Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 8718956, 10 pages
https://doi.org/10.1155/2017/8718956

Research Article

CNN-Based Pupil Center Detection for Wearable Gaze Estimation System

Warapon Chinsatit and Takeshi Saitoh

Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680–4 Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan

Correspondence should be addressed to Warapon Chinsatit; warapon@slab.ces.kyutech.ac.jp

Received 30 June 2017; Revised 16 October 2017; Accepted 25 October 2017; Published 4 December 2017

Academic Editor: Fouad Slimane

This paper presents a convolutional neural network- (CNN-) based pupil center detection method for a wearable gaze estimation system using infrared eye images. 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. In addition, this paper presents a process to create an eye image dataset using a wearable camera. This dataset, which was used to evaluate the proposed method, has approximately 20,000 images and a wide variation of eye states. We evaluated the proposed method from various perspectives. The result shows that the proposed method obtained good accuracy and has the potential for application in wearable device-based gaze estimation.

1. Introduction

People obtain various information through the human vision system. By observing eyes, we can observe changes in pupil size, eye direction, and changes in eye state, for example, opening, closing, blinking, and crying. This information can be used to estimate emotions, traits, or interests. To analyze the eye, eye image processing is an important task, and the development and availability of wearable cameras and recording devices have made eye image processing, including gaze estimation, increasingly popular.

A gaze estimation system (GES) involves multiple cameras, and such systems can estimate gaze direction and what a user is looking at. Thus, GESs can estimate objects of interest. One type of GES uses an inside-out camera [1, 2], 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 [3], daily activity recognition [4], reading [5], human-machine interfaces [6], and communication support [7].

It is difficult to detect the pupil center because the eye is a nonrigid 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 it is difficult to detect the pupil center point when a user blinks. Consequently, gaze direction errors can occur.

This research 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 higher 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 [8, 9]; 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.

2. Related Research

Several studies have focused on feature point detection based on eye images [10–13]. Li et al. proposed a hybrid eye-tracking method that integrates feature-based and model-based approaches [10]. 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 [11]. Moriyama et al. developed a generative eye region model that can meticulously represent the detailed appearance of the eye region for eye motion tracking [12]. Chinsatit and Saitoh proposed a fast and precise eye detection method using gradient value [13]. 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 [8]. 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, that is, yaw and pitch. Fuhl et al. proposed a dual CNN pipeline for image-based pupil detection [14]. 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 [15]. 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.

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, for example, when the eyelid overlays the pupil due to blinking. The proposed method is composed of two CNN models. The first model classifies the input image, as shown in Figure 1. The second model operates in a regression mode [16, 17]. Collectively, this CNN model outputs the X- and Y-coordinates of the pupil center point.

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.

3.1. Classification Model.

There are various CNN classification models, and each model has specific characteristics. AlexNet [18] is a well-known model for classification tasks. We selected this model to classify the eye state. We defined three states in eye images; that is, (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).

