from human face. We can guess a person's gender, age, and race from the face, and it also plays an important role in human communication. For instance, in a face-to-face conversation, we rely on interpreting facial expressions on each others' faces to understand more about what people are trying to say. When we see someone says "Is this job too difficult for you?" with a concerning face, we may consider he is really worried if the job can be too difficult for another person; however, if someone says the exactly same sentence with an angry or disappointed face, we may guess that he is angry with another person because he supposes the job should not be too difficult. If we can use our faces properly, we are able to send messages more efficiently. When a person is greeting with a smiling face, rather than a poker face, he is sending his greeting and kindness with a higher efficiency. Showing a smile is telling other people that he is a sincere person. Sometimes, also a simple wink can express messages more than words [4]. Imaging that your partner wink at you before an important presentation, without any words, you know he is saying "Don't worry, we can make it!" In some situations, it is difficult for us to have conversations in person; some people may need to fly overseas for work but still want to keep in contact with their families. Although we can solve this problem by talking on phones, sending e-mails or short messages, there is still an issue that they are not enough for us to deliver complete information to people we cannot see. When we receive messages only from words, we lose several pieces of information such as emotions revealed from facial expressions. The face is essential for human beings to understand each other.

Nowadays, our life is closely connected with computers and machines. Scientists are trying to make computers understand us or help us to know more about each other through our faces, just like what we can do. At the time when facial recognition technology was invented, scientists started applying it to possible applications. We have systems identifying different people for security purpose,

getting marketing feedback from customers' faces, detecting students' concentration levels at schools, and even tracking patients' expressions in a healthcare use case [8]. In the past decades, facial recognition technology is popularized in everyday usages. Facebook is one of the well-known social network services providing this technology. It can automatically recognize our friends' faces from the pictures we upload and suggest that if we want to tag their names on the pictures to share with more people. Furthermore, many photo/filming applications popular with the youth apply face detection technology, and provide entertaining effects such as masking faces with animations, morphing different faces, or deforming some facial parts. With the advance of facial recognition technology, our social life becomes more and more colorful.

Since computers are being able to capture massive information from human face, I wonder if facial recognition technology can be more functional and practical in an everyday usage. What if we can interact with computers by using the face consciously but not implicitly being detected by them? Scientists have been exploring explicit face inputs for different purposes. Some are for people with disabilities [7, 14, 16]. A famous English physicist, Professor Stephen Hawking diagnosed with Amyotrophic Lateral Sclerosis (ALS), relies on facial recognition technology to speak. In 2012, Intel Corporation, which supported Professor Stephen Hawking with technology allowing him to speak, provided infrared sensor to detect small movements on his cheek as an input modality and enabled him to write. Later, Intel kept trying different methods such as facial recognition cameras and eye gaze sensors to improve the interaction for people with motor neuron diseases and other disabilities. On the other side, scientists also found that face input is worthy to be implemented with systems for people without disabilities [22]. Looking at some prior works and the studies I conducted in this thesis, there are many situations in our everyday life that we want to do another task with both hands occupied: going to the next page while practicing the piano, checking the next step on recipe while cooking, or picking up phones while doing dishes. Also, in some situations, not using the hands can just be a better choice: checking an important e-mail during a meeting, changing TV channel while sitting far from any controllers, or going to the next slide while giving a presentation.

Face gesture is a potential input modality which can be useful in everyday situations. However, there are studies rarely discussing about how to make it practical. Although there are researches exploring face gestures as inputs, I would like to focus on practical face gestures and also a handy face gesture recognition

method.

## 1.1 Contribution

In this thesis, I present the exploration of face gesture mapping and a face interaction system implemented with the user-defined face gesture set. By conducting two focus groups on the topic of face gesture mapping for human-computer interaction, the potential usages of face gestures are indicated and 12 commands in computer applications are selected to be used for designing the experiment of defining face gesture mapping. Through the experiment, I find the evidence of users tending to use the eye region to perform face gestures as an explicit input modality. Following the result of this experiment, I implement a face gesture recognition system using a webcam to detect eye and eyebrow movements simultaneously. The contributions are summarized as below:

- Defining 12 face gestures mapping toward three everyday computer applications, which are e-mail checking, media playback, and a document reader, through a user-defined gesture study.
- The exploration of how users would like to use the face as an explicit input modality for computer interaction and presenting the finding that users tend to use the eye region when they perform face gestures.
- Implementation of face gesture recognition system using a webcam with the user-defined gestures in the study.
- Presenting a convenient and socially acceptable hands-free input modality for daily computer applications including the study result.

# Chapter 2

## Related Works

Scientists have been exploring face and gesture interactions in the past decades. However, understanding face gestures from an user's aspect is rarely seen. In this chapter, I will look into prior works related to the topic of human face, gesture interaction and user-defined gesture study to identify the subject area and contribution of this thesis.













### 2.1 Human Face

The face is essential for human beings. It is one of the most informative stimuli we can perceive from each other. From the face, we can receive informations about a person such as feelings, emotions, cognition load, and also health condition. Because of the distinctiveness of human face, scientists develop systems to capture informations from it. Nowadays, we have facial recognition system at airports for security, cameras with smile detection to take pictures automatically, and numerous facial recognition applications to make our lives better and more convenient.

#### 2.1.1 Facial Action Coding System

FACS (Facial Action Coding System) is widely used in various fields for facial behavior studies. It is a system to taxonomize visually discernible facial movements established by Paul Ekman. Every facial movement is called an Action Unit (AU). With FACS, scientists have a standard to describe facial movements; moreover, it can also help psychologists to understand facial expressions and emotions. In the book, "What the Face Reveals" edited by Paul Ekman and Erika Rosenberg [10], the authors introduced studies of spontaneous expression using FACS, such as showing the evidence for distinct affective displays of embarrassment, amusement, and shame [18]. Besides psychological studies, computer sci-

entists also apply FACS to encode facial behavior for building facial recognition systems [9, 24].

Upper Face Action Units					
AU1	AU2	AU4	AU5	AU6	AU7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU41	*AU42	*AU43	AU44	AU45	AU46
					
Lip Droop	Slit	Eyes Closed	Squint	Blink	Wink



















Lower Face Action Units					
AU9	AU10	AU11	AU12	AU13	AU14
					
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU15	AU16	AU17	AU18	AU20	AU22
					
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Pucker	Lip Stretcher	Lip Funneler
AU23	AU24	*AU25	*AU26	*AU27	AU28
					
Lip Tightener	Lip Pressor	Lips Parts	Jaw Drop	Mouth Stretch	Lip Suck

Figure 2.1: Examples of Action Units in Facial Action Coding System [9]

### 2.1.2 Facial Recognition

Technology of facial recognition is growing rapidly in recent year. For instance, iPhone X, released by Apple Inc.<sup>1</sup>, is implemented with depth sensing camera and said to have better facial recognition quality which is able to detect smaller facial features and movements in comparison with traditional recognition technology. Facial recognition can be achieved using different techniques. A commonly seen approach can be intensity image holistic analysis [15]. It can be applied with camera based input modality. Also, there are other signal-sensing based methods such as using EOG, EMG, or photo reflective sensors to detect facial muscle movements; however, these kinds of detections can only used to recognize facial behaviors but not features, in other words, they are not able to identify different people [24, 25].

## 2.2 Gesture Interaction

The use of gestures in HCI (Human-computer interaction) appeared in the 1970s. Wired glove was one of the early devices developed for hand gesture interaction. The first wired glove, Sayre Glove [34], was developed by Richard Sayre, Dan Sandin and Tom DeFanti at EVL (Electronic Visualization Laboratory) in 1977. It provided a multidimensional input method and was mostly used to control sliders. Besides wearable devices for hand gesture interaction, in 1980, Richard A. Bolt [6] in MIT (Massachusetts Institute of Technology) Architecture Machine Group presented a voice and gesture interaction system, "Put-That-There", with environmental recognition. It provided a natural user input modality with which users can intuitively point out a specific position by hand gestures.

In the past decades, gesture interaction has been used in various kinds of devices and systems. In our everyday life, for instance, touchpad for finger gesture input has become a basic equipment of laptops. Also, touch screens of tablet devices and smart phones support a similar gesture interaction replacing both traditional mouse and keyboard input for computers. Although touch interaction is widely used, touchless user interface [26], following the idea of wired glove and "Put-That-There" system, keeps being discussed by computer scientists in order to develop different approaches. Wii<sup>2</sup>, released by Nintendo, and Microsofts Kinect<sup>3</sup> are commercial product examples of touchless user interface which are mainly used with gaming systems. The interaction is based on motion capture of body gestures.

Gesture interaction with novel modality, such as hand-to-skin interaction, hand-to-face interaction, face gestures, and etc., is being proposed by scientists. The idea of hand-to-skin interaction [28,37] is using human skin as a touch input surface. On-skin gestures augment the interaction with haptic feedback which is given by the touching action on the skin surface. Furthermore, as a soft interface, the skin varies touch gestures with new modalities such as squeezing and pulling. Hand-to-face gesture can also be identified as on-skin interaction. Itchy Nose, presented by Lee et al. [21], recognizes hand gestures on the nose using EOG (Electrooculography) sensors equipped in smart glasses. By detecting electrical activities around the nose from EOG sensors, it allows users to perform flicking, pushing, and rubbing gestures on the nose to interact with computer systems.

Face gesture input, which is the modality used in this research, has been discussed in several studies. Matthies et al. [25] proposed a wearable face gesture input method using electric field sensing technology inside the ear canal as shown

in Figure 2.2. With the wearable sensing device, it allows users to use 25 face and head gestures. Another face gesture input can be camera-based input. It is also the input method I will talk about in the later chapters. FaceSwitch [31] as shown in Figure 2.3 is a webcam-based system supporting the users to interact with a computer GUI (Graphical User Interface) by using four face gestures, which are smiling, raising eyebrows, opening mouth, and snarling. In the system, users can select which gestures should be monitored and also map each gesture to different keyboard or mouse events.



Figure 2.2: EarFieldSensing proposed by Matthies et al. [25]

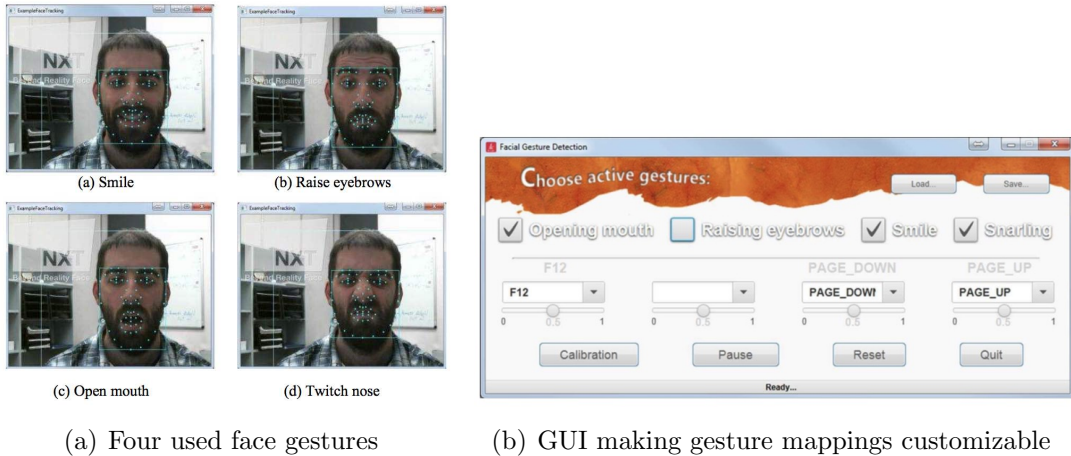


Figure 2.3: FaceSwitch [31]

Although studies have been done with creating new input method with face gestures, we have not had a well-understanding of how users tend to use face gestures to interact with a system. Once we have a better understanding, it will be helpful for scientists and researchers to create new technologies or systems with better interactions.

