

A Study on the Design of Eye and Eyeball Method based on MTCNN

Cheng-Yu Hsueh¹, Jason C. Hung², Jian-Wei Tzeng³, Hui-Chun Huang⁴, and Chun-Hong Huang^{5*}

¹ Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City 404348, Taiwan

² Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City 404348, Taiwan

³ Department of Information Management, National Taichung University of Science and Technology Taichung City 404348, Taiwan

⁴ Department of Innovation Application and Management, Chang Jung Christian University, Tainan City 71101, Taiwan

⁵ Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan City, 33306, Taiwan

s1811132007@nutc.edu.tw

jhung@nutc.edu.tw

tjw@nutc.edu.tw

a91223@gmail.com

ch.huang@mail.lhu.edu.tw

Abstract. Studies on eye tracking have relied on wearable eye trackers and chin-resting eye trackers, but the high cost of equipment and the need to wear devices during experiments can lead to less natural facial movement. This study collected eye-tracking data by using a raw-video camera without adjusting any parameters. Python was used as the primary programming language. Eye tracking was adjusted through calculations of facial distance, and multitask cascaded convolutional networks were used to collect eye-tracking data. The corrected results were visualized, and linear regression was used to determine correction error. The root mean square error was 221.66, and the mean squared error was 260.48.

Keywords: eye tracker, webcam, multitask cascaded convolutional networks, eye data analysis, eye tracker correction




1 Introduction

Eye-tracking technology has been widely applied in fields such as sports, marketing, and behavior detection. The technology captures eye movements, attention time, and areas of focus for experiments and analysis. Various stimuli are used as experimental variables. Because of the pandemic, students have been unable to take examinations on campus. Eye-tracking technology can be used to detect student behavior and prevent cheating. However, eye tracking requires equipment worn on the face, which may reduce students' willingness to use it. It may also cause cross-infection because of viruses on the equipment. This study designed a contactless eye-tracking method to address this problem. The method is safer and more convenient than chin-rest eye-tracking devices. This study used webcam eye-tracking and increased its accuracy experimentally.

1.1 Webcam Eye Tracking

Eye trackers can be chin resting, wearable, and webcam based. The chin-resting and wearable versions cover a large area of the face, limiting the device's ability to analyze eye movement. Neural models cannot be used to analyze facial features and other features, thereby reducing the effectiveness of data augmentation. During experiments, these eye trackers also restrict the movement of the head. Webcam eye trackers enable the collection of motion data through facial recognition and eye detection without any contact between the device and body. This method is both safe and easy to implement and has been increasingly adopted in eye-tracking research.

Table 1. Eye tracking device

Eye Tracking Devices	Chin-Resting Eye Tracker	Wearable Eye Tracker	Webcam Eye Tracking
Pictures			

1.2 Limitations of Eye Trackers in Experiments

Eye tracking must account for facial movement, which can cause calibration error, reduced accuracy, and experimental bias. Chin-resting eye trackers have been widely used because having the face placed on a frame and only focusing on eye tracking prevent errors. However, wearable trackers, which have cameras near the eye, are more widely used and ensure high accuracy by focusing an infrared light on the eye. The largest drawback of wearable devices is that the head cannot move naturally while the device is worn. Webcams have been increasingly used for eye tracking because of their low cost and safety, and they allow for facial movement during experiments.

Table 2. Advantages and Disadvantages of Eye Tracking Devices

Eye-tracking Device	Chin-resting eye tracker	Wearable eye tracker	Webcam eye tracker
Advantages	High accuracy facilitates analysis of eye movement	Positioned closer to the eye, allowing for accurate collection of data by using infrared technology	Eye tracking with a camera or webcam, enabling large-scale experiments
Disadvantages	The head is fixed on the chin rest and cannot move freely	High cost, making large-scale experiments difficult; prone to masking issues, making analysis challenging	Accuracy may not be as high as that of wearable devices; may require adjustment for high accuracy

1.3 The Universality of Eye Trackers

Webcam eye tracking is a form of web-based eye tracking. The tracking device can be calibrated by combining mouse and eye movements and then using computer calculations to adjust the position of the eyes. The 9-point calibration method is commonly used in experiments, but the accuracy around the edges of the screen is low (WebGazer) [1]. PyGaze is an eye-tracking module developed using the universal language Python, but it has not been updated since 2018 and is only compatible with Python2, making it prone to error. This study developed Python3.0, a universal version, to increase the accuracy of eye tracking.