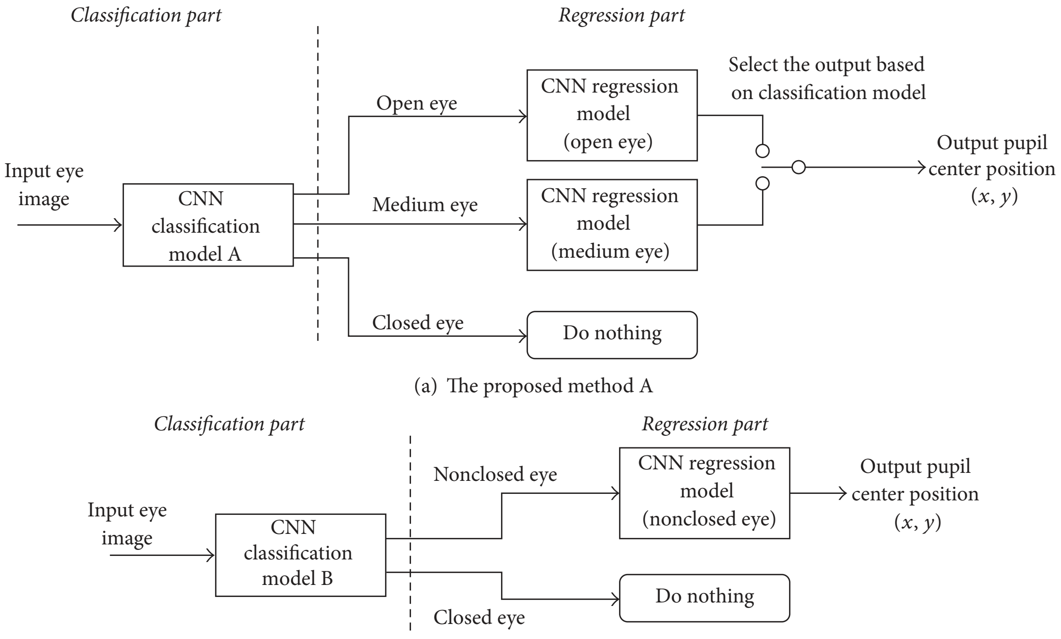
Some studies have used a separate CNN model to perform specific tasks. For example, Sun et al. created multiple models to detect each feature point [16]. We also propose using two methods, which we refer to as methods A and B. For method A, we create a CNN model to classify the input image as open, medium, or closed eye states, as shown in Figure 1(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 1 (row 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 1, row 2) to classify input images as closed or nonclosed eye (i.e., open eye and medium eye images, respectively). This classification model selects only nonclosed eye images and sends those images to the CNN trained using nonclosed eye images, as shown in Figure 1(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:

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

where o_i is an estimation output at i, d_i is a label at i, and N_o is the number of output classification results.


Classification part
Regression part
Input eye image
CNN classification model A
Open eye
Medium eye
Closed eye
CNN regression model (open eye)
CNN regression model (medium eye)
Do nothing
Select the output based on classification model
Output pupil center position (x, y)
Classification part
Regression part
Input eye image
CNN classification model B
Nonclosed eye
Closed eye
CNN regression model (nonclosed eye)
Do nothing
Output pupil center position (x, y)

(a) The proposed method A

(b) The proposed method B

Figure 1: Proposed two-part CNN model.

Table 1: Proposed CNN architectures.

Name	Item	Conv1	Conv2	Conv3	Conv4	Conv5	Full1	Full2	Out
Classification model A	Channel	48	128	192	192	128	1024	1024	3 classes
	Filter size	11 × 11	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	2 × 2	2 × 2	2 × 2	2 × 2	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—
Classification model B	Channel	48	128	192	192	128	1024	1024	2 classes
	Filter size	11 × 11	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	2 × 2	2 × 2	2 × 2	2 × 2	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—
Regression model	Channel	96	256	512	512	512	4096	4096	2 reg.
	Filter size	7 × 7	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	3 × 3	2 × 2	—	—	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—

3.2. Regression Model. The proposed CNN regression model (Table 1, row 3) is based on the pose regression ConvNet [17], 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 [17, 19–21]. 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 [18] for Conv1 and use $L2$ normalization for fully connected layers. $L2$ normalization is defined as follows:

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

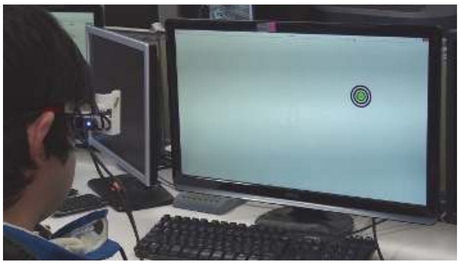


Figure 2: Collection experiment scene.

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:

$$e = \sqrt{\left(P_x - D_x\right)^2 + \left(P_y - D_y\right)^2}. \quad (3)$$

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

4. Experiment

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 [8] 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 developed a system to capture a dataset with appropriate variation and reliability using an inside-out camera [2].

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 Figure 2. Details about the data collection process are described in the following:

(i) 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.

(ii) 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.

(iii) 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. 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 × 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:

(i) An open eye image clearly shows the edge of the pupil, which makes it easy to estimate the pupil center position.

(ii) A medium eye image shows the eyelid overlaid on some part of the pupil, which makes it difficult to estimate the pupil position.

(iii) A closed eye image shows no pupil, which makes it impossible to estimate the pupil position.