## 2.3 User-defined Gesture Study

Scientists in HCI field devote their efforts to understand what is the best way for users to interact with computer systems. Wobbrock et al. [39] presented a user-centered approach to develop gestures in 2009. The method follows the authors' previous guessability study [38], which is able to maximize and evaluate how guessable the symbolic inputs are, and creates gestures relying on eliciting gestures from users by asking users to perform a gesture after the effect of it presented.

The method has been widely used by other researchers to understand different kinds of gestures in diverse interactions. There are various interactive systems studied with hand gestures. Weigle et al. [37] discussed how people use hand gesture on the other upper limb when the skin was used as an input surface. In the study, users did not only use finger movements which were similar to what we usually do with touch pads, but also hand motions such as grabbing and scratching, which were seldom seen in existed systems. Free-hand gestures have also been explored with the similar approach, such as how free-hand gestures can be generated to control a television or music playback [13, 35]. These studies allowed us to see that even the simple idea of hand gestures could be varied from different controls and interactive designs when researchers learned about them from the users. Moreover, when the users were not limited to use hand gestures only but told to control systems or devices with "gestures", the users still tended to use the hands [19, 27]. Besides hand gestures, some researchers use this method to understand foot gestures or even motion inputs with mobile devices. Researchers from University of Haifa studied about how people use foot gestures to interact with both GUIs and avatars on big screens [11]. Since we use our feet to interact with many devices in everyday situations, such as driving a car, playing the piano, playing dancing games, feet are looked as one of the possible modalities for human-computer interaction in daily life. The work from Ruiz et al. [32] is relatively different from the others. They explored motion gestures with a smart phone in the user's hand. Most of the time, we think about gestures with



either only touching the surface or without touching and moving the device itself. However, they proved a different way of thinking for the users to control a mobile device by giving the motion with it.

## Notes

- 1 iPhone X is a smartphone released by Apple Inc. in November of 2017.  
<https://www.apple.com/iphone-x/>
- 2 Wii is a home video game console released by Nintendo in November of 2006. Its remote controls allow the users use their body motion in games.  
<https://www.nintendo.co.jp/wii/>
- 3 Kinect is a line of motion sensing input devices developed by Microsoft for Xbox 360, Xbox One, and Microsoft Windows PCs.  
<https://www.xbox.com/en-US/xbox-one/accessories/kinect>

## Chapter 3

# Make a Face: System

In this chapter, I introduce how the system was built through the experiment we conducted and its result. We first conducted focus groups to discuss how the face could be used in human-computer interaction, and second, followed user-defined gesture approach to explore face gestures and the mappings. Based on the understanding of face gestures we gained from the experiment, we then implemented the system using a webcam as the input device for gesture recognition.

### 3.1 Approach

We used mainly two approaches to build the system. For the contents, we followed a user-defined gesture approach to understand and define what face gestures are best matched to the commands in certain computer applications. For the recognition technique, we decided to implement it using a webcam because of its low-cost and ability to detect face gestures.

#### 3.1.1 User-defined Gesture

User-defined Gesture is an approach through which researchers are able to understand how people would like to use gestures haven't been defined. Although many studies have been done with this approach, there are still differences for exploring varied types of gestures. In our study, we followed several main features of the approach and also added other factors in order to understand face gestures.

Similar to the pioneer work [39], we invited users to define their own face gesture sets by presenting them the effects of 12 computer application commands. After the user made a face gesture and decided that this was the one he/she would like to use to generate the command, the observer would ask the user several questions through both a questionnaire and a brief interview to understand the

gesture and mapping. In the pioneer work, Wobbrock et al. used two Likert scales for self-assessment to evaluate the mapping, which are "goodness of the match" and "ease of performing." We included these two factors in our study and also added the other ones. First, we built emotion factors with valence and arousal levels. Because the face is essential to showing human emotion, we wondered if face gesture could be related to emotion factors. Second, we created a Likert scale of "social acceptability." Since face gestures are visually discernible, we valued how acceptable the gestures might be to the users when they were aware that they might be seen by other people. In the end of each experiment trial, we had feedback about face gesture input modality and application scenarios from the users.

### 3.1.2 Webcam Based Facial Recognition

Although there are various methods to build facial recognition systems, webcam based system is one of the low-cost and simple methods. With OpenCV (Open Source Computer Vision Library) and Python machine learning libraries, we implemented a face gesture recognizer using webcam capture.

The process of different camera based facial recognitions are similar. In our system, we used dlib<sup>1</sup> face detector to identify the region of human face in webcam video streaming frame and processed the extracted face region with OpenCV. To make the face gesture easier be recognized in our gesture classifier, we transformed the face region to grayscale and did face alignment which made the eyes locations and face sizes nearly identical for every input frame. Finally, we sent the processed faces into gesture classifier, and it would show the recognized gesture names on the frame.

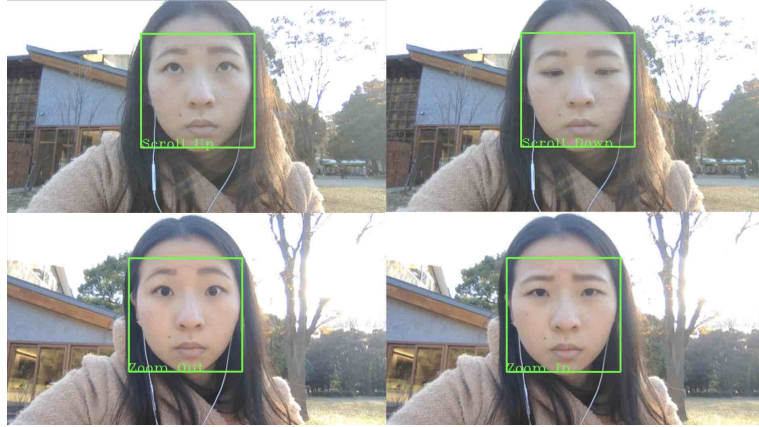
## 3.2 System Overview

We built "Make a Face" system based on the hypothesis that we can interact with computers by using face gestures but not implicitly being detected by them in everyday usages. We presented the system implementation using laptops with two computer applications, media playback and a document reader.

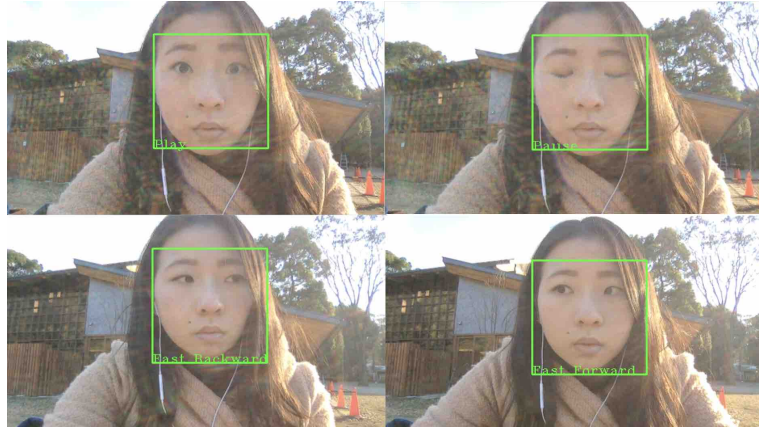
The system can be technically separated into two parts. The first part is a recognition program which detects the face through webcams and recognizes different face gestures. The second part is the GUIs which are built based on

several studies we conducted.

Our system allows the users to give commands with certain face gestures we defined through a study. As Figure 3.1 shows, it supports users to give the commands of *zoom in*, *zoom out*, *scroll down*, and *scroll up* in a document reader application with the four face gestures, *frowning*, *eyebrows raising*, *looking down*, and *looking up*. Also, in the other application of media playback, users are able to control *play*, *pause*, *fast forward*, and *fast backward* by using *eyebrows raising*, *blinking*, *looking at the right*, and *looking at the left* as the matched face gestures.



(a) Document reader application.



(b) Media playback application.

Figure 3.1: Controlled commands and the mapped gestures in "Make a Face" system.

### 3.3 Application Scenarios

There are two main features in the interaction supported by "Make a Face" system. Since face gestures are made by small facial movements, they can be performed within a short time and in a small region to make the interaction smoothly accomplished. Also, it provides an optional input method when the hands are occupied. Based on our studies, we proposed application scenarios of hands-free use case and subtle interaction.

#### 3.3.1 Hands-free Use Case

In various situations, we can find that our hands are not available to interact with computer devices in our everyday life. "Make a Face" provides an optional input modality when the hands are dirty, not able to reach the devices, or holding them, etc.. Although people may think that using hands is the most intuitive way to give commands, the hands are actually not a best choice for all interactions. For instance, while we are watching a entertaining movie on the computer, we may be having popcorn as well. Since we are in front of the computer, the webcam is able to capture our face gestures as inputs to control the video player. Furthermore, in another situation of using mobile devices such as smart phones and tablets, some functions are difficult to control with single-finger gestures. The finger gestures for "zoom in" and "zoom out" are usually using at least two fingers but they are difficult to perform while we are holding the devices with one hand. However, especially in Japan, people use mobile devices in crowded trains with one hand holding a strap. This makes it difficult to perform a proper gesture with the hand holding the device. With our system, users are able to give commands using the front camera with face gestures in order to replace multi-finger gestures.

Also, "Make a Face" is able to support users in a hand working situation such as craft making and hardware assembling. It is difficult to check manuals when the hands are occupied with components. Assembling bicycles, as an example, is a hand working situation that users may need to follow a manual but the hands are not able to turn the paper book to the next page or control a digital book in a tablet. With the system implemented in tablet devices, it can capture users' face gestures from front camera and help them to go to the next instruction.

### 3.3.2 Subtle Interaction

In our daily usage of computers, we use the hands to give commands with keyboards, the mouse, touch panels, or even buttons, etc.. And the actions are obvious since we have to put the hands on the keyboards or to touch the devices. However, there are some situations we may not be willing to do these actions obviously. As the working style changing, people are facing the situation that they have to take care of massive information from different people at the same time. They may have urgent messages when they are participating in another jobs. For instance, people may have to check few important e-mails during a meeting without making others uncomfortable. If they check the e-mails with their hands having actions which are not related to the meeting or making sounds because of typing and clicking on the computers, other people may consider that they are not paying attention and feel not respected, but the e-mails are very important. Using face gestures to open e-mails can let users check these information in a smooth and subtle way to reduce distraction in the situation.

