Convolutional Neural Network-Based Gaze Estimation Using Inside-Out Camera

（Inside-Out カメラを用いた畳み込みニューラル
ネットワークに基づく注視点推定）

Warapon Chinsatit

Abstract

The vision-based gaze estimation system (GES) involves multiple cameras, and such system can estimate gaze direction and what a user is looking at. The inside-out camera is the device to capture user eye and user vision. This system is widely used in many applications because the eye images with the pupil or cornea have much information. These applications have the capability to improve the quality of life of everyone especially a person with a disability. However, an end-user is difficult to access the ability of commercial GES device because of the high price and difficult to use. The budget GES device can be created with a general camera. The common method to estimate the gaze point from the vision-based GES is detected the pupil center position. However, the human eye has variable characteristics and the blinking makes reliable pupil detection is a challenging problem. The state-of-the-art method for the pupil detection is not designed for the wearable camera, the designed for the desktop/TV panels. A small error from the pupil detection can make a large error on gaze point estimation. This thesis presents the novel robust and accurate GES framework by using the learning-based method.

The main contributions of this thesis can be divided into two main groups. The first main contribution is to enhance the pupil center detection by creating an effective pupil center detection framework. The second contribution of this thesis is to create the calibration-free GES.

The first contribution is to enhance the accuracy of the pupil detection process. Handcraft and learning-based method are used to estimate the pupil center position. We design the handcraft method that using the gradient value and RANSAC ellipse fitting. The pupil center position was estimated by the proposed method and compared with the separability filter. The result shows the proposed method has a good performance in term of accuracy and computation time. However, when the user closes the eye, no eye present in the image, or a large unexpected object in the image, the accuracy will be decreased significantly. It is difficult for handcraft method to achieve good accuracy. The learning-based method has the potential to solve the general problem that becomes the focus of this thesis. This thesis presents the convolutional neural network (CNN) model to estimate the pupil position in the various situations. Moreover, this model can recognize the eye states such as open, middle, or closed eyes.

The second contribution is to create the calibration-free GES. The calibration process is the process to create the coordinate transfer (CT) function. The CT function uses for transfer the pupil position to the gaze point on-scene image. When the wearable camera moves during the use case, the static CT function cannot estimate

the gaze point accurately. The learning-based method has a potential to create a robust and adaptive CT function. The accurate calibration-free system can raise the accuracy of the GES. Furthermore, it makes the GES easy easier to use. We designed the CNN framework that has the ability to estimate the gaze position in the various situations. This thesis also presents the process to create the reliable dataset for GES. The result shows that proposed calibration-free GES can estimation the gaze point when glasses are moved.

Keywords: Gaze estimation system, Convolutional neural network, Inside-out camera, Eye state classification, Calibration-free gaze estimation system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of the research

An eye is an important organ for the human. This organ has an information to analyze the human condition. For example, the human gaze direction can be estimated by using the eyes image, human blink frequency has the ability to analyze the human mental condition [4]. Figure.1.1 shows the structure of human eye. The pupil is the dark aperture in the center of the eye that determines how much light is let into the eye. The iris is the colored part around the pupil of the eye that helps regulate the amount of light that enters the eye. The cornea is the clear front window of the eye that transmits and focuses light into the eye.

The gaze estimation system (GES) involves sensors, and such systems can estimate gaze direction and what a user is looking at. The desktop/TV panels gaze estimation device [20], [21] is an alternative human-computer interaction device. This well known GES device have to attach to the TV or monitor in order to estimate the gaze point on the screen. The wearable-based gaze estimation is one of the famous categories because of the mobility. A user can wear the device and do the everyday-life activity. This system enables the user to interact with the real-world environment not only in the monitor screen. Many wearable-based gaze estimation device has been developed by many researchers. The following is a sample of the wearable gaze estimation device.

- The scleral search coil [2]; This system is contact lens with conduction coils around sclera as shown in Fig.1.2. The conduction coils are connected to the amplifier by the signal cable. That signal send to the phase detector to detect the rotation of the eye. This device has a high accuracy and sensitivity. Designed for study the largest and the smallest movements of which the eye is capable. However, this system is used for laboratory experiments and not user-friendly.

- The Electrooculogram or Electrooculography is also known as EOG [3]; This system uses contact electrodes to stick on the skin around the user's eye to measure electric potential field from eyes as shown in Fig.1.3. This signal can estimate eye orientation even when the eyes are closed. However, every electrode has to stick on the user's skin and cannot detect eye blink frequency.

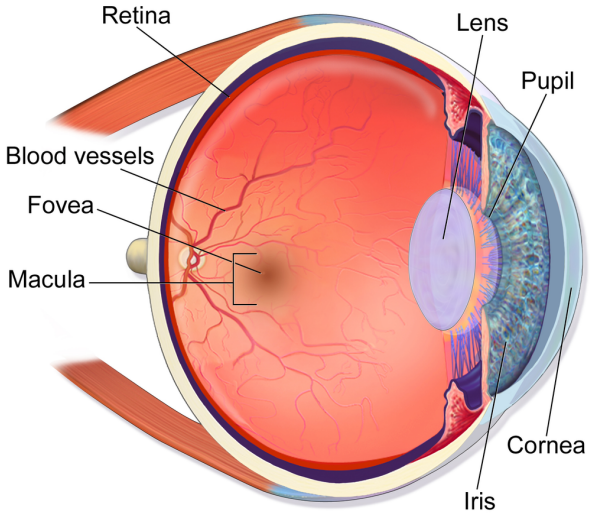


Figure 1.1: Structure of human eye [1].

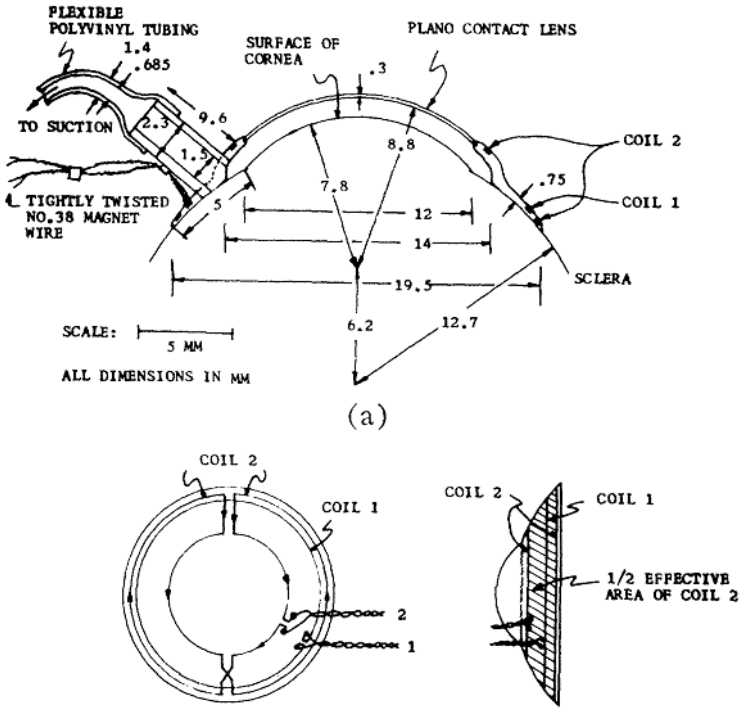


Figure 1.2: Scleral search coil [2]. The detail component of scleral search coil was shown in (a).

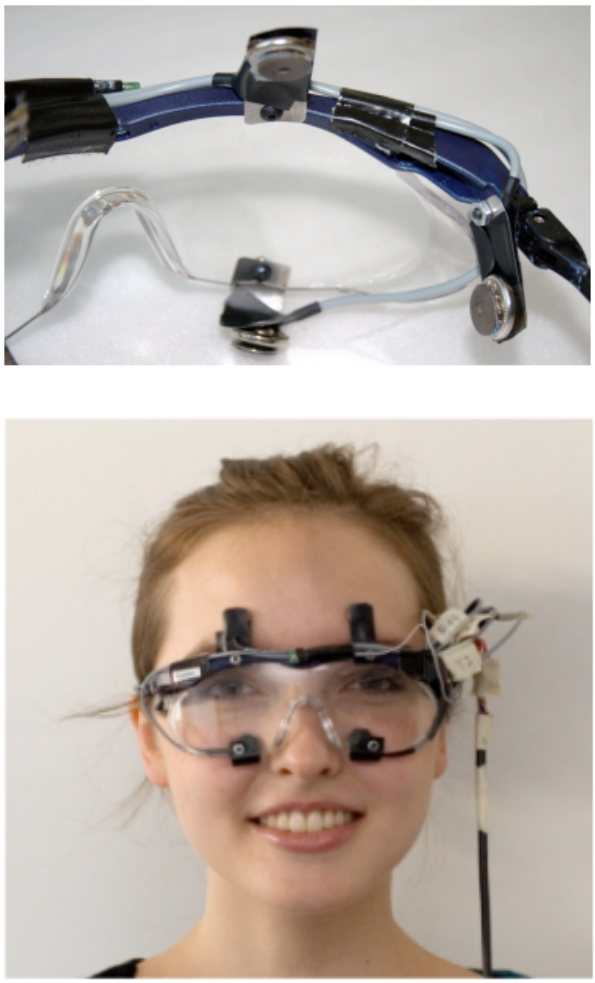


Figure 1.3: Electrooculography (EOG) [3].

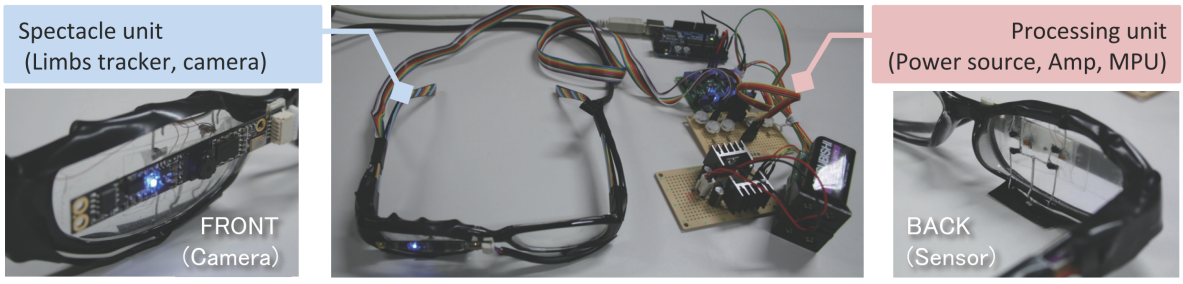


Figure 1.4: Eye tracking base infrared sensor [4].

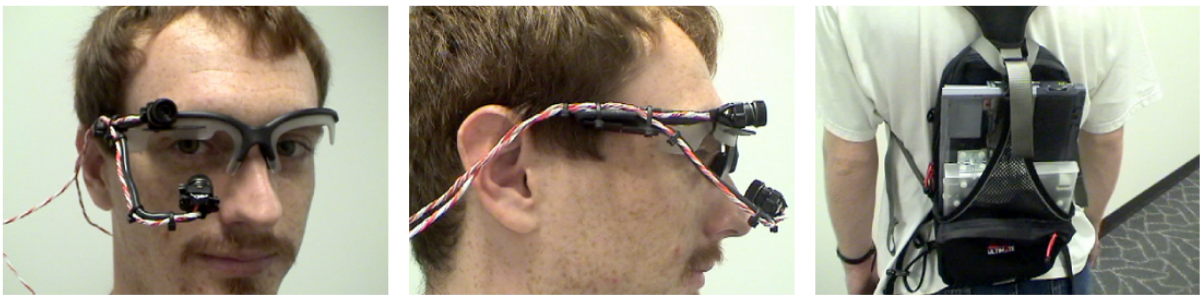


Figure 1.5: Mono inside-out camera [5].

- The simple gaze estimation device can be made by using the infrared sensor [4], [22] as shown in Fig.1.4. This device called Infrared corneal limbus tracker. This system uses the phototransistor to the detect infrared ray that reflects from the limbus. This eye tracker has two infrared LEDs and four phototransistors. By analyzing the four signal from phototransistors, the system can estimate the use gaze direction. However, this device is low accuracy compare with other GES device.

- The vision-based gaze estimation device use multiple camera to obtain the gaze direction also know as wearable camera base gaze estimation. This system is a famous for the GES. The inside-out camera [5] which is a wearable device and is mounted the eye camera and scene camera. The GES using the eye images from eye camera to estimate the user's gaze direction and use scene image to estimate what the user looking at. The simple inside-out camera use the single eye and scene camera to estimate the gaze point as shown in Fig.1.5. The inside-out camera has the potential to estimate the gaze point in the 3D coordinate [6], [23] by using the stereo camera as shown in Fig.1.6. This device has a mobility and user-friendly.

The vision-based GES or inside-out camera is widely used in many applications because the eye images with the pupil or cornea have much information. The GES has more benefit, for example, the patients with the disease, they cannot move the organ and difficult to communicate with others. The character input system can be made by applying the gaze estimation device to help the patients to communicate. For the people with near-sighted or far-sighted, the road signs recognition can be made by using gaze estimation device [24]. Furthermore, the patients with face blindness who cannot recognize the interlocutor, gaze estimation can be applied to build face recognition for helping the patients to recognize the interlocutor [25]. Moreover, the images from first-person vision device can be estimated the user position [26]. Recently, GESs have been used in various applications, such as video summarization [27], daily activity recognition [28], reading [29], human-machine interfaces [30], and communication support [31].

However, the commercial GES designed only for researcher and research marketing. The high accuracy GES device needs highly accurate and expensive sensor. The

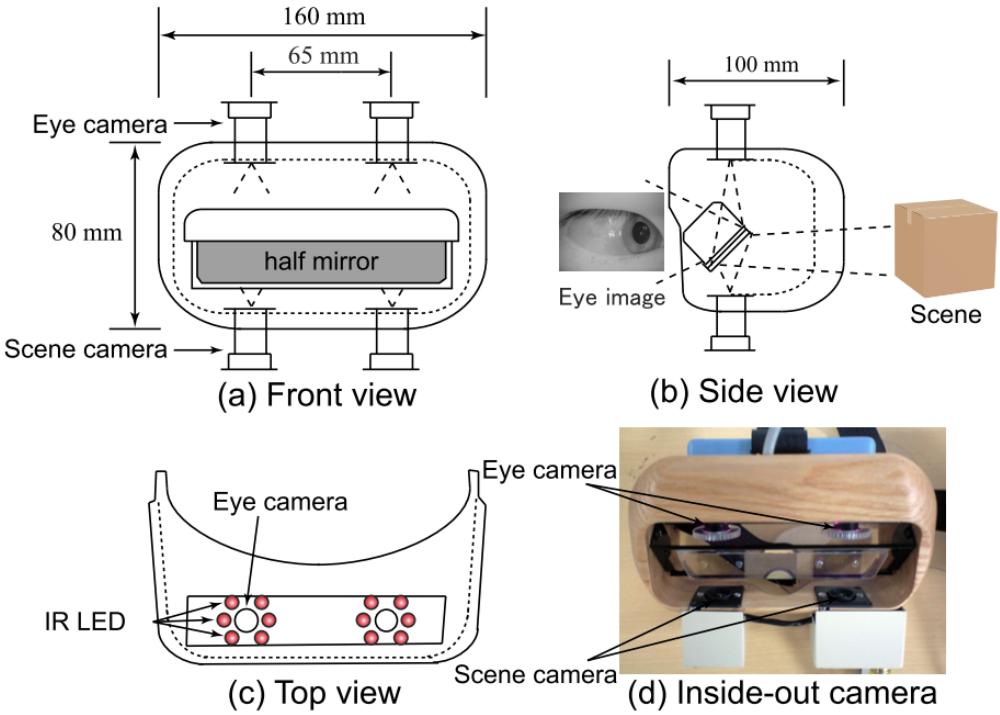


Figure 1.6: Stereo inside-out camera [6].

following is an example of commercial vision-based GES devices.

- The Tobii Pro glasses 2 [7] is the most popular GES on the commercial. This device have a four eye camera, two camera for left eye other for right eye. The single scene camera in the center of the frame as shown in Fig.1.7. This device also has many sensors, for example, microphone, gyroscrope, and accelerometer sensor. This designed for the researchers to observer human behavior by showing what a person is looking at in real time.

- The SensoMotoric Instruments (SMI) [8] is a company from German. The SMI provides many GES device, for example, The SMI Eye Tracking Glasses (ETG) 2 Observation, SMI ETG 2 Wireless Observation, SMI ETG 2 Wireless Analysis and SMI ETG 2 Wireless AnalysisPro. They have a wireless technology called remote eye tracking systems (RED) for communicate with the device in real time system. The SMI ETG 2 (Fig.1.8) using the two eye camera and single scene camera for user's gaze estimation.

- The Pupil Labs [9] provides the modular GES device as shown in Fig.1.9. This device can estimate the gaze direction by using only single eye camera. Additionally, the stereo eye camera can be use by option. This company provide the open source Application Programming Interface (API) for the developer and researcher to use in custom project [32].

Figure 1.7: Tobii pro glasses 2 [7].

- Nac image technology [33] provides the Eye-mark recoder (EMR-9). The EMR-9 [34] has a two version the cap and glasses as shown in Fig.1.10. The device only connects to the special controller and has no interface to the PC. This device has the stereo eye camera or mono eye camera. The controller has a capability to estimate the gaze point and show it on the screen.

Table 1.1 shows the specification of commercial gaze estimation device. The niche market [35] and expensive sensor make the cost of these device are high price. From this reason make the commercial device difficult for the end-user to access the ability of GES. However, the low cost and high accuracy GES can be created by using mono inside-out camera [5], [36]. This type of device allows everyone to access the ability of GES.

However, to estimate the gaze direction by using the camera is a challenging problem. Because the eye is a non-rigid object, users blink frequently. The duration of a blink is on average 100-150 milliseconds [37]. The actual rates vary by individual averaging around 10 blinks per minute. However, when the eyes are focused on an object for an extended period of time, such as when reading, the rate of blinking decreases to about 3 to 4 times per minute [38].

The human has many eye colors depend on two factors: the pigmentation of the eye's iris [39], [40] , and the way light is scattered as it hits the top layer of the iris [41]. The majority color is brown, blue, and green-hazel [11]. Some people develop a gray, white, or bluish circle around all or part of the iris called pupil ring as shown in Fig.1.11.

Figure 1.8: SMI Eye Tracking Glasses (ETG) 2 [8].

Single eye camera.

Stereo eye camera.

Figure 1.9: Pupil gaze estimation device [9].

Figure 1.10: Gaze estimation device from Nac image technology. (Left: EMR-9 CAP; Middle: EMR-9 Glasses; Right; EMR-9 Controller) [10]

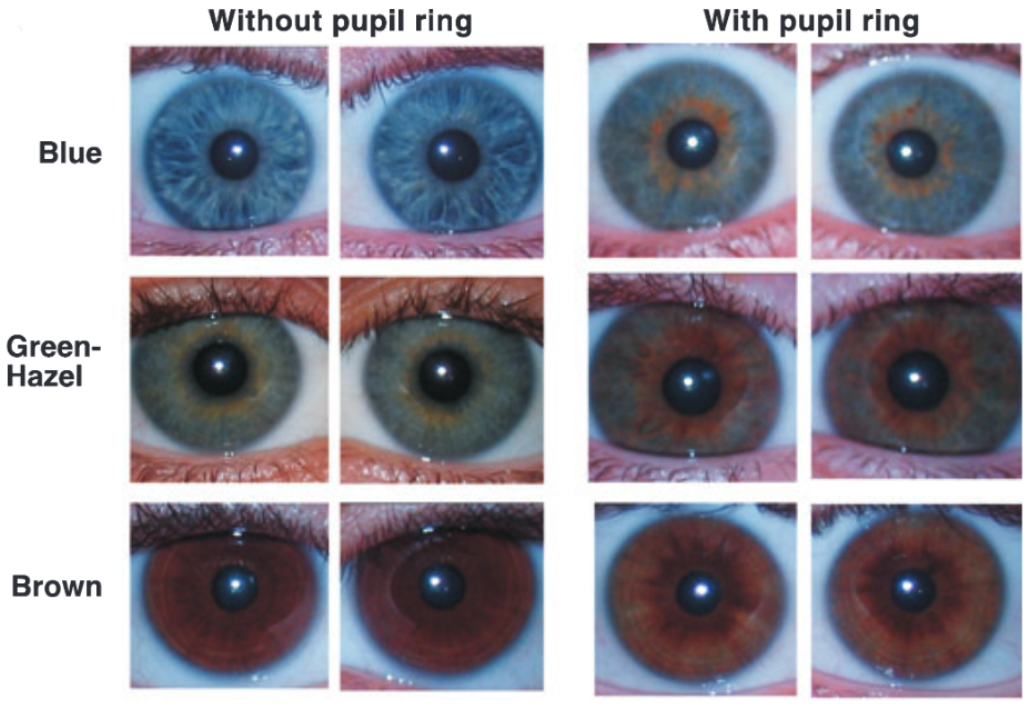


Figure 1.11: Human iris colour classification and patterns [11].

Table 1.1: Specification of commercial gaze estimation device.

Product	Scene camera	Scene camera FOV (Field of view)	Eye detection method	Recording method	SDK/API	Weight
Tobii Pro Glasses 2	25fps@1920x1080	90 ° 16:9	Stereo pupil detection	Internal SD card, Data cable to windows	HTTP Web service	45 g
SMI ETG 2 Observation	24fps@1280x960p, 30fps@960x720p	60 ° horizontal, 46 ° vertical	Stereo pupil detection	Data cable to Windows	MATLAB, PST E-Prime, Python, NBS Presentation, C/C++ .net	47 g
SMI ETG 2 Wireless Observation	24fps@1280x960p, 30fps@960x720p	60 ° horizontal, 46 ° vertical	Stereo pupil detection	Data cable to Windows, Wireless to windows or special portable device	MATLAB, PST E-Prime, Python, NBS Presentation, C/C++ .net	47 g
SMI ETG 2 Wireless Analysis	24fps@1280x960p, 30fps@960x720p	60 ° horizontal, 46 ° vertical	Stereo pupil detection	Data cable to Windows, Wireless to windows or special portable device	MATLAB, PST E-Prime, Python, NBS Presentation, C/C++ .net	47 g
SMI ETG 2 Wireless AnalysisPro	24fps@1280x960p, 30fps@960x720p	60 ° horizontal, 46 ° vertical	Stereo pupil detection	Data cable to Windows, Wireless to windows or special portable device	MATLAB, PST E-Prime, Python, NBS Presentation, C/C++ .net	47 g
Pupil	60fps@720p, 30fps@1080p, 120fps@vga	60 °, 90 °, 100 ° (Lens options)	Mono or Stereo pupil detection	Data cable to PC (Linux, MacOS, or Windows)	Python,C++	35-48 g (depend on option)
nac EMR-9 (CAP)	MPEG4, 640x480	44 ° (standard), 62 °, 92 ° and 121 ° (options)	Mono or Stereo pupil detection	Internal SD card	-	150 g
nac EMR-9 (Glasses)	MPEG4, 640x480	44 ° (standard), 62 °, 92 ° and 121 ° (options)	Stereo pupil detection	Internal SD card	-	75 g

Moreover, the eyelid or eyelashes can occlude the pupil. The Specular reflection point from environment hinders the GES. The position of the wearable device is effected by the user's gazes vector.

The vision-based GES uses the pupil center position from eye image to estimate the gaze point. To find this point, many researchers use the handcraft approach. From the above reason, it is difficult to detect the pupil center position from eye image. The learning-based approach has the capability to detect the object or classify the object from a digital image. Many researchers and the commercial company have participated in the image recognition challenge called ImageNet [16]. The result of this challenge shows the performance of a learning-based approach is to overcome the handcraft approach. The convolutional neural network (CNN) is a famous method and obtain the best accuracy. The CNN has the potential to detect the object with complex shape with high accuracy [42], [43]. From this advantage, this thesis considered the CNN to creating the proposed GES.

Figure 1.12 shows the category of the GES device. The desktop/TV panels is a famous type with the accuracy around three to one degree [51]〜[54]. These researches using the head pose compensated to achieve the good accuracy. This thesis focuses on the monocular base wearable device because of mobility and reasonable cost. However, the eye camera of inside-out camera cannot view the whole head image. Some researches present the GES using the appearance-based method that head pose compensated is not required [55]〜[57]. These approaches can obtain the accuracy around 2.5 degrees. Figure 1.13 shows that many algorithms can be used for GES, The handcraft and learning-based method be considered to this thesis. Our goal is to present the novel accurate CNN framework for GES.

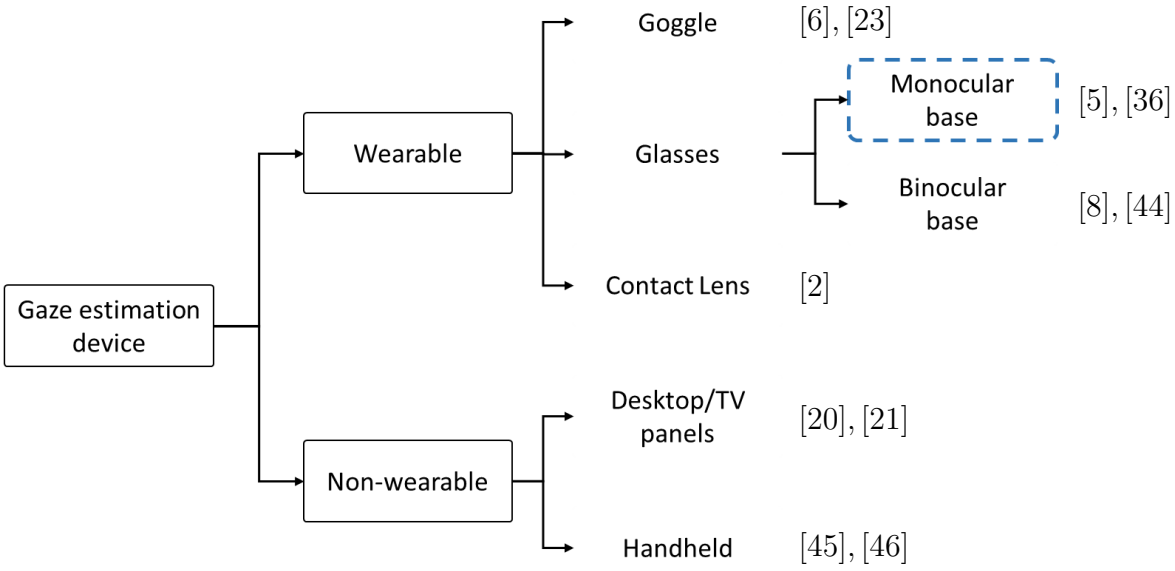


Figure 1.12: Category of gaze estimation device.

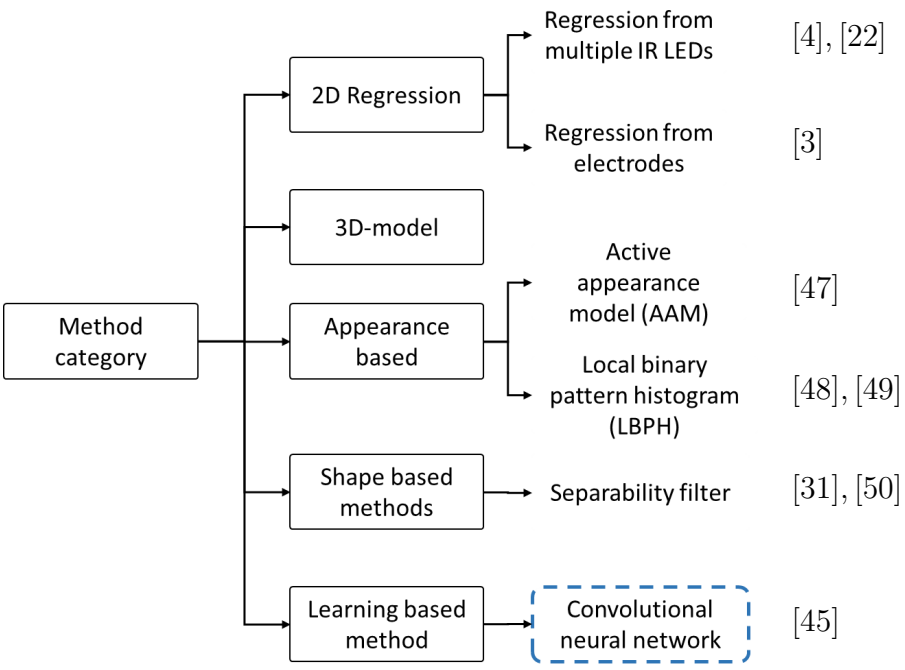


Figure 1.13: Classification of gaze estimation algorithm.

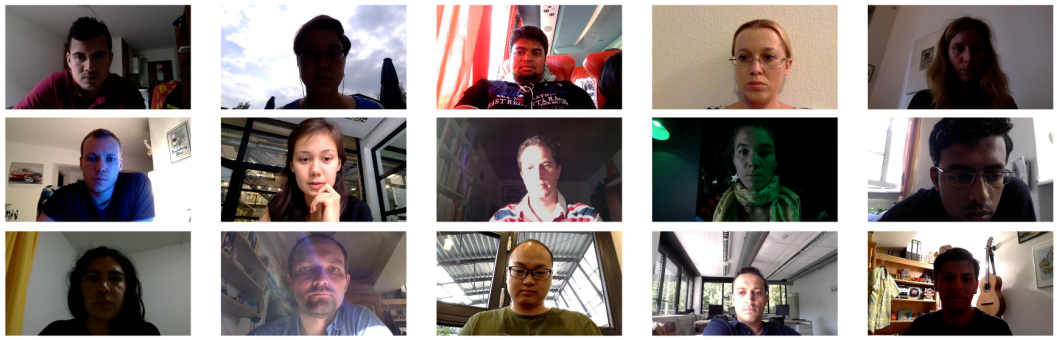


Figure 1.14: Sample image from MPIIGaze dataset [12].

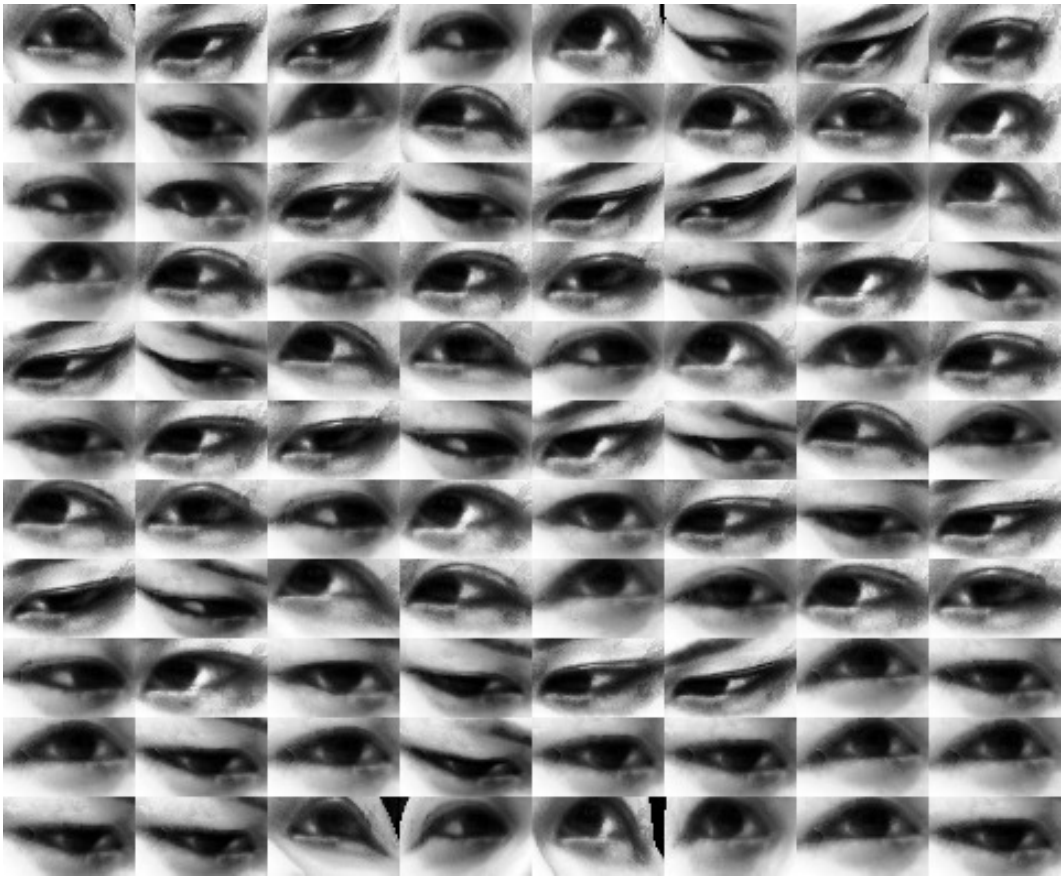


Figure 1.15: Sample image from UT Multiview dataset [13].