Figure 3 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 Figure 4. The distributions of open, medium, and closed eye images are shown in Figures 4(b), 4(c), and 4(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 Figure 5, 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 closed states; therefore, for such images, we used the ground truth from frame two.

4.2. Classification Evaluation. We evaluated the classification problem using leave-one-out cross-validation. We used a

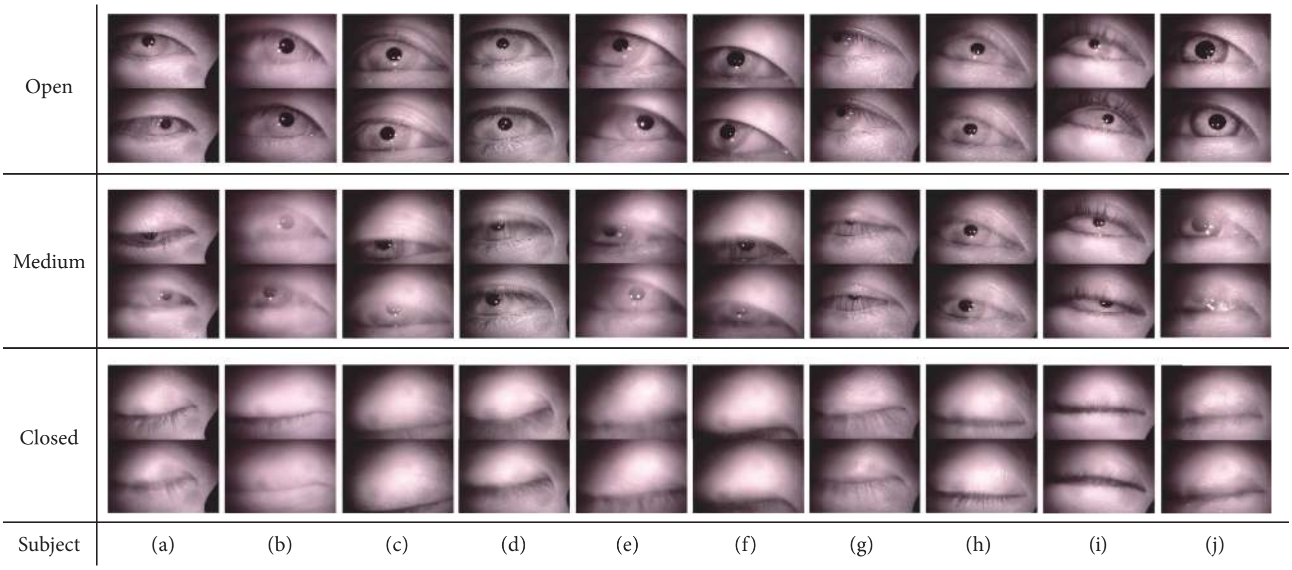
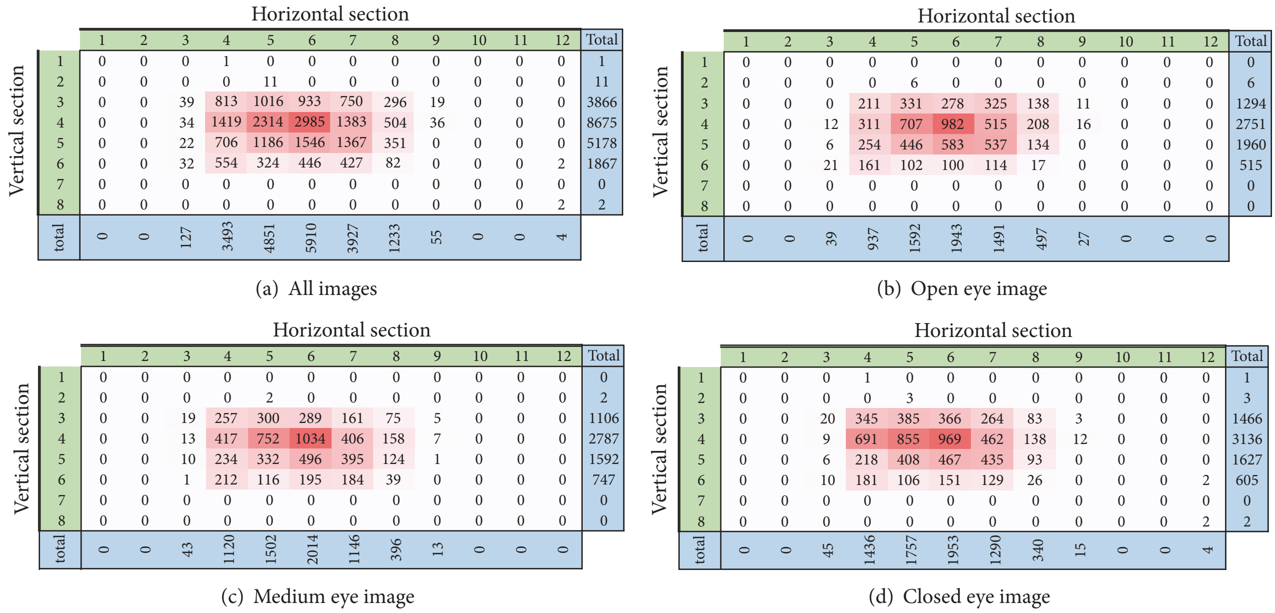