Another potential application of subtle interaction is using face gestures to control the computer while users are giving a presentation. Since the facial movements can be performed quickly and naturally, they can make users give the commands smoothly. If we implement the system with a camera set up in a presentation room to capture face gestures, it supports users to change slides, play videos, etc. when they are standing away from the computers and makes the presentation progress smooth.

## 3.4 Experimental Design

Before we designed the experiment for defining gestures, we conducted two focus groups with the goal of identifying face gestures and exploring face interaction for computer applications. And, we designed an user-defined face gesture experiment based on the result of focus group. Finally, we were able to have a face gesture set for the applications by analyzing most used gestures in the experiment.

### 3.4.1 Pilot Study: Focus Group

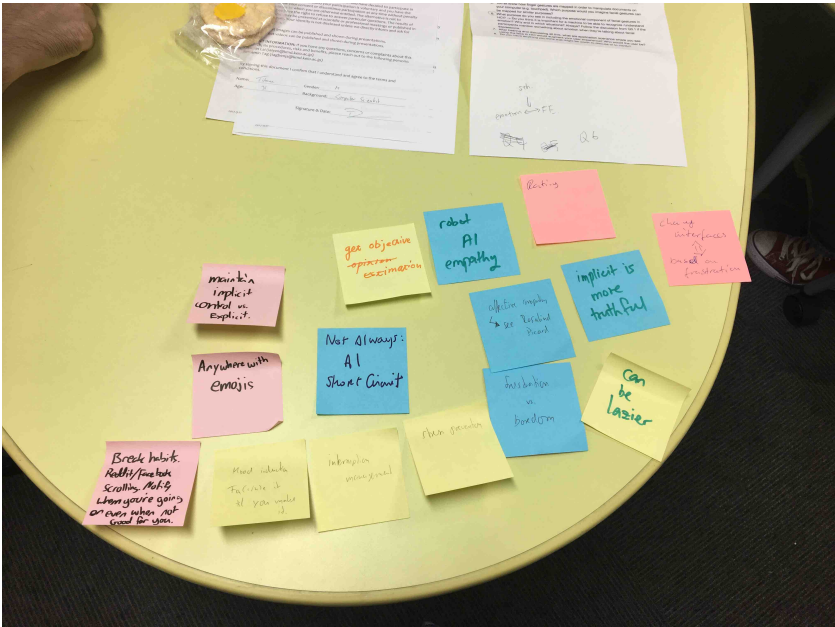
We recruited 11 participants (4 female) in KMD and invited them to the study in two groups of five and six respectively. The mean age of the participants was 27 years ( $SD = 4.95$ ). We earned the written consent from participants and

videotaped the discussion. We had seven main questions on the topic of face interaction. Starting with a warming up question of what the face could do in daily life, we then discussed questions including applications using the face as an input, devices to interact with, emotional components in face interaction, and the potential of generating computer commands with the face. Participants were given few minutes to write down ideas on sticky notes which then discussed by the entire group for every question. Figure 3.2 shows the ideas developed by participants.

Table 3.1: Categories and examples of ideas including face interaction application scenarios.

Category	Example Ideas
Triggers by Recognizing Human Emotion	Playing music when recognizing sadness, adding emotional context in messages automatically.
Detecting Human Condition/ Life Log	Reminding the users fatigue, attention, etc.
Auto Suggestion AI	Giving suggestions for dressing, movies, etc., depending on facial expressions.
Computer Controls	Lip movements to text, facial/head movements to control zoom level, screen brightness, etc.
Privacy and Security Use	Security checking, hiding private content on the screen with the face.
Game	Dynamic story telling depending on facial expressions, VR games with face interaction.
Smart House and Public Interaction	Interacting with the car, shower, digital signages etc., by using the face.
Hands-free Interaction	Refusing/picking up phone calls when babysitting, doing house chores, etc.
Playful Interaction	Turning on devices by kissing.

After the two discussion sessions, we grouped all ideas including face interaction application scenarios into nine categories in Table 3.1. Also, through the question, "Is there a difference between facial expressions and face gestures for you?" in focus group, we defined natural facial expressions as implicit inputs and face gestures as explicit inputs in face interaction. For instance, when the user



(a) Potential face interaction including emotional components.



(b) Possible devices to interact with by using face.

Figure 3.2: Ideas developed by participants.



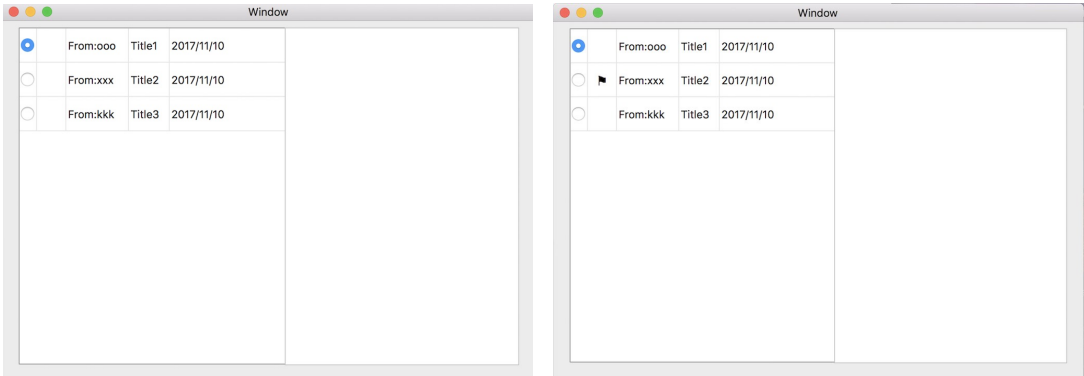
fakes a smile to give the command of playing a song, the smile is an explicit input, which is also a gesture; however, if the user smiles naturally because of happiness and the computer suggests a happy song for the user by recognizing the smile, it is considered as an implicit input. Since the category of computer controls had the most explicit input ideas, we decided to look into it and build application use cases for our gesture defining experiment.

### 3.4.2 Defining Gesture

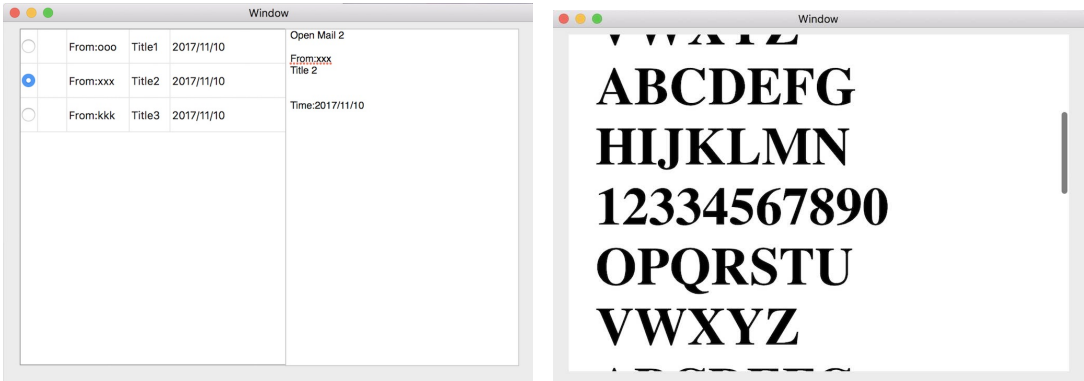
Following other user-defined gesture studies [11, 13, 19, 27, 32, 35, 37, 39] and the result from our pilot study, we designed an experiment to define face gesture sets for computer applications. We selected three application uses and four commands for each. There are 12 commands in total, shown in Table 3.2. To show participants the effects of commands, we built simple GUIs of each applications and recorded all effects into separated short videos. The GUIs used in experiment are shown in Figure 3.3. Besides, we edited questionnaires in which 12 commands were separated in different sessions by application uses and there were four questions following each command. An example of questionnaire is shown in Figure 3.4.

Table 3.2: Selected Applications and Commands.

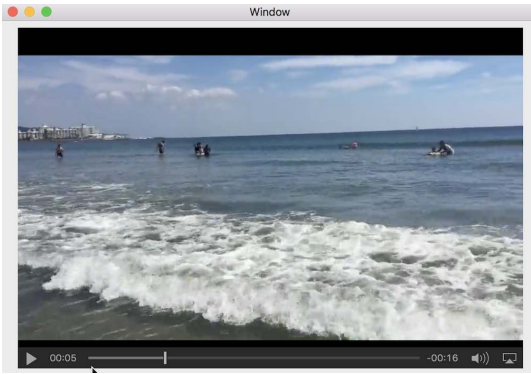
Application	Command
E-mail Checking	Next Mark Open Close
Media Playback	Play Pause Fast Forward Fast Backward
Reader	Zoom In Zoom Out Scroll Down Scroll Up



(a) "Next" command in E-mail Checking application for selecting the next e-mail. (b) "Mark" command in E-mail Checking application for giving a flag mark.



(c) "Open" command in E-mail Checking application for opening an e-mail. (d) GUI of Reader application.













(e) GUI of Media Playback application.

Figure 3.3: GUIs of 3 applications presented in the experiment.

Session 1 Reader

Zoom In

1. Emotion Valence/Arousal Level:

										N/A
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Valence (Positive-Negative)					Arousal(Excited-Calm)					

2. This face gesture is a good match for command Zoom In.

Entirely Agree	Mostly Agree	Somewhat Agree	Neither Agree nor Disagree	Somewhat Disagree	Mostly Disagree	Entirely Disagree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. This face gesture is easy to perform.

Entirely Agree	Mostly Agree	Somewhat Agree	Neither Agree nor Disagree	Somewhat Disagree	Mostly Disagree	Entirely Disagree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. I will use this face gesture in public.

Entirely Agree	Mostly Agree	Somewhat Agree	Neither Agree nor Disagree	Somewhat Disagree	Mostly Disagree	Entirely Disagree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 3.4: Example of questionnaire.

A pilot study was conducted on an excluded participant to confirm that there were no experiment design errors. The experiment was performed on 20 participants, aged between 22 and 30 years. Average age was 24.95 years ( $SD = 2.11$ ). Nine were female. After the participants signing up consent forms, the observer explained purpose and procedure of the experiment. In the beginning of each session, the four commands were informed to participants. And then the four effects of commands were randomly presented in short videos. After an effect was shown, participants were told to perform a gesture to match the effect and answer questions to explain the gesture and mapping. We earned the agreement from participants to videotape the experiment for recording gestures.

Before we ended the experiment, we interviewed participants with following questions:

- How do you think about using face gesture for computer input? How do you think about this input modality?
- In what situation do you think that you would like to use these applications?
- If there are any thing that you can change in these applications, what would you like to change?
- What other applications/scenarios would you like to use this kind of face gesture as input modality?
- Any other feedback for the study?