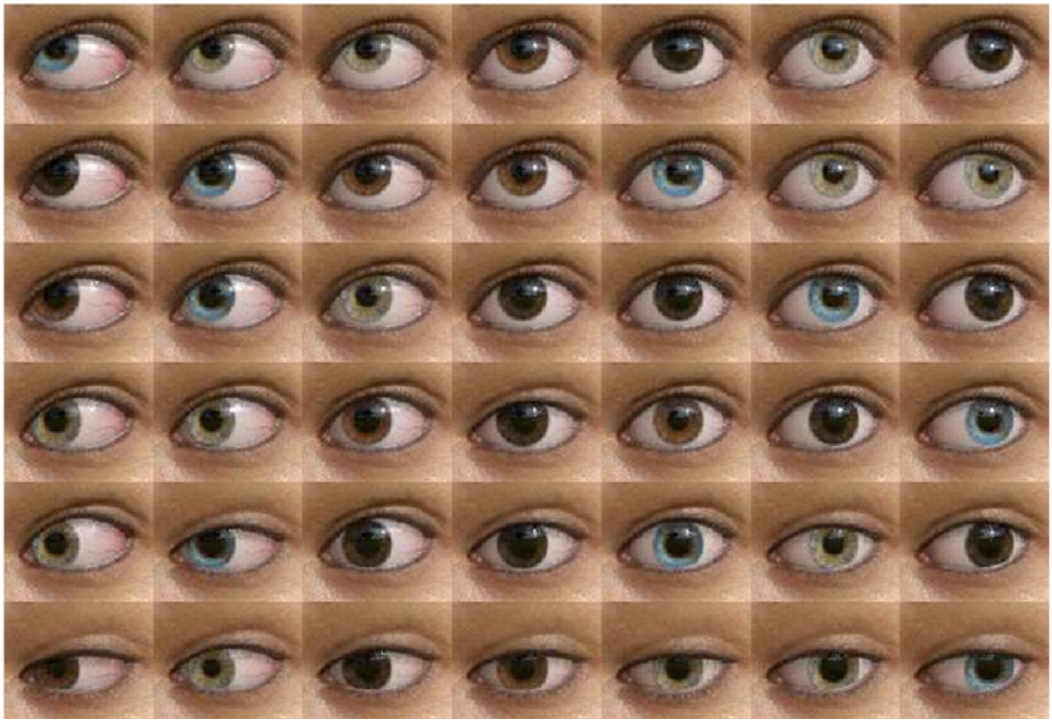


Figure 1.16: Sample image from SynthesEyes dataset [14].

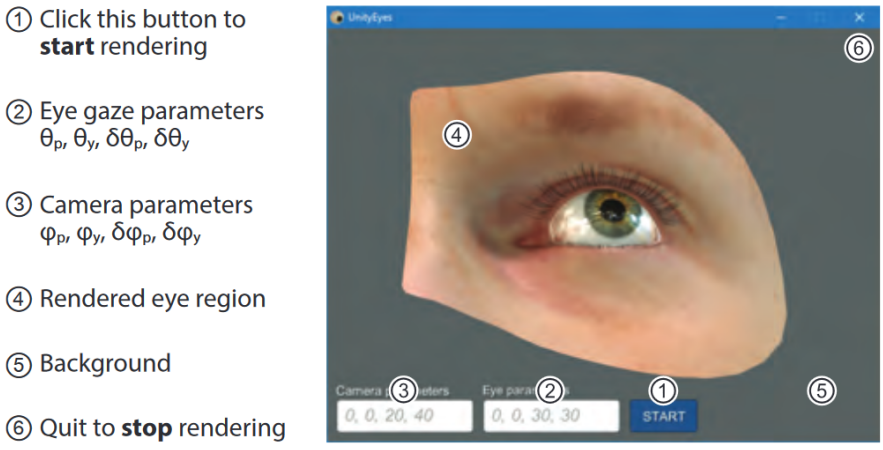


Figure 1.17: UnityEyes application for generate eye image [15].

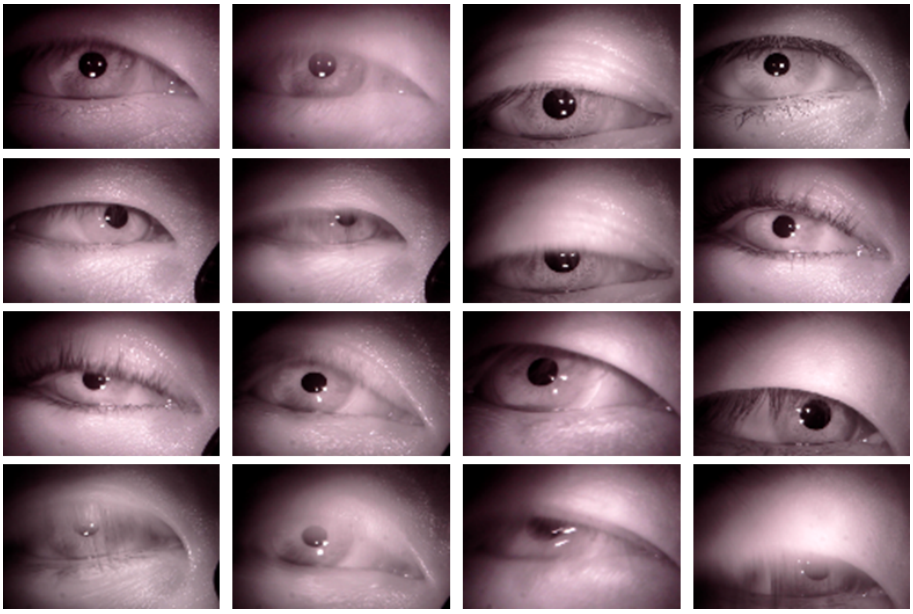


Figure 1.18: Original dataset that capture by our inside-out camera.

1.2 Dataset for gaze estimation

The dataset is very importance for the CNN to achieve the goal. To obtain the good accuracy, the CNN need to learn by using large dataset. For the gaze estimation, Many researchers release datasets to the public for further development.

- The MPIIGaze is present by Zhang et al. in 2015 [12]. This data is a famous dataset for the gaze estimation, it contains 213,659 eye images that capture by 15 subjects. The eye image takes in difference place and time between nine days to three months. The sample of the MPIIGaze image has shown in Fig.1.14.

- The UT Multiview dataset is present by Sugano et al. in 2014 [13]. They collect the largest and fully calibrated multi-view gaze dataset and perform a 3D reconstruction in order to generate dense training data of eye images. The dataset contains 64,000 eye images from 50 subject. The sample of UT Multiview image has shown in Fig.1.15.

- The SynthesEyes dataset is present by Wood et al. [14] in 2015. The image from this dataset are synthesized by 3D modeling. This eye image using the high accuracy 3D head scaner to scan human head. Next, the eye image is rendered by 3D computer application. The ambiance light and eye color are changed to make the variety of the images. This dataset have a high reliability because the ground truth are defined by using the computer. this dataset contain 11,382 images from ten 3D models. The sample of this eye image has shown in Fig.1.16.

- The UnityEyes dataset is present by Wood et al. [15] in 2016. This dataset also be the synthesis image. The UnityEyes researcher provide the computer

application to generate the eye images as shown in Fig.1.17. By random the camera orientation, eye direction, eye opening state, or eye color; this application can generate a million of eye images.

- The original dataset that capture by our device as shown in Fig.1.18. The data mismatch is a common problem when the CNN use the training and testing dataset that have high difference feature. The original dataset is necessary to train the CNN in our environment and specific problem.

1.3 Contributions of the thesis

This thesis focus on creating a high accuracy gaze estimation algorithm and easy to use. Many applications from GES can improve the quality of life for everyone, especially person with a disability. The pupil center position is an important information estimate the gaze point. The main contribution of this thesis can be divided into two main group. The first main contribution is to improve the pupil center detection by creating an effective pupil center detection framework. The system must be able to detect the pupil position with the variety situation and operate in a real-time system.

The handcraft method is a one of shape base algorithm, we present the process to detect the pupil point by using the gradient value and RANSAC method. In light of learning-based method, the CNN model for pupil detection is designed for robust pupil detection. Next, we have to prove the performance of the pupil detection system by creating the application called the character input system. This application helps the physical disorders patient to communicate with speech partner.

The second contribution of this thesis is to create the calibration-free GES. The calibration process is the process to create the coordinate transfer (CT) function. The CT function uses for transfer the pupil position to the gaze point on-scene image. When the wearable camera moving during the use case, the static CT function cannot estimate the gaze point accurately. The learning-based have a potential to create the robust and adaptive CT function. The accurate calibration-free system can improve the accuracy of the GES. Furthermore, it makes the GES easy easier to use and bring the power to GES to disorders patient and everyone.

1.4 Outline of the thesis

Chapter 2 presents the handcraft method for detecting the pupil center point. The pupil center position is a key information to estimate the gaze direction. The gradient value and RANSAC method are used to create the pupil center position and compare with the separability filter method.

Chapter 3 presents the fundamental of CNN. This method is the well-known learning-based method that use many of kernel to detect the specific feature information form image. The CNN saves the knowledge in term of weight and kernel by learning from training data also known as a dataset. Moreover, the public dataset for eye image is shown in this chapter.

Chapter 4 shows the pupil detection by using CNN. The proposed CNN model can estimate the pupil center position and eye state (open, medium, and close eye). The detail of this CNN model is explained in this chapter. Furthermore, The process to collect the dataset and annotation is also presented in this chapter.

Chapter 5 presents another powerful function of the CNN for GES. CNN can use to estimate the gaze direction in different camera angle. This ability makes the CNN useful to creating the GES with the calibration-free system. The chapter described the structure of the CNN model for calibration-free GES. The designed and training method is shown in this chapter.

Chapter 6 presents the character input system. This system is created by using the proposed CNN model for pupil detection. This chapter shows the system overall and algorithm to estimate the gaze point from pupil position. Moreover, the character recognition model is created and shown in this chapter.

Chapter 2

Pupil Detection using handcraft method

2.1 Introduction

This chapter presents a fast and precious eye detection technique by using gradient value for improve the performance of inside-out camera. The propose methods using gradient vector to estimate a temporary pupil position before using ellipse fitting to detect actual eye position.

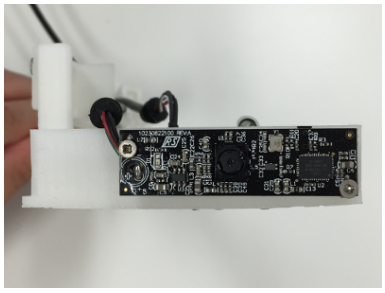
An inside-out camera uses image processing method for estimating the eye position. The advantages of the inside-out camera are high accuracy and low-cost [5]. The detail of inside-out camera will describe in Section **2.2**

Most of the research use the inside-out camera without the glasses. For the user with near-sighted or far-sighted, they need to wear a glasses. When a user is wearing the glasses, it is difficult to detect the pupil. Because the environment light is reflected by glasses. Environment light is a big problem for eye position detection. Because environment disturbs the eye tracking process. This eye detection problem makes the gaze estimation system cannot use on the outdoor operation. We need to implement the method to fix this problem. But the quickly processing time is important for using an inside-out camera in real-time mode.

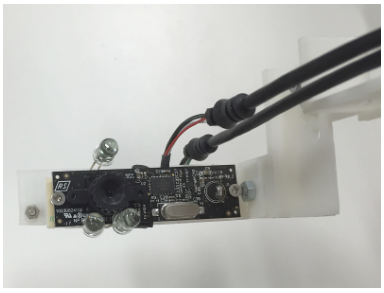
The previous research [58] uses Separability filter [50] to estimate the eye position. The problem of separability filter is the accuracy is low when the user is wearing the glasses. Moreover, the separability filter needs to set the appropriate filter size which depends on the user's pupil size. This chapter presents the new eye detection method by using gradient value with the aim of improving performance for the glasses pupil image.

2.2 Gaze estimation device

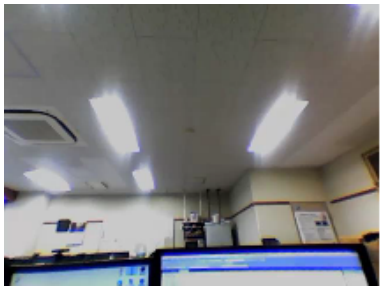
Inside-out camera in this chapter is shown in Fig.2.1. This inside-out camera composed of two USB web cameras. The first camera is called a scene camera as shown in Fig.2.1(a). Scene camera uses for capture the user field view. The image that captures



(a) Scene camera.



(b) Eye camera.



(c) Scene image.

(s) Eye image.

Figure 2.1: Inside-out camera.

from scene camera is called scene image as shown in Fig.2.1(c). The other camera called eye camera as shown in Fig.2.1(b). This camera is a USB camera and the three infrared LEDs to improve the accuracy of the measurements. Infrared LED makes the iris faded and easy to detect the center of the pupil. The image that captured from eye camera is called eye image as shown in Fig.2.1(d).

Before using the inside-out camera, the necessary process is calibration process. The calibration process creates a function to mapping eye position P_e^E into the position of the scene image. This function calls CT function. In the calibration mode, the system is recording the eye position P_e^E and marker position. Both of recording data use for calculating the coefficient of the CT function by using Random sample consensus (RANSAC) method. The marker in calibration mode can be anything such as the user's fingertip, object or marker on the monitor. After calibration mode, the system will run in gaze mode. From this process, the eye detection is an important process to make the gaze estimation work correctly.

2.3 Proposed method

From the previous section, we introduce the component of an inside-out camera and the process overview of gaze estimation. And know how the eye detection process is important. This section will describe the proposed method of eye detection by using gradient value. It is very simple and quick to find the gradient images. The gradient image is useful to find the edge point and can be used to find the pupil's

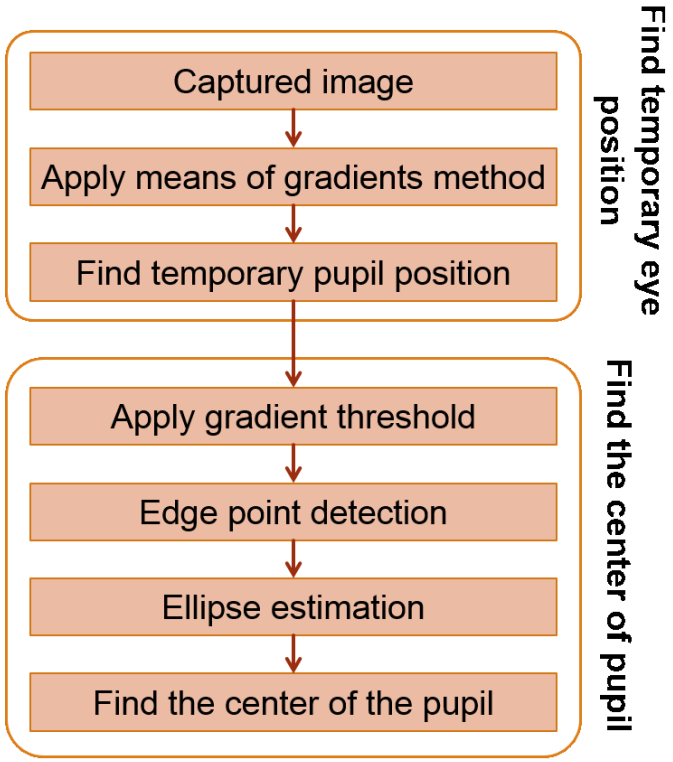


Figure 2.2: Diagram of eye detection.

position. Figure 2.2 has shown the processes of proposed eye detection method. These processes can be subdivided into two sections as follows.

2.3.1 Find temporary eye position

The goal of this research is to detect the center of the pupil in the eye image. In the eye image, the proposed method needs to limit the search area. Hence this section needs to find the temporary eye position to limit the search area around the temporary eye position point. The previous research uses separability filter to find the temporary eye position. From the problem of separability filter in section **2.1**, the proposed method use means of gradients [59] to find the temporary eye position. Because this method is faster than separability filter and does not need to set the filter size like the separability method. The means of gradients method, using the normalize gradient vectors $\hat{g}_i$ to find the pupil position by calculated the height-density value c^* as shown in following equation.

$$c^* = \arg\max_c \left\{ \frac{1}{N} \sum_{i=1}^{N} (\hat{d}_i^T \hat{g}_i)^2 \right\},$$

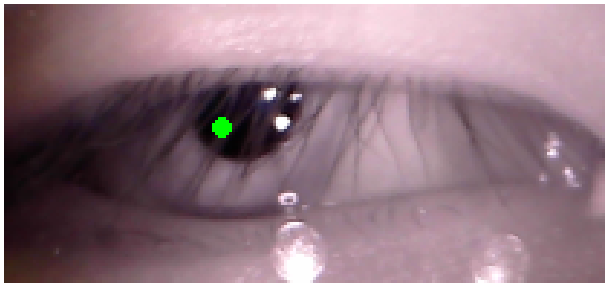


Figure 2.3: Temporary eye position from means of gradients method.

where N is a number of pixel, and g_i is the gradient vector at position x_i. The vector $\hat{d}_i$ can calculate by following equation.

$$\hat{d}_i = \frac{X_i - c}{\| X_i - c \|},$$

where X_i is the center of gradient vector.

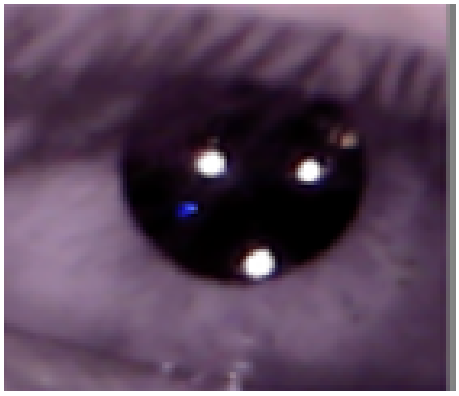
The green point in Fig.2.3 shows the temporary eye position P_c by means of gradients method.

2.3.2 Find the center of the pupil

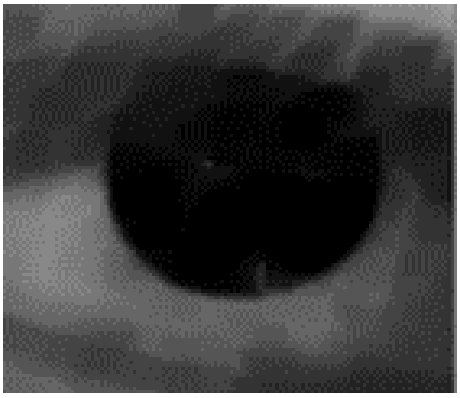
The pupil position from the previous section is not the center of the pupil. Because of, the pupil is not shown on full cycle from eye blink or ambient light as shown in Fig.2.3. After we found the temporary pupil position P_c, the proposed method need to find the center of the pupil by using the regression method to fitting the ellipse equation. To fitting the ellipse equation, so that, the next process is detection the pupil's edge point. We can simply use the gradient image to detect all of the edge points in the image. Because of the gradient image is the slope of the image, the slope of the image mean the edge image. But these points include most of the unwanted points. Therefore, the following method focuses on detecting only the pupil's edge points.

1. The infrared LEDs of eye camera make the reflection point in eye image. This point disturbs the eye tracking process. This research uses the erosion method to reduce the effect of specular reflection. The erosion method erode the white point in the image as shown in Fig.2.4. In Fig.2.4, Fig.2.4(a) is an original eye image, and Fig.2.4(b) is an image after applying erosion method.

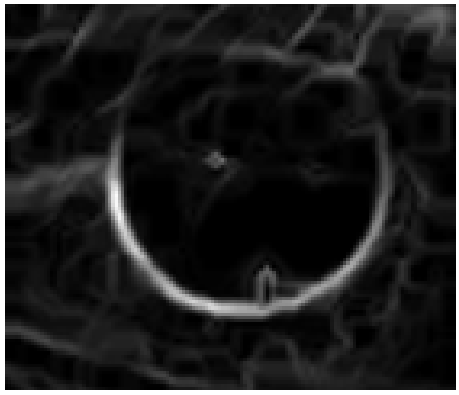
2. Calculate the gradient image as shown in Fig.2.4(c) from erosion image. This gradient image uses for finding the temporary edge point $\vec{P}^*$. It is not necessary to find all of the edge points in the image because it can make the unwanted edge point such as the edge point from the eyelid. So that, the proposed method scans the image by using polar coordinates and use the temporary eye position $\vec{P}_c$ to be the center point. Using the polar coordinates makes the edge point scope inside the pupil.



(a) Original image.



(b) Processed image.



(c) Gradient image.

Figure 2.4: Eye image from erosion method.

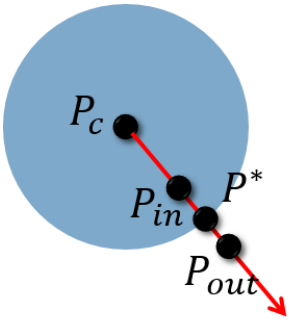


Figure 2.5: Edge point detection by using gradient value.

3. This step needs to confirm the temporary edge point. The idea of this method is the intensity of the pupil much lower than outer part. We can define the inside pupil point $\vec{P}_{in}$ and outside pupil point $\vec{P}_{out}$ by using the temporary pupil position $\vec{P}_c$. First, calculate the $\vec{P}_{in}$ and $\vec{P}_{out}$ by the following equation.

$$\vec{P}_{out} = (\vec{P}^* - \vec{P}_c) \times (1 + \sigma) \tag{2.1}$$

$$\vec{P}_{in} = (\vec{P}^* - \vec{P}_c) \times (1 - \sigma) \tag{2.2}$$

   By σ is a coefficient of pupil's edge distance. The $\vec{P}_{in}$ and $\vec{P}_{out}$ showed in Fig.2.5. If intensity at $\vec{P}_{in}$ lower than $\vec{P}_{out}$, we can confirm the $\vec{P}^*$ is the edge point.

After edge points detection process, next step is the calculation the center of the pupil by using regression method. This research uses the RANSAC method to fitting the ellipse model. This method can ignore the error from edge point detection process. The ellipse equation in polynomial form can be defined as following.

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

This equation composes of six degree-of-freedom (a, b, c, d, e, and f) and representing in the (x, y) plane. To find the solution of this equation we need to sample the six of pupil's edge point as shown in the following equation.

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ x_2^2 & 2x_2y_1 & y_2^2 & 2x_2 & 2y_2 & 1 \\ x_3^2 & 2x_3y_1 & y_3^2 & 2x_3 & 2y_3 & 1 \\ x_4^2 & 2x_4y_1 & y_4^2 & 2x_4 & 2y_4 & 1 \\ x_5^2 & 2x_5y_1 & y_5^2 & 2x_5 & 2y_5 & 1 \\ x_6^2 & 2x_6y_1 & y_6^2 & 2x_6 & 2y_6 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = 0$$

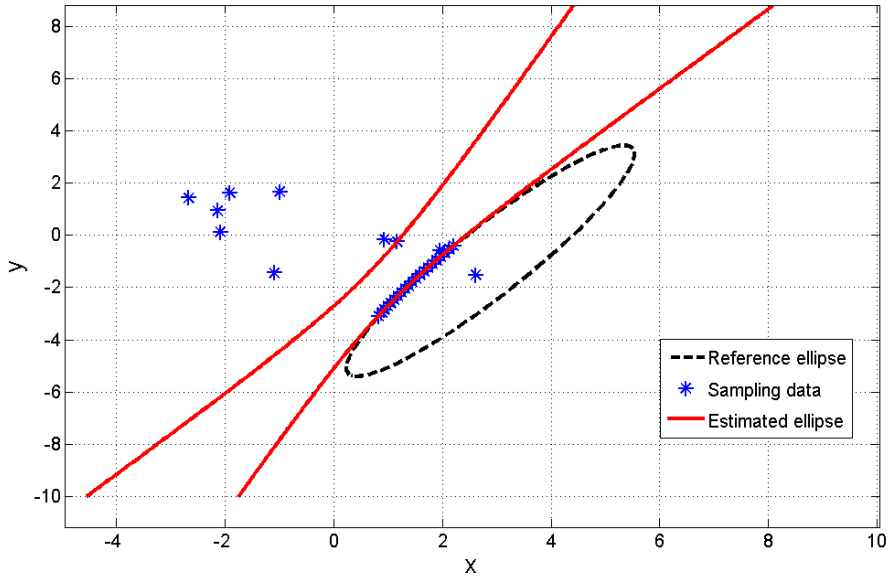
In a specific condition such as the pupil is not shown in full cycle or the edge points detection process can detect a part of the cycle. From this problem, the ellipse equation model can be the hyperbola or parabola as shown in Figs. 2.6(a) and 2.6(b). The cycle in Figs. 2.6 is the reference ellipse model, the point is sampling data and addition noise point, the light line is the estimated model from RANSAC. To fix this problem, the following condition must be added to the RANSAC process.

$$\begin{vmatrix} a & b \\ b & c \end{vmatrix} > 0$$

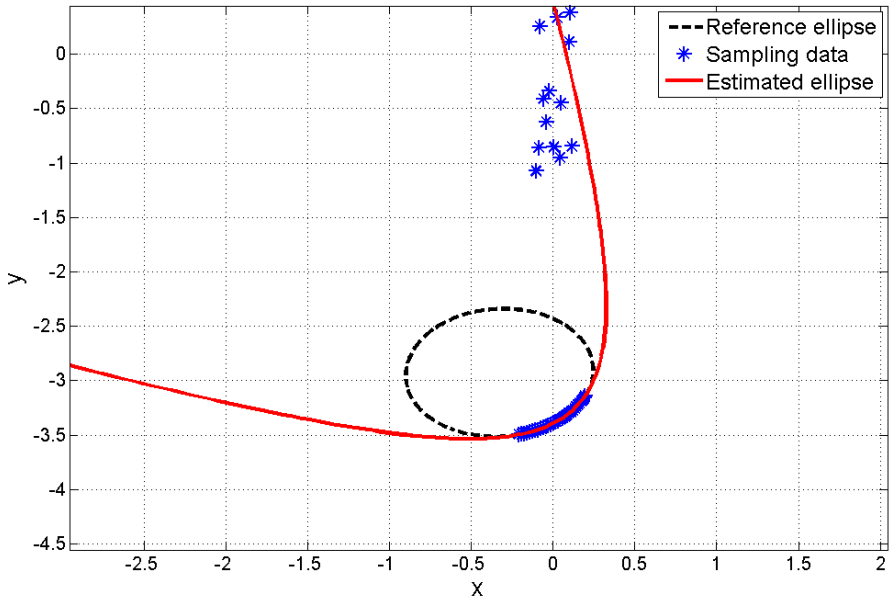
After found the best model, The center of the pupil $P_{pc} = (x_0, y_0)$ can calculate by the following equation.

$$x_0 = \frac{cd - be}{b^2 - ac}, \qquad y_0 = \frac{ae - bd}{b^2 - ac}.$$

Figure 2.7 has shown the result of the proposed method. The small points in the image are the edge points detection, the white cycle is the best ellipse model from RANSAC method, and the big point inside the cycle is the estimate pupil's center position P_{pc}.



(a) Hyperbola.



(b) Parabola.

Figure 2.6: RANSAC ellipse fitting.

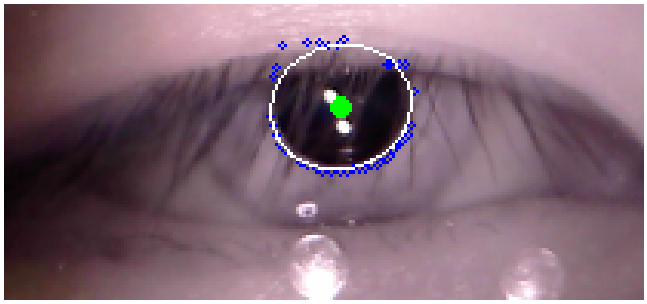


Figure 2.7: Pupil's center position result.

Table 2.1: Number of dataset.

Type	Without glasses image	Glasses image
Clear pupil images	109	48
Bleary pupil images	87	36
Closed eye images	14	6
Sum	210	90

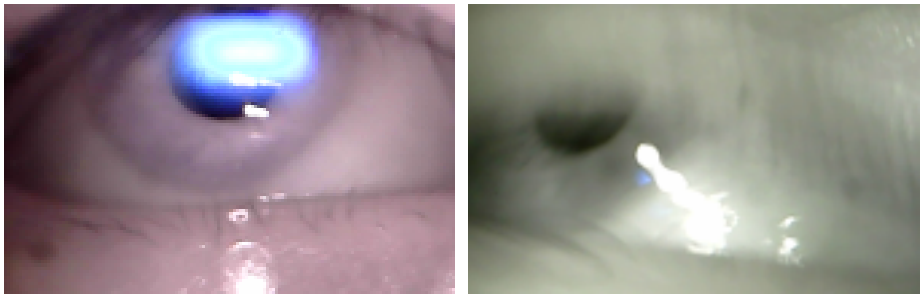
2.4 Dataset

The dataset using in this research is eye images that capture from our inside-out camera. The image size from eye camera is 320 × 240 pixel. After that, we crop the original image only the eye region. When user wears the inside-out camera and blink his eye, the system detects the blink region and defines as the eye region. This image call region of interest (ROI) image as shown in Fig. 2.8. We capture the eye image from ten peoples with various eye and pupil size and wearing glasses or not wearing glasses. The number of images has shown in Table 2.1. The data set composed of the clear pupil image, bleary pupil image and closed eye image. The clear pupil image is the image has shown the pupil in the full cycle and clearly pupil's edge as shown in Fig.2.8(a). The bleary images occur from any reason, for example, eye blinking, eye moving, or the reflection point of glasses. The bleary is the image does not show the pupil in the full cycle and bleary pupil's edge as shown in Fig. 2.8(b). The closed eye image is the image without showing the pupil as shown in Fig. 2.8(c).

2.5 Experiments

We compared the performance between proposed method and separability filter method [50]. This system ran on a personal computer with CPU Intel Core i7-3820, 3.60GHz, RAM 16GB. This experiment is divided into two parts. First parts are considered the performance of the temporary pupil detection P_c process. The second parts are considered the performance of the center of pupil P_{pc} detection process.

(a) Clear pupil image.



(b) Bleary pupil image.

(c) Closed eye image.

Figure 2.8: Dataset image (left: without glasses image, right: glasses image).

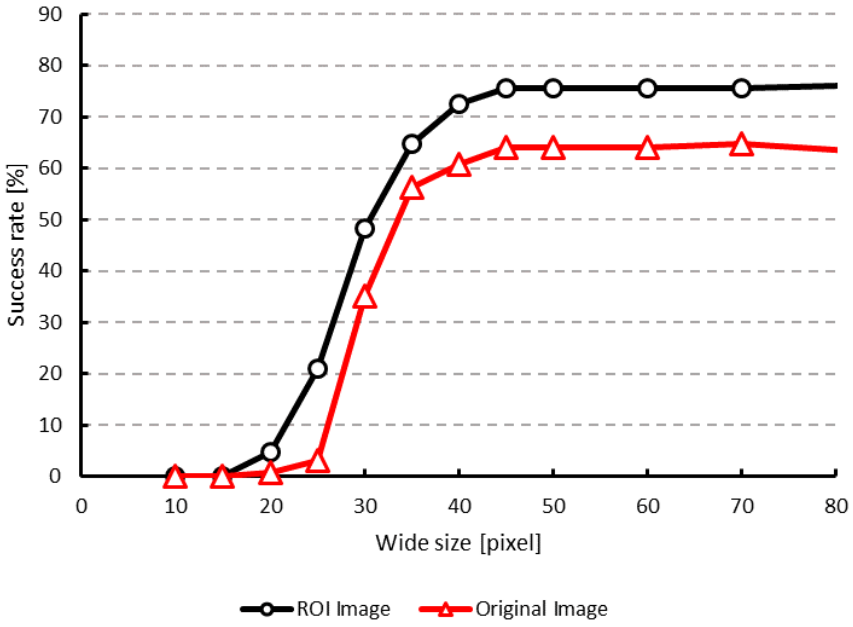


Figure 2.9: Mean of gradient success rate.

2.5.1 Performance of temporary pupil detection

In temporary pupil detection, the proposed method applied the means of gradients method. In technicality, the processing time of separability filter was faster than the means of gradients method in same image size. But the advantage of the means of gradients method is can detect the pupil in very small image size. In the temporary pupil detection process, we can resize the image to speed up the process time. From this reason, we need to consider the effect of image size. Furthermore, the proposed method uses ROI image for pupil detection. Therefore, the performance between the ROI image and original image has been considered by this experiment.

Figure 2.9 has shown the success rate between ROI images and original images in difference image size. The success rate of ROI image is higher than the original image. The success rate of the original images is decreased when the image's size increase. Because when image size increases the means of gradients method can detect more of the unwanted gradient vector. Figure 2.10 has shown the process time between ROI images and original images. The process time of ROI image is faster than original image because of ROI image's size is smaller than the original image. From this result, in the temporary pupil detection process, we choose 60-pixel wide image size. Because this image's size is appropriate in term of process time and success rate.