FIGURE 3: Sample eye images from our dataset.



		Horizontal section												
		1	2	3	4	5	6	7	8	9	10	11	12	Total
Vertical section	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

		Horizontal section												
		1	2	3	4	5	6	7	8	9	10	11	12	Total
Vertical section	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

		Horizontal section												
		1	2	3	4	5	6	7	8	9	10	11	12	Total
Vertical section	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

		Horizontal section												
		1	2	3	4	5	6	7	8	9	10	11	12	Total
Vertical section	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) Closed eye image

FIGURE 4: Distributions of our dataset.

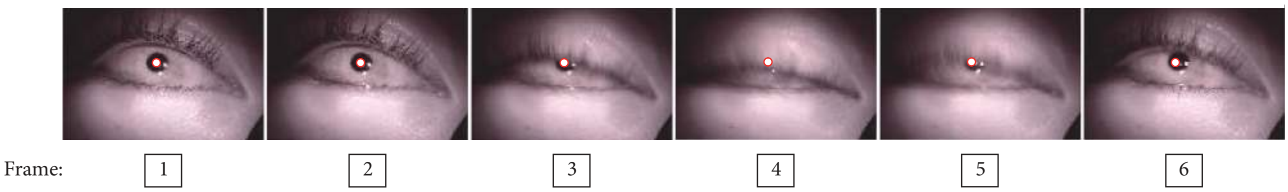


FIGURE 5: Annotation of medium and closed eye image.

Table 2: Confusion matrices of CNN classification model.

(a) Classification result of method A

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

(b) Classification result of method B

	Predict		
	Nonclosed	Closed	Accuracy
Actual			
Nonclosed	6596	244	96.43%
Closed	776	6064	88.65%

pretraining model trained using the ImageNet dataset [22] in order to avoid overfitting. The result from the pretraining model are better than without a pretraining model. The classification results of model A are shown in Table 2(a). The accuracy of this model was 82.58%. This result indicates that the accuracy of closed eye images is greater than that of the other classes. Some images for which classification failed are shown in Figure 6. The accuracy of the medium eye case (73.67%) is less than that of other classes because some of the medium eye images were difficult to classify, as shown in Figures 6(c) and 6(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 nonclosed eye images. To train model B, we randomly selected nonclosed eye images from medium and open eye images to ensure that the number of nonclosed eye images was the same as closed eye images. The classification results of this model are shown in Table 2(b). The overall accuracy of this model was 92.54%, and the accuracy of nonclosed 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 nonclosed eye images is easier than 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.