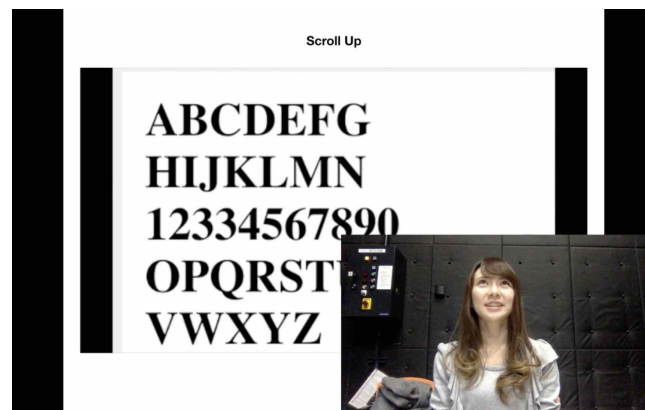


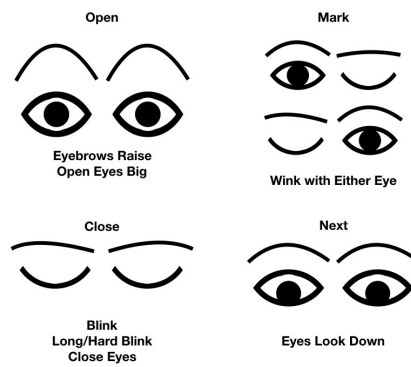
Figure 3.5: Participants made their own gestures for each command in the experiment.

### 3.4.3 Defined Gesture Set

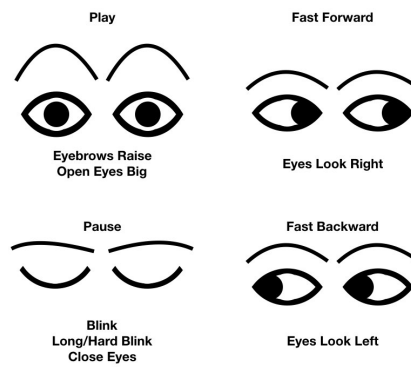
After conducting the experiment, we collected 240 gestures in total (12 commands \* 20 participants) and we got an user-defined face gesture set as shown in Figure 3.6. Also, Table 3.3 shows the gesture mapping with more information. First, if there is a " \* " mark after a gesture, it means that there is a mapping conflict occurred in this application due to that the same gesture is used to perform different commands. And the marked gesture is the second often used gesture to the command since the other command has a larger group using the most used one. Second, the table shows ratio of people performing the gesture. Also, following an agreement analysis method improved from user-defined gesture study by Wobbrock et al. [36], we computed the agreement rate (AR) of each mapping. The paper indicates that  $AR \leq .100$  represents a low agreement and  $.100 \leq AR \leq .300$  represents a medium agreement. Thus, "Pause" has the highest agreement rate, 0.23157, which shows a medium agreement and both "Next" and "Mark" have the lowest agreement rate, 0.05789, which shows a low agreement.

Table 3.3: User-defined face gesture set

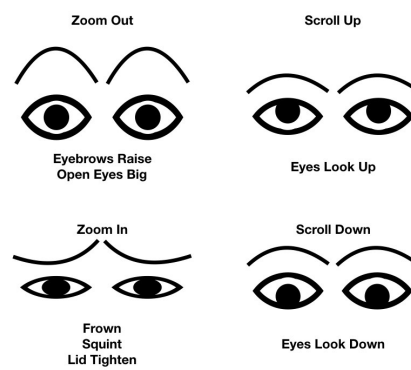
Application	Command	Gesture	Ratio	Agreement Rate
E-mail Checking	Next	Looking Down	25%	0.05789
	Mark	Winking(Either Eye)*	15%	0.05789
	Open	Eyebrows Raising	20%	0.06842
	Close	Blinking	40%	0.15789
Media Playback	Play	Eyebrows Raising*	20%	0.11578
	Pause	Blinking	45%	0.23157
	Fast Forward	Looking at the Right	25%	0.08947
	Fast Backward	Looking at the Left	25%	0.07894
Reader	Zoom In	Frowning	30%	0.09473
	Zoom Out	Eyebrows Raising	25%	0.06315
	Scroll Down	Looking Down	45%	0.19473
	Scroll Up	Looking Up	45%	0.19473



(a) Gesture mapping of "E-mail Checking"



(b) Gesture mapping of "Media Playback"



(c) Gesture mapping of "Reader"

Figure 3.6: User-defined face gesture set of 3 applications.

## 3.5 Face Gesture Analysis

In the experiment for defining face gestures, we collected gestures with further information for us to understand and interpret the gestures and mappings. We built a taxonomy of face gestures to present what kind of gestures users tended to use. As the questionnaire example shown in Figure 3.4, we also collected quantified self-assessment data of emotion, match goodness, ease of perform, and social acceptability of the gestures made by our participants for evaluating the mapping. Finally, we asked participants' feedback in the end of the experiment.

### 3.5.1 Taxonomy of Face Gestures

All face gestures performed in our experiment are classified by facial parts and movements as Table 3.4. The classification of facial parts is based on FACS. In our taxonomy of face gestures, upper facial part includes gestures performed with eyebrows, eye lids and eye gaze; middle facial part includes nose and cheeks movements; and, lips, mouth and chin movements are included in the category of lower facial part. Furthermore, in Figure 3.7, ratio of used facial parts is presented. It indicated that upper facial part, which contains movements around the eyes, is the most used category in our user-defined face gesture study. Taxonomy breakdown is also shown in Figure 3.8, which presents percentage of face gestures in each taxonomy category.

#### Descriptions of categories

In the category of upper facial part, single movement and repeated movement gestures are the ones performed with eyebrows and eye lids. Single movement means that a gesture contains only one movement such as a wink, a blink or raising eyebrows once. When a single movement gesture is repeated to represent another gesture, it is classified as a repeated movement. Eye gaze gestures are also separated into two kinds. Static gaze contains eye movements with looking at only one point or one direction, and dynamic gaze contains gestures performed with a continuous eye movements such as looking from the right to the left, looking along a clockwise circle, etc.. Also, there are gestures performed with both features. Some participants decided looking down and then blinking as a completed gesture.

As Figure 3.7 shows, only few gestures were performed with the middle facial part. In this category, gestures are only separated as nose movements and cheeks

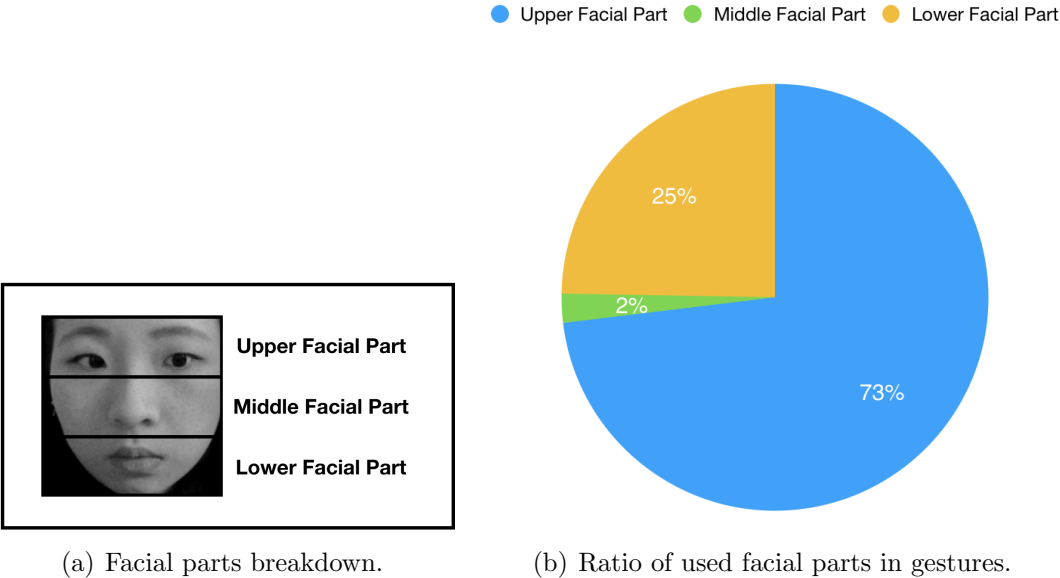


Figure 3.7: Facial Parts

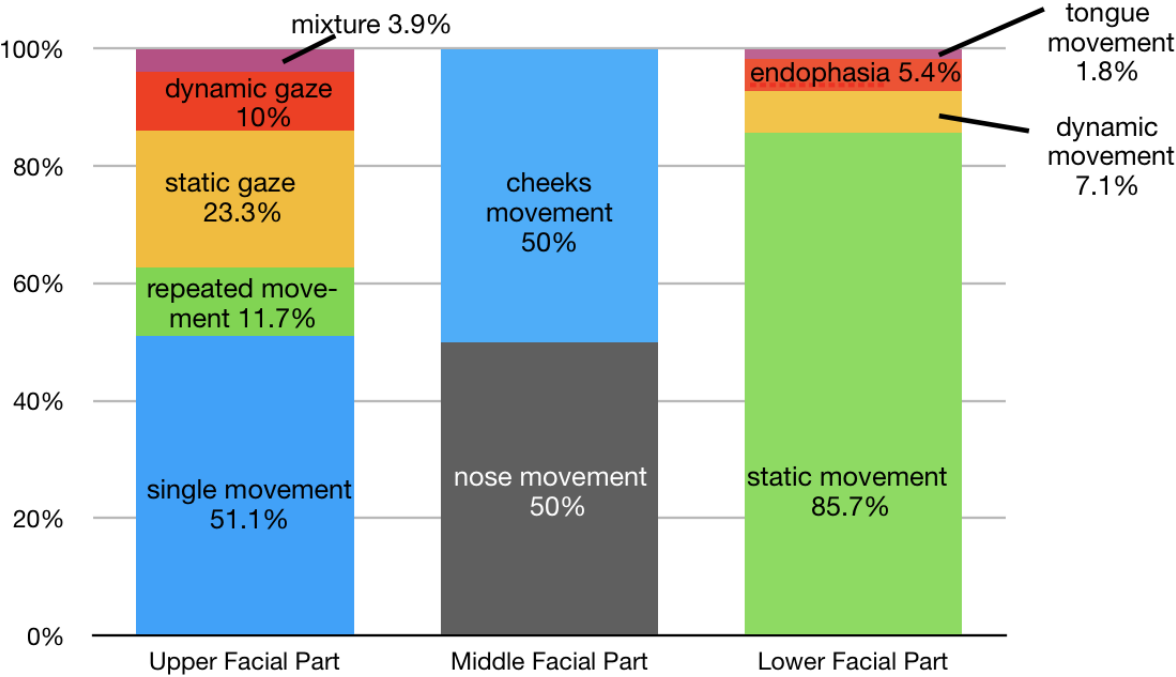


Figure 3.8: Taxonomy breakdown



Table 3.4: Taxonomy of Face Gestures

Upper Facial Part	Single Movement	Gesture is performed with a single muscle movement.
	Repeated Movement	Gesture is performed by repeating the same muscle movement.
	Static Gaze	Eyes look at a certain point or direction.
	Dynamic Gaze	Eyes look at different locations continuously.
	Mixture	Gesture is performed with both muscle movement and eye gaze.
Middle Facial Part	Nose Movement	Gesture is performed with nose.
	Cheek Movement	Gesture is performed with cheeks.
Lower Facial Part	Static Movement	A muscle movement is held.
	Dynamic Movement	A series of muscle movements is continuously performed.
	Endophasia	Gesture is performed with mute spoken words.
	Tongue Movement	Gesture is performed with tongue.

movements. A participant used wrinkled nose to pause a video and another participant used sucking cheeks in and puffing cheeks out to give the commands of zoom in and zoom out.