2 Literature

2.1 Face Detection

Facial recognition typically involves extracting information regarding facial features such as the eyes, nose, and mouth to identify human faces. Large quantities of data are collected and labelled to increase accuracy. Integral image and feature detection combined with an AdaBoost classifier can be used to quickly identify facial features from a large data set and to remove the background [2]. Dlib facial recognition can also be used to identify faces with obstructed regions, and by extracting feature points and vectors, the accuracy and sensitivity of dynamic image face recognition can be increased [3]. Multitask cascaded convolutional networks (MTCNNs) use a neural network and the P-Net, O-Net, and R-Net methods to segment facial regions into eyes, nose, and mouth corners for facial recognition [4].

2.2 Eye Tracker

When light enters the eye through the pupil, the cornea and lens focus the light onto the retina, and the diameter of the pupil controls the amount of light entering the eye and the resulting image intensity, ensuring clear images [5]. When engaged in cognitive

processing, the fovea (the central indentation in the eye) points to the stimulus being processed and denotes the behavior of fixation, but the eye does not continuously fixate on the same stimulus, and saccades occur. Eye tracking can be based on point-of-gaze estimation, three-dimensional (3D) gaze estimation, and pupil center detection. The algorithm based on point-of-gaze estimation [7] uses the vector between the center of the iris and the eye corner as a feature to estimate the gaze point and analyze the positional relationship of eye features in images during head movement. The data collection for the eye tracking method is based on the model approach to the appearance of eye images [8].

Eye tracking involves the use of various mapping functions to calculate gaze fixation. Deep learning techniques yield more accurate results for eye tracking when camera-captured images are used. 3D gaze estimation is based on a novel deep neural network architecture designed specifically for monocular gaze estimation, which simplifies the direct regression of the eye in 3D by regressing two angles for the pitch and yaw of the eyeball [9]. An unsupervised learning-based method is used to estimate the eye gaze in 3D space; geometric spectral photometric consistency constraints and spatial consistency constraints are applied to multiple views in video sequences to refine the initial depth values on the detected iris landmark [10]. Pupil centers are detected by combining the You Only Look Once model with a convolutional neural network (CNN), resulting in an eye-tracking method based on deep learning with a detection accuracy of up to 80% and a recall rate approaching 83% in experimental designs [11]. Deep learning involves training a CNN with eye images as the input, segmenting the pupil area in IR images by using the UNet model, finding the pupil center, and using the pupil center as the regression result [12].

3 Experimental

3.1 Experimental Framework

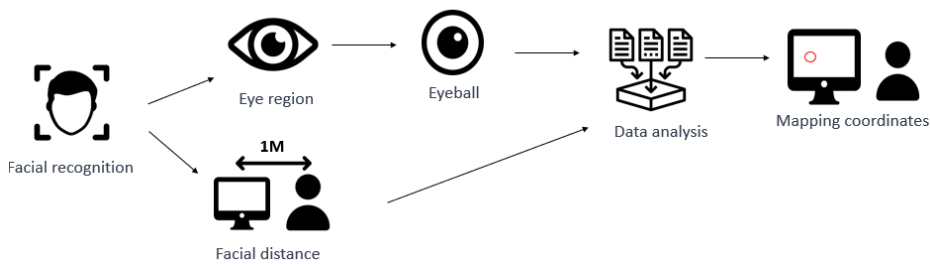


Fig. 1. Experimental Design

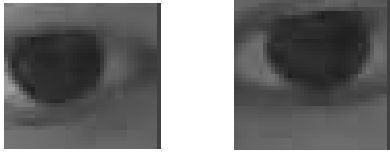

To address generalizability issues in eye tracking, facial information and distance must be determined. The experimental design is shown as figure 1. First, each frame of the video feed is captured, and eye regions are generated through the MTCNN architecture, which captures the position of the eyeballs and then processes the image data. To

address facial anomalies in the video in real time, facial distance calculations exclude abnormal values, thereby increasing data accuracy. The collected eye data are compared with the facial distance data to determine eye size and maximum eye movement distance and to ultimately project them onto a screen to visualize the coordinates of the eye's range of sight.

3.2 Face Detection

Face detection is a mature technology that enables the extraction of facial information through the collection of facial features. First, an MTCNN model is used to extract facial features, including the eyes, nose, and mouth corners. After the eye area feature is extracted, eye area feature segmentation is combined with HoughCircles to locate the circular shape of the eyeball. The eyeball can be monitored by collecting eye position information, including information regarding both the eye area and the eyeball coordinates.