After we select the means of gradients image resize. The next process is the comparison of the performance between the means of gradients method and separability filter [50]. The results has shown in Table 2.2. The process time in this table is the average processing time of 300 images. This result shows the success rate of means of gradients method is 75.66%, on the other hand, the separability filter is 46.80%. Fur-

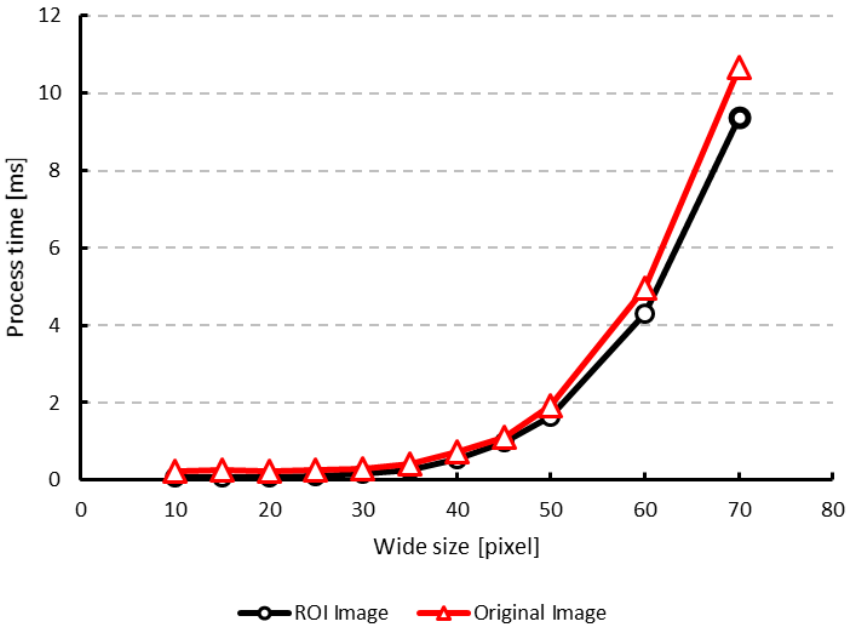


Figure 2.10: Mean of gradient process time.

Table 2.2: Performance in term of temporary pupil detection.

Method	Process time [ms]	Success rate [%]
Mean of gradient	4.21	75.66
Separability filter [50]	18.12	46.80

thermore, the process time of the means of gradients method is faster than separability filter.

2.5.2 Performance of the pupil center detection

This section we will evaluate the performance of the pupil center detection. The previous method [58] uses the region growing and label region for ellipse fitting. The proposed method find the pupil's center by using RANSAC method. The process time of RANSAC method depends on the number of maximum iteration. In this experiment, the maximum iteration of RANSAC method is 500 iteration and threshold value is 0.02.

Figure 2.11 has shown the comparison between actual center point and temporary pupil position. The green point is the estimation center point from RANSAC method. The red point is the temporary pupil position. The temporary pupil position often not in the same position with the center point. This experiment defines the acceptation length is 5 pixels (the average size of the pupil in the dataset is around 45 pixels). The acceptation length is the length between estimation points to ground truth point.

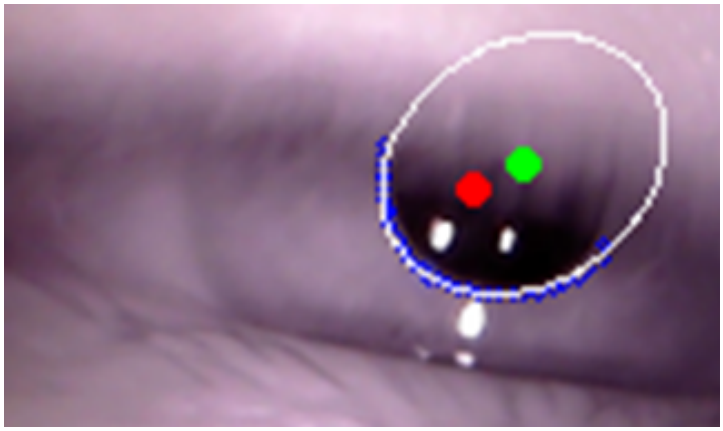


Figure 2.11: Comparison between actual center point (green) and temporary pupil position (red).

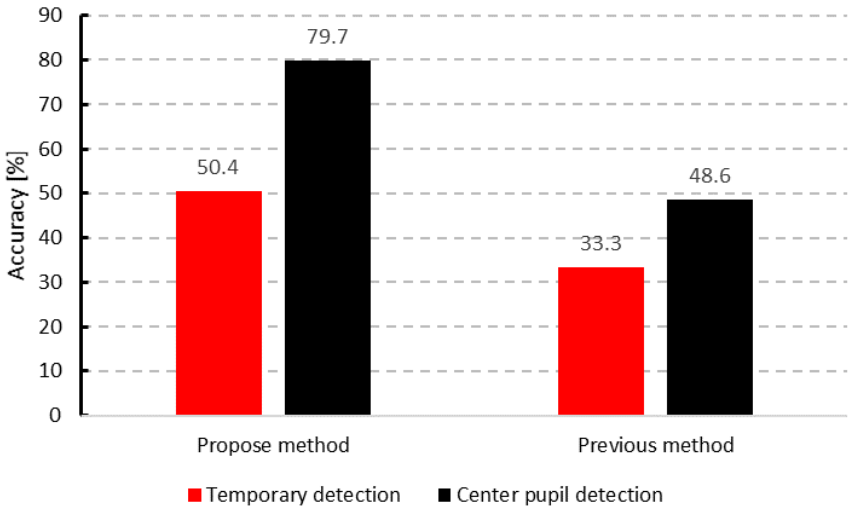
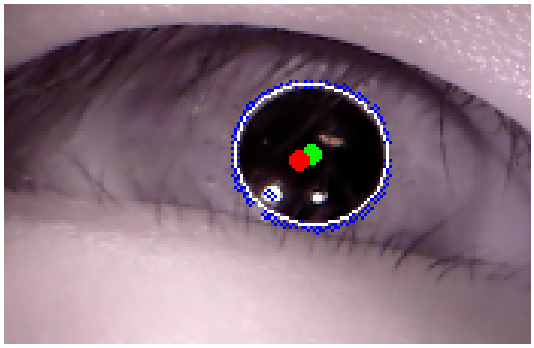
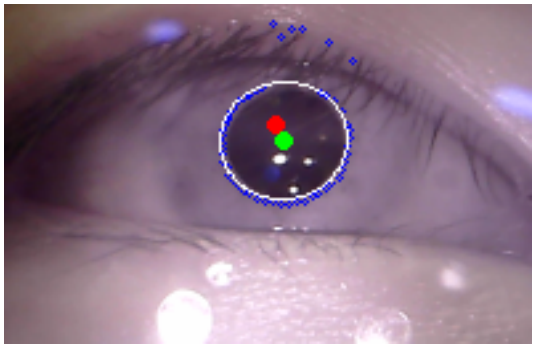


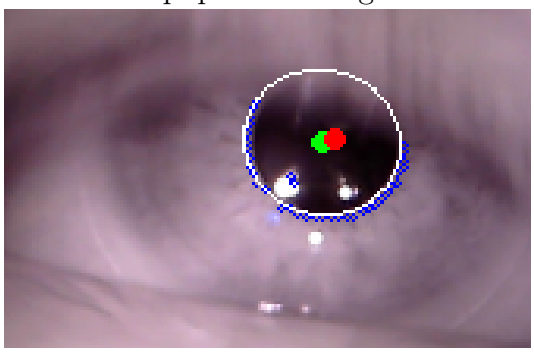
Figure 2.12: Accuracy result of every step.



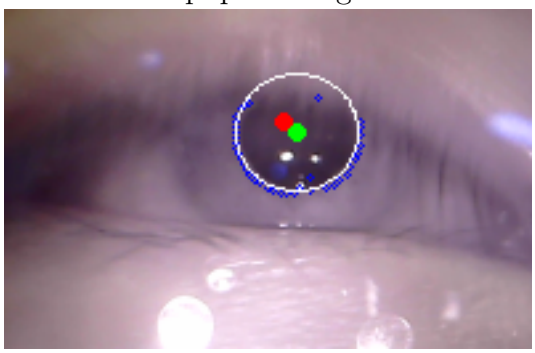
Clear pupil without glasses.



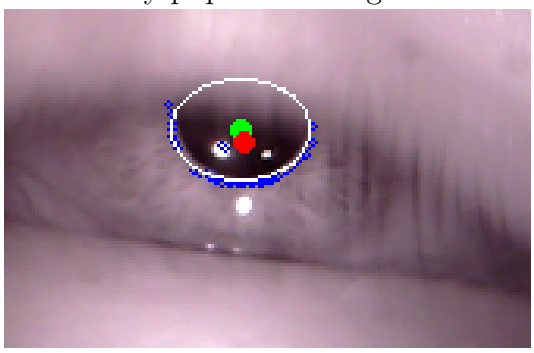
Clear pupil with glasses.



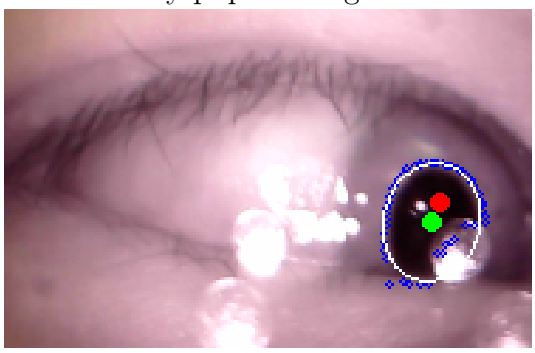
Bleary pupil without glasses.



Bleary pupil with glasses.



Bleary pupil without glasses.



Bleary pupil with glasses.

Figure 2.13: Result of eye detection.

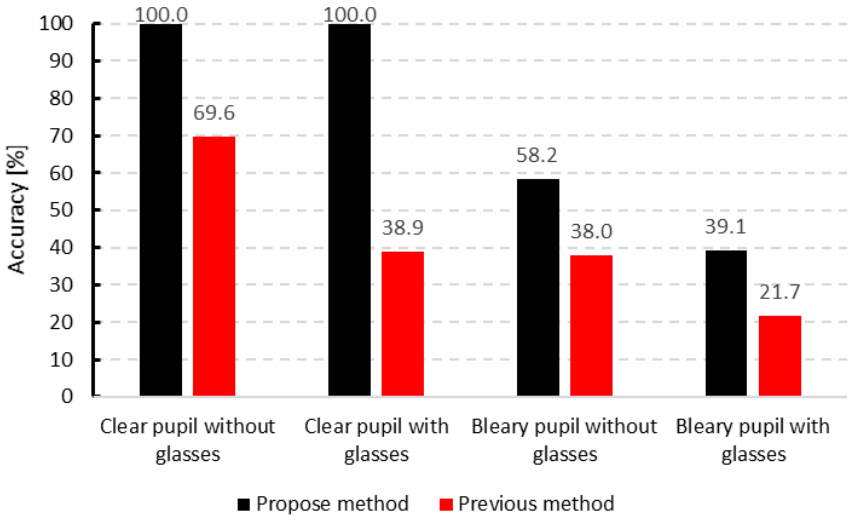


Figure 2.14: Comparison of success rate between proposed method and separability filter.

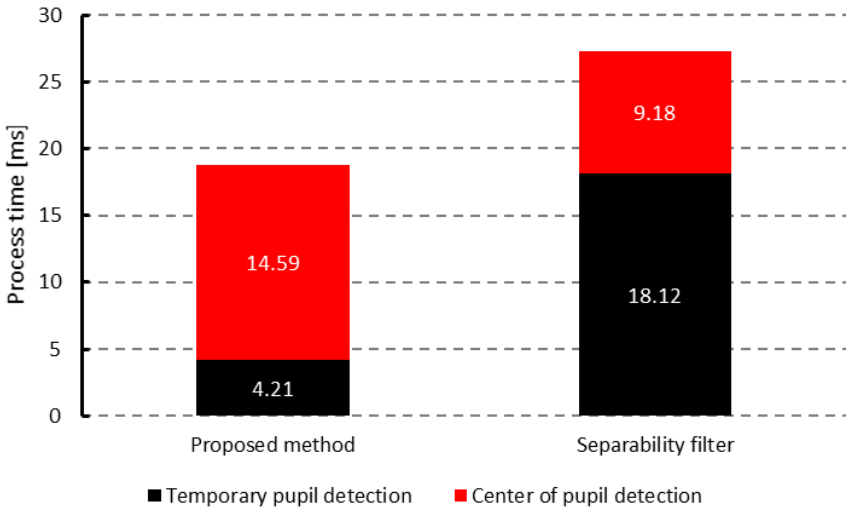


Figure 2.15: Comparison of process time between proposed method and separability filter.

The accuracy of every process has shown in Fig. 2.12. This result shows the accuracy of temporary pupil position detection process is low. Hence, it is necessary to apply the pupil center detection process.

Figure 2.13 has shown the final result of eye detection. Figure 2.14 has shown the comparison of accuracy between proposed method and previous method. In term of clear pupil images, proposed method can detect the pupil very well. In term of glasses image, the accuracy of the proposed method is better than previous method. For bleary pupil image, our method cannot detect the edge of the pupil in full cycle. As a result, the number of unwanted points more than detected edge points. Consequently, the ellipse fitting process by using RANSAC method cannot estimate the correct model. Therefore, the accuracy of bleary pupil image is lower than clear image. However, the accuracy of proposed method is higher than previous method in every condition of dataset. From this result, the proposed method have more accuracy than previous method and suitable when the user wearing the glasses. Figure 2.15 has shown the comparison of process time. The average processing time of RANSAC method is 14.6 milliseconds. The process time of RANSAC is more than the previous method. But total process time is faster than the previous method.

2.6 Conclusion

This chapter presents the eye detection method that can improve the performance of the inside-out camera. These experiments show the proposed method have more performance than the previous method. In the case of using an inside-out camera in an indoor environment or without the wearing the glasses, the bleary image is occasional to occur. Separability filter is able to detect the pupil in clear pupil image. But when the user wearing the glasses, the bleary are frequent to occur. This research proves the proposed method have the ability to operate when the user wearing the glasses. Feature more, the process time is faster than separability method. To operate in real-time mode, it is possible to add more image processing method to improve the accuracy of eye detection such as a specular-free method.

Chapter 3

Convolutional neural network

3.1 Overview of Convolutional Neural Network

The convolutional neural netowrks (CNN) are very famous for image classification. The CNN is a learning-based method that needs more data to train the model. This method obtains the good result on ImageNet [16] challenge. Recently, many researchers present the new architecture of the CNN that obtain good accuracy in a specific problem. The General artificial neural network did not design to classify the image data. The CNN use the convolution method to detect the feature point and sent that feature to the artificial neural network. Figure 3.1 shown the general architecture of CNN model. The input images send to the convolution layer to detecting the feature information. Next, the feature is down-scale by sub-sampling layer and pass that value to the next convolutional layer. This process repeats until the feature size is small enough to be the input of the artificial neural network layer also known as a fully connected layer.

3.2 Fully connected layer

The fully connect layer also known as the multi-layer perceptron (MLP) or artificial neural network. The general definition of the MLP is a mathematical modeling tool. The standard MLP model the neurons in a biological brain. The neuron is a basic component of MLP as shown in Fig. 3.2. The neuron is composed of the connection weight and activation function $\phi(v)$. The artificial neural network is composed of multi neural that connected as shown in Fig. 3.3.

3.2.1 Connection weight

The connection weight model the synaptic of the brain. In the biological, the synaptic is a structure that permits a neuron (or nerve cell) to pass an electrical or chemical signal to another neuron or to the target effector cell. The mathematic model of synaptic simplifies by multiply the output of other neural with the weight. The artificial neural network has stored the knowledge from the learning process in the weight of

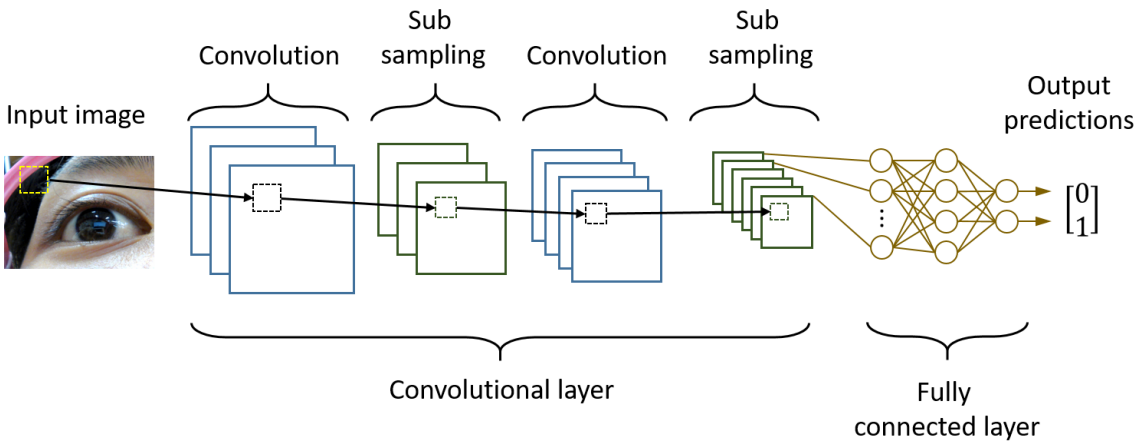


Figure 3.1: Structure of convolutional neural network.

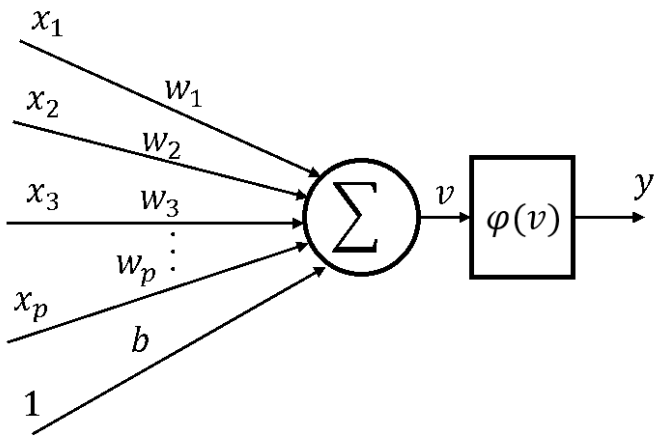


Figure 3.2: Mathematical model of neuron.

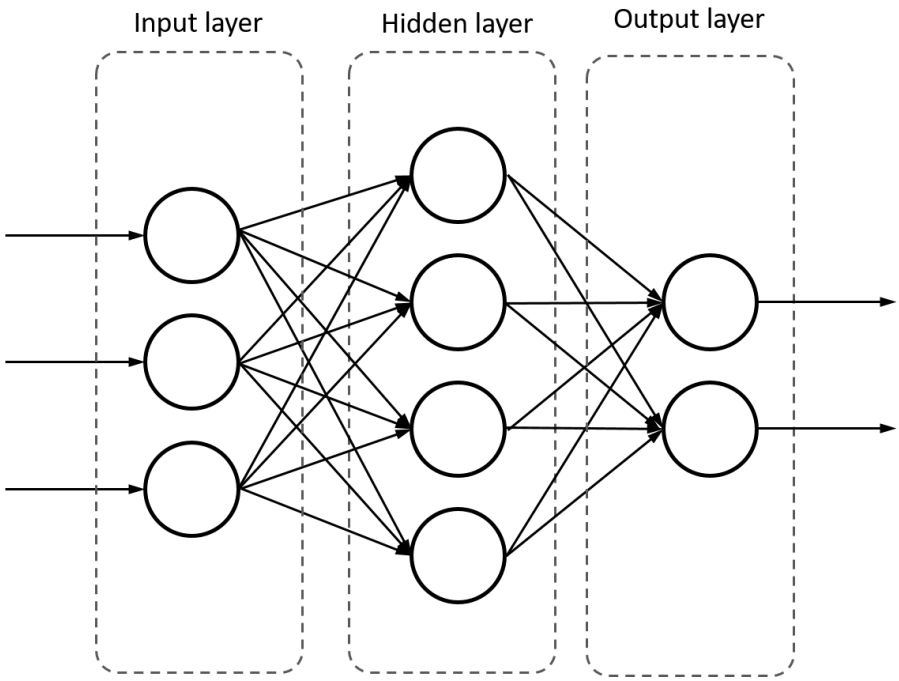


Figure 3.3: Fully connected layer.

the model. The learning process used for transfer the weight from initial state to the final state. The initial state is the state when the random weight was assigned to each connection. The equation of the connection weight has shown as following:

$$v = \sum_{i=1}^{p} w_i x_i + b$$

The function V called net internal activity level, by w_i is a connection weight of the neuron i. The x_i is the input of the node i and p is the number of the input node.

3.2.2 Activation function

The activation function used for convert the net internal activity level v to the neuron output. The activation function can be the linear function, for example, the amplification function or scaling function. Moreover, the non-linear function can be the activation function, for example, the logistic function or exponential function. The famous activation function has shown in the following:

1. The first famous activation is sigmoid function. The mathematical equation of sigmoid functions shown as follows:

$$y = \text{sigmoid}(v) = \frac{1}{1 + e^{-v}}.$$

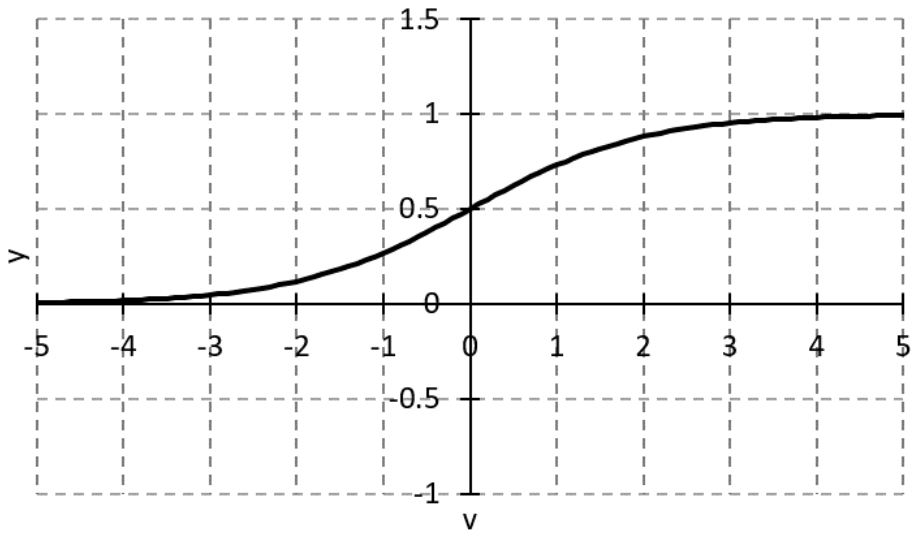


Figure 3.4: Sigmoid function.

When the input is very small, the output of a function is very close to zero. When the input is very large, the output of the function is close to one. In-between these two extremes, the function assumes an S-shape, as shown in Fig. 3.4.

2. The hyperbolic tangent (tanh) use a similar kind of S-shaped nonlinearity, but instead of ranging from zero to one, the output of this function range from minus one to one. The mathematical equation of *tanh* shown as follows:

$$y = \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}.$$

The relationship between the output and input of this function is shown in Fig. 3.5

3. ReLU or restricted linear unit function is shown as follows:

$$y = \text{ReLU}(v) = \begin{cases} v, & \text{if } v > 0 \\ 0, & \text{otherwise} \end{cases}$$

This function is a different kind of nonlinearity activation function. The ReLU has recently become the most choice activation function for the computer vision or regression problem. The relation between the output and input of ReLU is shown in Fig. 3.6

4. The soft plus function is smooth of ReLU. The transition between zero has replaced with smoothness as shown by Fig. 3.7. The equation of soft plus is shown as follows:

$$y = \text{softplus}(v) = \log(1 + e^v).$$

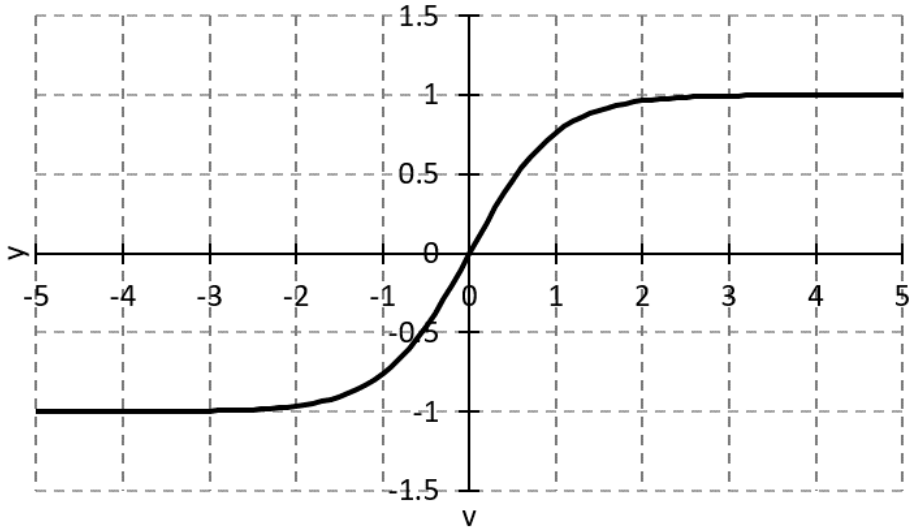


Figure 3.5: Hyperbolic tangent.

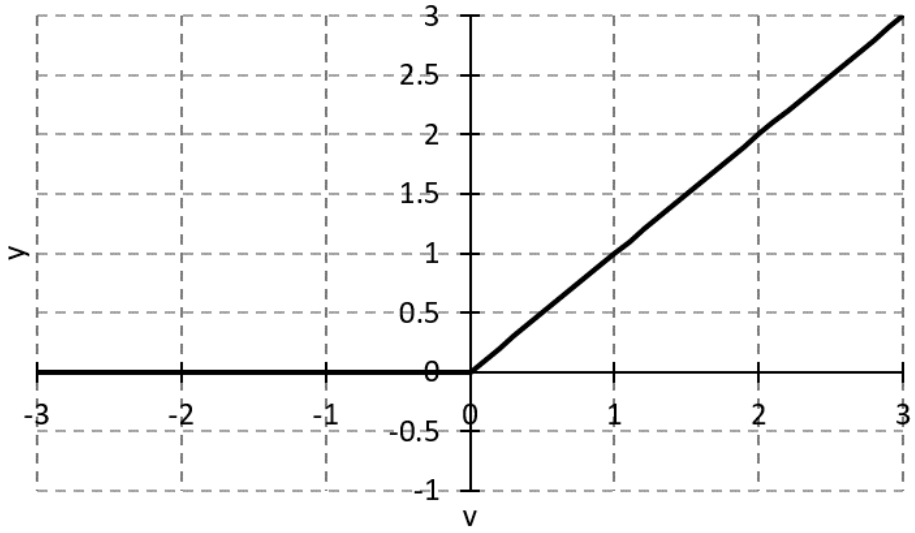


Figure 3.6: ReLU function.

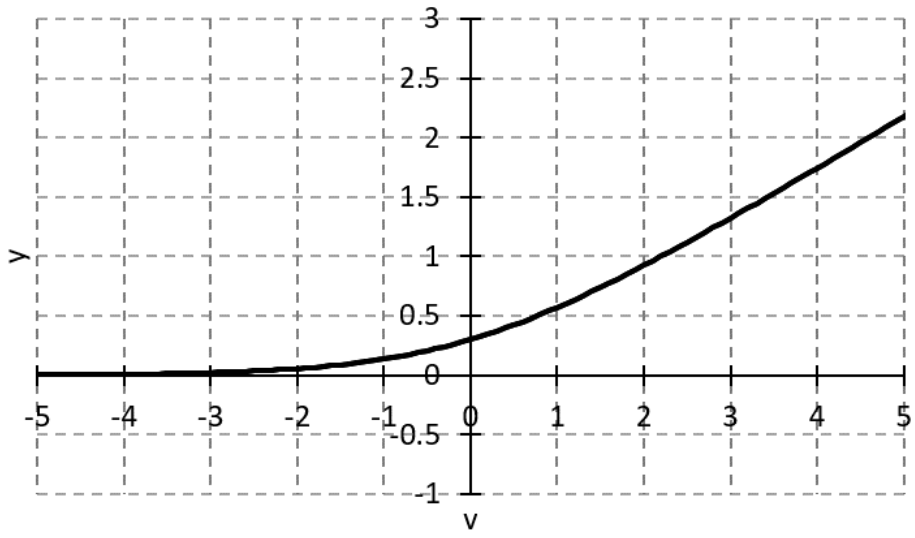


Figure 3.7: Soft plus function.

3.2.3 Learning process

The initial state of the neuron network is a random weight. The learning process makes the weight connection to save the knowledge. The learning process of an artificial neuron network has many categories. The supervised learning is a most popular for the classification or regression problem. The supervised learning or error-correction learning use the external teaching data to train the model. This learning process tries to minimize the error signal of the model by modifying the weight connection. The error signal is the differences between the output of the initial state and teaching data. The well-known algorithm to minimize the error signal is the gradient descent method. By error-correction the output layer and continue to hidden layer to the input layer, this process also known as backpropagation.

3.3 Convolutional layer

3.3.1 2D Convolution

The convolution ($*$) is a mathematical operation on two functions to produce a third function. The basic convolution is generally used in signal processing to creating the filter. The 2D convolution is widely used for a filter in the digital image processing. The convolution result of image f and image g at pixel (x, y) as shown at the following equation:

$$y(x,y) = f(x,y) * g(x,y) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f(n_1,n_2) \cdot g(x-n_1, y-n_2).$$

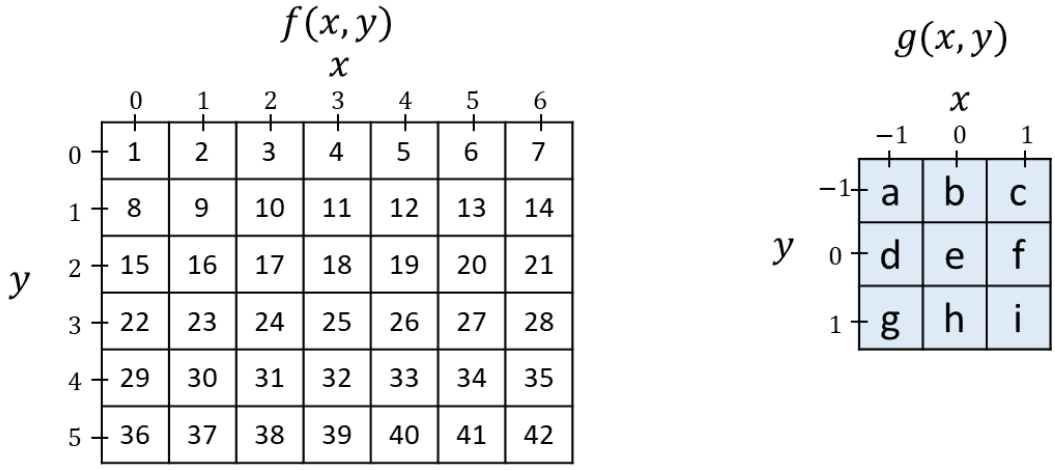


Figure 3.8: Sample data of the input image $f(x, y)$ and the kernel $g(x, y)$.

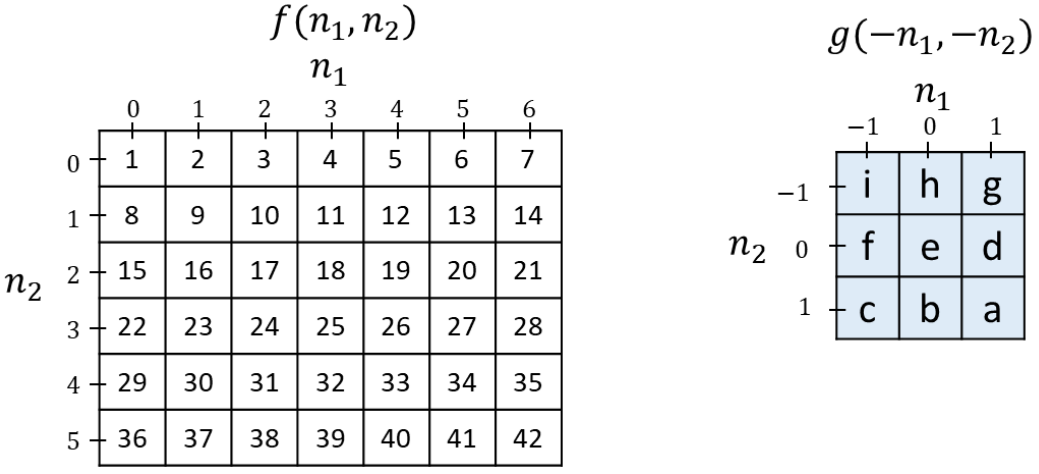


Figure 3.9: Transferring the image to n and kernel to $-n$.