In the category of lower facial part, gestures performed with a single held pose are considered as static movement. Some participants made a smile, opening mouth, or stretching lips as gestures. Dynamic movement means the gesture contains a series of movements performed continuously such as closing mouth and then opening it bigger or opening the mouth and then closing it like making a bite. However, when the continuous movement represents a mute spoken word, it is classified as an endophasia gesture. There is also a gesture performed with the tongue as a sticking tongue out gesture.

### Grouped Gestures

Among of the 240 gestures, we consider several similar movements as the same gestures. Table 3.5 shows how the gestures are combined as the same groups.

Table 3.5: Grouped Gestures

Blinking	blink / long blink / hard blink /close eyes /close eyes for three seconds
Frowning	squinting / lid tightened / frowning
Eyebrows Rais- ing	eyebrows raising / opening eyes big
Blinking Twice	blink twice / fast blink twice
Looking Up	looking up / rolling eyes
Looking Down	looking down / looking down once rapidly
Smile	smile /wild smile
Lips Stretching Left/Right	lips stretching left or right / lips raising left or right
Bite	open mouth and then close / a bite

We combined these gestures for three reasons. One is when the same movement is held in different duration time. Since the sense of time duration is varied from people to people, we grouped *blink* and *long blink* together. And, *blink twice* and *fast blink twice* are also grouped as a same gesture because "fast" is an ambiguous word to describe time. The second reason to combine gestures is when one performed gesture may cause the same movement in another one. When we squint or tighten our lids, the muscles also bring the eyebrows closer and cause the similar movement to frowning. Also, it is difficult for us to raise eyebrows without opening eyes big. The last reason is that the definitions of gestures are ambiguous or they can mean the same movements. The definitions of a smile and a wild smile can be different, and lips stretching and lips raising can have various performances due to the users. Although all performed gestures in our experiment are recorded as videos, to create gesture mappings, we consider that we do not have a strong purpose to separate gestures by slight differences since it may be confusing when users try to learn gestures.

### 3.5.2 Emotions in Face Gesture Mapping

Since human face is essential to emotional expressions, we have tried to understand if there are correlations between face gesture mappings and emotions through our questionnaire. We designed an emotion assessment scale with Self Assessment Manikin (SAM) [23], which is considered to be able to give participants a better description of emotions with images than text. We put arousal level (high arousal - low arousal) and emotion valence (positive - negative) as X and Y axis to name each emotion with a X-Y coordinate. In the book "Affective Computing" written by Rosalind W. Picard, the author describes mood with the dimensions of valence and arousal [29]. Although what we collected in the experiment here is not mood but emotions, since it also says that mood is a background process of emotions, we decide to use the same dimensions for representing emotions.

Among the 240 gesture mappings picked by our participants, 36.25% of them were answered as that there were emotions in the mappings. The distribution of the 84 named emotion coordinates is shown in Figure 3.9. Although there is no significant principal components in the distribution, it still indicates that the emotions tend to be at a middle and low level of arousal. Furthermore, when we only look at the emotions in defined gestures, there are actually only nine coordinates indexed by our participants. However, as Figure 3.10 shows, no coordinate is at the 4th quadrant; thus, we can interpret from the graph that users do not tend to have high arousal - negative valence emotions, where includes emotions such as anger, when they are making a face gesture to give a command. Also, the commands of Pause, Zoom In, and Zoom Out are all have two coordinates indicated by different participants; however, the coordinates for the same commands are actually not at the same quadrants. To sum up, we consider that there may not be a significant relation between emotions and face gesture mappings.

### 3.5.3 Mapping Assessment

We tried to understand the mappings through the self-assessment from participants and the planning time. We used three 7-point Likert scales of "goodness of match", "ease to perform", and "social acceptability" to evaluate the mapping. As shown in Figure 3.4, the first read, "This face gesture is a good match for command (command name)." The second read, "This face gesture is easy to perform." And the last read, "I will use this face gesture in public." We scored the

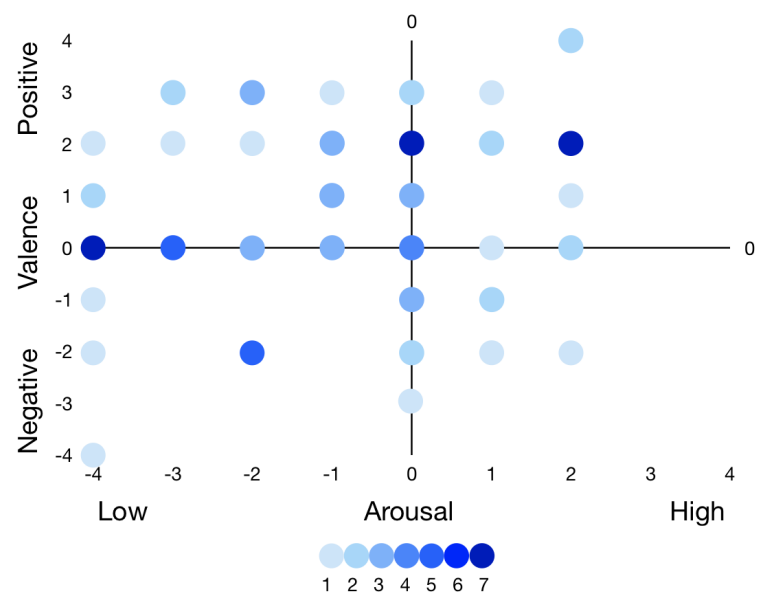


Figure 3.9: The distribution of the 84 named emotion coordinates: the numbers of the coordinates are marked with colors.

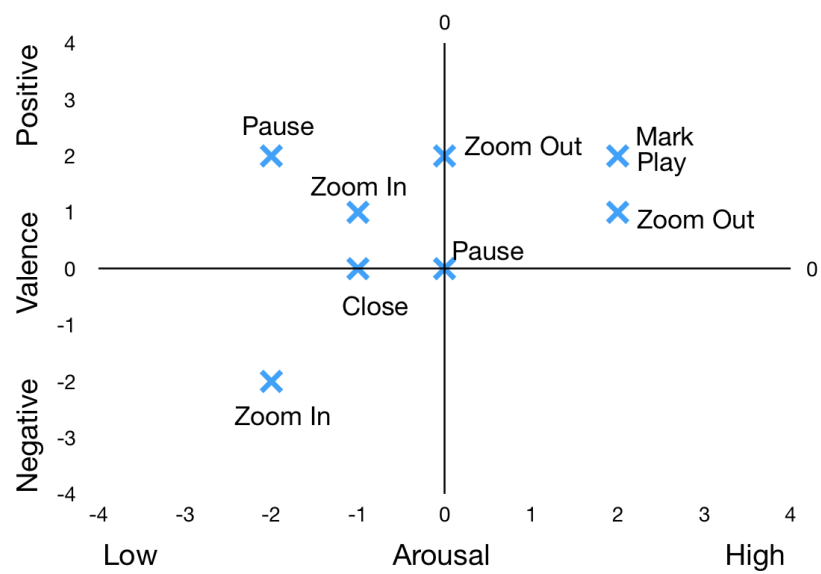


Figure 3.10: Emotion distribution of defined gestures.

answers from 1:entirely disagree to 7:entirely agree. We collected 240 mappings and 72 of them were the most used ones. We analyzed the assessment data from only the 72 defined mappings.

To understand participants' feelings toward their gestures and mappings, we computed the mean of self-assessment scores on each gesture mapping in Table 3.6. Since the score of *goodness* represents how good the participants feel about the defined gesture mappings and the agreement rate (as shown in Table 3.3) represents the agreement among all participants to the defined gesture mappings, we wonder that if the *goodness* and the agreement rate can affect each other. However, we did not find a correlation between them.

Table 3.6: The mean of self-assessment scores on each gesture mapping.

	Goodness	Ease to perform	Social acceptability
Next - Looking Down	4.4	6	6.4
Mark - Winking(Either Eye)*	5.7	5.7	5.3
Open - Eyebrows Raising	5.8	5.8	6
Close - Blinking	5	5.8	5.8
Play - Eyebrows Raising*	5.3	6.5	6.3
Pause - Blinking	5.8	6.6	6.2
Fast Forward - Looking at the Right	4.8	6	6
Fast Backward - Looking at the Left	4.8	6	6
Zoom In - Frowning	6.2	6.6	6.6
Zoom Out - Eyebrows Raising	5.6	6.2	5.8
Scroll Down - Looking Down	4.4	5.8	4.6
Scroll Up - Looking Up	4.4	5.8	4.6

## Notes

- 1 Dlib is a toolkit for making machine learning and data analysis applications.  
<https://pypi.python.org/pypi/dlib>

# Chapter 4

## Implementation

We implemented the system in Python 2.7 with OpenCV for face gesture recognition on macOS Sierra 10.12.6. We had the first prototype to understand that the Fisherface algorithm is possible to classify certain face gestures on different faces and the other one implemented based on our defined face gestures.

### 4.1 Prototypes

We first built a prototype which could detect four face gestures. However, after our gesture defining experiment, we found the prototype was not enough for detecting several gestures such as eye movements; therefore, we improved the system with a similar method to recognize the user-defined set.

#### 4.1.1 First Prototype

Following several face recognition works and their descriptions [3, 5, 30], we built our recognition system which could detect different face gestures. In the book "Mastering OpenCV with Practical Computer Vision Projects", a face recognition system which could recognize different people was built in C++. The system labels face images by different people; however, based on our understanding in facial recognition, when we label face images by different face gestures, we are supposed to make the system recognize gestures performed by different users.

We started with capturing the face region from webcam using `imutils` and `dlib` libraries. `Imutils` is developed by Adrian Rosebrock. It provides a series of functions to make basic image processing functions which makes it more convenient to build our system. After making the face region into grayscale and equalized, we could save the images as png files into a data folder with filenames labeled with numbers. In this prototype, we tried to recognize four gestures: eyebrows

raising, frowning, mouth stretching right, and mouth stretching left (Figure 4.2). We saved total 70 images as a training data set. There were ten images for three of the gestures performed by two people, and the other one performed by only one person due to a data saving miss. The data set then trained into a yml file as a model with an open source code<sup>1</sup>. Finally, we used Fisherface recognizer function provided in OpenCV and loaded the model to recognize the four gestures from webcam capture in real-time.

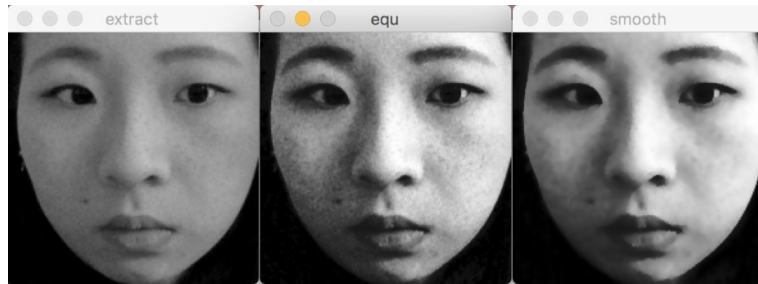


Figure 4.1: Face region processing in the first prototype.