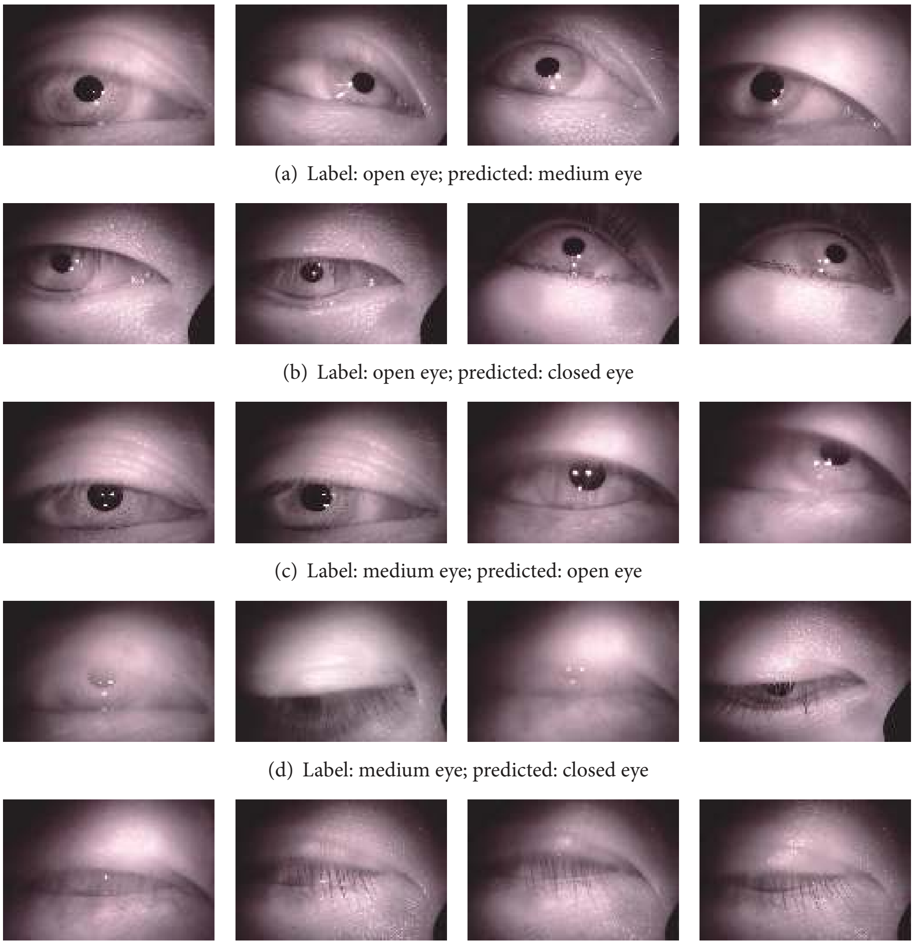
4.3. Regression Model Evaluation. We employed leave-one-out cross-validation to evaluate the regression model. As with the classification model, we used models pretrained using the ImageNet dataset [22] before training with our eye dataset. As discussed in Section 3, the input to the regression model 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 methods A^* and B^*. The regression model was trained using methods A^* 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 3.

Table 3: 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

Methods A^* and B^* are the situation of classification model having 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 7 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. [16]; Zhang et al. [23]). Sun et al. presented multiple CNN models to detect facial feature points. Zhang et al. presented Coarse-to-Fine Auto-Encoder Networks, which are used to detect multiple facial feature points. 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 8 and 9 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



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

(b) Label: open eye; predicted: closed eye

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

(d) Label: medium eye; predicted: closed eye

(e) Label: closed eye; predicted: medium eye

Figure 6: Sample images from the failed classification model.

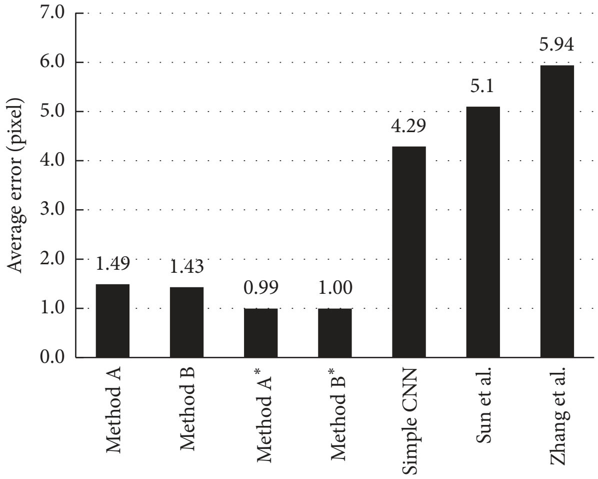


Figure 7: Average error of each CNN model.