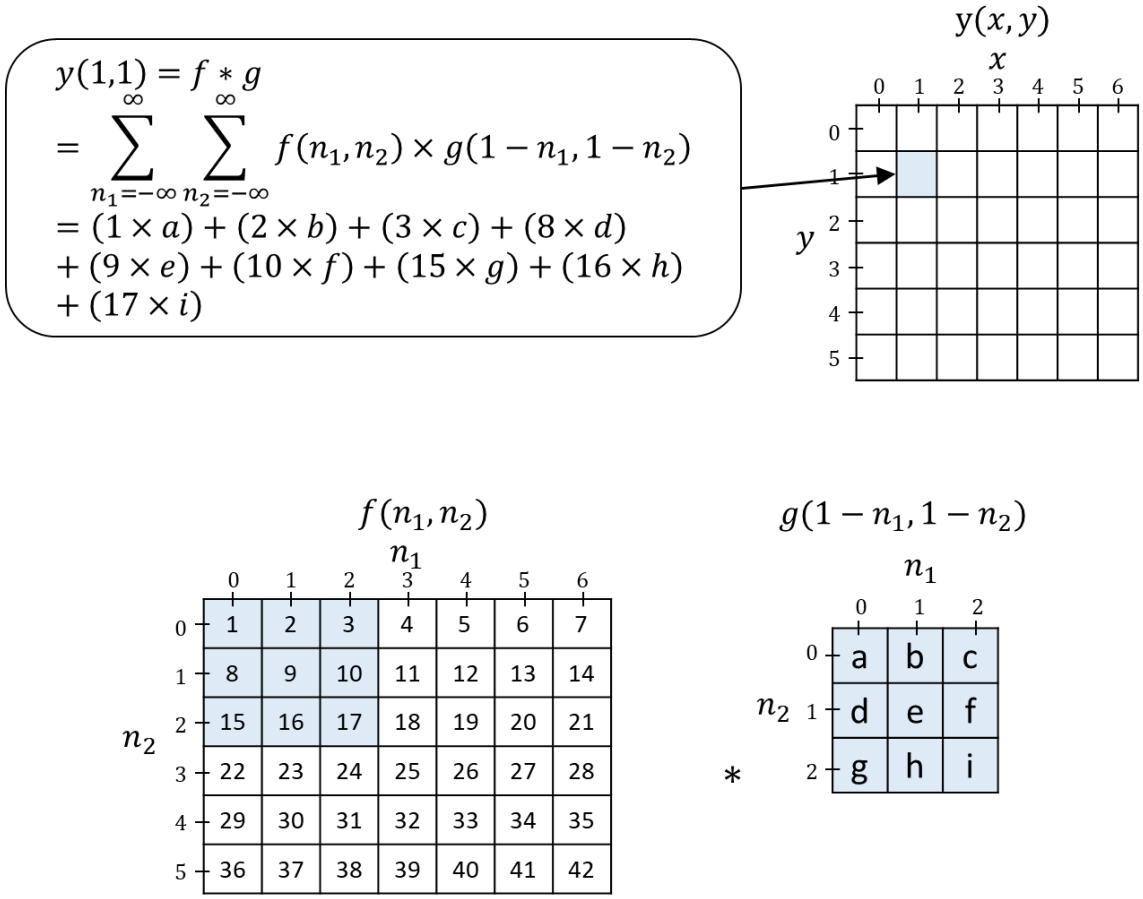


Figure 3.10: Convolution result at point $y(1, 1)$.

For example, the image f and kernel g shown as the Fig.3.8. To make the data easy for convolution process, the image $f(x, y)$ transfer to $f(n_1, n_2)$ space and the kernel $g(x, y)$ transfer to $g(-n_1, -n_2)$ as shown in Fig.3.9.

Figure 3.10 shows the convolutional result at the pixel $x = 1, y = 1$. The general convolution process decreases the output image's size because of the intersection size of two data. However, most CNN model uses the padding algorithm to make the output images' size the same as the input image. By assuming every out of bound pixel is zeros as shown in Fig. 3.11.

As we mentioned above, the application of the 2D convolution is used for an image filter. This method can create the image effect, for example, sharpen, edge detection or blur image by convolution the input image with the kernel as shown in Fig. 3.12.

3.3.2 Convolutional layer

The objective of the convolutional layer is detecting the significant feature of the input image. The appropriate filter also known as the kernel can detect the specific feature point from the input image. The structure of the convolutional layer is similar to

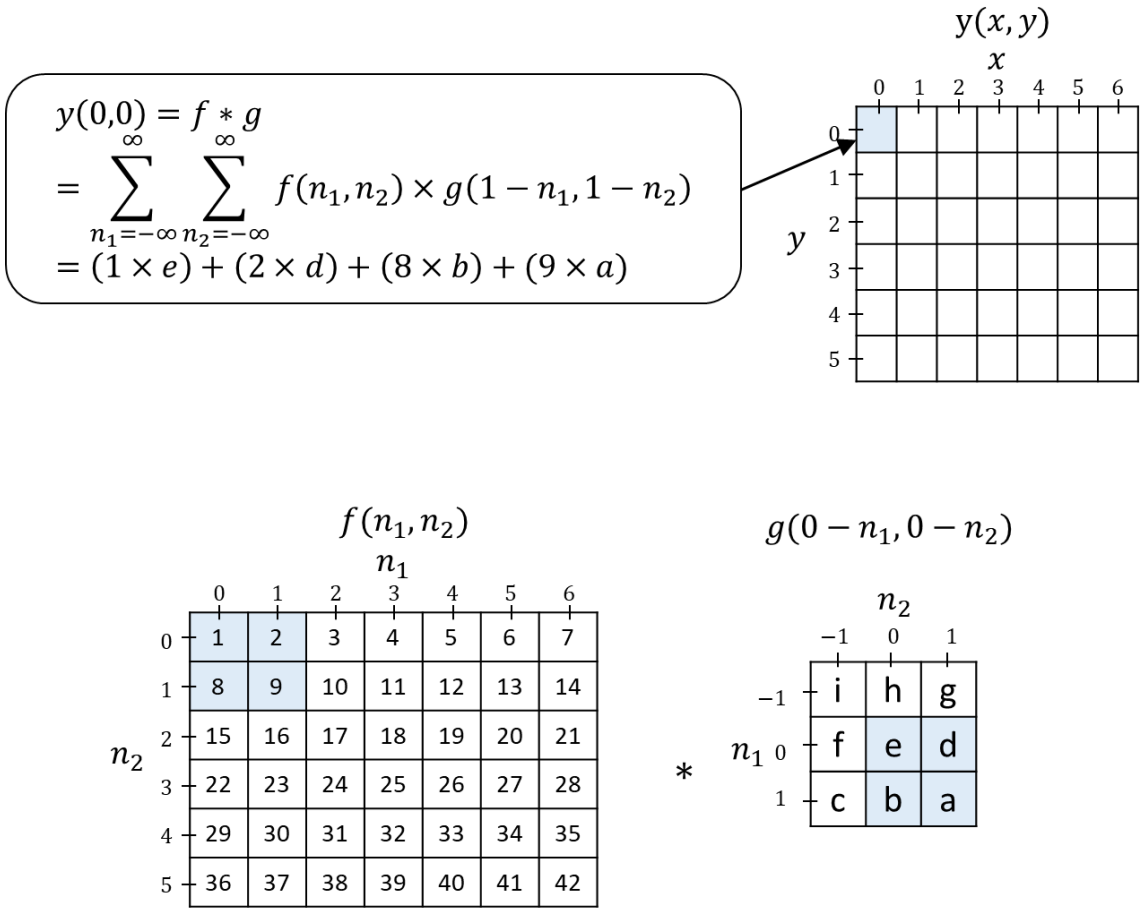


Figure 3.11: Convolution result at point $y(0, 0)$.

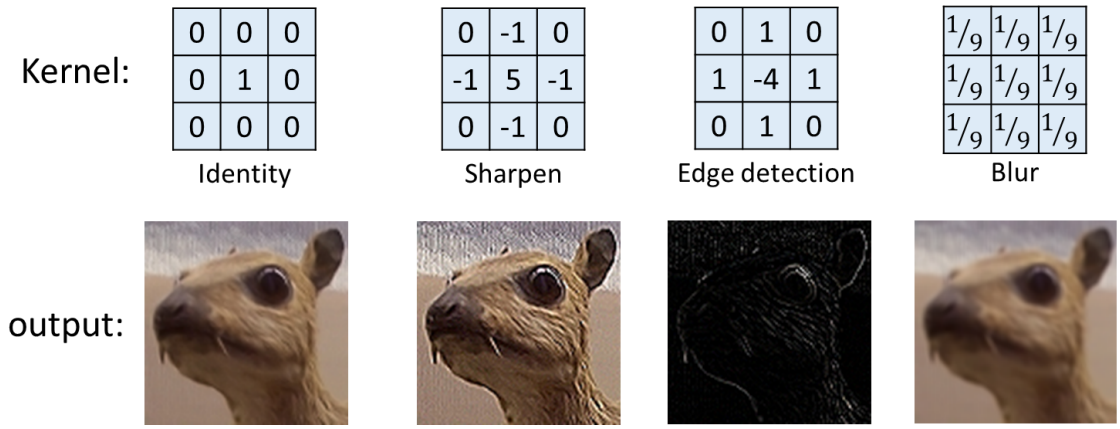


Figure 3.12: The sample of convolution result [60].

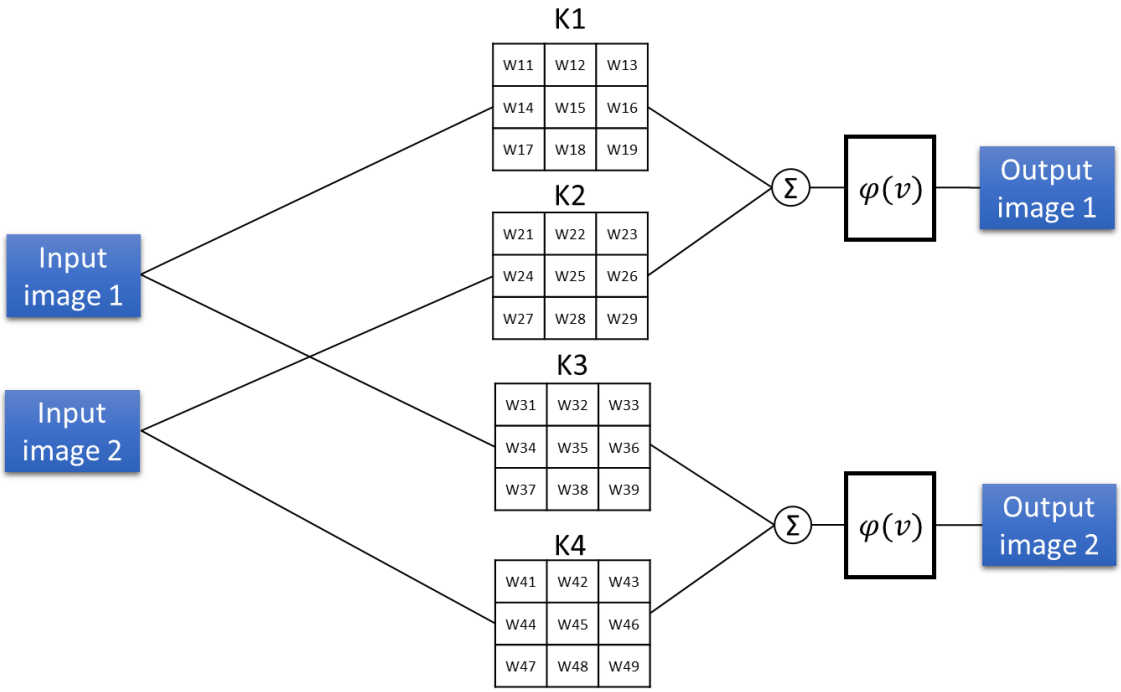


Figure 3.13: Detail of convolutional layer.

the multi-layer perceptron. However, the process of connection weight is replaced by the convolution process. The scalar weight is replaced with the kernel. The training process tries to optimize the kernel to detect the most effective feature point. The structure of convolutional process is shown of Fig.3.13.

The input image can divide into three channel (RGB or HSV). Each input channel is convoluted with the kernel matrix and applies the matrix addition. The activation function is applied to regulate the pixel value. The most CNN model uses the ReLU function in the convolutional layer.

3.4 Subsampling layer

The sub-sampling (also known as pooling layer) is designed for skipping over pixels during convolution. Many CNN model use pooling layers to decrease the input channel. There are two famous of pooling type: max pooling and average pooling. Figure 3.14 shows the sample of 2×2 tile size. The result of average pooling is the average value of all elements inside the pooling size. The max pooling is the process to find the max feature value inside the pooling tile. The essential idea behind max pooling is to break up each feature map into equally sized tiles. Then we create a condensed feature map. The max pooling uses low computation resource and make the learning and predict process faster and robust than average pooling. For this reason, the max pooling is the most useful method for the subsampling layer of the general CNN model.

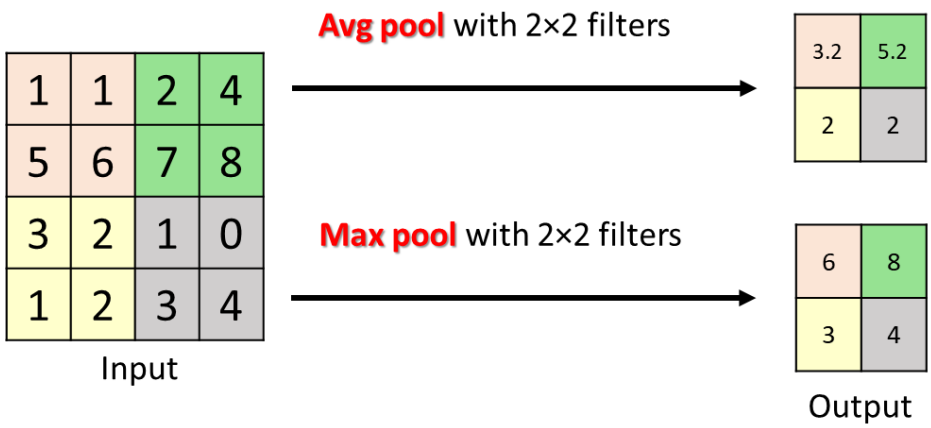


Figure 3.14: Average pooling and max pooling.

3.5 State-of-the-art CNN model

Many researchers and the commercial company have participated in the image recognition challenge called ImageNet [16]. This challenge provides the dataset that containing the 1000 categories and 1.2 million images. Figure. 3.15 shows the sample image from ImageNet challenge. The participator creates a model to classify these image and obtain the best accuracy. They create differ in complexity and architecture, though for image classification. The following is a some of famous CNN model that get the good accuracy in this challenge.

3.5.1 AlexNet

The AlexNet has designed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton from the University of Toronto [61] on 2012. This model is present the basic structure of the CNN model. AlexNet contained eight layers; the first five were convolutional layers, and the last three were fully connected layers as shown in Fig. 3.16. This model shows the simple structure of CNN can obtain the good accuracy of image classification. This research shows that by using ReLU activation in the fully connected layer is improve the training performance than sigmoid or tanh. The structure of AlexNet inspire many CNN in the modern age.

3.5.2 VGGNet

The VGGNet presents by Visual Geometry Group from Oxford university on 2014 [62]. This model is composed of many layers of the convolutional layer. The research shows the relationship between the number of convolutional layers and the accuracy of image recognition. The increasing the convolutional layer is increasing capability of the CNN model to do the difficult task. The structure of the VGG-11 and VGG-19 shown in Fig. 3.17.

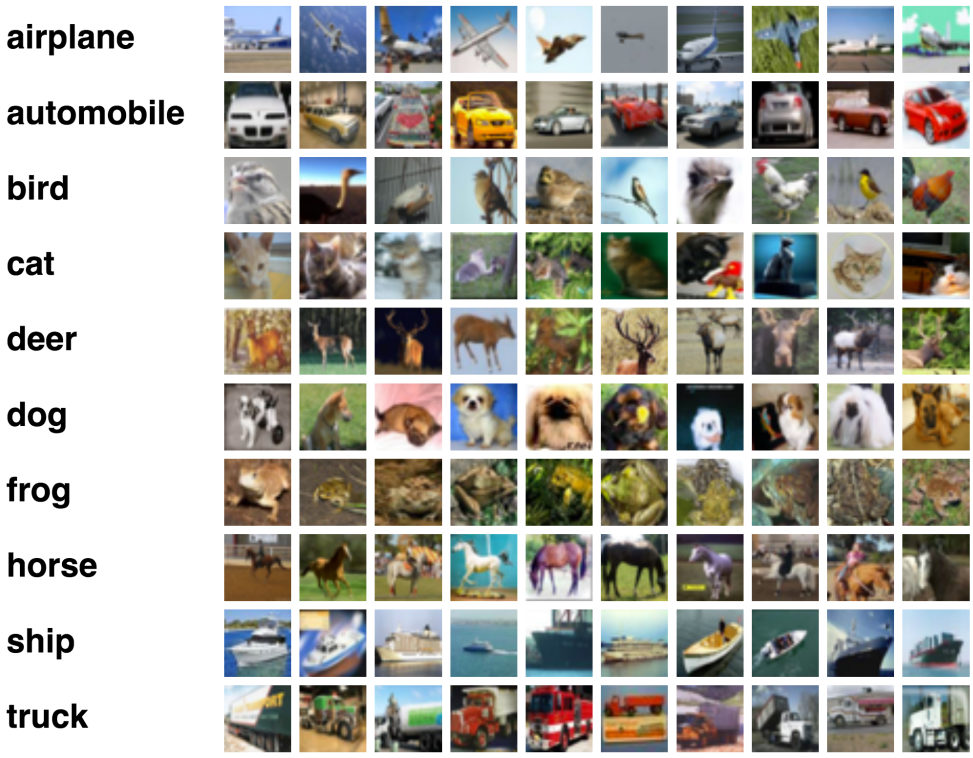


Figure 3.15: Sample images from ImageNet dataset [16].

3.5.3 ResNet model

The ResNet [17] presents by Microsoft research group on 2015. This model presents a residual learning framework. The basic structure of this model is similar withe the VGGNet. however, they have a residual image to bypass to some layer. The structure of the ResNet shown in Fig. 3.18. The VGGNet show that more convolutional layer increase the accuracy relatively. However, the vanishing gradient problem, the back-propagation can not updated the weight of first layer correctly. The error gradient values is lower than one then, when the gradient error comes to the first layers, its value goes to zero. The residual function preserve the gradient error.

3.5.4 Inception model

The Inception model first presents by Google company was also known as GoogLeNet [63]. The structure of the CNN model is an effect with the accuracy of the output. The idea of this model is using many kinds of the CNN structure by parallel all of the model. The learning process will use the effective branch by automatically. The structure of the Inception model shown in Fig. 3.19.

Input	224×224	Channel: 3 (RGB)	
Convolutional	11×11	Output Ch.: 48	Stride: 4
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Convolutional	5×5	Output Ch.: 128	Stride: 1
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 192	Stride: 1
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 192	Stride: 1
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 128	Stride: 1
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Fully-connected		Output Ch.: 2048	
		ReLU	
Dropout	50%		
Fully-connected		Output Ch.: 2048	
		ReLU	
Dropout	50%		
Fully-connected		Output Ch.: 1000	
		ReLU	
		Soft-max	
Output		Channel: 1000 (Classification)	

Figure 3.16: Structure of AlexNet.

VGG-11 model

Input	224×224	Channel: 3 (RGB)	
Convolutional	3×3	Output Ch.: 64	Stride: 1
		ReLU	
Normallization	LRN		
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 128	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Fully-connected		Output Ch.: 4096	
		ReLU	
Fully-connected		Output Ch.: 4096	
		ReLU	
Fully-connected		Output Ch.: 1000	
		ReLU	
		Soft-max	
Output		Channel: 1000 (Classification)	

VGG-19 model

Input	224×224	Channel: 3 (RGB)	
Convolutional	3×3	Output Ch.: 64	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 64	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 128	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 128	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 256	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Convolutional	3×3	Output Ch.: 512	Stride: 1
		ReLU	
Maxpool	2×2		Stride: 2
Fully-connected		Output Ch.: 4096	
		ReLU	
Fully-connected		Output Ch.: 4096	
		ReLU	
Fully-connected		Output Ch.: 1000	
		ReLU	
		Soft-max	
Output		Channel: 1000 (Classification)	

Figure 3.17: Structure of VGGNet.

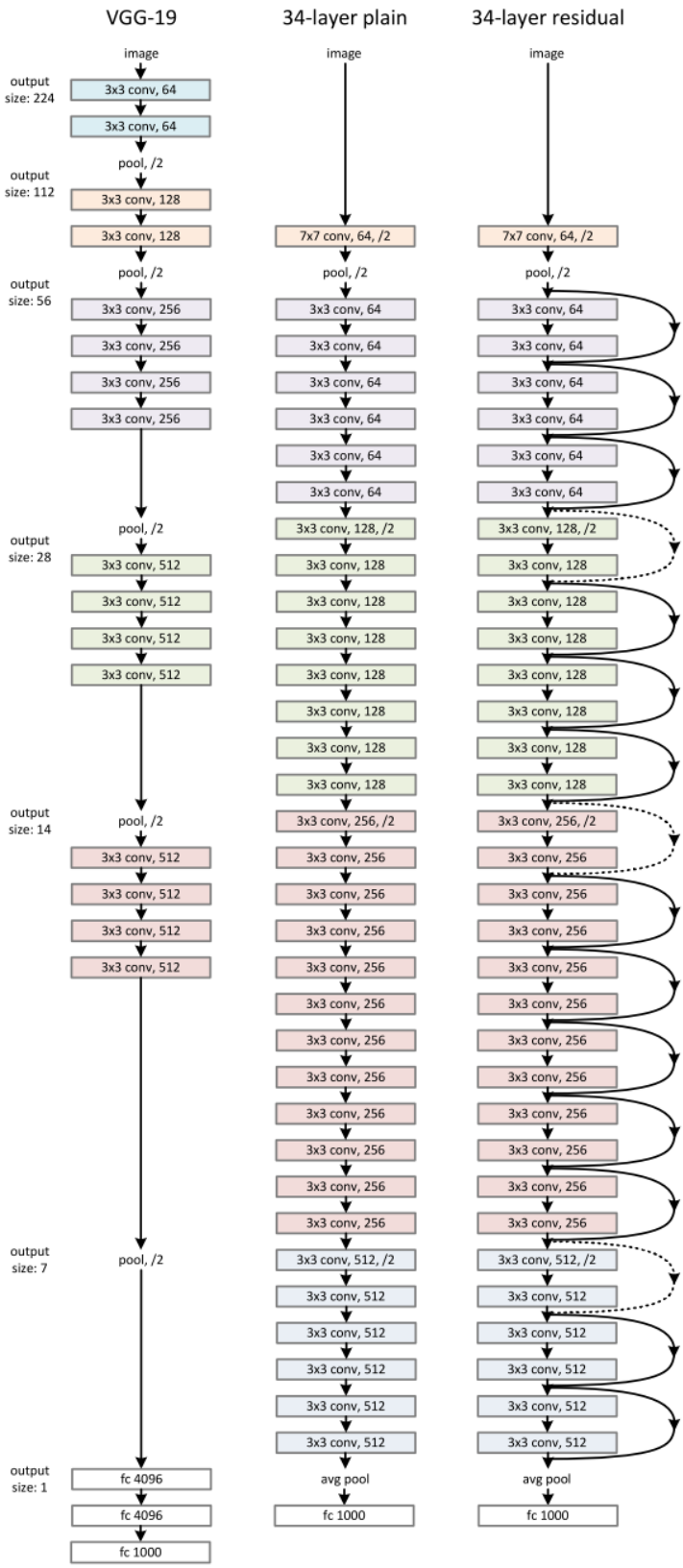


Figure 3.18: Compare the structure of ResNet and VGGNet [17].

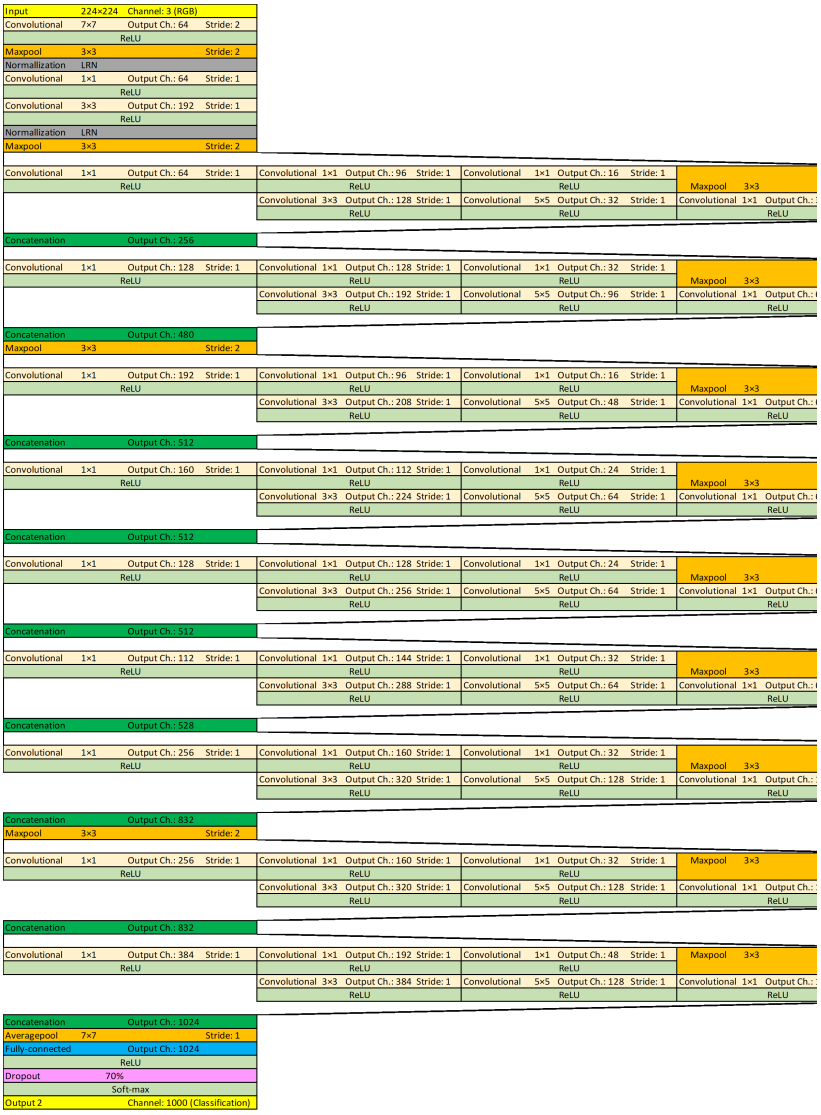


Figure 3.19: Structure of Inception model.

3.6 Conclusion

The CNN has had a large impact on the computer vision and pattern recognition. The impressive accuracy of the CNN makes many researchers apply this method in many application including gaze estimation. The CNN is a learning-based method, the good dataset is required to train the model. The good dataset is composed of having enough amount of the image and diversity of the training images.

Chapter 4

Pupil detection using CNN method

4.1 Introduction

This chapter presents a convolutional neural network (CNN)-based pupil center detection method. Potentially, the pupil center position of a user's eye can be used in various applications, such as human-computer interaction, medical diagnosis, and psychological studies. However, users tend to blink frequently; thus, estimating gaze direction is difficult. The proposed method uses two CNN models. The first CNN model is used to classify the eye state and the second is used to estimate the pupil center position. The classification model filters images with closed eyes and terminates the gaze estimation process when the input image shows a closed eye.

Thus, GESs can estimate objects of interest. One type of GES uses an inside-out camera [23], [58], which is comprised of an eye camera and a scene camera. The eye camera captures images of the user's eyes. Such a GES detects the pupil center and maps it to a point in the scene image. Recently, GESs have been used in various applications, such as video summarization [27], daily activity recognition [28], reading [29], human-machine interfaces [30], and communication support [31].

It is difficult to detect the pupil center because the eye is a non-rigid object, users blink frequently, and eyelid or eyelashes can occlude the pupil. Furthermore, the iris has various colors, such as blue, brown, and black. However, when an infrared camera is used to capture eye images, the iris fades out, which makes the pupil clearer. This approach makes the eye image easy to work with. However, blinking remains problematic because is difficult to detect the pupil center point when a user blinks. Consequently, gaze direction errors can occur.

This chapter focuses on pupil center detection using infrared eye images captured by a wearable inside-out camera and proposes an accurate detection method that uses a convolutional neural network (CNN). The proposed method is composed of two CNN models. The first determines whether it is possible to detect a pupil in an input image. The second CNN model detects the pupil center in an input eye image. This model outputs the pupil center X- and Y-coordinates. We evaluated the proposed method using a dataset of infrared eye images captured by our inside-out camera. The results demonstrate that the proposed method demonstrates high accuracy than other methods.

Typically, CNNs are trained using supervised learning; thus, they require a large training dataset. There are some public datasets of eye images [12], [13]; however, such datasets do not typically include images of eyes in the blink state. We describe a process to capture a sufficiently large image dataset with good distribution and variety of pupil position and eye state.

4.2 Related research

Several studies have focused on feature point detection based on eye images [64]〜[67]. Li et al. proposed a hybrid eye-tracking method that integrates feature-based and model-based approaches [64]. They captured eye images using an inexpensive head-mounted camera. Their method detects pupil edge points and uses ellipse fitting to estimate the pupil center. Zheng et al. proposed an algorithm to detect eye feature points, including pupil center and radius, eye corners, and eyelid contours [65]. Moriyama et al. developed a generative eye-region model that can meticulously represent the detailed appearance of the eye region for eye motion tracking [66]. Warapon and Saitoh proposed a fast and precise eye detection method using gradient value [67]. However, if the eye image contains unexpected objects with a high gradient or intensity, such as an eyelash with mascara or a specular point, it is difficult for such methods to detect the pupil.

CNNs outperform traditional algorithms in various research fields, such as artificial intelligence, image classification, and audio processing. Zhang et al. proposed a CNN-based gaze estimation method in an unconstrained daily-life setting [12]. In that method, the input data are an eye image and the 2D head angle, and the output is a 2D gaze angle vector that consists of two gaze angles, i.e., yaw and pitch. Fuhl et al. proposed a dual CNN pipeline for image-based pupil detection [68]. Here, the input is an eye image, and the output is an estimated pupil center position. In the first pipeline stage, an input image is downscaled and divided into overlapping subregions. A coarse pupil position is estimated by the first shallow CNN. In the second stage, subregions surrounding the initial estimation are evaluated using a second CNN, and the final pupil center position is detected. Choi et al. proposed a CNN model to categorize driver gaze zones [69]. Here the input image is an eye image, and the outputs are the probabilities of nine gaze zones. As mentioned previously, most related studies that employ CNNs attempt to detect only the center point of a pupil.

Sun et al. [18] presented the cascade CNN model for facial point detection. This structure is composed with multiple CNN models to detect each facial feature points as shown in Fig. 4.1. Zhang et al. [19] presented Coarse-to-fine auto-encoder networks (CFAN), which are based on Sun et al. model that used to detect multiple facial feature points as shown in Fig. 4.2. This chapter uses both models to be the comparison model.

The objective of this study is to apply the proposed method to a GES. The proposed method is designed for daily life; thus, it must be robust because it is not always possible to detect the pupil center position, e.g., when the eyelid overlays the pupil due to blinking. The proposed method is composed of two CNN models. The first

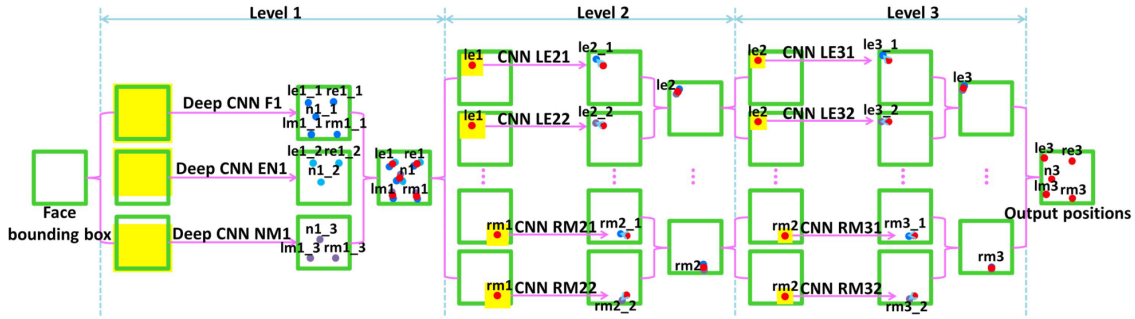


Figure 4.1: Cascade CNN model for facial point detection from Sun et al. [18].