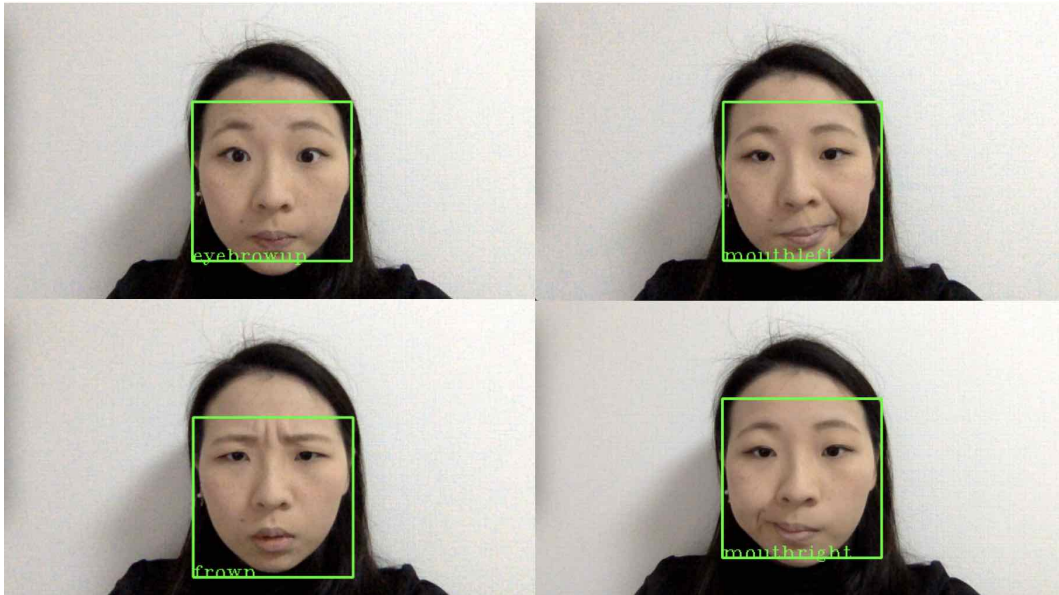


Figure 4.2: Four gestures recognized in the first prototype.

### 4.1.2 Defined Gesture Recognition

After we got the face gesture set from our user-defined gesture experiment, we tried to detect the gesture set. However, gestures using eye movements such as looking up and down are difficult to be recognized in the same setup. We then tried several ways to improve the detection.

First, we narrowed down the Region Of Interest (ROI) to two eyes and eyebrows (Figure 4.3(a)) since our gesture set is focusing on this region, but it didn't improve the recognition much. Second, we focused on detecting gestures from only one eye and eyebrow (Figure 4.3(b)), and we got a better recognition. However, it was still unstable when the lights changed or using data collected from different people. Since most of eye tracking methods processed ROI on only the eye, we tried to narrow our ROI to one eye without eyebrow (Figure 4.3(c)). Also, making the contrast level higher would help detecting eye movements because it could make pupil and iris areas more clearly separated from the sclera; thus, we applied different contrast levels and compared which could best improve ROI for detection (Figure 4.3(d)), and indeed, higher contrast level made eye movement recognition have a better performance in our prototype. Since we realized how contrast level could effect training data, we processed Figure 4.3(b) with higher contrast level and trained with gesture data collected from one person. The result (Figure 4.3(e)) showed that it was much improved so we decided to train data with this image processing method.

In the final implementation, we collected data from nine people (four males) after we received their written consent of using their face images as training data. To avoid the light affecting image processing for recognition, the data was collected at six different places. Since we had an issue of detecting the winking gesture of e-mail checking application, we implemented the other two of the computer applications we presented in our user-defined gesture experiment. We discuss about the detection problem in the later chapter.

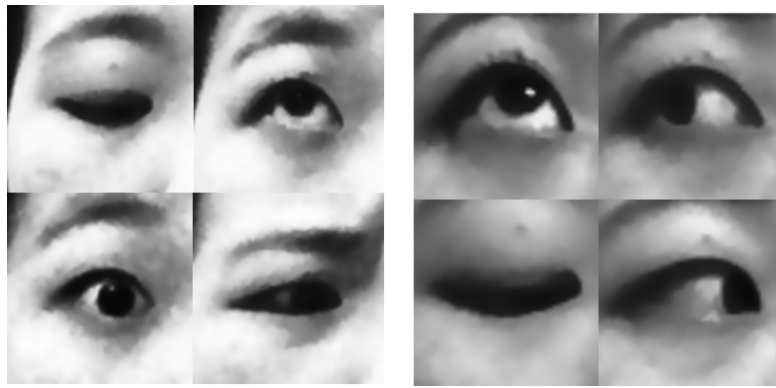
### 4.1.3 Communication

The interface was built in Swift 3.0. We used SwiftSocket<sup>2</sup> library to build the communication between our recognition system and the interface. In order to distinguish natural movements and face gestures, we controlled the among of sending-messages by a filter. It also made the system need a few time to distinguish the detected facial movements as gestures to give the commands.





(a) ROI on two eyes and eyebrows



(b) ROI on single eye and eye-brow

(c) ROI on single eye without eye-brow



(d) ROI on single eye with higher contrast level

(e) ROI on single eye and eyebrow with higher contrast level

Figure 4.3: Image processing.

## 4.2 Algorithm

Before the system is able to recognize our defined face gestures, we have to capture the face from webcam streaming frames and process the captured face images. The detection algorithms are provided in `dlib` and `imutils` libraries in Python. To detect the defined face gestures, we trained our own machine learning model and analyzed input images with Fisherface algorithm provided in OpenCV.

### 4.2.1 Face Detection and Processing

To help us processing the face region, we used `imutils` to align the faces and it was based on `dlib`'s facial landmarks. In `dlib` library, the face detector is made with a combination of the classic Histogram of Oriented Gradients (HOG) <sup>3</sup> and a linear classifier [17]. It creates sliding window detection scheme with trained model to find faces in the images. Next, with a given shape predictor file, we are able to have 68 face landmarks as shown in Figure 4.4. By using these landmarks, `imutils` has a `FaceAligner`<sup>4</sup> class providing functions to align the face regions. It first extracts left and right eye coordinates from landmarks. And then it computes the center of each eye and the angle between eye centroids. Next, it finds the desired eyes placement by calculating the desired x-coordinate of the right eye from the right edge of the image to be equidistant as the desired x-coordinate of the left eye from its left edge. Finally, it computes the center (x,y)-coordinates between the two eyes in order to rotate and rescale the eyes to the desired placement. (Figure 4.5)

### 4.2.2 Face Gesture Recognition

We trained two machine learning models and recognized our defined gestures in media playback and reader applications respectively with Fisherface algorithm. Fisherface is developed by Ronald Fisher [12]. It is an algorithm for facial recognition based on Linear Discriminant Analysis (LDA) and provided in OpenCV library. According to the LDA, it maximizes the ratio of between-classes scatter and minimizes within-classes scatter in order to cluster same classes tightly and make different classes further away from each other in the lower-dimensional representation [1, 20]. Fisherface helps the machine to classify face images with the understanding of facial features. Unlike usual image classification, it makes the facial features clearer in order to improve facial recognition.

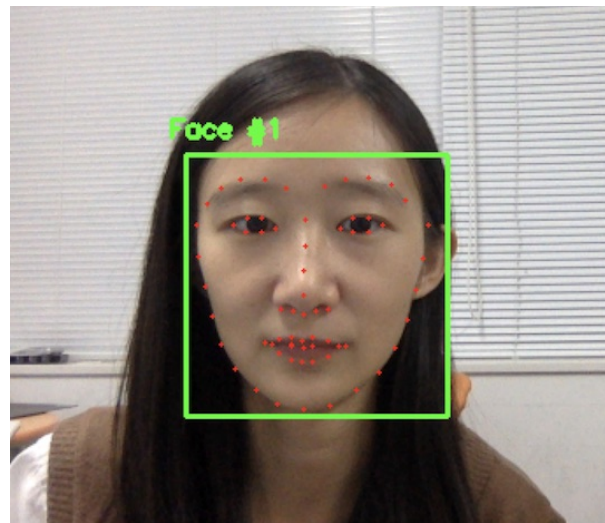
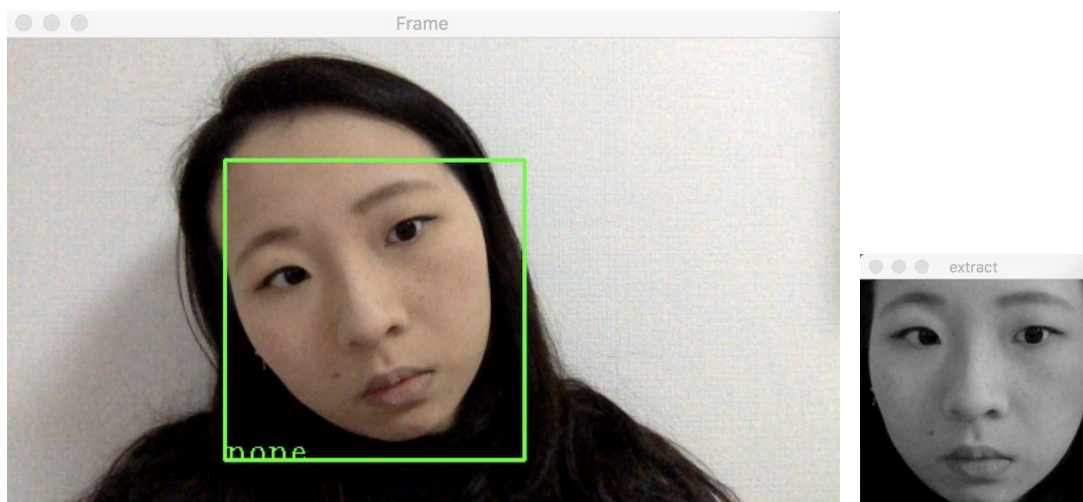


Figure 4.4: 68 face landmarks.



(a) Capturing the face.

(b) Aligned face.

Figure 4.5: Face Alignment.

## Notes

- 1 Face-Recognition-Train-YML-Python  
<https://github.com/AsankaD7/Face-Recognition-Train-YML-Python>
- 2 SwiftSocket  
<https://github.com/swiftsocket/SwiftSocket>
- 3 Histogram of Oriented Gradients (HOG) is a feature descriptor for object detection in computer vision and image processing.
- 4 Adrian Rosebrock gives a introduction of how to use this class in the article "*Face Alignment with OpenCV and Python*" on his website.  
<https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/>

# Chapter 5

## Evaluation

”Make a Face” system is valuated in two aspects. We first tested the face gesture recognition system through a detection accuracy test. Next, we designed a usability study based on the accuracy test result to valuate the interaction we built.

