RODET: Estimating Gaze Position on Mobile Device using Real-time On-device Eye-tracking Model

Dohwa Kim*, Yejin Jang*, Eunji Park‡
Chung-Ang University

kimdohwa2@cau.ac.kr, jangyejin1@gmail.com, eunjipark@cau.ac.kr

Abstract

Eye-tracking is a prominent technology used for analyzing users' gaze patterns, estimating users' intentions, and assessing their cognitive load. Previous studies have mainly focused on analyzing gaze patterns on desktop monitors using commercial eye-tracking devices, but eye-tracking on mobile devices has been understudied. In this research, we propose a RODET (Real-time On-Device Eye-Tracking) model for mobile devices based on deep learning, aiming to overcome the limitations of expensive and desktop-centric eye-trackers. Additionally, we developed a finger-free scrolling mobile application utilizing the user's gaze estimated by the lightweight eye-tracking model. As a result, this study provides a benchmark using an accuracy metric with prior eye-tracking models. Also, evaluation through user studies demonstrates the effectiveness of our eye-tracking-based scrolling application, with the potential for further calibration improvements and dataset diversification in future work.

Ⅰ. Introduction

Currently, most commercial eye trackers are primarily designed for large computer monitors, making them inaccessible for mobile devices. Although there are wearable eye trackers, they are expensive and thus not suitable for everyday use by regular users. In addition, most existing eye tracking technologies based on deep learning models run on a desktop environment, not on-device. Therefore, the development of a stable and efficient on-device model is crucial for daily basis mobile eye tracking for normal users.

To address these limitations, we introduce the RODET model. We lightweighted the existing deep learning model [1] using *TensorFlow Lite* [2] to allow detecting of eye features. Also, we adopted *Google ML Kit* [3] allowing estimating users' gaze on mobile devices. To validate the performance of our model, we utilized the dataset provided by Krafka et al. [4], which includes user face images captured by mobile devices and ground truth information for gaze position. We used the data from a total of 10 users, and these datasets contain images of diverse races captured from various angles.

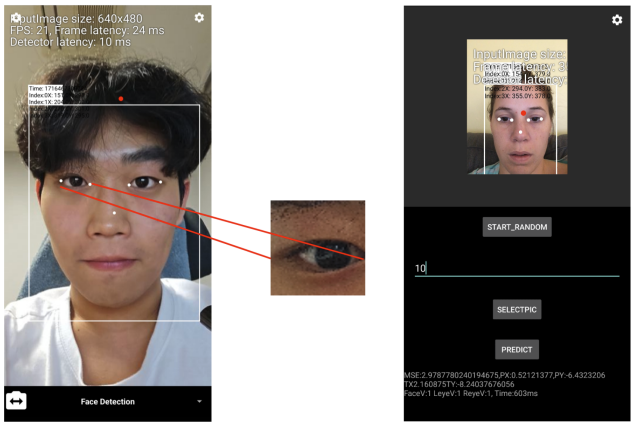
As a result, RODET demonstrated performance similar to prior deep learning-based eye-tracking models operating in desktop environments (Accuracy of the model by Krafka et al. = 2.04 cm, Valliappan et al. = 1.92 cm, RODET (our model) = 2.48 cm). We developed a finger-free scrolling application based on the RODET model and conducted a pilot study with a total of four participants. On average, RODET achieved a 70% success rate in estimating the user's intent for vertical scrolling.

RODET was performed without calibration to enhance user convenience and model generalizability. We anticipate that the performance would improve further if individual calibration data were added. By demonstrating the capability of real-time eye-tracking using only the camera on a mobile device, we showcased the potential for on-device eye-tracking expansion.

Ⅱ. Method

i. Implementation of Real-time On-Device Eye-Tracking System

The main goal of implementing the RODET is to make the model light enough to run on-device with real-time gaze estimates. First, to extract features from the facial image data, we utilized *Google ML-Kit*. It provides precise real-time eye feature extraction with 16 eye contours from both eyes on a 30fps real-time user camera. Based on the eye contour data, we crop the eye image (see Figure 1.(a)) to 128 * 128 pixels precisely and use the eye image data as input data for the multi-layer feed-forward convolutional neural network (ConvNet). The detailed model architecture is described in this paper [1].



(a) (b)

Figure 1. Screenshot of an application with RODET model. (a) Real-time estimation screen and example frame of the contoured eye image. (b) Performance checking screen using existing dataset.

Additionally, each eye's leftmost and rightmost coordinates are given as input data as well. For the model, we reproduced the model proposed by Valliappan et al. [1, 5] using *TensorFlow Lite* to run it on mobile devices.

As a result, RODET provides the coordinates of the estimated user's gaze on the screen. The prediction point is based on a coordinate system in which the camera is the origin (i.e., (x,y) = (0,0) at the camera position) and the unit is a centimeter. So, the y coordinate's prediction point is always negative because our prediction space is only in the device screen under the camera. Since RODET was performed without calibration to enhance user convenience and model generalizability, we expect an additional calibration process will not be necessary with a huge dataset.