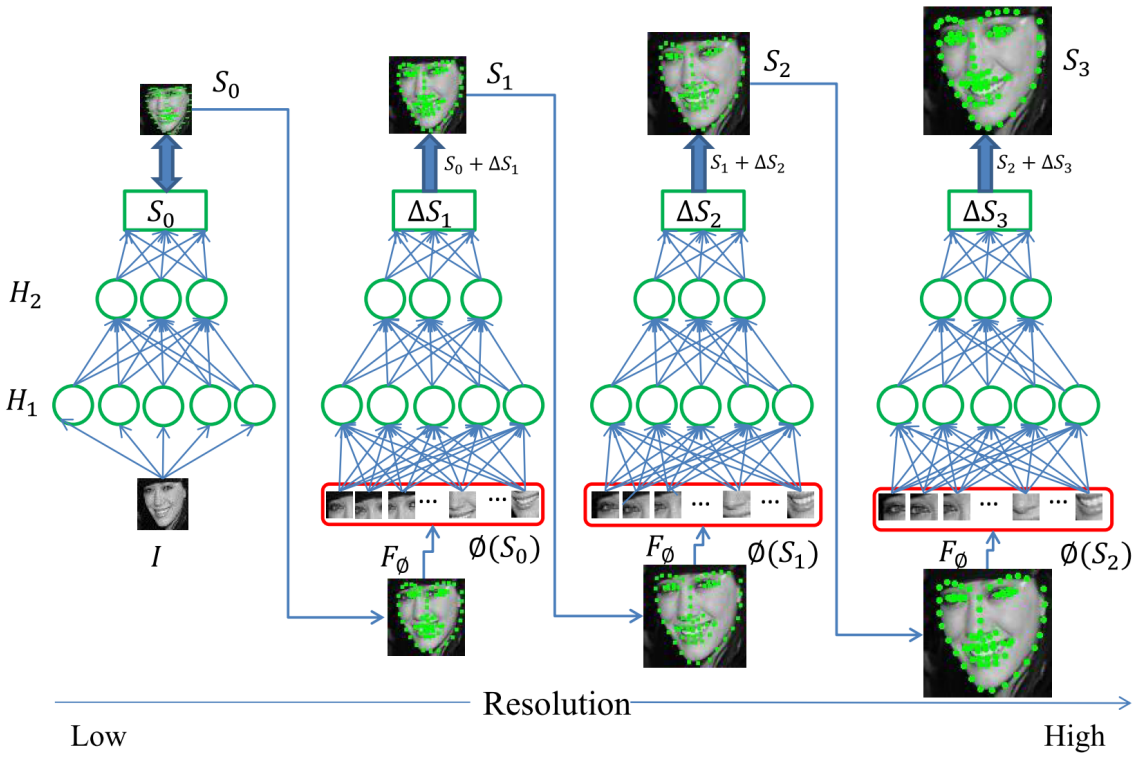
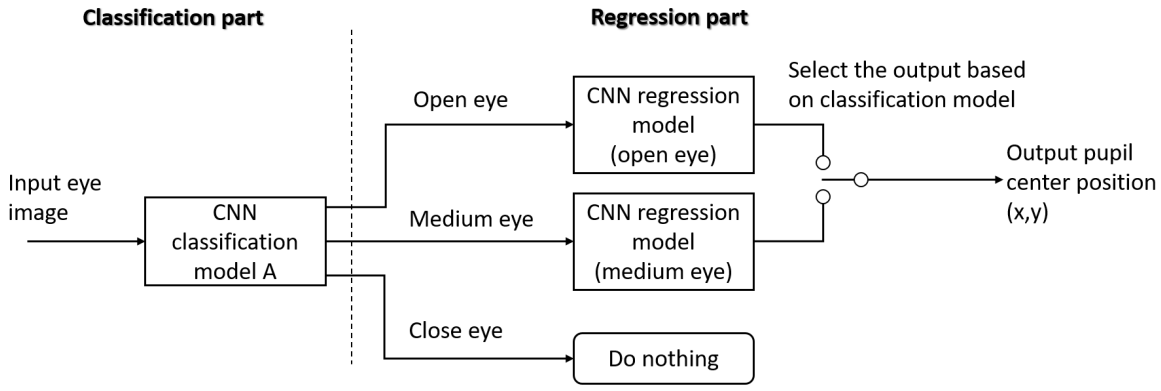
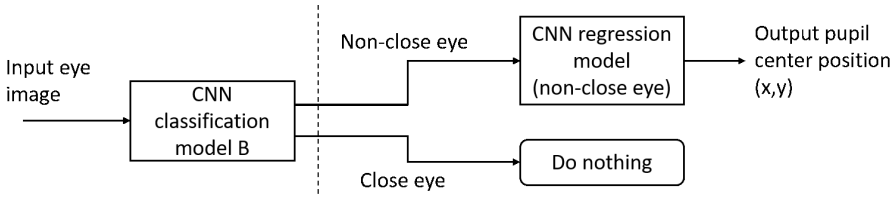


Figure 4.2: Coarse-to-fine auto-encoder networks (CFAN) from Zhang et al. [19].



(a) Method A.



(b) Method B.

Figure 4.3: Proposed two parts CNN model.

model classifies the input image, as shown in Fig.4.3. The second model operates in a regression mode [18], [70]. Collectively, these CNN models output the X- and Y-coordinates of the pupil center point.

4.3 Proposed method

A CNN is composed of a convolutional layer and a fully-connected layer. Typically, the fully-connected layer is a feed-forward neural network. The effective layer between the input data and the fully-connected layer is the convolutional layer, which is used to detect the significant feature point in the input data prior to sending it to the fully-connected layer. If the convolutional layer cannot detect the target feature point, it inputs zeros to the fully-connected layer. Under this condition, the fully-connected layer outputs only the bias effect of each layer. In other words, a CNN outputs a value regardless of the quality of the input data. We employ a CNN model to classify the input data prior to sending it to the detection model.

We describe the classification and detection models in the following subsections.

4.3.1 Classification model

There are various CNN classification models, and each model has specific characteristics. AlexNet [61] is a well-known models for classification tasks. We selected this

Table 4.1: Proposed CNN architectures of classification model A.

Layer	Channel	Filter size	Pooling size	Normalization	Drop
Conv1	48	11×11	2×2	yes	-
Conv2	128	5×5	2×2	-	-
Conv3	192	3×3	2×2	-	-
Conv4	192	3×3	2×2	-	-
Conv5	128	3×3	3×3	-	-
Full1	1024	-	-	yes	yes
Full2	1024	-	-	yes	yes
Out	3 class	-	-	-	-

Table 4.2: Proposed CNN architectures of classification model B.

Layer	Channel	Filter size	Pooling size	Normalization	Drop
Conv1	48	11×11	2×2	yes	-
Conv2	128	5×5	2×2	-	-
Conv3	192	3×3	2×2	-	-
Conv4	192	3×3	2×2	-	-
Conv5	128	3×3	3×3	-	-
Full1	1024	-	-	yes	yes
Full2	1024	-	-	yes	yes
Out	2 class	-	-	-	-

Table 4.3: Proposed CNN architectures of regression model.

Layer	Channel	Filter size	Pooling size	Normalization	Drop
Conv1	96	7×7	3×3	yes	-
Conv2	256	5×5	2×2	-	-
Conv3	512	3×3	-	-	-
Conv4	512	3×3	-	-	-
Conv5	512	3×3	3×3	-	-
Full1	4096	-	-	yes	yes
Full2	4096	-	-	yes	yes
Out	2 Reg.	-	-	-	-

model to classify the eye state. We defined three states in eye images, i.e., 1) the image shows the pupil as a full circle (open state), 2) an eyelid overlays the pupil (medium state), and 3) no pupil is observable in the image(closed state).

Some studies have used a separate CNN model to perform specific tasks. For example, Sun et al. created multiple models to detect each feature point [18]. We also propose using two methods, which we refer to as methods A and B. For method A, we create a CNN model to classify the input image as open, medium, or closed eye states, as shown in Fig.4.3(a). For medium and open eye images, we create two CNN regression models to detect the feature points from each image type. The details of method A's classification and regression models are listed in Table 4.1. If the input image is an open eye image, it will be sent to a CNN model trained using only open eye images. Similarly, if the input image is a medium eye image, it is sent to a CNN model trained using only medium eye images.

The proposed CNN models can potentially solve multiple problems. Note that most previous studies employed an end-to-end CNN model to solve multiple problems. We use method B (Table 4.2) to classify input images as closed or non-closed eye (i.e., open eye and medium eye images, respectively). This classification model selects only non-closed eye images and sends those images to the CNN trained using non-closed eye images, as shown in Fig.4.3(b). Note that we compare the performance of both methods.

A cost function must be defined prior to training the CNN. The training process attempts to minimize this cost function. In the proposed CNN classification model, we use the mean of the sum of squared errors as the cost function, which is expressed as follows:

$$cost = \frac{\sum_{i=1}^{N_o}(o_i - d_i)^2}{N_o},$$

where o_i is an estimation output at i, d_i is a label at i, and No is the number of output classification results.

4.3.2 Regression model

The proposed CNN regression model (Table 4.3) is based on the pose regression ConvNet [70], which consists of five convolutional layers and three fully-connected layers. The collection of convolutional layers is followed by pooling and local response normalization layers, and the fully-connected layers are regularized using dropout. All hidden weight layers use a rectification activation (i.e., ReLU) function. Most CNN architectures for object localization use five convolutional layers [70]〜[73]. A difference between pose regression ConvNet and the proposed regression model is the normalization layer. ConvNet has a normalization layer after the last convolutional layer (Conv5). However, in a preliminary experiment, we found that training using the eye image dataset does not converge when the normalization layer is applied after the final convolutional layer. Thus, we do not employ this architecture. This difference also applies to the fully-connected layers. In our architecture, we use local response normalization [61] for Conv1 and use L2 normalization for fully-connected layers. L2

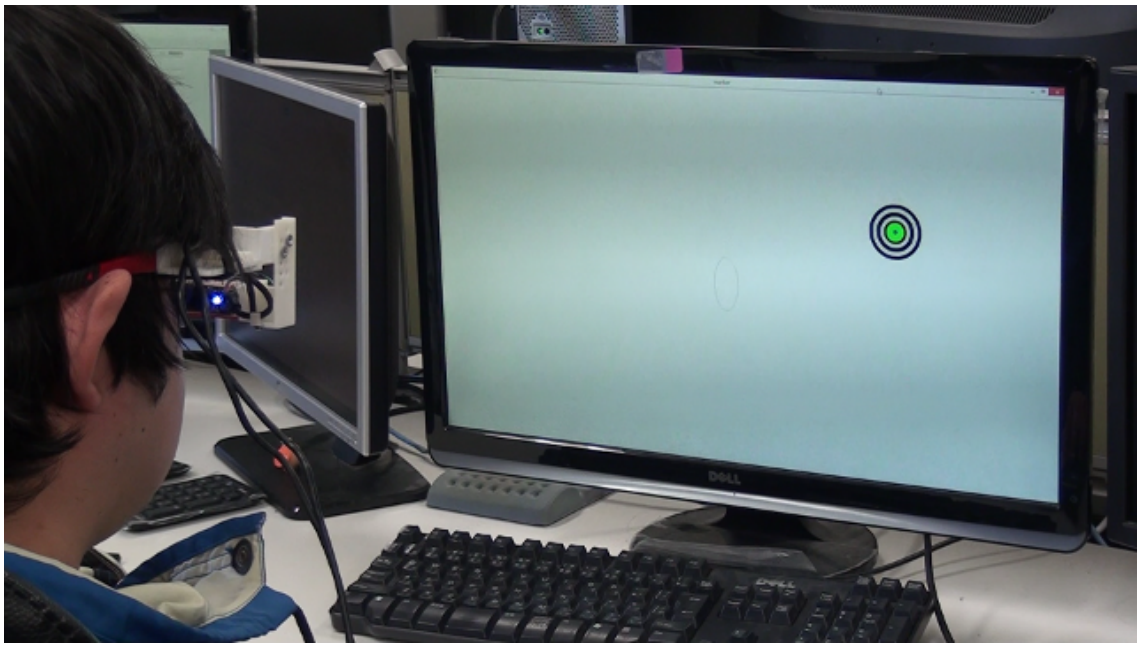


Figure 4.4: Collection experiment scene.

normalization is defined as follows:

$$x'_k = \frac{x_k}{\sqrt{\sum_{i=1}^{N_i} x_i^2}},$$

where k is the index of input nodes, x_k is the input data at node k, x'_k is the output from the normalization process at node k, and N_i is the number of data elements in the layer. This normalization process is required for training to converge.

We remove the activation function to make the output value linear. The input to the proposed CNN is an eye image (120×80 pixels). The error function e of the CNN regression model is defined as follows:

$$e = \sqrt{(P_x - D_x)^2 + (P_y - D_y)^2}.$$

This function is the distance between ground truth D and estimated point P.

4.4 Experiment

4.4.1 Dataset

A CNN is a supervised learning method that requires a large dataset to train a model. Moreover, a variety of ground truths are required to make the model more accurate. MPIIGaze [12] is a well-known eye image dataset composed of medial canthus, lateral canthus, and pupil points. However, the pupil points are not center points. For a GES, pupil center points are required to calculate gaze direction. In this study, we

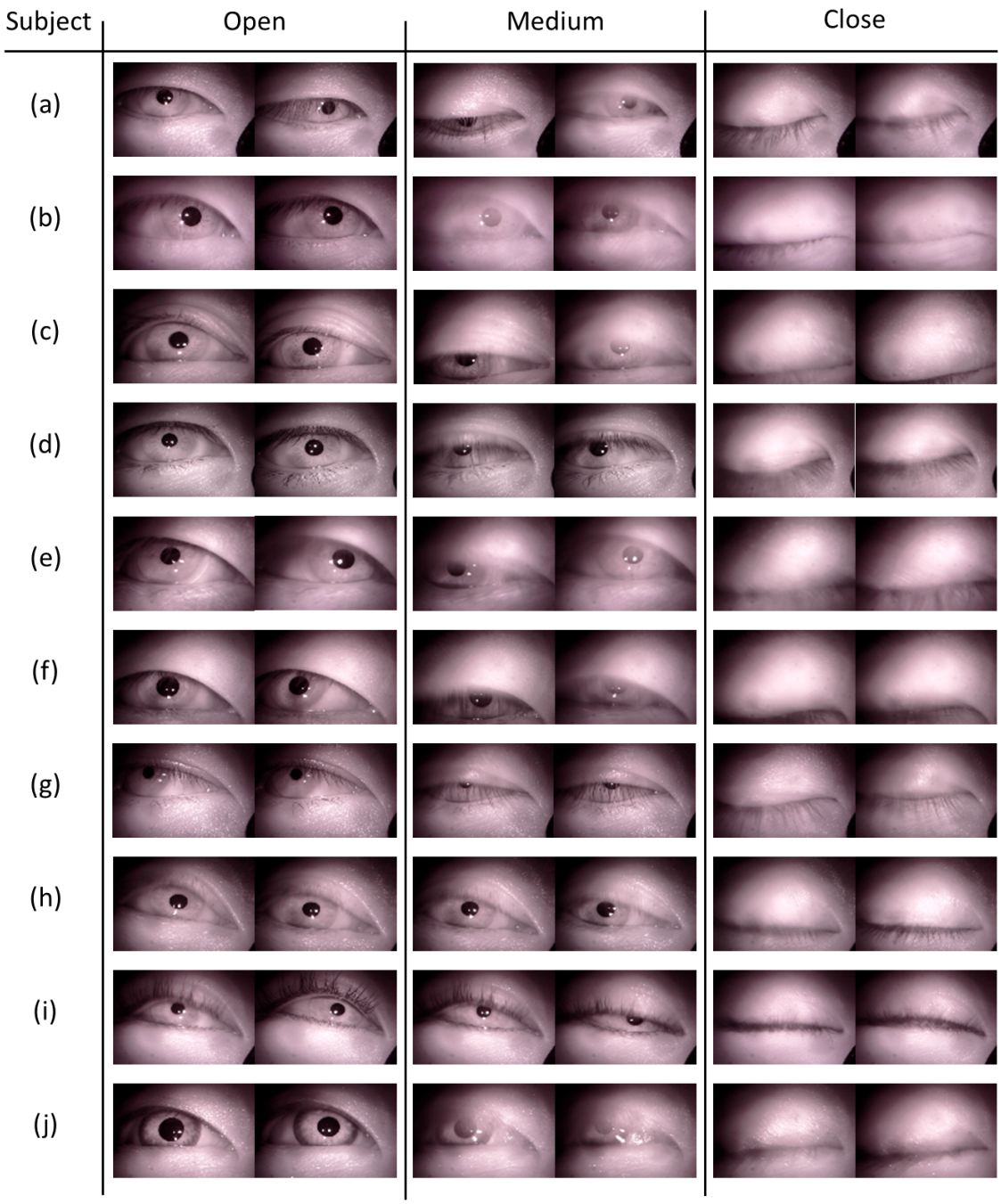


Figure 4.5: Sample eye images from our dataset.

Vertical section \ Horizontal section	1	2	3	4	5	6	7	8	9	10	11	12	total
1	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	11	0	0	0	0	0	0	0	11
3	0	0	39	813	1016	933	750	296	19	0	0	0	3866
4	0	0	34	1419	2314	2985	1383	504	36	0	0	0	8675
5	0	0	22	706	1186	1546	1367	351	0	0	0	0	5178
6	0	0	32	554	324	446	427	82	0	0	0	2	1867
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	2
total	0	0	127	3493	4851	5910	3927	1233	55	0	0	4	

(a) All images.

Vertical section \ Horizontal section	1	2	3	4	5	6	7	8	9	10	11	12	total
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	6	0	0	0	0	0	0	0	6
3	0	0	0	211	331	278	325	138	11	0	0	0	1294
4	0	0	12	311	707	982	515	208	16	0	0	0	2751
5	0	0	6	254	446	583	537	134	0	0	0	0	1960
6	0	0	21	161	102	100	114	17	0	0	0	0	515
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
total	0	0	39	937	1592	1943	1491	497	27	0	0	0	

(b) Open eye image.

Vertical section \ Horizontal section	1	2	3	4	5	6	7	8	9	10	11	12	total
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	2	0	0	0	0	0	0	0	2
3	0	0	19	257	300	289	161	75	5	0	0	0	1106
4	0	0	13	417	752	1034	406	158	7	0	0	0	2787
5	0	0	10	234	332	496	395	124	1	0	0	0	1592
6	0	0	1	212	116	195	184	39	0	0	0	0	747
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
total	0	0	43	1120	1502	2014	1146	396	13	0	0	0	

(c) Medium eye image.

Vertical section \ Horizontal section	1	2	3	4	5	6	7	8	9	10	11	12	total
1	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	3	0	0	0	0	0	0	0	3
3	0	0	20	345	385	366	264	83	3	0	0	0	1466
4	0	0	9	691	855	969	462	138	12	0	0	0	3136
5	0	0	6	218	408	467	435	93	0	0	0	0	1627
6	0	0	10	181	106	151	129	26	0	0	0	2	605
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	2
total	0	0	45	1436	1757	1953	1290	340	15	0	0	4	

(d) Close eye image.

Figure 4.6: Distributions of our dataset.

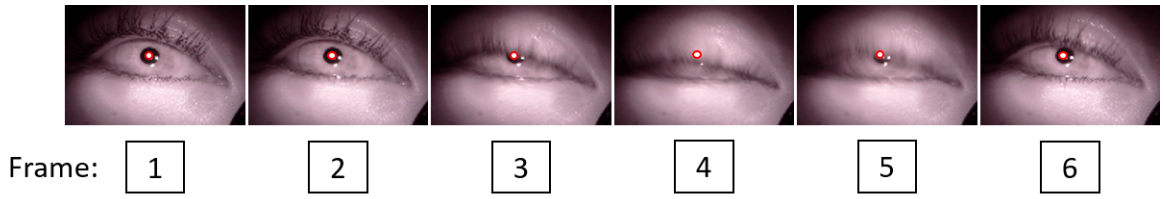


Figure 4.7: Annotation of medium and close eye image.

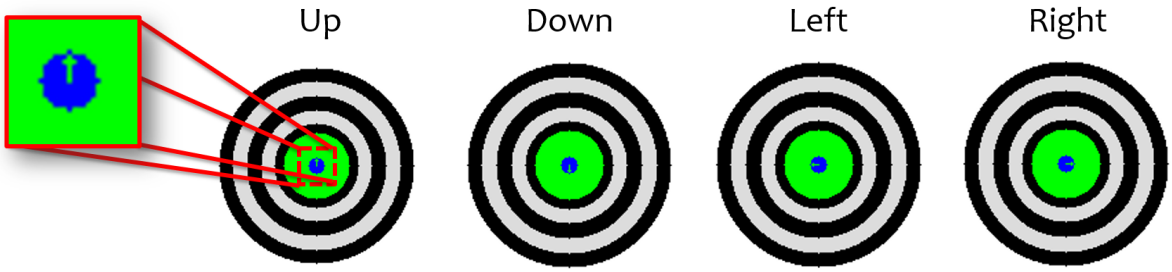


Figure 4.8: Four markers with arrow.

developed a system to capture a dataset with appropriate variation and reliability using an inside-out camera [58].

We required a dataset that contains blinking eye images to test the performance of the proposed CNN method. Thus, we had to design a system to capture multiple eye images under appropriate conditions. Note that the center of the pupil's position depends on gaze direction. To create the dataset, subjects wore an inside-out camera and observed a marker displayed on a monitor. Next, the system captures an image from the eye camera. We designed an additional process to ensure that the subject focused on the marker position. This capture system selects an arrow (up, right, down, and left) at random and displays it at the center of the marker. The subjects were tasked with pressing a corresponding arrow key. We asked the subjects to blink approximately five times before pressing the key. If the subject pressed the correct key, the capture system saved the eye images to the dataset. This process improved the variation of eye images in the dataset. The image collection environment is shown in Fig.4.4 . Details about the data collection process are described in the following:

- We used a 24-inch widescreen display for this experiment, and the distance between the subject and the display was 60 cm. We captured the images for the dataset in a room with sufficient light from both natural and fluorescent light sources.

- We divided the display area into 49 (7×7) sections and show the marker in that section respectively. First, we shuffle the order of the marker position, in

order to make the unpredictable position. The subject has to gaze at the marker without moving the head.

- Then, the user was asked to blink approximately five times. Next, the subject pressed the direction key corresponding to the arrow shown in the center of the marker (Fig. 4.8). The capture program stored 20 eye images captured approximately one second prior to the subject pressing the key. After the eye images were saved, the marker was moved to the next position automatically. This process was repeated 49 times to collect $49 \times 40 = 1960$ eye images.

After collecting all eye images, we manually annotated the pupil center position by one person for avoiding wrong categorization by multiple persons. We categorized the eye images into three classes: open, medium, and closed eyes. Each class is described as follows:

- An open eye image clearly shows the edge of the pupil, which makes it easy to estimate the pupil center position.
- A medium eye image shows the eyelid overlaid on some part of the pupil, which makes it difficult to estimate the pupil position.
- A closed eye image shows no pupil, which makes it impossible to estimate the pupil position.

Figure 4.5 shows sample eye images. Ten subjects (seven males (a)-(g); three females (h)-(j)) participated, and a total of 19,600 eye images were collected. All subjects were normally sighted and did not wear glasses. This dataset has 6,526 open eye images, 6,234 medium eye images, and 6,840 closed eye images.

The distribution of the pupil center position in our dataset is shown in Fig.4.6. The distributions of open, medium, and closed eye images are shown in Figs4.6(b), 4.6(c), and 4.6(d), respectively. These distributions show that the number of image types is approximately equal for each section. Note that the pupil center positions were annotated manually. For medium and closed eye images, the exact pupil center position is unknown. We assume the pupil does not move during blinking; thus, we use the same annotation point from a previous open eye image frame, as shown in Fig.4.7, where the red dot shows the manually annotated ground truth. At frames one and two, the eye is open and easy to annotate. However, in frames three to five, the eye is in the medium or close states; therefore, for such images, we used the ground truth from frame two.

4.4.2 Classification evaluation

We evaluated the classification problem using leave-one- out cross-validation. We used a pre-training model trained using the ImageNet dataset [16] in order to avoid overfitting. The result from the pre-training model are better than without a pre-training model. The classification results of model A are shown in Table 4.4(a). The accuracy of this model was 82.58%. This result indicates that the accuracy of closed eye

Table 4.4: Confusion matrices of CNN classification model.

(a) Classification result of method A.

		Predict			
		Open	Medium	Close	Accuracy
Actual	Open	5465	1013	48	83.64%
	Medium	784	4595	855	73.67%
	Close	0	707	6133	89.66%

(b) Classification result of method B.

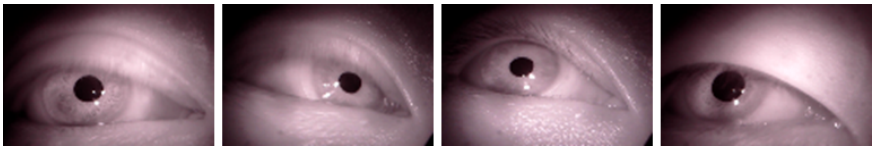
		Predict		
		Non-close	Close	Accuracy
Actual	Non-close	6596	244	96.43%
	Close	776	6064	88.65%

images is greater than that of the other classes. Some images for which classification failed are shown in Fig 4.9. The accuracy of the medium eye case (73.67%) is less than that of other the classes because some of the medium eye images were difficult to classify, as shown in Figs.4.9(c) and (d). However, this level of accuracy is reasonable.

Next, we created a model to classify two classes for method B, which we refer to as classification model B. This model was designed to classify closed and non-closed eye images. To train model B, we randomly selected non-closed eye images from medium and open eye images to ensure that the number of non-closed eye images was the same as closed eye images. The classification results of this model are shown in Table 4.4(b). The overall accuracy of this model was 92.54%, and the accuracy of non-closed and closed eye images was 96.43% and 88.65%, respectively. This indicates that the classification accuracy of model B is better than that of model A. Classifying closed and non-closed eye images is easier that doing so for the three classes of eye images because classification model B only classifies two classes, which improves accuracy compared to classification model A. However, all proposed classification models were designed to identify input images for which it is impossible to detect the pupil center position. Thus, both classification models can potentially identify closed eye images effectively.

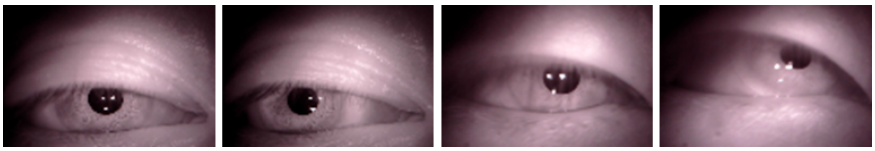
4.4.3 Regression model evaluation

We employed leave-one-out cross-validation to evaluate the regression model. As with the classification model, we used models pre-trained using the ImageNet dataset [16] before training with our eye dataset. As discussed in Section 2.3, the input to the regression model is an eye image selected by the classification model. For the regression model, we had to train and evaluate the model using manually annotated eye images, we called the method A* and B*. The regression model was trained using methods A*

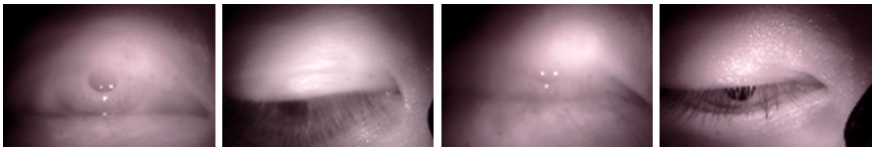


(a) Label: open eye; predicted: medium eye.

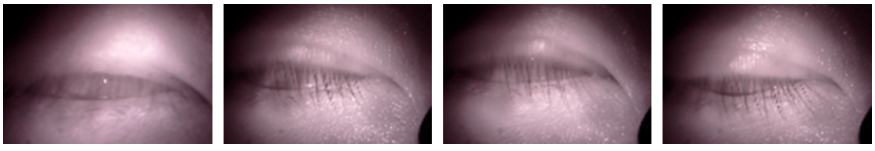
(b) Label: open eye; predicted: close eye.



(c) Label: medium eye; predicted: open eye.



(d) Label: medium eye; predicted: close eye.



(e) Label: close eye; predicted: medium eye.

Figure 4.9: Sample images from the failed classification model.

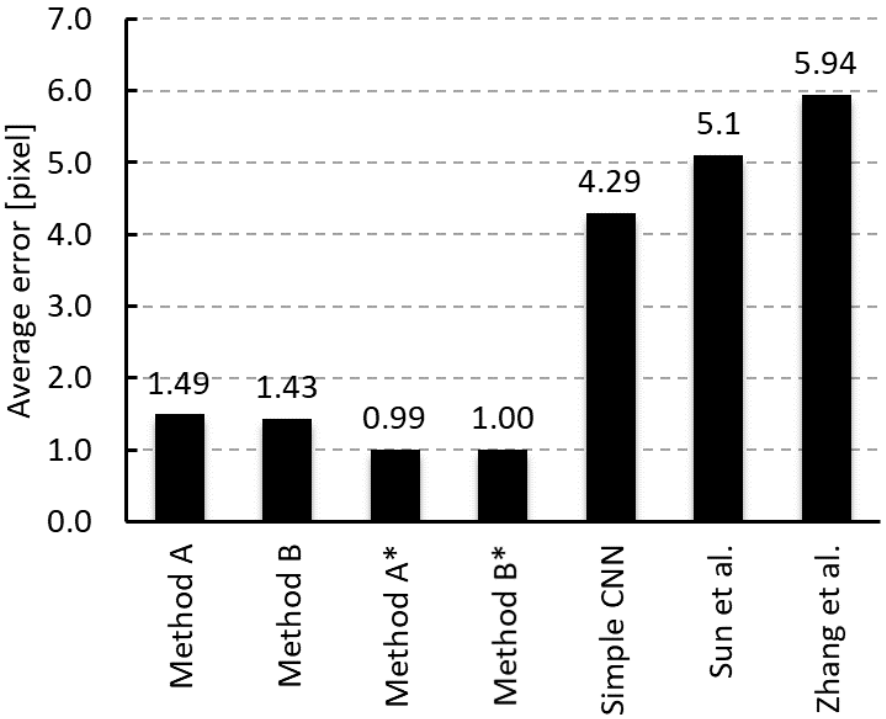


Figure 4.10: Average error of each CNN model.

Table 4.5: Confusion matrix of CNN classification model.

	Average error [pixel]			
Method	A	B	A*	B*
Open eye	0.79	–	0.80	–
Medium eye	2.19	–	1.21	–
Total	1.49	1.43	1.00	0.97

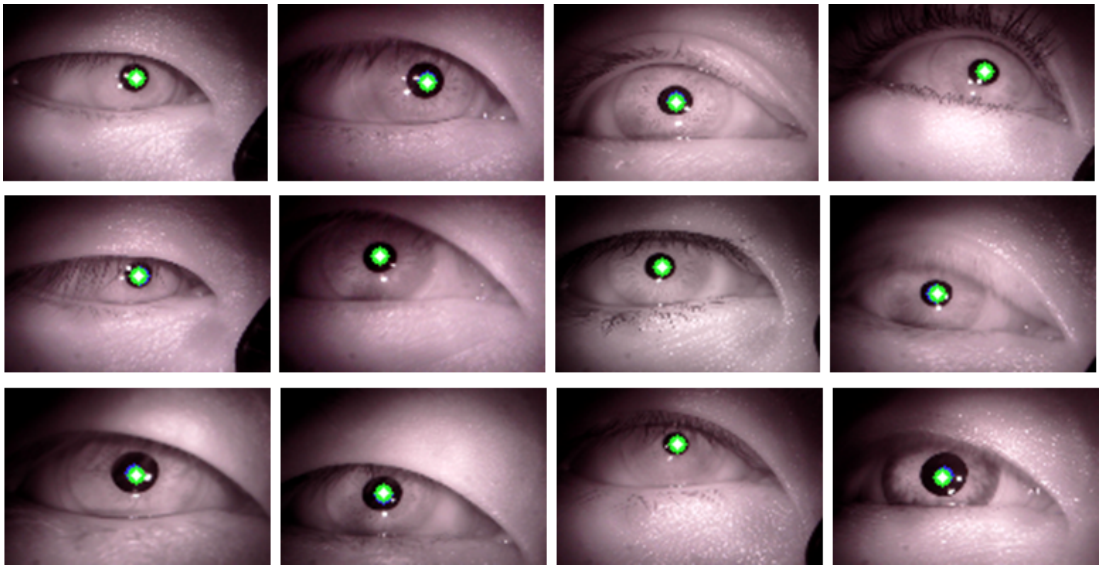


Figure 4.11: Success samples of open eye image.

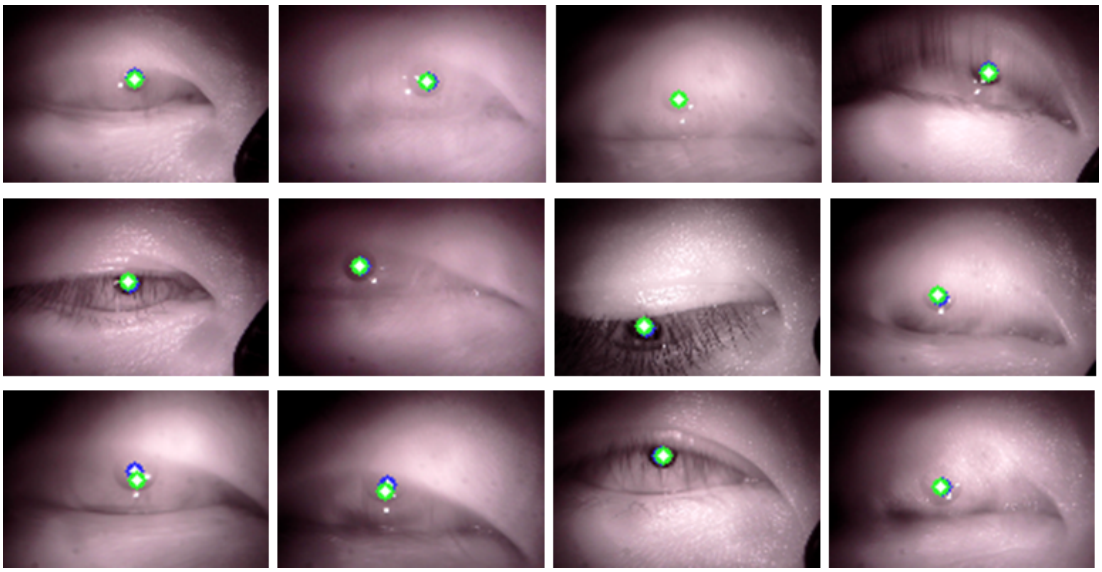


Figure 4.12: Success samples of medium eye image.