### 5.1 Detection Accuracy

To test a machine learning system, scientists usually use ”Confusion Matrix” to understand accuracy, precision, recall, and F1-score of it. There are two classes in a confusion matrix, one is predicted class, and the other is actual class. When we predict the condition is true and the condition is true, we call it True Positive (TP). But if the condition is false, it is called False Positive (FP). On the other hand, when we predict the condition is false and the condition is false, then we call it True Negative (TN). But if the condition is true, it is called False Negative (FN). However, to test the recognition program of our system, we did not use confusion matrix. Since the detection is continuous and we can control how much detection data we want to use to trigger the commands when we send the data to our GUI program, we only calculated the percentage of TP happening in this test.

Table 5.1: Confusion Matrix

		Actual Class	
		Condition Positive	Condition False
Predicted Condition	Predicted Condition Positive	<b>True Positive</b>	False Positive
	Predicted Condition Negative	False Negative	True Negative

We conducted the accuracy test on five participants (two male) who were not included in our training data. The age is between 24 and 28 years old ( $SD = 1.52$ ). After earning participants' agreement and explaining the test with a consent form, the observer instructed participants to try on the system with different gestures from the two implemented applications. The participants were told to perform each gesture in 8 seconds in a random order and had a 8-second break in between. We collected data from terminal output and extracted continuous 60 outputs from the 8-second performing time. Accuracy of detecting different gestures from two application gesture sets are shown in Table 5.2. There is a data missing issue on P3 due to a data collecting error.

Table 5.2: Detection Accuracy

	P1	P2	P3	P4	P5	Total
Media Playback	77.5%	86.25%	98.89%	100%	64.58%	85.44%
Eyebrow Raising	96.67%	70%	98.33%	100%	98.33%	79.33%
Looking Left	45%	93.33%	98.33%	100%	60%	92.67%
Looking Right	96.67%	91.67%	N/A	100%	0%	72.09%
Blinking	71.67%	90%	100%	100%	100%	92.33%
Reader	64.58%	87.5%	98.33%	100%	74.58%	85%
Eyebrow Raising	0%	93.33%	100%	100%	3.33%	59.33%
Frowning	58.33%	63.33%	N/A	100%	95%	79.17%
Looking Up	100%	100%	95%	100%	100%	99%
Looking Down	100%	93.33%	100%	100%	100%	98.67%

As the result, detection accuracy from five participants in total is 85.44% on media playback application and 85% on reader application. Although there is eyebrow raising gesture in both applications, in reader application, accuracy of the gesture is only 59.33%. From the collected data, we can interpret that it is because of the confusion with looking up gesture.

## 5.2 Usability Study

To evaluate this interaction, we conducted a usability study using a "Wizard of Oz" method. Although we have the implemented system, according to the result of our detection accuracy test, the detection may work variously from different

people, and in order to evaluate the "interaction" rather than the "implementation", we decided to use "Wizard of Oz" approach to design our usability study.

### 5.2.1 Study Design

In a "Wizard of Oz" experiment, the system is controlled by a person who plays the role of a wizard, while the participants believe that the system is working automatically. Figure 5.1 shows the environment of our usability study. The participants sit in front of a computer and the observer, who was also the "wizard" in this experiment, sit behind them in order to control the system secretly. The participants' screen showed the interface, gestures, and tasks. On the other hand, the observer could see both the participants' screen and their faces from the webcam streaming.

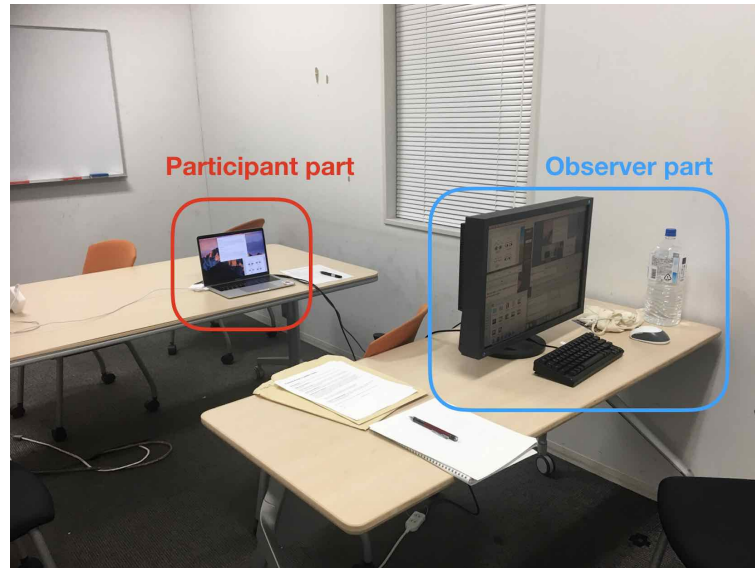


Figure 5.1: Wizard of Oz experiment setup.

The study was conducted on eight participants (three male) who did not participate in the user-defined gesture study. The mean age was 24.38 years ( $SD = 1.41$ ). After receiving the participants' written consent, the observer explained the procedure of the experiment. The participants were informed that there would be two applications and two tasks for each in this usability study. They had to use the face gestures shown in the beginning of each session to control the system and finish the tasks. The applications and tasks are shown in Table 5.3.

The contents used for the usability study in both applications were different from the ones presented in our previous user-defined face gesture study in order to create better tasks. In the reader application, we used a part of the short paper written by the author. And in the media playback application, we imported a 1:18 minutes long drone-filmed scenery video which was provided with CC0 license on PEXELS VIDEOS<sup>1</sup>.

Table 5.3: Applications and Tasks

Application	Task
Media Playback	P1: Play the video and skip it over 0:50 then pause it.
	P2: Play the video and play it back to the first scene from the second scene.
Reader	R1: Find the name of the "Haptic Controller" in the Implementation section, tell me what's the name. And find the term "SMASH" in Related Works section, tell me what does "H" represent for.
	R2: Try to adjust the size of the words and then find the best size for you.

After the participants finished the four tasks, they were asked to fill in a on-line standard usability questionnaire. We used the USE questionnaire [2,33] to do the evaluation. There are 30 questions in four aspects, which are *Usefulness*, *Ease of Use*, *Ease of Learning*, and *Satisfaction*. The questionnaire is presented in 7-point Likert scales. The participants could select "NA" if they did not consider the question applicable. Also, below the questionnaire, it instructs the participants to "List the most **negative** aspect(s)" and "List the most **positive** aspect(s)" with three fill-in blanks after each.

### 5.2.2 Results

The result of the usability study is shown in Table 5.4 by the four aspects of the USE questionnaire. We received generally good scores towards all aspects. Furthermore, we collected the listed negative and positive aspects filled in by the participants. We gained total 33 feedback statements and classified them into four issues: *Gesture*(13), *Interaction*(13), *Scenario*(6), and *Content*(1).



Table 5.4: Usability Score

Aspect	Score
Usefulness	5.3
Ease of Use	5.3
Ease of Learning	6
Satisfaction	5.4

The *Gesture* issue contains the statements regarding to the defined gestures. There are 12 negative statements and one positive statement. It is mentioned in five statements that using looking up and down as gestures makes it difficult to see the interface and the contents while controlling the system. The only positive statement says that the opening eyes big gesture is the easiest one; however, another statement which is negative says that it is uncomfortable to use opening eyes big as a gesture.

In the *Interaction* issue, six negative and seven positive statements are listed. Three statements indicate that the contents cannot be controlled how much it skips, scrolls, and zooms when the users use fast forward/backward, scrolling, and zooming functions. On the other hand, five of the positive statements mention that the interface is convenient and the interaction is easy to remember and understand.

The six statements in the *Scenario* issue are all positive and mentioning use case scenarios such as "the interaction frees both hands" and "it can help me checking the navigator while driving." And the only *Content* issue is a positive feedback of the video content.

## Notes

- 1 PEXELS VIDEOS  
<https://videos.pexels.com>

# Chapter 6

## Discussion

In this chapter, we discuss about the issues in our studies and implementation. We discuss about the limitation of our system and look at the issues in the defined face gestures. Furthermore, we give discussions on the qualitative data collected in our studies.

### 6.1 Limitation of the Implementation

While we were implementing the system, we found that there were some limitations in our system. We had a difficulty in detecting small face movements on the whole face region. And, it was difficult for the recognizer to detect the natural face when there was the label of natural face in the trained data. The continuous detection was also an issue for triggering the commands though the problem was possible to be solved after gestures were recognized.

#### 6.1.1 Detection of Both Eye Regions

We conducted the user-defined face gesture experiment with three application use cases; however, this time, we only implemented two of them. In the e-mail checking application, which we did not implement yet, the winking gesture was difficult to be detected by our system. When the ROI was set as the whole face region or the both-eye region, it was possible to detect the winking gestures. However, it was difficult to detect smaller gestures such as looking up and down correctly with the participants' own data either the model trained with data from another eight participants. Although there is no theory indicating how many images we need in order to detect small movements on the facial region, we still consider that the lack of data may be a reason why the detection does not work well.

### 6.1.2 Natural Face and Continuous Detection

There is no natural face label in our final implementation. When we trained the model with a natural face label using data collected from a single person, we found the recognition worse than without natural face label. We consider that there are two possible reasons causing this issue. First, the data may not be enough to classify the number of labels when we added natural face label in. Since classifying seven gestures in a single trained model did not work as well as detecting only four gestures, the lack of data may be the cause of it. Second, unlike most of the gestures in a single model, the natural face is more similar to each gesture. As the result of our detection accuracy test, when the eyebrows raising gesture and the looking up gesture were trained in the same model, the machine was confused by the two gestures because they had similar features. Since the natural face does not have the significant features, it may confuse the machine on its classification.

While we do not have a natural face label in the trained model, we control the detection by its confidence level. When the confidence is either too high or too low, the machine tells us that it is a natural face, which actually means that there is no target gesture in the ROI. In other words, it causes the problem that the machine continuously detects random gestures easily in the ROI when there is actually no target gesture performed. Eventually, we use a filter to control the among of information which says that a target gesture is detected before they are sent to trigger the commands in GUI program. Thus, we did not focus much on the issue of natural face and continuous detection.

## 6.2 Gestures and Daily Usages

In our user-defined gesture experiment, we had a brief interview after each trial in order to understand how the users think about using face gestures as an input modality. We found an issue on using head movements which were not included in the definition of face gesture in our experiment. Also, besides the application scenarios we mentioned in Chapter 3, participants actually gave us more ideas about the advantage using face gestures.

### 6.2.1 Limitation of the Gestures

When we were conducting the experiment, participants were naturally giving the observer their ideas of how they were thinking, such as why certain movements were not appropriate in their opinion. Summing up with the interview result, we found several issues are worthy to discuss since we received the same ideas from different participants.

#### Eye Movement and Head Movement

As mentioned in Chapter 3, simple eye movements, such as looking up, down, left, and right, are the often used ones in the defined face gestures. Eye movements are easy to use because users can give the commands including directional components by looking at the direction. Many participants chose to use looking up and down to control scrolling up and down. Also, in the media playback application, with the timeline shown on the interface, participants picked the gesture of looking at the right while having an image of moving the slider to the right, by which we usually controlled the video to go forward. However, there is also a difficulty of using eye movements. The movements distract users while they are looking at the contents shown on the screen. One participant said that he did not really want to use the gesture of looking down to give the command of scroll down because after he looked down to scroll, he had to go back to the text and find again where was the line he was reading.