* Authors contributed equally
‡ Corresponding author

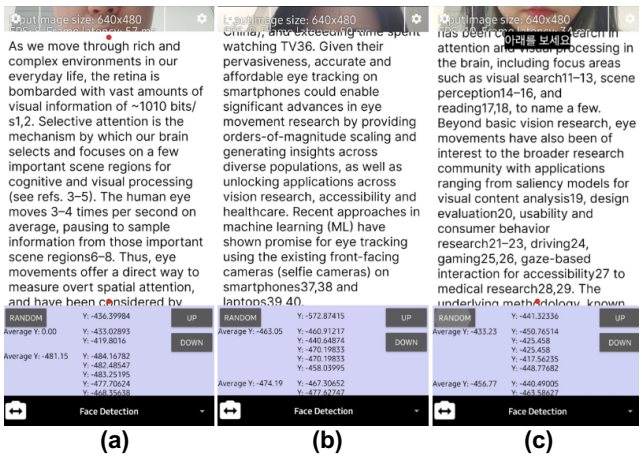
ii. Implementation of an Eye-Tracking-based Scrolling Application

We have developed an application that controls screen scrolling by detecting user eye movements based on Kotlin and Google ML Kit. The application utilizes the device's front-facing camera to detect user eye movements in real-time at 30 frames per second (fps). To ensure accurate detection, we performed the calibration process before carrying out the task. During the calibration, we asked users to stare at red dots located 6dp from the top boundary of the screen and another red dot located 3dp from the bottom boundary (See Figure 2(a)). Then, we measured the user's estimated gaze position in (x,y) five times and averaged the three values (excepting maximum and minimum y values) to set the thresholds. The averaged values are designated as the thresholds that enable scrolling up and down the screen.

The application estimates the user's gaze distance from the camera in centimeters and analyzes gaze direction to understand scrolling intent (scrolling up or down). It is designed to automatically scroll if the user's estimated gaze position exceeds the top or bottom scrolling thresholds. This application enhances user experience by allowing screen navigation without finger usage by utilizing eye movements. It is particularly useful for individuals such as chefs, engineers, people with disabilities, or multitaskers. We believe the application based on the RODET will provide a more convenient and efficient screen navigation experience across various scenarios.

iii. User Study

To evaluate the effectiveness of our eye-tracking-based scrolling application, we conducted a pilot user study with 5 participants. Participants first undergo calibration, then they are randomly instructed to look either up or down while reading a given text document (see Figure 2.(c)). Each participant performed 10 trials, and each trial was considered a success trial if the instruction and the estimated gaze position matched. For example, if the instruction was to 'look up' and the estimated gaze exceeded the upper threshold, the trial was recorded as a success. Conversely, if the instruction and estimated gaze position did not match, the trial was considered a failure.



(a) **(b)** **(c)**

Figure 2. Screenshots of eye-tracking-based scrolling application. (a) Calibration stage, where participants follow a red dot, (b) Post-calibration stage where scrolling is freely performed, (c) Task stage where participants are randomly instructed to look either up or down, displaying guidance text and a red dot indicating where to look.

iv. Results

Performance of RODET

We evaluated the performance of the RODET model based on the dataset provided by Krafka et al. [4]. We considered only images where both eyes were clearly detected and captured in portrait orientation. The images where one eye is out of the screen or when a user hides their eye with their finger are excluded. As a performance metric, we deploy RMSE (Root Mean Square Deviation) value in centimeters. We used all frames that satisfy the above condition user by user, and calculated the mean value of RMSE. Each frame's MSE and user's individual RMSE can be used to check the individual differences between users.

Accuracy (cm)	Krafka et al.	Valliappan et al.	RODET
without calibration	2.04	1.92±0.20cm	2.48
with calibration	1.34	0.45±0.03cm	None

Table 1. Accuracy comparison with prior models

We also measure the speed of the model performing predictions on-device. The delay could exist during extracting eye features using ML-Kit and predicting gaze coordinates based on TensorFlow Lite. On the Samsung Galaxy23 (SM-S911N), the average time taken to estimate gaze from an image of a each single frame was 165ms.

Performance of Eye-Tracking-based Scrolling Application

Each participant was given a random instruction to look either up or down (see Figure 2.(c)). The application recorded 10 estimated gaze coordinates while participants looked in the designated direction for 5 seconds. A task was considered successful if the majority of the data points indicated the correct action. All participants in this experiment succeeded in 7 out of 10 trials (i.e., 70% success rate on average), demonstrating its effectiveness in accurately detecting and responding to user eye movements.

Ⅲ. Conclusion

We propose RODET, a lightweight deep learning-based real-time eye-tracking technology designed for mobile devices. Our model, which operates entirely on-device, showed similar performance to models from previous studies. In addition, we have developed a scrolling application to enhance user experience. In the future, we aim to diversify datasets and implement personalized calibration processes for further improvement.

Acknowledgements

This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [**5011-2024-000015**, User requirements analysis based on a prototype of visual restoration technology]

References

[1] Valliappan, N., Dai, N., Steinberg, E. et al. "Accelerating eye movement research via accurate and affordable smartphone eye tracking." Nat Commun 11, 4553. 2020

[2] https://www.tensorflow.org/lite

[3] https://developers.google.com/ml-kit

[4] Krafka, Kyle, et al. "Eye tracking for everyone." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[5] https://github.com/s0mnaths/Gaze-Tracker