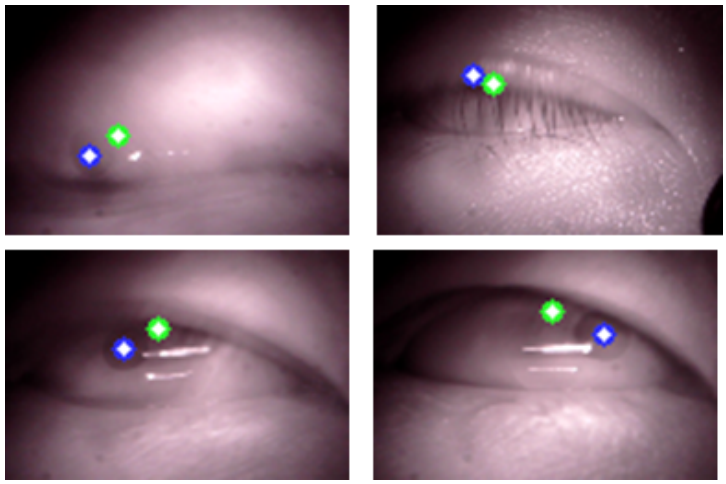


Figure 4.13: Failure samples.

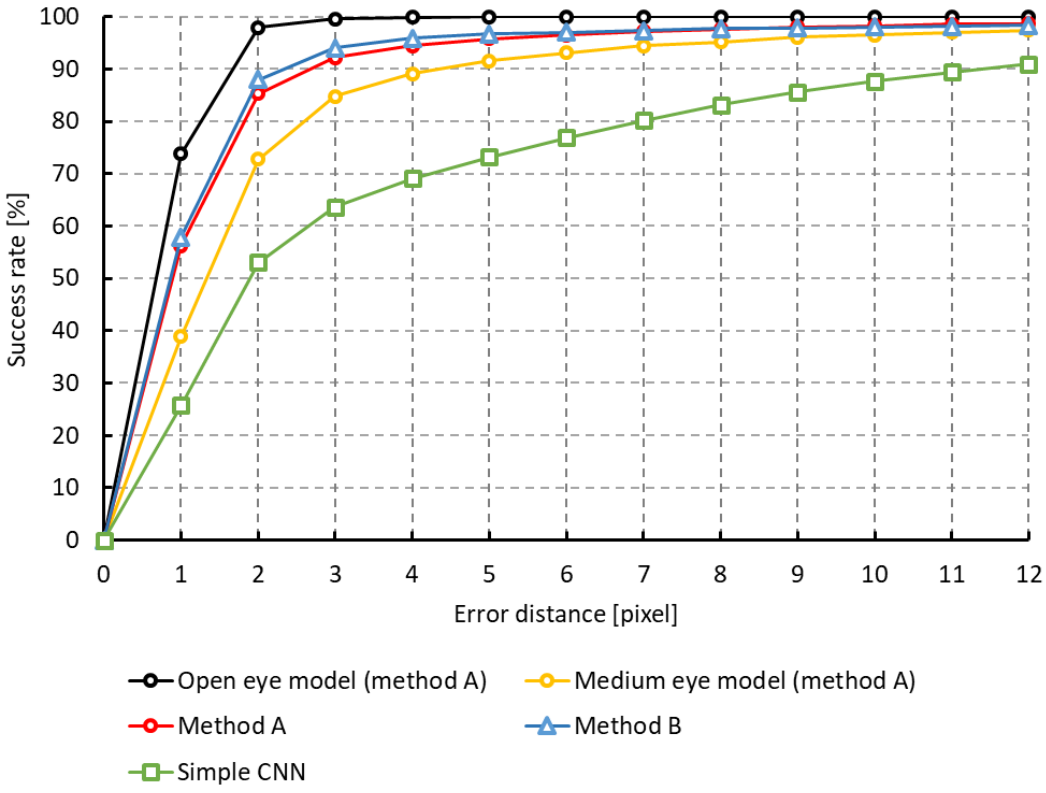


Figure 4.14: Performance curves.

and B* before the regression model was integrated into the CNN classification model. Next, we evaluated the estimated point using an image from the classification model (methods A and B). Methods A and A* have two CNN regression models to estimate the pupil center position in the specific input image (open and medium eye images). The average errors are shown in Table 4.5 .

The average errors of methods A* and B* these are results in the case of classification model have a 100% accuracy. However, when we attempted to detect the pupil position in an image classified by the CNN classification model (methods A and B), the average error was somewhat high. Next, we compared the proposed method to a CNN with no classification model, which we refer to as the simple CNN. This model architecture is the same as the regression model of methods A and B. We trained this model using all eye images in the dataset. Figure 4.10 shows that the average errors of methods A and B are better than those of the regression model with no classification model. Moreover, we compared the proposed method to other well-known CNNs used in feature point detection research (Sun et al. [18]; Zhang et al. [19]). We trained the compared models under the same conditions as the simple CNN. The results show that the proposed simple CNN model obtained good accuracy compared to the other models.

Figures 4.11 and 4.12 show sample results for the estimated point obtained by method A. Here, the green point is the estimated pupil point, and the blue point is the ground truth from our dataset. As can be seen, these points are very accurate, and the estimated point nearly overlays the ground truth. However, for some difficult images in which the pupil is shown in the small part, the CNN generates more errors, as shown in Fig.4.13.

4.5 Discussion

We compared the proposed method to the simple CNN model. We also compared the different effects between method A and method B. Methods A* and B* represent methods A and B when the classification model achieves 100% accuracy. The results shown in Fig.4.10 indicate that the success rate of method A* is better than that of method B*. This result proves that when we allow the CNN model to learn a specific problem, the model can obtain better results than the single model. However, when we use an input image from the CNN classification, the success rate of method A is slightly less than that of method B because the classification accuracy of method B is better than that of method A. When we consider the difficulty of the classification problem, classifying non-closed and closed eye images is easier than classifying eye states with three classes (i.e., open, medium, and closed). The single regression model (method B) was trained using both types of image (open and medium). Method B has robustness relative to classification error compare with method A.

However, the success rate of both models is better than that of the CNN model with no classification model (i.e., the simple CNN) and the compared models. Figure 4.14 shows the success rate of the proposed method. These results are the ratio of successful images compared to failed images when the distance between the ground

truth and estimated point is less than the error distance. When the error distance is greater than four pixels, the success rate of methods A and B is greater than 90%. This shows that the proposed method has the potential for application in gaze estimation tasks.

4.6 Conclusion

This chapter has presented methods to detect the pupil center position using a CNN model. We have focused on a wearable camera-based GES. When using a GES in daily life, it is sometimes impossible to detect the pupil center position from an eye image; thus, this chapter has considered avoiding this situation, e.g., when blinking obscures the pupil. For supervised learning of the CNN, the dataset required specific features, i.e., effective variety, appropriate distributions of image types, and sufficient amounts of data, to make the training process successful. Thus, we created a capture system to construct an original dataset. This original dataset provided closed, open, and medium eye images with good distribution. Using pre-trained models, the dataset contained approximately 20,000 images, which is sufficient to train the CNN model effectively.

The proposed CNN method has two parts. The first is the CNN model, which is used to classify the eye state, and the other is the CNN regression model, which detects the pupil center position. The results show that the proposed CNN model has the potential to classify the eye state. Moreover, the accuracy of the pupil detection is better than that of the simple CNN model.

Chapter 5

Calibration free approach for GES

5.1 Introduction

The main GES challenge is to solve the relationship between the eye point on the eye-camera image and the gaze point on the scene-camera image. Usually, GES has required individual calibration before estimating the gaze point. For the calibration process in general, the user has to look at several markers across the operating plane. During the calibration, the GES saves the positions of these markers on the scene image and pupil-center positions on the eye image; after the calibration, the GES applies a coordinate-transfer function to these saved data to map pupil-center positions from the eye image as gaze points on the scene image.

The calibration process varies depending on the device, however, this process is required for all commercially available GES devices when the user starts to use these devices. Among wearable GES devices, Tobii Pro Glasses 2 requires a one-point calibration process [7]. and NAC Image Technology EMR-9 requires viewing two to nine markers for the calibration process [34]. Gazo's device [74] is another example: it not only requires four calibration points but requires recalibration if the device is removed or is used for a long time, because the positional relationship between the camera and the eye changes. This inconvenient process makes the GES difficult to use.

Calibration is important for a GES to achieve high-precision performance. However, because the eye is not rigid, it is difficult to detect a feature point from eye images. Furthermore, the iris has various colors, such as blue, brown, and black. Especially in the case of black eyes, it is difficult to detect the center of the pupil. Thus, GES devices commonly use several infrared LEDs and an infrared camera to brighten the iris so that the pupil can be detected easily. Although processing is simplified, the development cost of such devices is higher.

To solve the problems of existing GES devices and make GES available for use in many applications, we have developed a GES that does not require calibration. We propose a convolutional neural network (CNN) model that detects a pupil-center point from an eye-camera image and automatically estimates a gaze point on the scene-camera image. Because the proposed system does not require calibration, it can be used even when the position of the device moves during use. Moreover, since a color eye image is acceptable to our GES, infrared LEDs and an infrared camera

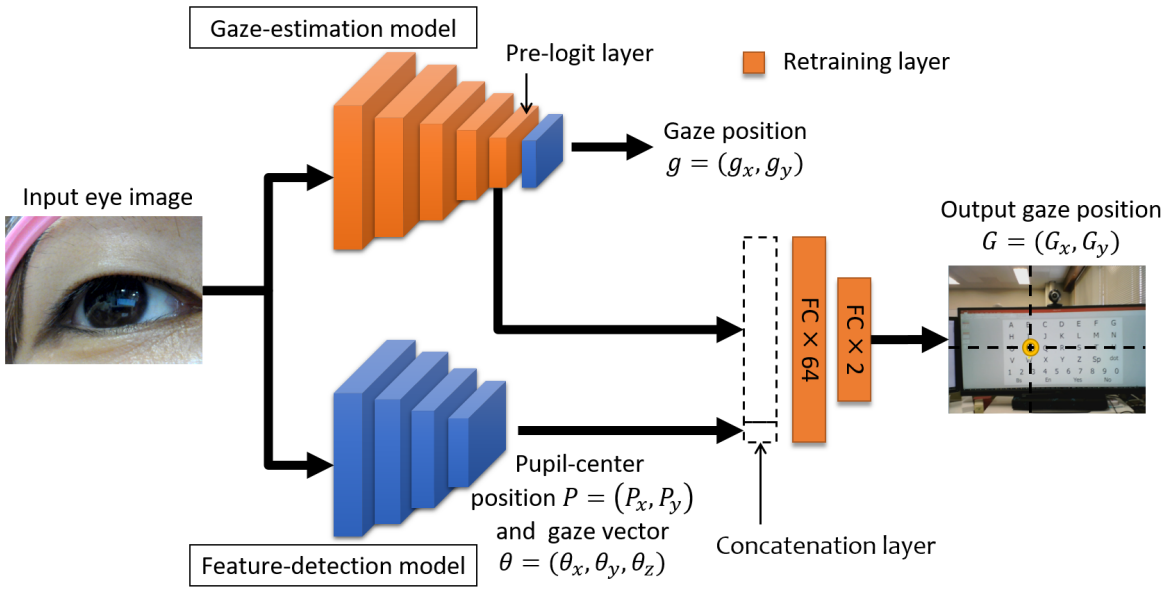


Figure 5.1: Proposed gaze estimation model.

are unnecessary. To verify the effectiveness of the proposed method, we developed a prototype device for demonstration and carried out evaluation experiments using other state-of-the-art methods. As a result, we confirmed that our proposed method estimates gaze point with high accuracy.

5.2 Related research

As mentioned above, commercially available GES products require calibration, but simpler calibration and calibration-free systems also have been proposed by various researchers.

Guestrin and Eizenman proposed a single-point calibration for which the user only has to fixate on a single point [75]. This method uses the centers of the pupil and two corneal reflections that are estimated from eye images captured by cameras. This method needs at least two cameras and some infrared light sources. Alnajar et al. proposed a different type of calibration process [76], which is based on gaze patterns obtained from other viewers. When new viewers are looking at an image, their method first estimates a topology of gaze points, and then these points are transformed so that they match others gaze patterns to correctly locate gaze points. Although special equipment is unnecessary, their method requires more than one person to determine gaze patterns in advance. Krafka et al. devised a GES for mobile application that uses whole-face images captured by built-in mobile cameras to estimate on-screen gaze points [45]. They proposed a new CNN model called iTracker and created a large eye-tracking dataset captured via crowdsourcing. Because their input images are only of whole faces, we were more interested in their use of a wearable camera to estimate

gaze points on scene images.

CNN outperforms traditional algorithms in various research fields. This method has the potential to solve complicated classification and regression problems. To obtain high accuracy by CNN, a large and reliable dataset is needed. Because our input images have the specific requirement to be a matched pair of eye image and scene image, preparing an adequately large dataset would be very costly. Wood et al. proposed UnityEyes [15], a method to generate a very large eye dataset based on 3D images and incorporating other useful information, such as gaze vector and pupil-center position. This chapter uses our own dataset and eye images generated by UnityEyes.

5.3 Proposed GES

5.3.1 Proposed CNN architectures

In recent years, deep learning technologies have greatly advanced the performance of many state-of-the-art visual-recognition tasks. In particular, CNN has been established as a powerful class of models for image recognition tasks. Through preliminary experiments, we confirmed that the standard CNN model can be used to estimate gaze point (see Sec. 5.4.3). However, in order to obtain a higher estimation accuracy, we propose a new architecture based on the CNN model.

Our proposed model combines two CNN models as shown in Fig. 5.1, for which the input data is a color eye image.

The first model is called the gaze-estimation (GE) model. This is an end-to-end CNN model to estimate the gaze point $g = (g_x, g_y)$ in the scene image. It is based on two well-known CNN models, Inception-v3 [77] and VGG-19 [62], for the GE model. The performance of both models will be discussed in Sec. 5.4.3. The GE model is trained using eye images and gaze points taken with our prototype device. The loss function $loss_g$ of GE model is defined as follows:

$$loss_{GE} = \|(g_x - \hat{g}_x)^2 + (g_y - \hat{g}_y)^2\|_2.$$

This function is the distance between the estimated gaze point g and ground truth $\hat{g} = (\hat{g}_x, \hat{g}_y)$ of the gaze point.

The second model of the proposed CNN architecture is called the feature-detection (FD) model. This model is used to detect the gaze feature information of a gaze vector $\theta = (\theta_x, \theta_y, \theta_z)$ and a pupil-center position $P = (P_x, P_y)$. It is based on the Inception-v3 model and is composed of multiple modules, each having many types of convolutional layers. If we have a large enough dataset for training, Inception-v3 can attain good accuracy compared with other models [77]. The input and output data for the FD model are a color eye image and gaze features, respectively. Since it is difficult to prepare a large dataset of actual eye-image and gaze-vector data for training the FD model, estimating the gaze vector from the actual eye image is difficult. To address this, we use the UnityEyes dataset [15], which can generate a million eye images and gaze vectors.

Table 5.1: Two models constructed proposed method.

Method	GE model	FD model
A	Inception-v3	Inception-v3
B	VGG-19	Inception-v3

Regarding the FD model, the Euclidean distance between the estimated gaze feature and ground truth is defined as the loss function:

$$loss_{FD} = \|(\theta_x - \hat{\theta}_x)^2 + (\theta_y - \hat{\theta}_y)^2 + (\theta_z - \hat{\theta}_z)^2 + (P_x - \hat{P}_x)^2 + (P_y - \hat{P}_y)^2\|_2,$$

where $\hat{\theta} = (\hat{\theta}_x, \hat{\theta}_y, \hat{\theta}_z)$ is the ground truth of the gaze vector in camera space, and $\hat{P} = (\hat{P}_x, \hat{P}_y)$ is the ground truth of pupil-center position in the eye image.

As mentioned above, the proposed architecture incorporates two models, the GE and FD models, arranged in parallel. The latter structure of the proposed model is combined with these two models. The pre-logit layer of the GE model and output of the FD model are combined as a concatenation layer as shown in Fig. 5.1. Finally, two fully-connected layers are added at the last stage of the proposed model. Output from the proposed model is an estimated gaze point $G = (G_x, G_y)$.

As for the training process, we train the GE and FD models separately. After training these models, we retrain the GE model and the fully-connected layer with our dataset, as shown in Fig.5.1. The loss function of this process $loss_O$ is defined as follows:

$$loss_O = \|(G_x - \hat{g}_x)^2 + (G_y - \hat{g}_y)^2\|_2.$$

The unit number of the output layer of the FD model is five. However, we present two methods, A and B as show in Table 5.1, and the unit number of the pre-logit layer of GE model depends on the structure: Method A has 2,048 units, and method B has 4,096 units. Thus, the number of units of the concatenation layer depends on whether the GE model uses method A or B.

5.3.2 Prototype device

We developed a prototype of an inside-out camera to demonstrate the proposed method described above. The general structure of our new device is similar what we used in our previous research [78]. The most important difference of the new device is the fixed positional relationship between the eye-imaging camera and the scene-imaging camera. Therefore, our device is designed to place both cameras on the same part as shown in Figs. 5.2, 5.3.

The prototype device simultaneously captures the eye image and the scene image. The dimensions of the prototype device, the distance d between the device and the operating plane, and other interrelationships are illustrated in Fig. 5.3. The device has two joints; the user can adjust the camera in along yaw and pitch axes. The eye camera and scene camera are arranged on the same axis so that both cameras move

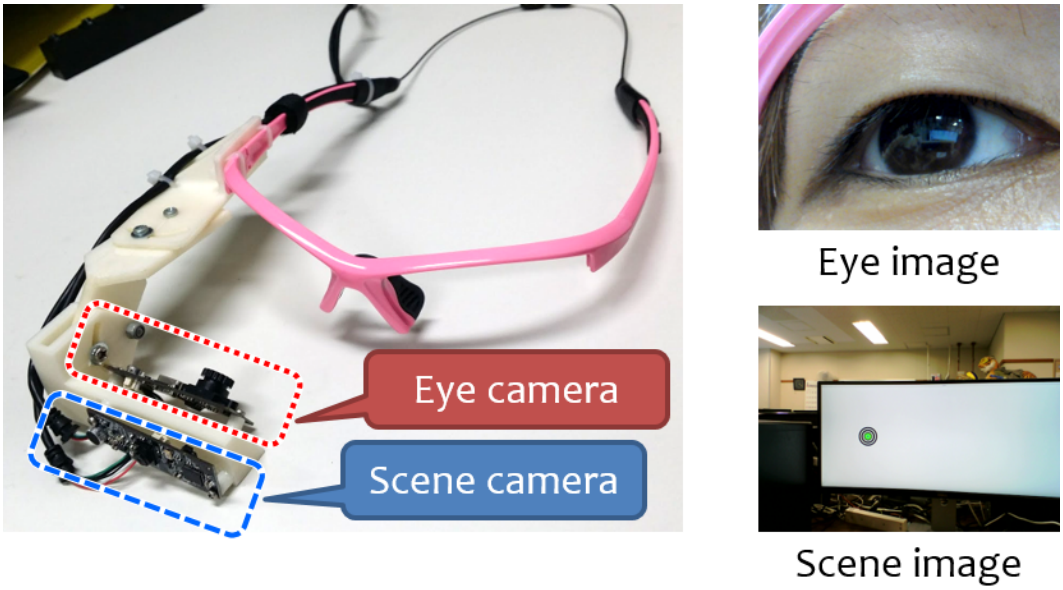


Figure 5.2: Overview of developed prototype inside-out camera.

together when the glasses are moved or when the user reorients the camera. The distance between the user's eye and the eye-imaging camera and the angles of the eye and scene cameras may be changed. The restriction when using the device is that the user's eye must be in the eye image. In other words, if this restriction is satisfied, the position of the device may be moved. Further, the device does not require any infrared light source, which also makes it more lightweight.

5.4 Experiment

5.4.1 Dataset

A CNN is a supervised learning method that requires a large dataset to train a model. Also, various ground-truthing measurements are required to make the model more accurate. One of the most famous datasets for gaze estimation is the MPIIGaze dataset [12], which contains 10,848 full-face images with manually annotated facial landmarks from 15 subjects. However, the input for our proposed method is not a full-face image but an eye image (Fig. 5.1). For this study, we originally collected more than 51,000 eye images from 13 subjects.

For the calibration-free GES, the dataset has to contain various camera angles. Our new dataset was constructed by the same process as used in our previous study [78]. Details about the dataset collection process are described as follows:

1. We used a 34-inch widescreen display for the data-collection experiment. The distance between the subject and the display was set to $d = 90$[cm]. We captured

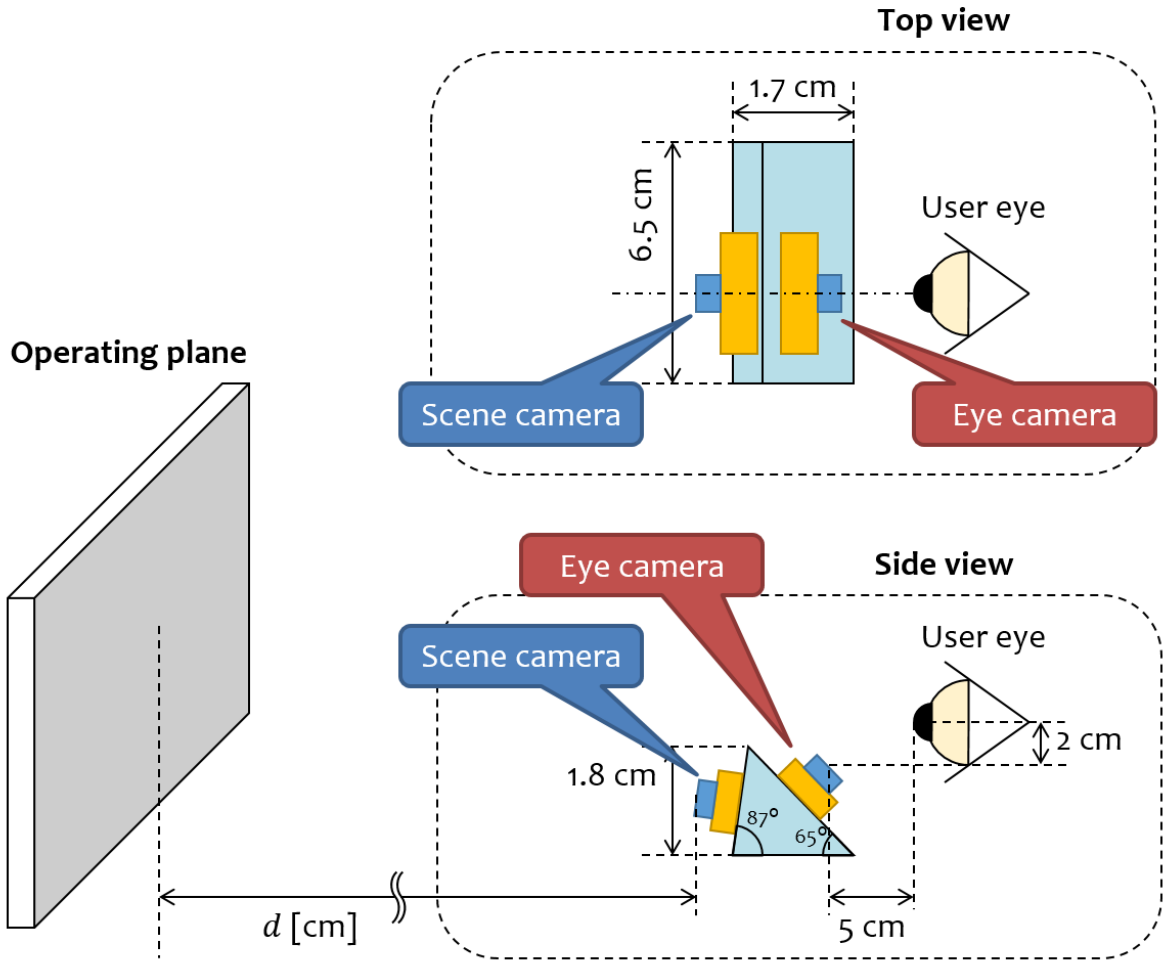


Figure 5.3: Dimension of prototype device.

the images for the dataset in a university laboratory room with sufficient light from both natural and fluorescent light sources.

2. We developed a capture program to efficiently collect various eye images.

3. The display that we used was divided into 49 ($= 7 \times 7$) sections. In a section of the display randomly selected by the capture program, a marker was shown, and at its center was an arrow.

4. First, the capture program shuffled the order of the marker positions to make their positions unpredictable. The subject had to gaze at the marker without any head movement.

5. The subject pressed the direction key corresponding to the marker arrow. The capture program stored the eye and scene image after a subject pressed the key.

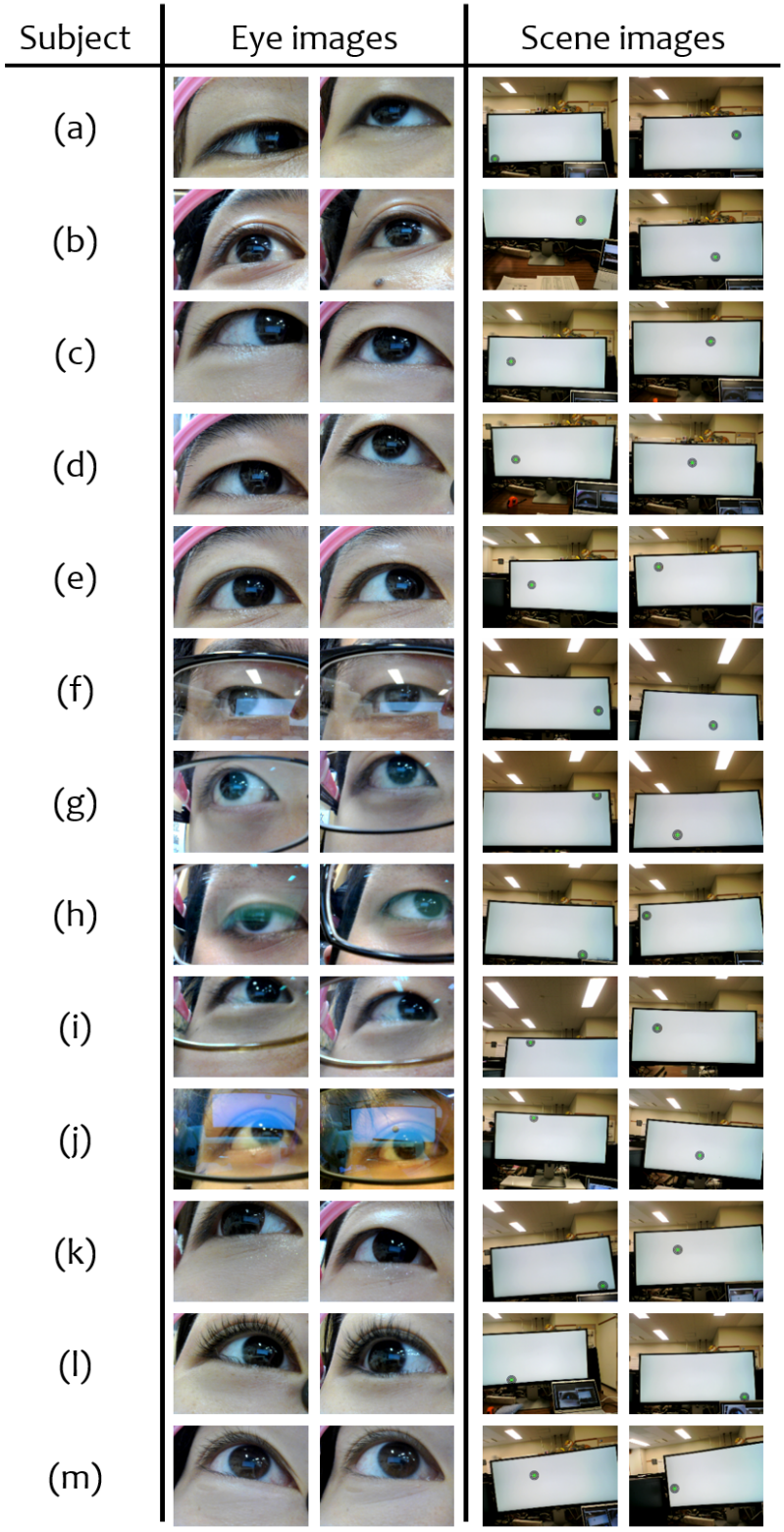


Figure 5.4: Sample pairs of eye image and scene image.

After the eye and scene images were stored, the marker was moved to the next random position automatically.

6. It was necessary to collect eye images from various camera angles, assuming that glasses were removed. The subject first wore glasses at standard nose-pad position as the reference position. After collecting 49 pairs of eye and scene images, the subject moved the glasses position 1cm from the reference position. Each subject was asked to repeat the previous process nine times to collect $49 \times 9 = 441$ image pairs, which we called one set.

7. Each subject had to collect nine sets on different days and at different times of day. Thus, the total images from each subject was $441 \times 9 = 3,969$ pairs. The collection task involved 13 subjects, who therefore generated 51,597 pairs.

After collecting all eye and scene images, we excluded eye images in which the subject was blinking or closed an eye. The number of remaining images were 50,591. Next, we performed annotation tasks. For ground truth of the gaze point $\hat{g}$, we used the center of the marker from the scene image because the subject had gazed at the marker during the collection process. To avoid wrong annotation and to make the label reliable, first, we converted the scene image from red-green-blue (RGB) color space to hue-saturation-value (HSV) color space and applied the thresholding method to detect the green region, which is the color of the marker centers. Finally, two persons visually checked all the scene images fixed errors where found. For ground truth, pupil-center positions $\hat{P}$ of the eye image were annotated manually.

Sample eye images and scene images from our dataset are shown in Fig. 5.4. They were generated by 13 subjects-five males without glasses (a)-(e), five males with glasses (f)-(j), and three females with contact lenses (k)-(m). The distribution of the ground truth of gaze points is given as percentages in Table 5.2. This distribution shows that our dataset has a good distribution across the operating plane.

To train the FD model, we used the UnityEyes dataset [15]. We set the distance between eye model and camera position as far, medium, or near as shown in Fig. 5.5. Then various eye images (totaling 640,297) were generated randomly by our original program.

5.4.2 Evaluation experiment of FD model

For the training of the FD model, we used a pretrained model of the ImageNet dataset [16] and set the dropout rate to 50%. Then we fine-tuned the model training process to avoid overfitting. For this task, we prepared eye images from the UnityEyes dataset and used the Adam optimization method [79] to train all the proposed models using a batch-size set to 20 images. We also applied the data augmentation (DA) method to improve the detection accuracy. Because of the difference between the eye images captured by our prototype inside-out camera and the UnityEyes images, we randomly adjusted the training eye images with values of hue (-0.5 to 0.5), saturation (0.2 to 1.2), brightness (-0.4 to 0.4), and contrast (0.2 to 1.2).

Table 5.2: Distribution of ground truth of gaze point on the scene image [%].

	Horizontal section										
	1	2	3	4	5	6	7	8	9	10	Total
1	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.03	0.05	0.01	0.19
2	0.06	0.07	0.10	0.13	0.16	0.13	0.14	0.18	0.22	0.04	1.22
3	0.38	0.46	0.49	0.63	0.62	0.51	0.50	0.67	0.66	0.13	5.04
4	0.67	1.17	1.39	1.49	1.47	1.25	1.30	1.50	1.46	0.24	11.92
5	1.08	1.88	2.14	2.31	2.23	1.89	1.99	2.19	2.11	0.42	18.26
6	1.09	2.07	2.36	2.61	2.59	2.15	2.23	2.40	2.24	0.47	20.19
7	1.10	2.07	2.43	2.48	2.31	2.07	2.16	2.34	2.05	0.50	19.51
8	0.83	1.51	1.70	1.84	1.67	1.45	1.56	1.68	1.50	0.43	14.18
9	0.49	0.79	0.87	0.88	0.80	0.69	0.87	0.85	0.74	0.32	7.29
10	0.17	0.25	0.25	0.24	0.25	0.25	0.23	0.23	0.21	0.09	2.18
Total	5.87	10.27	11.74	12.62	12.11	10.42	11.00	12.06	11.24	2.66	100.0

Table 5.3: Average errors of FD model.

Features	Without DA	With DA
θ_x [deg.]	7.28	2.84
θ_y [deg.]	8.33	3.47
θ_z [deg.]	5.88	2.07
P_x [pixel]	2.21	0.69
P_y [pixel]	2.70	1.01

As for the evaluation, we randomly selected approximately 61,000 eye images from the UnityEyes dataset and fed them into the proposed FD model. Average loss was mitigated by applying DA, which reduced errors and thus increased the accuracy of every feature as shown in Table 5.3.

The estimated gaze vector θ and pupil-center position P are shown in Fig. 5.6, in which red points are detected pupil-center positions and yellow lines are estimated-gaze vectors.

5.4.3 Evaluation experiment with other state-of-the-art models

Of the 13 subjects, we selected three for test data and ten for training data, thereby addressing the potential problem of subject interdependency. To evaluate gaze-point estimation accuracy of the proposed method, we also used several well-known CNN models: AlexNet [61], Inception-v3 [77], VGG-11 [62] and VGG-19 [62], and ResNet-101 [17] and ResNet-200 [17]. These methods each used only one CNN model, compared to our method's use of two. Averaging error as the distance between the estimated gaze point g and gaze-point ground truth $\hat{g}$ for each model is shown in Fig. 5.7.