This problem is related to another issue of using head movements. When our observer explained the definition of face gestures in the beginning of experiment, many participants had the question that if they could use head movements to make face gestures. We did not include head movements in our experiment this time because the movements were usually more obvious than other facial movements and we defined face gestures as "gestures made with any visible facial movements" in this study. Although head movements may be more intuitive for making directional commands, we have not yet understood that if using head movements as gestures is better than using eye movements and other facial movements.

#### The Concern of Detection

While participants were deciding the gestures, they were often thinking about if the gesture would be possible to be detected even our observer told them we could assume that all visible facial movements were able to be detected. Thus,

the picked gestures were affected by participants' concerns toward an detection issue. One participant considered that she had very slant eyes so she mentioned that she tried not to use eye movements as gestures because it might be difficult to detect.

Another concern of the detection issue we understand from our participants is that some of them worry the machine can not tell when to detect the gestures. If users do a certain movement which is same as the defined gesture and the machine can not recognize whether it is a natural movement or a explicit input gesture, it will stop the system working. Some participants also mentioned that they tried not to use the movements they used often in daily life as gestures because they did not want to confuse the machine and they wanted to make special movements for gestures. In fact, this problem is possible to solve by giving a trigger for gesture detection before giving a gesture itself. Imaging that we can have "Ok, Google" and "Hey, Siri" to wake our intelligent personal assistants up in computer devices, it may be possible for us to have a trigger like these voice commands in order to call the gesture detection function.

### 6.2.2 Alternative Use Cases

Besides the application scenarios we mentioned in the earlier chapter, there were other use cases mentioned by the participants in our studies. Since they may not be the main purpose to have face gesture interaction, we only discuss about these ideas as alternative applications.

Although defining new face gestures was difficult for the participants since people have not yet been used to have face gestures for interacting with computers, many of the participants in our user-defined gesture experiment still considered it would be convenient, interesting, and helpful. They mentioned that using face gestures would be convenient because sometimes they felt lazy to leave the bed, chair, or couch and grab the remote control or reach the televisions and computers while watching movies. In fact, it indicated a possible situation that when users were too sick to leave the bed and to speak but they had to control certain devices at home.

# Chapter 7

## Conclusion

From the finding in this thesis, users tend to use the eye region to make face gestures in the situation that they have to give the commands to computer devices only with the face. In the user-defined face gesture experiment we conducted, participants chose to use the eye region because of the following reasons:

- Eye movements can have directional gestures which are difficult to make with other facial parts.
- It is easy to perform face gestures using the eye region with the subtle movements.
- Compared with other facial parts, it is socially acceptable that using the eye region to make tiny and quick gestures

While using the eye region is the best way that our study indicated to perform face gestures, it still has its limitation. Face gestures made with eye movements may distract users' attention away from the contents where they are looking. If users look up to scroll up a document, the gesture moves users' focus on the lines which they are reading. However, it is helpful to have face gestures as an optional input modality when users' hands are occupied.

Similar results were found in a usability study. We conducted a usability study with eight participants who were different from the initial user-defined gesture study. Through a standard self-assessment questionnaire, the interaction we built was rated as useful, easy to use, and easy to learn. Also, it satisfied the users' needs. From the participants' feedback, we received five statements indicating that it was difficult to see the contents clearly while controlling the system when they used the looking up and down gestures.

Furthermore, based on the finding in our face gesture study, we developed a face gesture recognition system using webcams with the defined gestures. The

result of detection accuracy test shows it 85.44% accurate on media playback application and 85% accurate on reader application. We used SwiftSocket library to build the communication between the recognition system and the interface. In order to solve the problem of the confusion between natural movements and defined gestures, we adjusted the among of sending-data to the interface program. Although some gestures are not detected well in our final implementation, we understand the possible issues and that it is possible to improve the detection by collecting more data for the machine learning system.

In this thesis, we describe three application scenarios including 12 commands discussed in the study and two of them implemented with eight commands in our recognition system. According to our findings in this study, it is a potential aspect for face gesture studies to explore other commands in various use cases. If we have more understandings toward what commands the users want to control, it may help us to develop suitable face gesture recognition system on different devices in order to meet the users' needs.

# References

- [1] Face recognition with opencv. Retrieved December 11, 2017 from [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html#face-recognition](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition).
- [2] Use questionnaire: Usefulness, satisfaction, and ease of use. Retrieved January 04, 2018 from <http://garyperلمان.com/quest/quest.cgi?form=USE>.
- [3] Baggio, D. L., Emami, S., Escriva, D. M., Ievgen, K., Mahmood, N., Saragih, J., and Shilkrot, R. *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing, Limited, 2012. recommended: advanced OpenCV project support/examples inc. iOS and Android examples.
- [4] Bell, S. S. body language of winking, 2012. Retrieved November 28, 2017 from <http://readingbodylanguagenow.com/bodylanguageofwinking/>.
- [5] Beyeler, M. *OpenCV with Python Blueprints*. Packt Publishing Ltd., 2015.
- [6] Bolt, R. A. "put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, ACM (New York, NY, USA, 1980), 262–270.
- [7] Charlie. Stephen hawkins improved speech software is now free to download, 2015. Retrieved November 28, 2017 from <https://bltt.org/stephen-hawkings-improved-speech-software-is-now-free-to-download/>.
- [8] Cox, L. 10 uses of facial recognition technology, 2017. Retrieved November 28, 2017 from <https://disruptionhub.com/10-uses-facial-recognition-technology/>.
- [9] De la Torre, F., Chu, W.-S., Xiong, X., Vicente, F., Ding, X., and Cohn, J. Intraface, 05 2015.



- [10] Ekman, P., and Rosenberg, E. L., Eds. *What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system(FACS)*, first ed. Series in affective science. Oxford University Press, 1997.
- [11] Felberbaum, Y., and Lanir, J. Step by step: Investigating foot gesture interaction. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, ACM (New York, NY, USA, 2016), 306–307.
- [12] FISHER, R. A. The precision of discriminant functions. *Annals of Eugenics* 10, 1 (1940), 422–429.
- [13] Henze, N., Löcken, A., Boll, S., Hesselmann, T., and Pielot, M. Free-hand gestures for music playback: Deriving gestures with a user-centred process. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, MUM '10, ACM (New York, NY, USA, 2010), 16:1–16:10.
- [14] Intel. Assistive context-aware toolkit (acat). Retrieved November 28, 2017 from <https://01.org/zh/acat?langredirect=1>.
- [15] Jafri, R., and Arabnia, H. R. A survey of face recognition techniques. *JIPS* 5, 2 (2009), 41–68.
- [16] Kaplan, K. How intel keeps stephen hawking talking with assistive technology, 2014. Retrieved November 28, 2017 from <https://iq.intel.com/behind-scenes-intel-keeps-stephen-hawking-talking/>.
- [17] Kazemi, V., and Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), 1867–1874.
- [18] Keltner, D. The signs of appeasement: Evidence for the distinct displays of embarrassment, amusement, and shame. *Journal of Personality and Social Psychology* (1995), 441–454.
- [19] Lee, G. A., Wong, J., Park, H. S., Choi, J. S., Park, C. J., and Billingham, M. User defined gestures for augmented virtual mirrors: A guessability study. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, ACM (New York, NY, USA, 2015), 959–964.

- [20] Lee, H.-J., Lee, W.-S., and Chung, J.-H. Face recognition using fisherface algorithm and elastic graph matching. In *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 1 (2001), 998–1001 vol.1.
- [21] Lee, J., Yeo, H.-S., Dhuliawala, M., Akano, J., Shimizu, J., Starner, T., Quigley, A., Woo, W., and Kunze, K. Itchy nose: Discreet gesture interaction using eog sensors in smart eyewear. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers, ISWC '17*, ACM (New York, NY, USA, 2017), 94–97.
- [22] Lyons, M. J. Facial gesture interfaces for expression and communication. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 1 (Oct 2004), 598–603 vol.1.
- [23] M. Bradley, M., and J. Lang, P. Measuring emotion: The self-assessment manikin and the semantic differential. 49–59.
- [24] Masai, K., Sugiura, Y., Suzuki, K., Shimamura, S., Kunze, K., Ogata, M., Inami, M., and Sugimoto, M. Affectivewear: Towards recognizing affect in real life. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC'15 Adjunct*, ACM (New York, NY, USA, 2015), 357–360.
- [25] Matthies, D. J. C., Strecker, B. A., and Urban, B. Earfieldsensing: A novel in-ear electric field sensing to enrich wearable gesture input through facial expressions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, ACM (New York, NY, USA, 2017), 1911–1922.
- [26] Minagawa, A., Odagiri, J., Hotta, Y., Nakashima, S., Wei, L., and Wei, F. Touchless user interface utilizing several types of sensing technology. 34–39.
- [27] Obaid, M., Kistler, F., Kasparavičiūtė, G., Yantaç, A. E., and Fjeld, M. How would you gesture navigate a drone?: A user-centered approach to control a drone. In *Proceedings of the 20th International Academic Mindtrek Conference, AcademicMindtrek '16*, ACM (New York, NY, USA, 2016), 113–121.

- [28] Ogata, M., Sugiura, Y., Makino, Y., Inami, M., and Imai, M. Senskin: Adapting skin as a soft interface. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, ACM (New York, NY, USA, 2013), 539–544.
- [29] Picard, R. W. *Affective Computing*. MIT Press, Cambridge, MA, USA, 1997.
- [30] Rosebrock, A. Pyimagesearch. Retrieved December 4, 2017 from <https://www.pyimagesearch.com>.
- [31] Rozado, D., Niu, J., and Duenser, A. Faceswitch - low-cost accessibility software for computer control combining gaze interaction and face gestures. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, OzCHI '15, ACM (New York, NY, USA, 2015), 197–201.
- [32] Ruiz, J., Li, Y., and Lank, E. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (New York, NY, USA, 2011), 197–206.
- [33] Sauro, J., and Lewis, J. R. *Quantifying the User Experience: Practical Statistics for User Research*, 1st ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.
- [34] Sturman, D. J., and Zeltzer, D. A survey of glove-based input. *IEEE Computer Graphics and Applications* 14, 1 (Jan 1994), 30–39.
- [35] Vatavu, R.-D. User-defined gestures for free-hand tv control. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroITV '12, ACM (New York, NY, USA, 2012), 45–48.
- [36] Vatavu, R.-D., and Wobbrock, J. O. Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. In *Proceedings of the 33rd ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM (2015), 1325–1334.
- [37] Weigel, M., Mehta, V., and Steimle, J. More than touch: Understanding how people use skin as an input surface for mobile computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, ACM (New York, NY, USA, 2014), 179–188.

## REFERENCES

---

- [38] Wobbrock, J. O., Aung, H. H., Rothrock, B., and Myers, B. A. Maximizing the guessability of symbolic input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, ACM (New York, NY, USA, 2005), 1869–1872.
- [39] Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, ACM (New York, NY, USA, 2009), 1083–1092.