Table 3. Eye Area and Eyeball

	
Eye Area	Eye Ball Detection

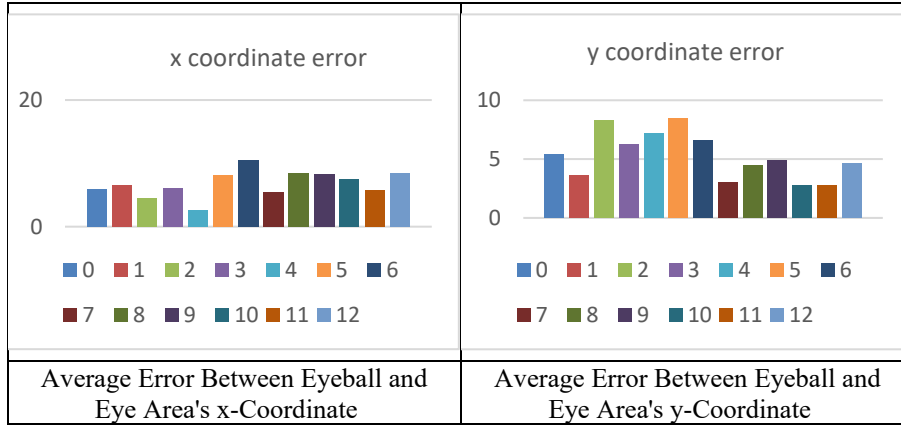
3.3 Face distance

$$D(fs, f, as) = (as * f) / f \quad (1)$$

Function $D(fs, f, as)$ represents the distance between the face and the camera, where fs is the size of the face, f is the focal length of the camera, and as is the actual size of the face. When these parameters are input, the distance between the object (with a face as an example) and the camera can be calculated. The principles of geometry and optics are used for this calculation, with the face distance being calculated to identify significant deviations in the face. Large deviations can lead to a decrease in the accuracy of correction, making the collection of facial distance information particularly important.

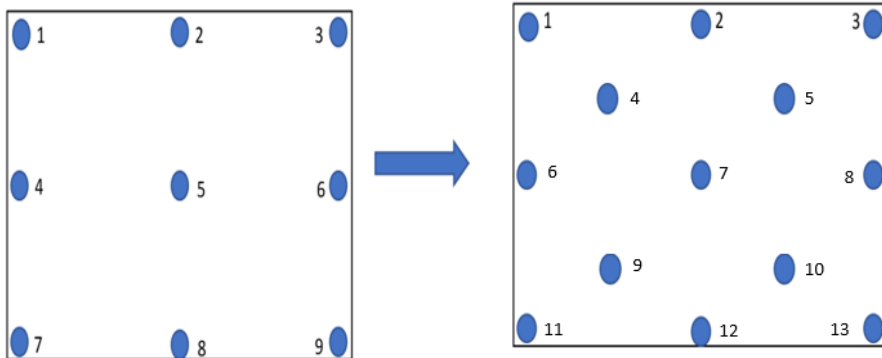
3.4 Eye and Eyeball

The eye area is usually extracted using face detection, but this method does not perfectly extract eye position. Table 4 indicates the average error of the eye area and eyeball position during calibration. The center of the eye area may not be the true center of the eyeball. When the eyeball and eye area are observed with 13 calibration points, the average error on the x-axis is 2.6 to 10.5, and the average error on the y-axis is 2.7 to 8.5.

Table 4. Average Error Between Eye Area and eyeball

3.5 Eye tracker correction

Eye tracking calibration methods usually rely on 5 points. As shown as figure 2. Webgazer's design improves accuracy by using 9 points; the circle turns red after being clicked five times. In this experiment, to further improve accuracy, a 13-point design was used, and calibration and eye-tracking information collection were performed by clicking 15 points in a specified order.

**Fig. 2.** calibration point

3.6 Eye Mapping

The mapping method is based on using the maximum number of coordinates to which that the eyeball can move. The maximum distance from the eyeball to the eye corner is projected onto a 1920×1080 panel, and the eyeball coordinates (blue dot) of the clicked coordinates are displayed. This ensures that the coordinates correspond to the x-y coordinates of the eyeball when they are viewed on the screen. The average coordinates are marked using a green "x".

4 Conclusions and Directions for Future Research

With the third calibration point in Figure 3 as an example, most of the values are close to the error range of the linear regression, except for one point in the bottom-left corner that has a larger error. This may have been due to saccades, which caused the point in the bottom-left corner to fall outside the error range of the linear regression. The root mean square error and mean squared error for this point are 221.66 and 260.48, respectively.