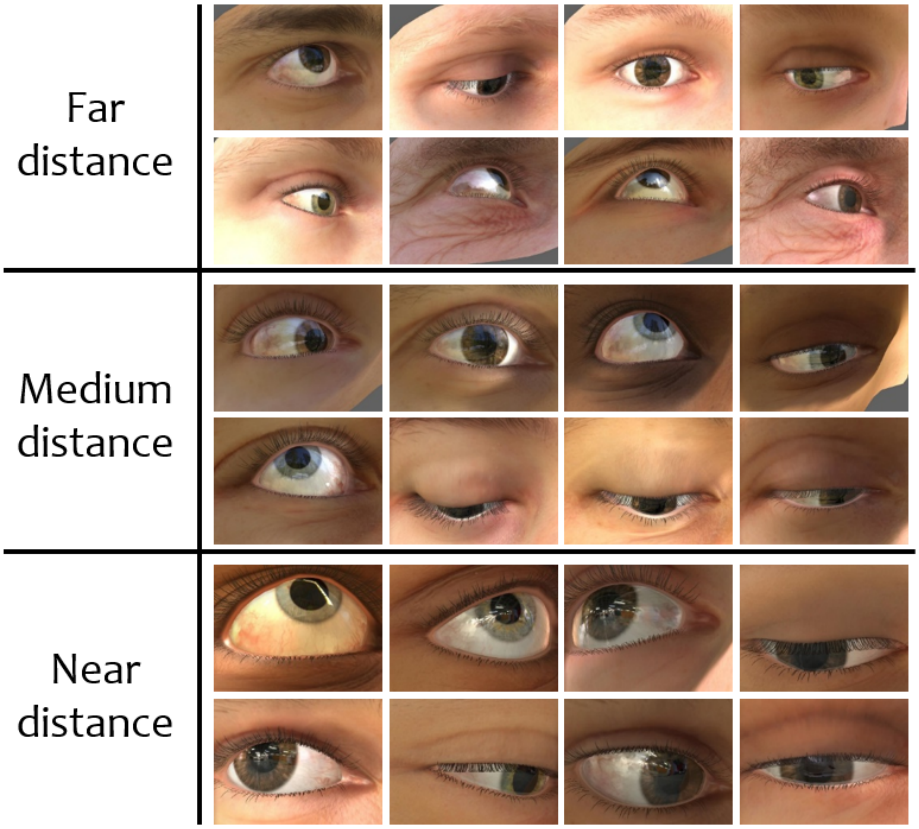


Figure 5.5: Generated eye images by UnityEyes.

Among the six comparable methods that each use one CNN model, the two VGGNets (VGG-11 and VGG-19) attained the highest estimation accuracies, 7.50cm and 7.36cm, respectively. From the gaze-point estimation results, we decided to use Inception-v3 and VGG-19 as GE models. Also, we found that estimation accuracy was improved by integrating the FD model into our proposed method. Method A, which used Inception-v3 for both the GE and FD integrated models, improved estimation accuracy from 9.25cm to 7.17cm; in contrast, method B, which used VGG-19 as the GE model and Inception-v3 as the FD model, attained the highest estimation accuracy of 1.62cm.

Error distance is plotted against success rate in Fig. 5.8. These performance curves show the relationship between the number of successful images for which the error is less than a threshold, and the number of all test images. From these curves, the proposed method B clearly has the highest accuracy of all the methods.

5.4.4 Comparison experiment with commercial products

We compared the developed prototype GES with two commercially available GES products, EMR-9 and Gazo. The evaluation process was almost the same as the

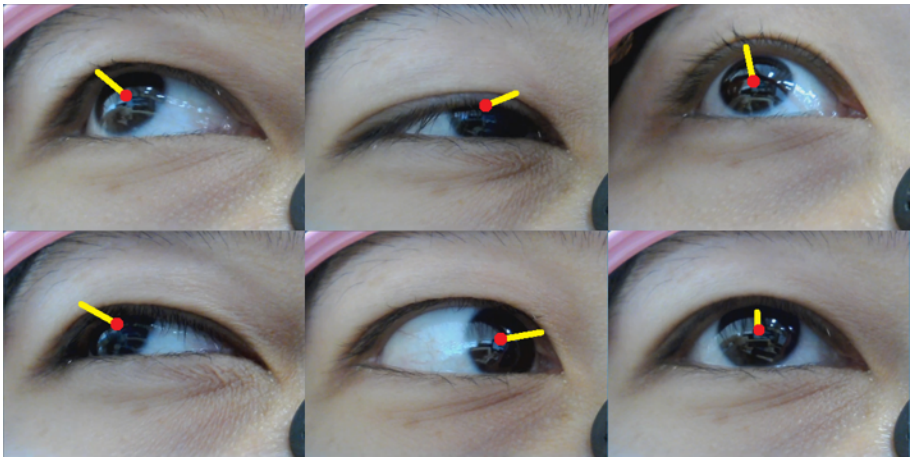


Figure 5.6: Estimated gaze vector by FD model.

Table 5.4: Comparison experiment with commercial products.

	Average error [cm]	
Device	No shift	Shift
Gazo [74]	3.24	10.48
EMR-9 [34]	4.13	8.60
Ours (method A)	—	7.17
Ours (method B)	—	**1.62**

collection process described in Sec. 2.4. In general, this process was as follows: (1) A subject wore the device. (2) A marker automatically appeared at a random position on the display. (3) The subject pressed one of the direction keys. (4) The last two steps were carried out 49 times. (5) The subject slightly shifted the device and repeated the process (steps 2 through 4) nine times. Thus, each subject generated data for 441 trials.

Average errors using each device are shown in Table 5.4. In the table, "shift" means that the device was shifted nine times, and "no shift" means that the device was kept in the standard position. We confirmed that the operating error of the comparison products is 4cm. However, when the device is shifted, which increases the error, the user has to recalibrate the device. In contrast, our proposed device is designed for calibration-free operation and is robust toward device shifts. Our device, using method B, attained the smallest error, 1.62cm.

5.4.5 Discussions

Based on the results of past experiments, we confirmed that our proposed method has high potential as a calibration-free GES. However, in those experiments, the distance d between the scene camera and operating plane was set to 90cm. Therefore, as an additional experiment, we verified the gaze-point estimation accuracy when d was reset

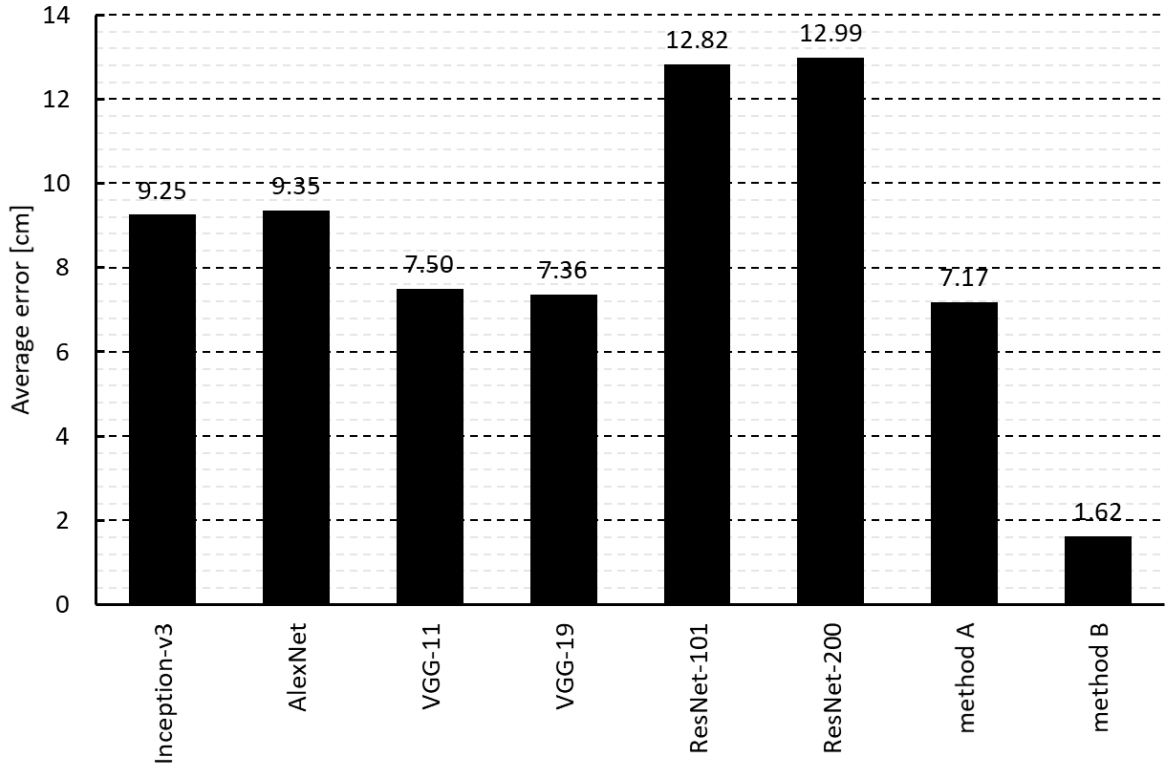


Figure 5.7: Gaze point estimated results.

to 80cm and 100cm, given that the model had been trained with the dataset assuming $d = 90$[cm]. The experimental results as shown in Table 5.5 show that when we used a different d for the test data ($d = 80$[cm] or 100[cm]) from that of the training data ($d = 90$[cm]), accuracy decreased. However, the error is still almost equivalent to that of comparable products as shown in Table 5.4.

We next investigated the error distribution of gaze estimation for the scene image. We divided the scene image into 100 sections (10 rows by 10 columns) and calculated the average error for each section, using proposed method B. The error distribution shown in Table 5.6 shows that errors in the upper half of the scene image area are smaller than in the lower half, which indicates that gaze position is difficult to estimate for the lower area. We presume that when the user looked at the lower area, the gaze

Table 5.5: Estimated error on different operating distances.

	Average error [cm]	
Distance [cm]	Method A	Method B
80	7.52	4.35
90	7.17	1.62
100	7.23	4.85

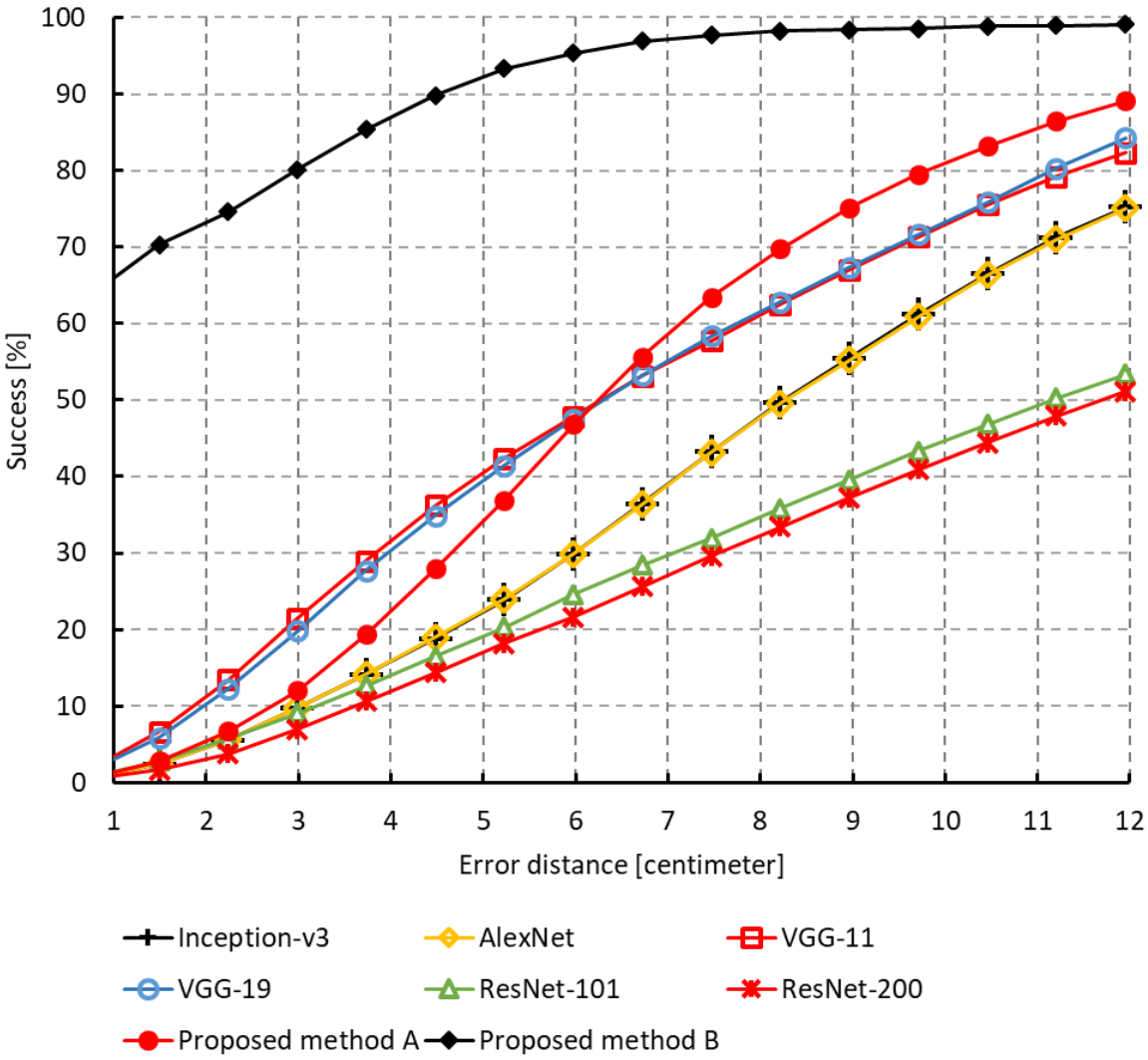


Figure 5.8: Performance curves.

vector was difficult to detect because the eyelid obscured the pupil. In contrast, when the user looked at the upper part, the eyelid was opened up, the pupil was clear, and therefore it was easy to detect the gaze vector.

Our dataset contains three types of eye images: without glasses, with glasses, and with contact lenses (Sec. 2.4), and we investigated estimation accuracy by eye-image type. Three test subjects were selected for each eye type to generate the test data, and average error was assessed by eye-image type. We excluded some images, such as those in which eyes were closed. In the results shown in Table 5.7, the average errors show that high accuracy was attained by subjects using no glasses or by those using contact lenses, whereas using glasses resulted in low accuracy. The difficulty caused by glasses is that lens distortion strongly affects the eye image. Also, we assume that appropriate training data-eye images involving glasses-are very limited.

Table 5.6: Average error of each section (model B)[cm].

	Horizontal section									
	1	2	3	4	5	6	7	8	9	10
1	—	0.51	0.23	0.05	0.39	—	0.68	0.52	—	0.74
2	0.01	0.16	0.32	0.38	0.48	0.53	0.59	0.60	0.71	0.58
3	0.43	0.25	0.29	0.32	0.35	0.57	0.53	0.58	0.62	0.63
4	0.72	1.39	1.03	0.99	0.75	0.85	0.89	0.87	0.99	0.77
5	3.07	1.79	1.38	1.36	1.13	1.21	1.19	1.08	1.25	1.04
6	1.74	2.07	1.75	1.32	1.14	0.96	1.13	1.52	1.56	0.98
7	2.98	2.33	1.86	1.76	1.30	1.25	1.57	1.77	1.60	0.93
8	3.68	2.83	2.64	1.81	1.60	1.51	1.59	1.89	1.60	0.70
9	7.57	3.36	3.25	3.33	2.48	2.25	2.40	2.50	2.62	2.98
10	6.34	7.54	3.56	5.41	4.93	3.58	2.78	5.20	2.67	4.48

Table 5.7: Average error of three types of eye images.

		Average error [cm]	
Image type	# of images	Method A	Method B
Without glasses	11,729	7.17	1.62
With glasses	11,579	15.28	9.47
With contact lens	11,522	5.62	1.86
Total	34,830	9.35	4.31

5.5 Conclusion

We propose a CNN-based calibration-free gaze-estimation method for a wearable inside-out camera. Calibration of existing devices has proven difficult and time consuming. Also, in most situations, the positional relationship changes between camera and eye, and therefore, the user has to go through the recalibration process. Our proposed method, which does not require calibration, consists of multiple CNN models to detect the gaze point on the scene-camera image from the eye-camera image. Using the prototype device that we developed to demonstrate the proposed method, we compiled an original dataset that contains more than 51,000 images. We also ran evaluation experiments using other state-of-the-art methods. Experimental results show that our proposed method attained the highest accuracy among all the methods that we assessed.

Chapter 6

Character input system

6.1 Introduction

This chapter presents the character input system by using wearable camera based GES. For physically disorders patient, who cannot move the organ and difficult to communicate, two-way communication is necessary for improving quality of life. The character input system [30], [31] can be used to help the patient to communicate with other.

We developed a text input system by using an inside-out camera [31]. The handcraft method can be input the text, however, there are several problems such as the gaze point cannot estimate accurately and the user cannot input character correctly. The goal of this chapter is creating the robust character input system by using Convolutional Neural Network (CNN). To solve the problem, we designed the CNN for detecting the pupil center and eye state [78] as mention in chapter 4. For the character recognition process, our system uses the Google Cloud Vision service for input character correctly. This service provides many functions. For example, the optical character recognition, object recognition, facial detection, and landmark detection. We can use these function via Google Cloud Vision Application Programming Interface (GCV-API).

6.2 Proposed system

A user wears the inside-out camera and looks at a character such as the character board as shown in Fig. 6.1. When user gazes the same character more than two seconds, the system determines that the character is desired character. As for the user, he cannot visually check whether the desired character recognized correctly. Them, our system sounds an inputted character to user in order to make the user check the input character. After the user finishes inputting the text, the system outputs the text by speech synthesis and communicate the message to speech partner. The main window of our system is shown in Fig. 6.2. The proposed system is composed of two systems. The first is gaze estimation, and the other is character recognition. We describe the detail of the proposed system in the following subsections.

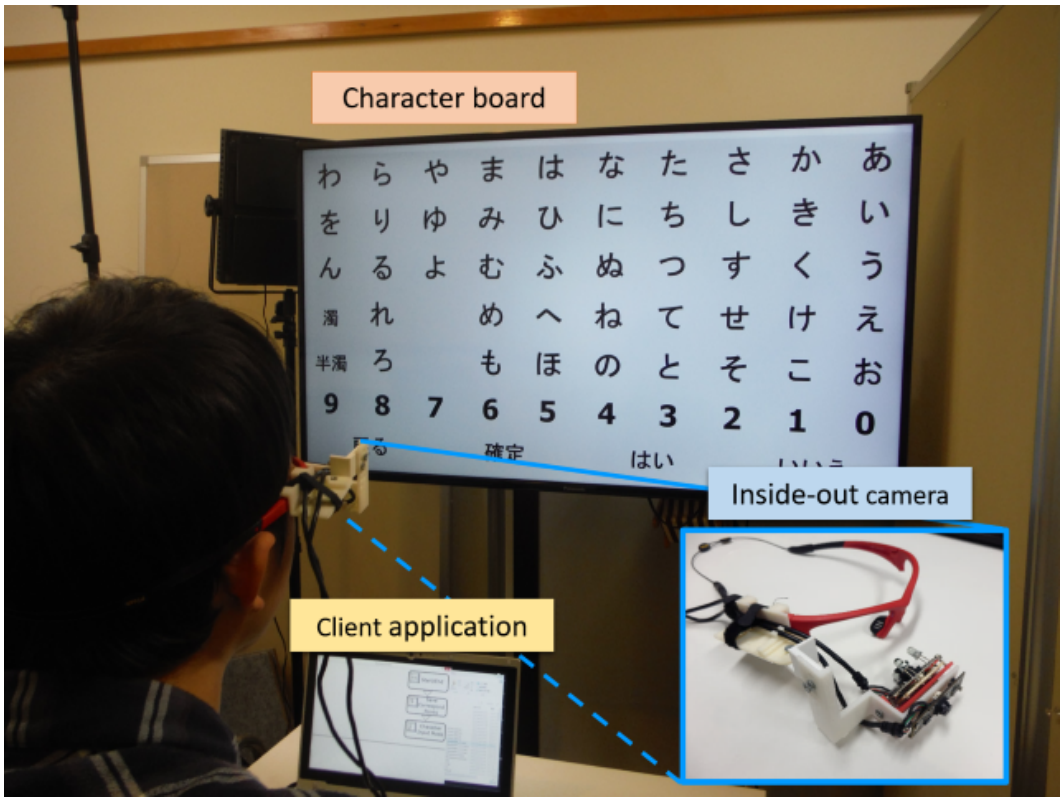


Figure 6.1: Demonstration scene.

Our system designed to classify the Hiragana character board as shown in Fig. 6.5. The designed board is composed of 46 Hiragana character, number and command ("yes", "no", "backspace", and "enter"). There are more sounds in Japanese than just the 46 Hiragana characters. In order to make the other sounds, the voiced sound and semi-voiced mark was added to the character board. The "濁" is used when entering voiced sound, "半濁" is used when entering semi-voiced sound.

6.2.1 Gaze estimation

The gaze estimation system using CNN detects the pupil center position. We train the CNN model with the 19,600 eye images that we correct form the inside-out camera as shown in Fig. 6.3. However, when we use the gaze estimation in the actual situation, the pupil may show in the different position that there is no similar sample in the dataset. Thus, we apply the data argumentation to improve the accuracy of the CNN pupil center detection model. We apply the data argumentation for the eye image and the ground truth (the pupil center position) as shown in the following:

- Rotating: The camera can be rotated by weight, we apply the random rotation between -75 and 75 degrees.
- Scaling: To simulate when the eye camera near and far away from an eye. We

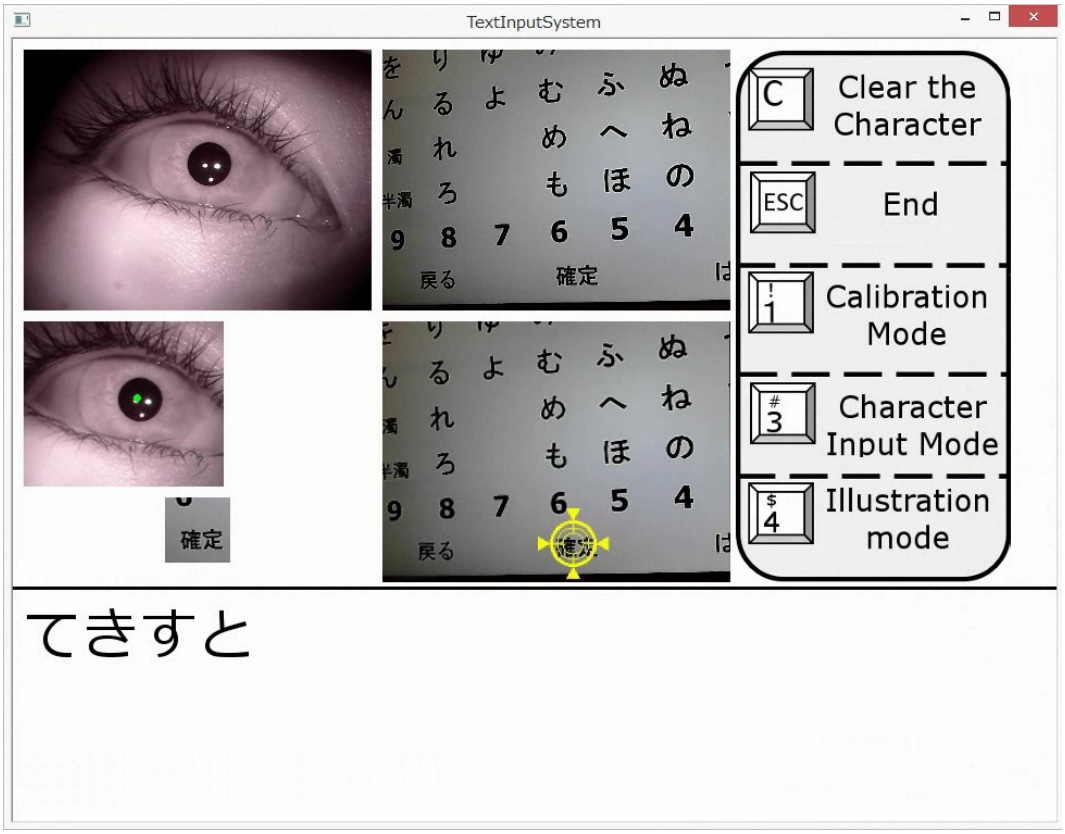


Figure 6.2: Main window of client application.

apply the random scale the image from the previous process between 0.5 to 1.4 time from the original size.

- Shifting: To make the variation of the pupil center position. We apply the random shift the image from the previous process between -20 and 40 pixels in the vertical and horizontal axis.

The important process is calibration process which calculates the coordinate transformation (CT) function between eye image and scene image [31]. In this process, user has to look at the marker point when marker moving around the monitor. We use the marker and pupil center point to create the CT function. This function is used for transfer the pupil center position in eye image into gaze point in the scene image.

To use the character input system in everyday life, the fast response system is required. However, CNN spends many of computational resources. To solve this problem, we set up the server with high computational resource and run the CNN model on this server. We use the representational state transfer (REST) technique based on Hypertext Transfer Protocol (HTTP) to communicate with the client application as shown in Fig. 6.4. Moreover, we proposed new CNN model that has two functions; classify the eye state (close eye or non-close eye), and detect the pupil center position [78]. When user blinks or closes his eyes, our system estimates the close eye state,

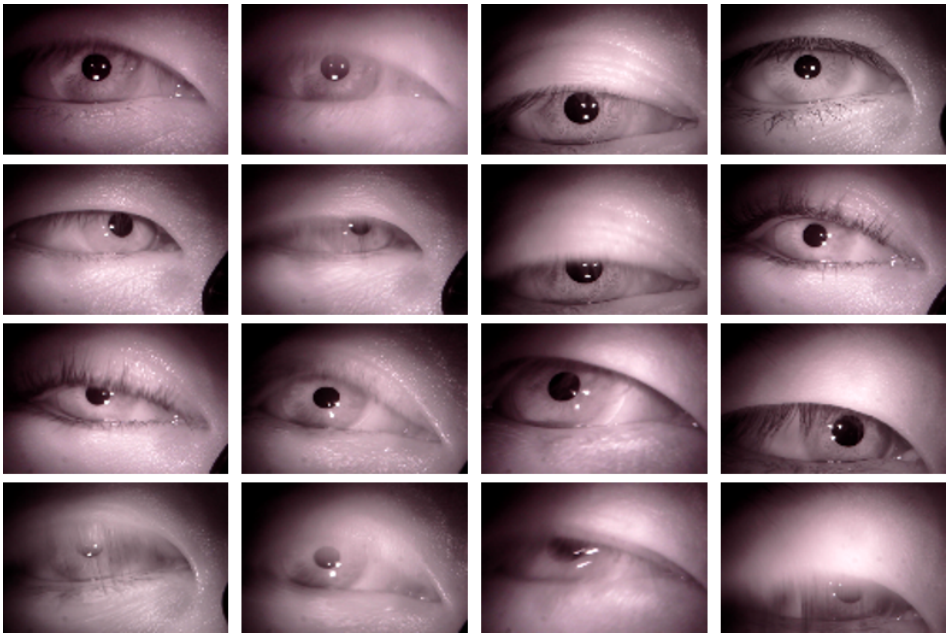


Figure 6.3: Sample eye images from our inside-out camera.

and stop the subsequent character recognition process to avoid wrong input.

6.2.2 Character recognition

After estimating the gaze point, a region of interest (ROI) image around the gaze point is extracted from scene image. When user gazes the same character more than two seconds, the system selects ten ROI images from this two seconds as shown in Fig. 6.6. We transform the ROI images into the binary image. We set the left and right space to 3 pixels from the character and concatenate this images into one big image as shown in Fig. 6.7. Then, this image is sent to the Google Cloud Vision API (GCV-API)

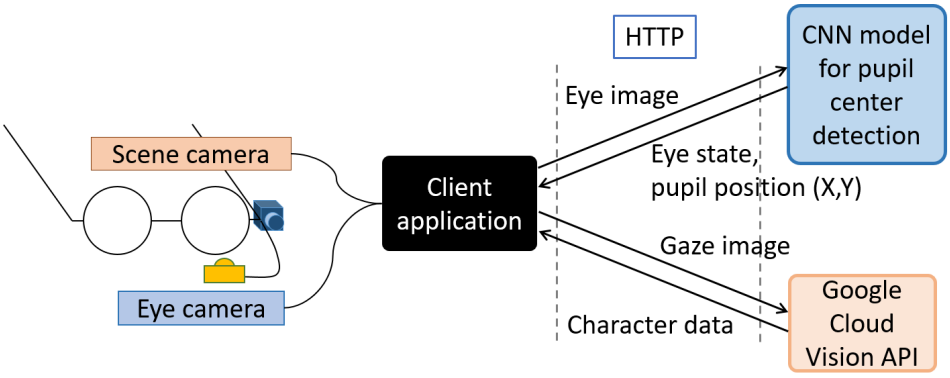


Figure 6.4: System overview.

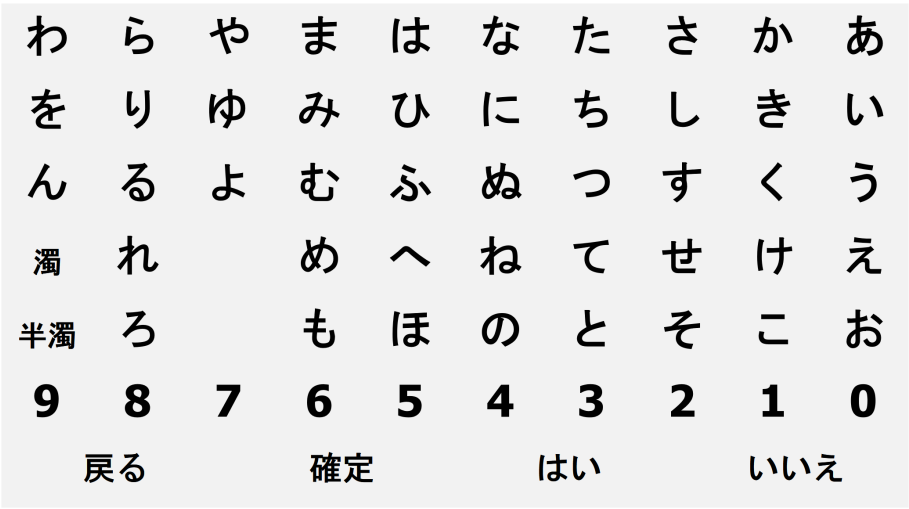


Figure 6.5: Hiragana character board.

Table 6.1: Testing the character recognition.

Transformation method	Range	Image number
Original	-	1
Scale (x,y)	0.8 to 1.2	16
Shifting (x,y)	-10 to 0	24
Rotation	-5, -3, 3, 5	4
Affine (x axis)	-10 to 10	4
Affine (y axis)	-3	1

service to recognize the character. The REST also applies to communications with GCV-API.

The GCV-API will return the multiple results via JSON format. We apply the voting method to decide the estimated character. If we have more than one character that has the same largest voting score, we select the first character.

6.3 Experiment

We need to evaluate the performance of our system. First, we test the character recognition process. We capture the Hiragana character board from scene camera. Next, we split that image to individual character images with 60 by 60 pixels. To test the character recognition process in variation situation, we use the multiple transformation methods as shown in Table 6.1 to generate 50 images for each character. Figure 6.8 shows the result of character recognition process of each character. The result showed the average error of 78.1% was obtained. This result indicated the GCV-API

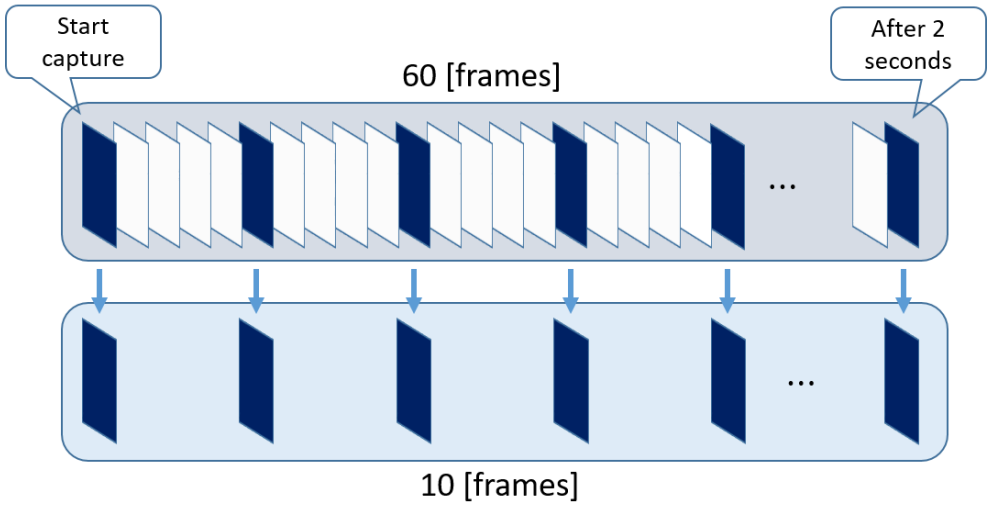


Figure 6.6: Selection the frames.

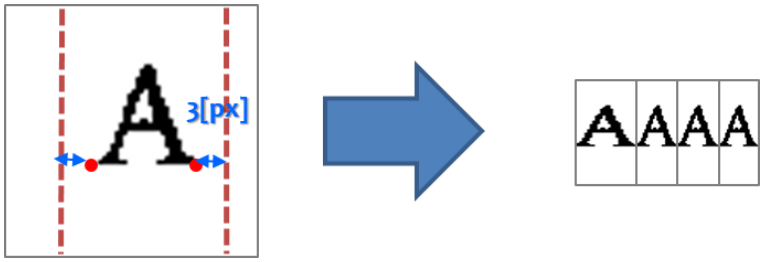


Figure 6.7: Concatenation the character.

is suitable for character input system.