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 Figure 10.

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 Figure 7 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 nonclosed 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 compared with method A.

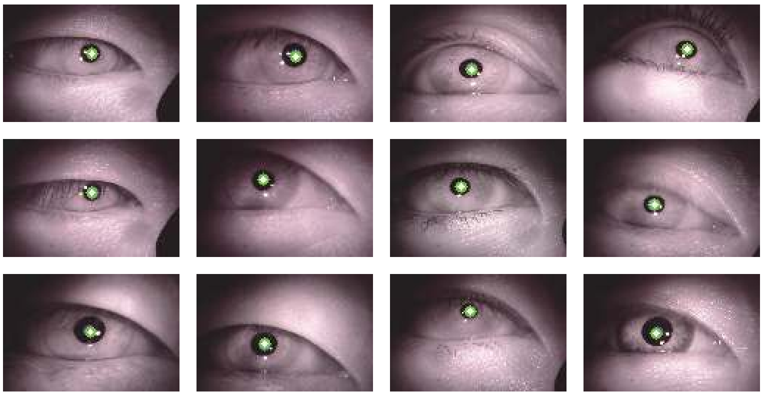


Figure 8: Success samples of open eye image.

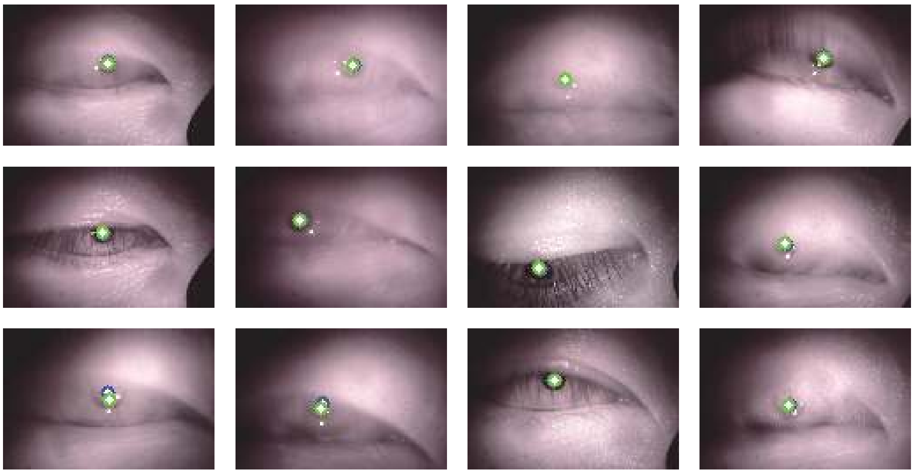


Figure 9: Success samples of medium eye image.

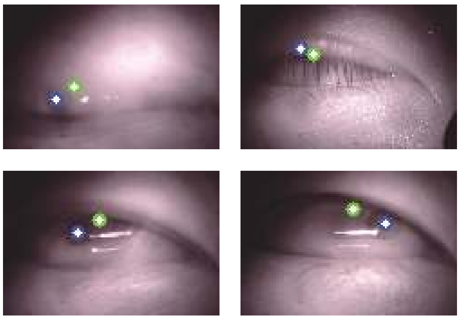


Figure 10: Failure samples.

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 11 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.

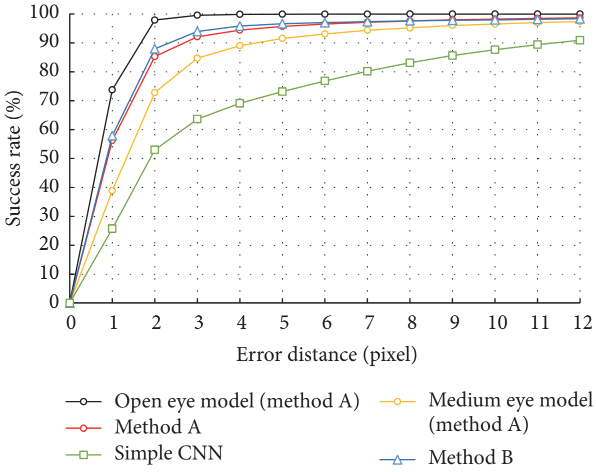


Figure 11: Performance curves.