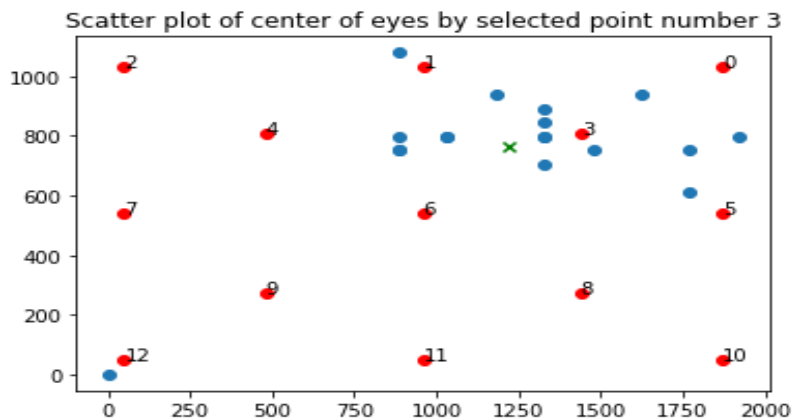


Fig.3. Accuracy of Left Eye Calibration

The experiment involved capturing facial images using a camera and accurately assessing the movement of the eye-ball from the eye. This process also determined the distance of the face and addressed any abnormalities or abnormal values during calibration. However, real-time video can be affected by environmental factors, leading to issues in face recognition. To overcome this problem and improve accuracy, a solution is proposed. It involves increasing the number of calibration points, carefully evaluating their results, and continuously collecting the eye coordinates of the selected points in real-time. The resulting mapping is then compared to the screen to determine its accuracy. This process aims to establish a straightforward and real-time method for collecting eye data within a Python-based system. By incorporating more calibration points and monitoring the eye coordinates, it seeks to enhance accuracy and address abnormalities during face recognition.

The accuracy is acceptable, but the deviation is uncontrollable when an insufficient number of eye positioning points are collected. The design of calibration points will be based on seconds to ensure more precise calibration points are collected for experimental testing and design and to determine whether the deviation during calibration decreases. In addition to improving accuracy, we will consider facial movement deviation. This study only explored facial deviation and excluded abnormal values during facial recognition. We will consider extending our research to the rotation of the face and parameter design to address the problems in recalibration when the head area moves excessively.

References

1. Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediya Daskalova, Jeff Huang, and James Hays. (2016). Webgazer: scalable webcam eye tracking using user interactions. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16). AAAI Press, 3839–3845.
2. Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I).
3. Zhang, D., Li, J., & Shan, Z. (2020, November). Implementation of Dlib deep learning face recognition technology. In 2020 International Conference on Robots & Intelligent System (ICRIS) (pp. 88-91).
4. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10), 1499-1503.
5. Wade, N., & Tatler, B. W. (2005). *The moving tablet of the eye: The origins of modern eye movement research*. Oxford University Press, USA.
6. Rayner, K., & Reingold, E. M. (2015). Evidence for direct cognitive control of fixation durations during reading. *Current opinion in behavioral sciences*, 1, 107-112.
7. Hu, D., Qin, H., Liu, H., & Zhang, S. (2019, July). Gaze tracking algorithm based on projective mapping correction and gaze point compensation in natural light. In 2019 IEEE 15th International Conference on Control and Automation (ICCA) (pp. 1150-1155).
8. Modi, N., & Singh, J. (2022, October). A comparative analysis of deep learning algorithms in eye gaze estimation. In 2022 International Conference on Data Analytics for Business and Industry (ICDABI) (pp. 444-447).
9. Park, S., Spurr, A., & Hilliges, O. (2018). Deep pictorial gaze estimation. In Proceedings of the European conference on computer vision (ECCV) (pp. 721-738).
10. Lu, Y., Wang, Y., Xin, Y., Wu, D., & Lu, G. (2021). Unsupervised Gaze: Exploration of Geometric Constraints for 3D Gaze Estimation. In *MultiMedia Modeling: 27th International Conference, MMM 2021, Prague, Czech Republic, June 22–24, 2021, Proceedings, Part II* (pp. 121-133). Springer International Publishing.
11. Ou, W. L., Kuo, T. L., Chang, C. C., & Fan, C. P. (2021). Deep-learning-based pupil center detection and tracking technology for visible-light wearable gaze tracking devices. *Applied Sciences*, 11(2), 851.
12. Han, S. Y., Kwon, H. J., Kim, Y., & Cho, N. I. (2020). Noise-robust pupil center detection through CNN-based segmentation with shape-prior loss. *IEEE Access*, 8, 64739-64749.