As for the pupil center detection process by using CNN, the average error of 1.5 pixel distance between the estimated point and ground truth was obtained. The computational time is 16 millisecond by using single GPU. We compare the pupil detection with the mean of gradient method [59]. By using mean of gradient, the average error of 2.12 pixel was obtained. This result shows the proposed method obtained higher accuracy.

We implemented the proposed system and demonstrated. We prepared four target texts consists of four to five character length. Next, we asked the six subjects to input that text. Table 6.2 shows the accuracy and time to input the character. This result showed that our proposed system could help the user to communicate with the other.

6.3.1 Improve the character input system

In order to improve the accuracy of the character input system, the device and some method has been added to the system.

The device is changed to the Gazo device has shown in Fig. 6.9. This device is a

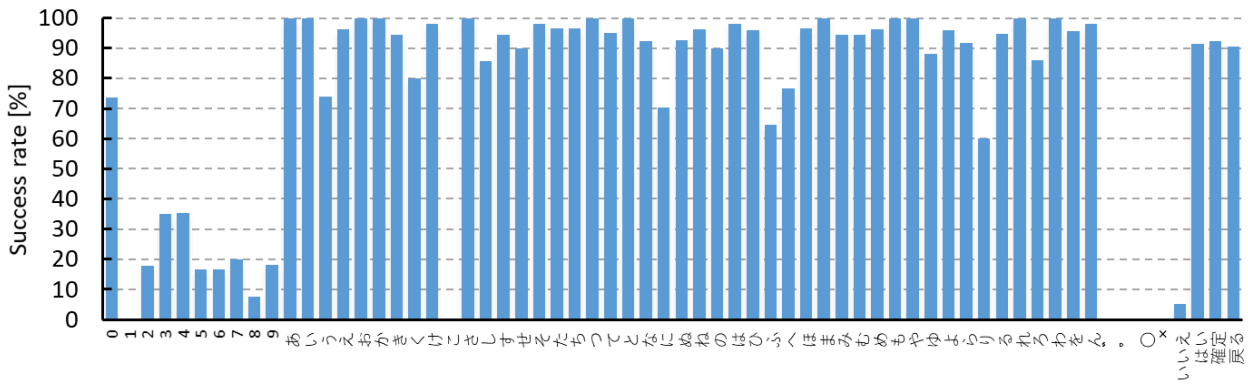


Figure 6.8: Character recognition result.

Table 6.2: Accuracy and time to input the character.

Subject	Times [sec]	Miss rate[%]
1	4.8	34.7
2	4.8	13.4
3	4.9	11.4
4	4.9	3.6
5	6.1	45.1
6	5.7	37.3

product from Gazo company [74]. The camera is more light weight than the previous camera. The eye camera and scene camera is arranged following the user's gaze line. The eye camera is an infrared camera with the infrared LEDs is installed at the glasses frame. When an infrared camera is used to capture eye images, the iris fades out, which makes the pupil clearer. This approach makes the eye image easy to work with.

6.3.2 Improve the gaze estimation method

The CNN model is used to detect the gaze point. The original gaze image is a 80 × 80 pixels around the gaze point. Some gaze image contain the unexpected object, for example, the monitor's boundary or unintentional character. To increase the accuracy of character recognition, the following process describes to create the clear gaze image.

1. First, the binalization method was applied to detect the black character.
2. Next, the contour detection method was applied to detect the edge of the character.
3. The system finds the boundary box of each character and that center point P_C.

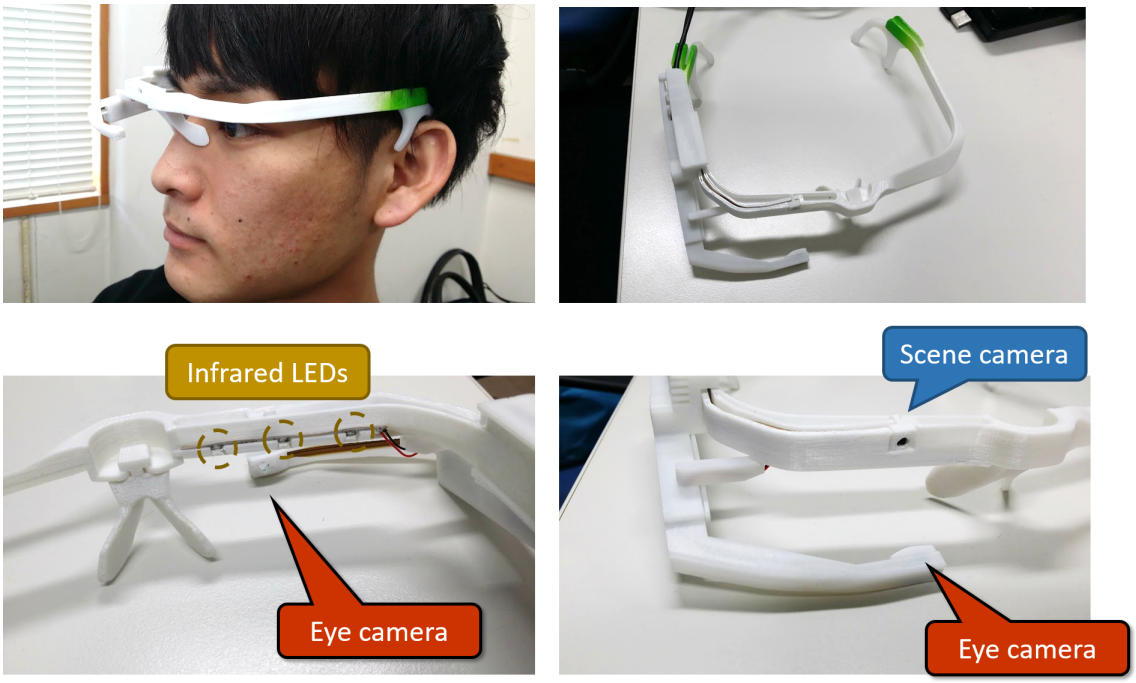


Figure 6.9: Inside-out camera (Gazo camera).

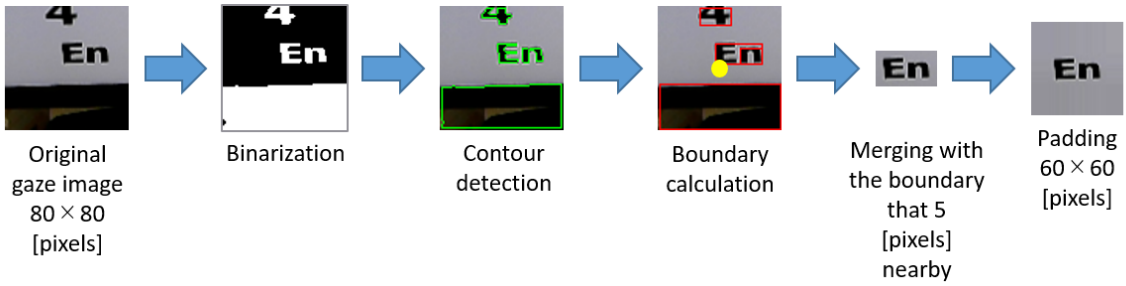


Figure 6.10: Process to create the clear gaze image.

4. The system detects the character point P_C that nearest distance from gaze point P_g^S.

5. Merging with the neighbor character that have a five-pixels nearby the character from previous process.

6. Padding the previous image to 60 × 60 pixels by using replicated method.

Figure 6.10 shows the process to create the output gaze image.

6.3.3 Improve the character recognition method

The previous system using the GCV-API for classify the character. The OCR function of GCV-API do not designed to detect the separate character. In order to improve

Figure 6.11: Experimental scene.

the character recognition, the proposed character recognition model uses the Inception V3 [77] for classify the character from gaze image. This model is a well known CNN model that obtained the impressive result. We train the model with the national diet library (NDL) Lab dataset [80]. The NDL is the national library of Japan and among the largest libraries in the world. They collected this dataset from the material of the library. This dataset is composed of Hiragana character image from handwriting and printed text. The total images is about 80,000 images. For the number, we use the Chars74K dataset [81] for training the model. This dataset consists of English alphabet and number that generate by multiple font. They also have a handwriting and character's image from street photo around the city. The proposed CNN model uses only the number to learn to classify the number. For the command and voiced modification marks, we create the original dataset capture by our inside-out camera. Moreover, the data argumentation method is applied to avoid the over fitting and improve the accuracy of the model.

6.4 Comparison result

Our CNN model was installed on the server computer with CPU Intel Core i7-8700K 3.70GHz and GPU GeForce GTX 1080ti. The client application runs on laptop and communication with the server over the Internet. The proposed system was evaluated by five subjects. They need to sit in front of the character board and create the text by looking at the Hiragana board that shown on the monitor. Figure 6.11 shows the

Table 6.3: Target words, the number of letters is include enter command.

Number of letter	Japanese word		
3	ねつ	そと	よむ
4	ひとつ	ひじ	せなか
5	てれび	でんき	のみもの
6	はみがき	めぐすり	ほとんど
7	むしねがね	しんかんせん	かんがえる

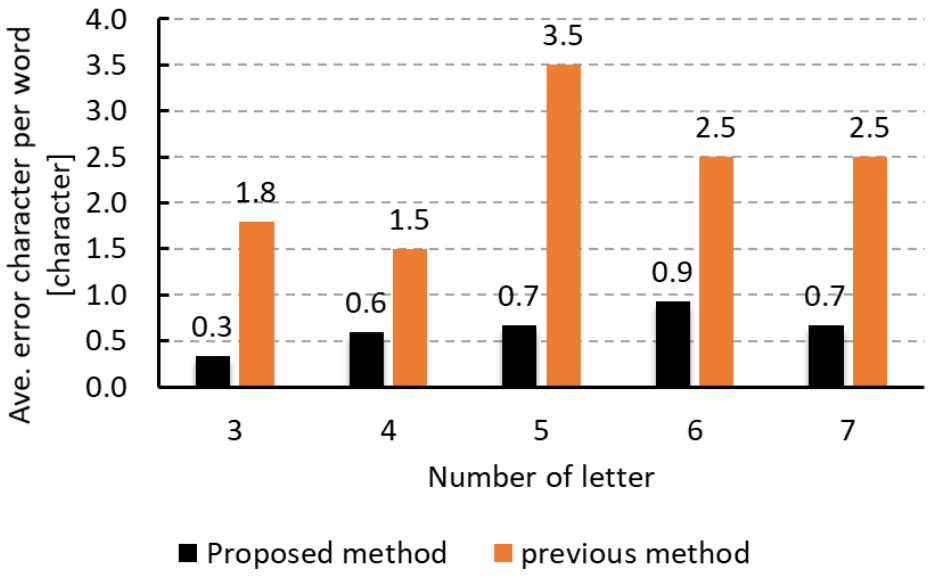


Figure 6.12: Average error character per word.

experimental scene. The monitor is used in this system is 50-inch LCD (screen size is 109.6cm × 61.6cm). The subjects were tasked with creating the word as shown in Table 6.3.

Figure 6.12 shows the average error character per word that compares with the proposed and previous method. This result shows that the proposed method makes more accurate than previous method. The result of Table 6.4 shows the average time consumption by entering the character. By applying the RESTful method, the average time per character is almost the same as the handcrafted method.

6.5 Conclusion

This chapter presented the gaze-based character input system which has a potential to input the high accuracy of pupil center estimation and reliable text input. Furthermore, by applying REST approach, our system can use the power of CNN on low

Table 6.4: Average time consumption per character.

Number of letter	Proposed method [second]	Previous method [second]
3	6.1	4.5
4	5.1	4.2
5	5.3	4.3
6	4.9	4.4
7	5.0	4.4
Average	5.3	4.4

computational resource computer. The GCV-API has many functions, for example, object recognition, facial detection, and landmark detection. It is possible to apply these functions to our system.

Next, the improve method is applied to the system. The CNN model for character recognition is created. By using this method the result shows that the accuracy is improved. The data argumentation technique is applied in the training process to avoid the over-fitting and improve the accuracy of the model.

Chapter 7

Conclusions

7.1 Summary and discussion

The vision-base gaze estimation system (GES) has the capability to improve the quality of life for everyone, especially person with a disability. The vision-base GES involves multiple cameras, and such systems can estimate gaze direction and what a user is looking at. However, an end-user is difficult to access the ability of commercial GES device because of the high price and difficult to a user. The budget GES device can be created with a general web camera and attach into the glasses frame. We create the GES device called inside-out camera by using the budget camera and 3D printer. The common method to estimate the gaze point from the vision-based GES is detected the pupil center position. However, the human eye has a variable characteristic and the blinking make the reliable pupil detection is a challenging problem.

The first contribution of this thesis is to create the frameworks to effectively estimate the pupil center position. The first section of this thesis, the handcraft method is used to estimate the pupil center position. In the chapter 2, The separability filter, and gradient method are implemented for detecting the pupil point. The separability filter is the most useful for shape-based recognition. This method can be used to detect the object from the image. This method using the two-dimensional filter to detect the circle shape from the image. However, when human blink this method obtains the low accuracy. The proposed handcraft approach use the gradient value and RANSAC method to create ellipse fitting. First, the approximate pupil position is estimated by using the gradient value of the image. The edge detection was applied around the estimate pupil point to detecting the pupil edge. From the pupil edge points, the RANSAC method was applied to minimize the ellipse equation. The result shows the proposed handcraft method has improved the performance in term of computation time and accuracy compare with separability filter.

The proposed handcraft approach show good performance. This method does not the static shape to detect the pupil. However, when the user closes the eye, have no eye present in the image, or have a large unexpected object in the image, the accuracy will be decreased significantly. For the handcraft method, the researcher has to create a straightforward algorithm to solve the specific problem. On the other hand, the learning-base method has the potential to solve the general problem that becomes the

main focus of this thesis. The CNN is a most famous learning-based method that proven in many problem and challenge. This method can detect or recognize the object with a complex shape and get the impressive accuracy to compare with other learning base method. The CNN using a lot of kernels convolute with the input image to detect the specific feature point and send to the neural network layer to obtain the result. This thesis uses the CNN to create the accurate GES.

The dataset is the important factor for the learning-based method. The amount and variety of the dataset have to cover the use case. Furthermore, the reliable ground truths also relate with the accuracy of the CNN. In the chapter 4, the original eye image dataset was collected. This dataset contains the blinking image and gaze direction. The ground truths include pupil center positions and eye states (open, medium or close the eyes). The process of annotation the reliable ground truth was created. The proposed CNN framework contains two CNN model. The first called classification model used for classifying the eye state. The second model called the regression model used for detecting the pupil center position. The AlexNet was modified to have a compatibility with the eye detection problem. Moreover, we compared the proposed CNN framework to other well-known CNNs used in feature point detection research. The results show that the proposed CNN model has the potential to classify the eye state. Moreover, the accuracy of the pupil detection is better than that of the other CNN model.

After we got an accurate pupil center position, the character input system was created. This application is very useful for physically disorders patient, who cannot move the organ and difficult to communicate. A user wears the inside-out camera and creates the word by looks at the character board, the system outputs the text by speech synthesis and communicate the message to speech partner. The CNN is consumed high computation resource. The proposed character input system uses the representational state transfer (REST) technique to bring the power of CNN to the low-resource PC. The client application communicates with the CNN server via Hypertext Transfer Protocol (HTTP). Chapter 6 presents proposed system. The proposed system is composed of two systems. The first is gaze estimation, and the other is character recognition. First, we use the Google Cloud Vision API (GCV-API) for the character recognition. Next, we create own CNN model for character recognition by using the public dataset and data argumentation technique. The result shows the proposed pupil center detection model has the potential to create the character input system.

The second contribution of this thesis is to create the calibration-free GES. The gaze position can be estimated by using the pupil center position. The coordinate transfer (CT) function was created by the process called calibration process. However, when the wearable camera moving during the use case, user have to recalibrate to update the CT function. The static CT function cannot estimate the gaze position in this situation. However, the CNN has the capability to estimate the gaze point without the calibration process. Chapter 5 show the process to collect the dataset. We use the original dataset with the synthesis of eye images called UnityEyes to train the model. The proposed calibration free mode is composed of two CNN models, for which the input data is only an eye image. The first model is called the gaze-estimation (GE) model. It is based on two well-known CNN models, Inception-v3 and

VGG-19. The second model called the feature-detection (FD) model. This model is used to detect the gaze feature information of a gaze vector and pupil center position. The FD model learns to estimate the gaze vector and pupil center position by using UnityEyes dataset. We also carried out evaluation experiments using other state-of-the-art methods. Experimental results show that our proposed method obtained the highest accuracy among all the methods tested.

7.2 Future research

The CNN has shown the impressive result in GES. The calibration-free is an important feature for improve the accuracy and bring the GES to everyone. However, the dataset to train the calibration-free model only 15 subjects. We use the UnityEyes to overcome the lagging of vanity in the original dataset. However, the accuracy can be improved by adding the subjects and number of the dataset. However, the structure of the CNN model effect with the gaze estimation performance. This thesis using the state-of-the-art model, this model designed for general object classification. We can improve the accuracy by creating the optimize CNN model. The AutoML [82] that automates the design of machine learning models is one of the interesting methods to create the optimize CNN model for the calibration-free system.

References

[1] Blausen.com staff (2014). Medical gallery of blausen medical 2014, 2014. [Online; accessed 20-August-2018].

[2] David Robinson. A method of measuring eye movement using a scleral search coil in a magnetic field. *IEEE Trans. on Bio-Medical Electrics*, 10(4):137–145, 1963.

[3] Bulling Andreas, Roggen Daniel, and Trster Gerhard. Wearable EOG goggles: Eye-based interaction in everyday environments. In *Extended Abstracts on Human Factors in Computing Systems*, pages 3259–3264, 2009.

[4] Yoshio Ishiguro, Adiyan Mujibiya, Takashi Miyaki, and Jun Rekimoto. Aided eyes: Eye activity sensing for daily life. In *1st Augmented Human International Conference (AH2010)*, pages 169–175, 2010.

[5] Dongheng Li, Jason Babcock, and Derrick J. Parkhurst. openeyes: a low-cost head-mounted eye-tracking solution. In *2006 Symposium on Eye Tracking Research and Applications*, pages 95–100, 2006.

[6] Goto Yuto and Fujiyoshi Hironobu. Recovering 3-D gaze scan path and scene structure from inside-out camera. In *4th Augmented Human International Conference (AH2013)*, pages 198–201, 2013.

[7] Tobii AB. Eye tracking technology for research - tobii pro. https://www.tobiipro.com/.

[8] SensoMotoric Instruments (SMI). Eye tracking solutions by smi. https://www.smivision.com/, 2018.

[9] Pupil Labs GmbH. Pupil labs - pupil. https://pupil-labs.com/pupil/, 2018.

[10] Scientific figure on researchgate. https://www.researchgate.net/NACs-EMR-9-eye-mark-recorder-left-photo-by-NAC-reproduced-with-permission-and-the_fig2_283247791.

[11] Richard A Sturm and Mats Larsson. Genetics of human iris colour and patterns. *Pigment cell & melanoma research*, 22(5):544–562, 2009.

[12] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 4511–4520, 2015.

[13] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3D gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014.

[14] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *IEEE International Conference on Computer Vision (ICCV2015)*, pages 3756–3764, 2015.

[15] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, pages 3476–3483, 2013.

[19] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *Proceedings of the European Conference on Computer Vision*, pages 1–16, 2014.

[20] The EyeTribe. The eye tribe. http://theeyetribe.com.

[21] Tobii AB. Tobii pro x3-120 screen-based eye tracker. https://www.tobiipro.com/product-listing/tobii-pro-x3-120/.

[22] Cihan Topal, Atakan Dogan, and Omer Nezih Gerek. A wearable head-mounted sensor-based apparatus for eye tracking applications. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2008. VECIMS 2008. IEEE Conference on*, pages 136–139. IEEE, 2008.

[23] Hironobu Fujiyoshi, Yuto Goto, and Makoto Kimura. Inside-out camera for acquiring 3D gaze points. In *Proceedings of the Workshop on Egocentric (First-Person) Vision in conjunction with CVPR*, 2012.

[24] Masanari Takaki and Hironobu Fujiyoshi. Traffic sign recognition using SIFT features. *IEEJ Trans. on Electronics, Information and Systems*, 129(5):824–831, 2009.

[25] Michael S. Devyver, Akihiro Tsukada, and Takeo Kanade. A wearable device for first person vision. In *3rd International Symposium on Quality of Life Technology*, July 2011.

[26] Yoshinari Kameda and Yuichi Ohta. Image retrieval of first-person vision for pedestrian navigation in urban area. In *International Conference on Pattern Recognition (ICPR)*, pages 364–367, 2010.

[27] Jia Xu, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M. Rehg, and Vikas Singh. Gaze-enabled egocentric video summarization via constrained submodular maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 2235–2244, 2015.

[28] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012)*, pages 2847–2854, 2012.

[29] Andrea Mazzei, Shahram Eivazi, Youri Marko, Frederic Kaplan, and Pierre Dillenbourg. 3D model-based gaze estimation in natural reading: a systematic error correction procedure based on annotated texts. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 87–90, 2014.

[30] Kiyohiko Abe, Yasuhiro Nakayama, Shoichi Ohi, and Minoru Ohyama. A support system for mouse operations using eye-gaze input. *IEEJ Trans. on Electronics, Information and Systems*, 129(9):1705–1713, 2009 (written in Japanese).

[31] Warapon Chinsatit, Masataka Shibuya, Kenji Kawada, and Takeshi Saitoh. Character input system using gaze estimation. In *Proceedings of the International Conference on Communication Systems and Computing Application Science (CSCAS2016)*, 2016.

[32] Github - pupil-labs/pupil: Open source eye tracking. https://github.com/pupil-labs/pupil.

[33] nac Image Technology. nac EMR-9 eye tracking measurement and analysis system data sheet. EMR9-Data-Sheet-June-09.pdf.

[34] EMR-9 data sheet. http://www.nacinc.com/datasheets/archive/EMR9-Data-Sheet-June-09.pdf.

[35] Niche market - wikipedia. https://en.wikipedia.org/wiki/Niche_market.

[36] Michał Kowalik. How to build low cost eye tracking glasses for head mounted system, 2010.

[37] Jenny Gimpel of University College London. Blink and you miss it! http://www.ucl.ac.uk/media/library/blinking/, 2006.

[38] Anna Rita Bentivoglio, Susan B Bressman, Emanuele Cassetta, Donatella Carretta, Pietro Tonali, and Alberto Albanese. Analysis of blink rate patterns in normal subjects. *Movement Disorders*, 12(6):1028–1034, 1997.

[39] Albert R Wielgus and Tadeusz Sarna. Melanin in human irides of different color and age of donors. *Pigment cell research*, 18(6):454–464, 2005.

[40] Giuseppe Prota, Dan-ning Hu, Maria R Vincensi, Steven A Mccormick, and Alessandra Napolitano. Characterization of melanins in human irides and cultured uveal melanocytes from eyes of different colors. *Experimental eye research*, 67(3):293–299, 1998.

[41] Denis Llewellyn Fox. *Biochromy, natural coloration of living things*. Univ of California Press, 1979.

[42] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[44] Tobii AB. Tobii pro glasses 2 wearable eye tracker. https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/.

[45] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[46] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*, 2015.

[47] Ioana Bacivarov, Mircea Ionita, and Peter Corcoran. Statistical models of appearance for eye tracking and eye-blink detection and measurement. *IEEE transactions on consumer electronics*, 54(3):1312–1320, 2008.

[48] Hu-chuan Lu, Chao Wang, and Yen-wei Chen. Gaze tracking by binocular vision and lbp features. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. Citeseer, 2008.

[49] Cagatay Murat Yilmaz and Cemal Kose. Local binary pattern histogram features for on-screen eye-gaze direction estimation and a comparison of appearance based methods. In *Telecommunications and Signal Processing (TSP), 2016 39th International Conference on*, pages 693–696. IEEE, 2016.

[50] Yasuhiro Ohkawa, Chendra Hadi Suryanto, and Kazuhiro Fukui. Fast combined separability filter for detecting circular objects. In *12th IAPR Conference on Machine Vision Applications (MVA2011)*, pages 99–103, 2011.

[51] David Beymer and Myron Flickner. Eye gaze tracking using an active stereo head. In *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, volume 2, pages II–451. IEEE, 2003.

[52] Sheng-Wen Shih and Jin Liu. A novel approach to 3-D gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):234–245, 2004.

[53] Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. A calibration-free gaze tracking technique. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 201–204. IEEE, 2000.

[54] Rhys Newman, Yoshio Matsumoto, Sebastien Rougeaux, and Alexander Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 122–128. IEEE, 2000.

[55] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32(3):169–179, 2014.

[56] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2033–2046, 2014.

[57] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. Gaze estimation from eye appearance: A head pose-free method via eye image synthesis. *IEEE Transactions on Image Processing*, 24(11):3680–3693, 2015.

[58] Junki Iwagami and Takeshi Saitoh. Easy calibration for gaze estimation using inside-out camera. In *Proceedings of the 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV2014)*, pages 292–297, 2014.

[59] Fabian Timm and Erhardt Barth. Accurate eye center localisation by means of gradients. In *International Conference on Computer Theory and Applications (VISAPP)*, pages 125–130, 2011.

[60] Michael Plotke. Kernel (image processing), 2013. [Online; accessed 20-August-2018].

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS2012)*, pages 1097–1105, 2012.

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[63] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[64] Dongheng Li, David Winfield, and Derrick J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2005)*, pages 99–103, 2005.

[65] Zhonglong Zheng, Jie Yang, and Limin Yang. A robust method for eye features extraction on color image. *Pattern Recognition Letters*, 26:2252–2261, 2005.

[66] Tsuyoshi Moriyama, Takeo Kanade, Jing Xiao, and Jeffrey F. Cohn. Meticulously detailed eye region model and its application to analysis of facial images. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 28(5):738–752, 2006.

[67] Warapon Chinsatit and Takeshi Saitoh. Eye detection by using gradient value for performance improvement of wearable gaze estimation system. In *IEICE Technical Report*, volume 115, pages 149–154, 2016.

[68] Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, and Enkelejda Kasneci. Pupilnet: Convolutional neural networks for robust pupil detection. In *Computing Research Repository (CoRR)*, volume abs/1601.04902, 2016.

[69] In-Ho Choi, Sung Kyung Hong, and Yong-Guk Kim. Real-time categorization of driver's gaze zone using the deep learning techniques. In *Proceedings of the International Conference on Big Data and Smart Computing (BigComp2016)*, pages 143–148, 2016.

[70] Tomas Pfister, Karen Simonyan, James Charles, and Andrew Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proceedings of the Asian Conference on Computer Vision (ACCV2014)*, 2014.

[71] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR2014)*, 2014.

[72] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2014)*, pages 1717–1724, 2014.

[73] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 685–694, 2015.

[74] Ltd. Gazo Co. Gazo corporation. http://www.gazo.co.jp/index.html, 2017.

[75] Elias Daniel Guestrin and Moshe Eizenman. Remote point-of-gaze estimation with free head movements requiring a single-point calibration. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4556–4560, 2007.

[76] Fares Alnajar, Theo Gevers, Roberto Valenti, and Sennay Ghebreab. Auto-calibrated gaze estimation using human gaze patterns. *International Journal of Computer Vision*, 124:223–236, 2017.

[77] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[78] Warapon Chinsatit and Takeshi Saitoh. CNN-Based pupil center detection for wearable gaze estimation system. *Applied Computational Intelligence and Soft Computing*, 2017, 2017.

[79] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[80] National Diet Library (NDL). Research and development for next-generation systems office national diet library (NDL Lab). https://lab.ndl.go.jp/cms/hiragana73, 2016.

[81] Teófilo Emídio De Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. *VISAPP (2)*, 7, 2009.

[82] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*, 2017.

Publications

- **Journal**

  1. Warapon Chinsatit and Takeshi Saitoh, "CNN-Based Pupil Center Detection for Wearable Gaze Estimation System", Applied Computational Intelligence and Soft Computing, Hindawi, Article ID 8718956, Vol.2017 (10pages)

- **International conference**

  1. Warapon Chinsatit, Masataka Shibuya, Kenji Kawada, and Takeshi Saitoh, "Character Input System using Gaze Estimation", 2nd International Conference on Communication Systems and Computing Application Science (CSCAS2016), #21, (5 pages), 2016.
  2. Warapon Chinsatit and Takeshi Saitoh, "Improve the Performance of Eye Detection Method for Inside-Out Camera", 15th IEEE/ACIS International Conference on Computer and Information Science (ICIS2016), pp.415-420, 2016.
  3. Warapon Chinsatit and Takeshi Saitoh, "CNN for pupil center detection", International Conference on Information and Communication Technology Robotics (ICT-ROBOT2016), ThBT2.1, (3 pages) 2016.
  4. Nagisa Kondo, Warapon Chinsatit, and Takeshi Saitoh, "Pupil Center Detection for Infrared Irradiation Eye Image Using CNN", Proc. of SICE Annual Conference, WeA05.2, pp.100-105, 2017.
  5. Naoyuki Kan, Nagisa Kondo, Warapon Chinsatit, and Takeshi Saitoh, "Effectiveness of Data Augmentation for CNN-Based Pupil Center Point Detection", Proc. of SICE Annual Conference, WeA02.4, pp.41-46, 2018.
  6. Warapon Chinsatit, Nagisa Kondo, and Takeshi Saitoh, "Character input system based on gaze input using wearable camera", 10th International Conference on Graphics and Image Processing (ICGIP 2018), (7 pages),2018.

- **Domestic conference**

  1. Warapon Chinsatit, Takeshi Saitoh, "Eye Detection by using Gradient Value for Performance Improvement of Wearable Gaze Estimation System", IEICE Technical Report, vol.115, no.456, pp.149-154. 2016

2. Warapon Chinsatit, Takeshi Saito, “Pupil Center Estimation by using Deep Convolutional Neural Network”, 19th Meeting on Image Recognition and Understanding (MIRU2016), 2016.
3. Rinko Komiya, Warapon Chinsatit, Takeshi Saito, “Eye Detection using CNN for Eye Contact Analysis”, The 69th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu, p.419, 2016.
4. Nagisa Kondo, Warapon Chinsatit, Takeshi Saito, “Study on Effectiveness Gaze Input System using Wearable Camera”, The 69th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu, p.436, 2016.
5. Nagisa Kondo, Warapon Chinsatit, Takeshi Saito, “Study on Pupil Detection from Eye Image using CNN”, 2017 ”Hinokuni - Land of Fire” Information Processing Symposium, 2017.
6. Warapon Chinsatit, Nagisa Kondo, Takeshi Saito, “Gaze-based Text Input System using Wearable Camera”, Dynamic Image processing for real Application workshop 2018 （DIA2018）, IS1-11, pp.100-104, 2018.

Acknowledgements

Foremost, I would like to express my sincere appreciation to my advisor, Associate Professor Dr. Takeshi SAITOH who gave me many precious opinions and guidance in doing this research. I appreciate for encouraging, motivating, endless help and support. I could not have imagined having a better advisor and mentor for my Ph. D study.

Besides my advisor, I would express to thank my committee members, Professor Dr. Takashi OKAMOTO, Professor Dr. Takahiro OKABE, Associate Professor Dr. Masaki OSHITA for serving as my committee members.

I sincerely acknowledge the support from the SGH Foundation Scholarship that selected me as the recipient. Your generosity allows me to take my goals and dreams a reality and this scholarship has afforded me the opportunity to continue my educational pursuits.

Additionally, I would like to thank Mr. Kenji KAWADA, Mr. Masataka SHIBUYA, Miss Rinko KOMIYA, Mr. Keisuke HASHIMURA, Mr. Tomoya KODAMA, Mr. Keishi KAMITAMARI, Miss Nagisa KONDO, Miss Michiko KUBOKAWA and Mr Naoyuki KAN who gave me important help when I am in trouble. I also want to thank everyone in the Takeshi SAITOH Laboratory for your kind cooperation and your advice.

Special thanks for all member of Eiji HAYASHI Laboratory and Mario KOEPPEN Laboratory for kind cooperation to collect the dataset in this research.

My heartfelt thanks to my father Mr. Panom CHINSATIT and mother Mrs. Benyapa WANGPHOL for giving birth to me at the first place. I owe so much thanks to my aunt Miss Jira CHINSATIT who were continuously supporting me throughout my life and leaving me free in all my decisions.

I am very much thankful to Mr. jiraphan INTHIAM for the help and advice to designed the inside-out camera in this research. I also thank my friends Mr. Natthaphon BUN-ATHUEK, Mss Manthana TIAWONGSUWAN, Mr. Chawanut SAIRUK, Mss Wongpramot WALAIKORN, Mr. Sansuwan CHANSATHAPORNKUL, Mr. Supachai SIRISAWAT, Mr. Thitipat PERMPATDECHAKUL and Mr. Teedanai PRAMANPOL for encouraging and support.

I would like to thank Miss Yuka KOBAYASHI and Mrs. Apiradee HORIE who have made a comfortable environment for me during my study time. I wish to extend my thanks to all the departmental staff for helping me during these years of my study both academically and officially. I also thank for everyone I met at Kyushu Institute of Technology (KIT).

Above all I thank Miss Sunan BOONARD my fiancee for all her love, support and assistance in every step of my thesis. The love and support that you gave has motivated me to work harder and to strive towards our goal.

Warapon Chinsatit

NOTE