6. Conclusion

This paper 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 paper has considered avoiding this situation, for example, when blinking obscures the pupil. For supervised learning of the CNN, the dataset required specific features, that is, 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 pretrained 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.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] H. Fujiyoshi, Y. Goto, and M. Kimura, "Inside-out camera for acquiring 3D gaze points," in *Proceedings of the in Proceedings of the Workshop on Egocentric (First-Person) Vision in conjunction with CVPR*, 2012.

[2] J. Iwagami and T. Saitoh, "Easy calibration for gaze estimation using inside-out camera," in *Proceedings of the in Proceedings of the 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV2014)*, pp. 292–297, 2014.

[3] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2235–2244, USA, June 2015.

[4] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2847–2854, USA, June 2012.

[5] A. Mazzei, S. Eivazi, Y. Marko, F. Kaplan, and P. Dillenbourg, "3D model-based gaze estimation in natural reading: A systematic error correction procedure based on annotated texts," in *Proceedings of the 8th Symposium on Eye Tracking Research and Applications, ETRA 2014*, pp. 87–90, USA, March 2014.

[6] A. Kiyohiko, N. Yasuhiro, O. Shoichi, and O. Minoru, "A support system for mouse operations using eye-gaze input," *IEEJ Transactions on Electronics, Information and Systems*, vol. 129, no. 9, pp. 11–1713, 2009.

[7] W. Chinsatit, M. Shibuya, K. Kawada, and T. Saitoh, "Character input system using gaze estimation," in *Proceedings of the in Proceedings of the International Conference on Communication Systems and Computing Application Science (CSCAS2016)*, 2016.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 4511–4520, Boston, Mass, USA, June 2015.

[9] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1821–1828, IEEE, Columbus, Ohio, USA, June 2014.

[10] D. Li, D. Winfield, and D. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 79-79, San Diego, Calif, USA.

[11] Z. Zheng, J. Yang, and L. Yang, "A robust method for eye features extraction on color image," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2252–2261, 2005.

[12] T. Moriyama, T. Kanade, J. Xiao, and J. F. Cohn, "Meticulously detailed eye region model and its application to analysis of facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 738–752, 2006.

[13] W. Chinsatit and T. Saitoh, "Eye detection by using gradient value for performance improvement of wearable gaze estimation system," IEICE Technical Report 115, no. 456, 2016, pp. 149-154.

[14] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci, "Pupilnet: Convolutional neural networks for robust pupil detection," *Computing Research Repository (CoRR)*, 2016, https://arxiv.org/abs/1601.04902.

[15] I.-H. Choi, S. K. Hong, and Y.-G. 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, BigComp 2016*, pp. 143–148, China, January 2016.

[16] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3476–3483, IEEE, Portland, Ore, USA, June 2013.

[17] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9003, pp. 538–552, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of the in Proceedings of the International Conference on Learning Representations (ICLR2014)*, 2014.

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1717–1724, IEEE, Columbus, Ohio, USA, June 2014.

[21] M. Oquab, L. Bottou, I. Laptev, and J. 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, CVPR 2015*, pp. 685–694, June 2015.

[22] O. Russakovsky, J. Deng, H. Su et al., “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] J. Zhang, S. Shan, M. Kan, and X. Chen, “Coarse-to-Fine Auto-encoder Networks (CFAN) for real-time face alignment,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8690, no. 2, pp. 1–16, 2014.

Advances in Multimedia
The Scientific World Journal
International Journal of Distributed Sensor Networks
Journal of Industrial Engineering
Applied Computational Intelligence and Soft Computing
Advances in Fuzzy Systems
Modelling & Simulation in Engineering
Journal of Computer Networks and Communications
Advances in Artificial Intelligence
Advances in Computer Engineering
International Journal of Computer Games Technology
International Journal of Biomedical Imaging
Advances in Artificial Neural Systems
Advances in Software Engineering
Journal of Robotics
Advances in Human-Computer Interaction
Computational Intelligence and Neuroscience
International Journal of Reconfigurable Computing
Journal of Electrical and Computer Engineering