# EyeGesener: Eye Gesture Listener for Smart Glasses Interaction Using Acoustic Sensing

TAO SUN* and YANKAI ZHAO*, Research Institute of Trustworthy Autonomous Systems, Department of Computer Science and Engineering, Southern University of Science and Technology, China
WENTAO XIE, Southern University of Science and Technology, and The Hong Kong University of Science and Technology, China
JIAO LI, YONGYU MA, and JIN ZHANG†, Research Institute of Trustworthy Autonomous Systems, Department of Computer Science and Engineering, Southern University of Science and Technology, China

The smart glasses market has witnessed significant growth in recent years. The interaction of commercial smart glasses mostly relies on the hand, which is unsuitable for scenarios where both hands are occupied. In this paper, we propose *EyeGesener*, an eye gesture listener for smart glasses interaction using acoustic sensing. To mitigate the Midas touch problem, we meticulously design eye gestures for interaction as two intentional consecutive saccades in a specific direction without visual dwell. The proposed system is a glass-mounted acoustic sensing system with two pairs of commercial speakers and microphones to sense eye gestures. To capture the subtle movements of the eyelid and surrounding skin induced by eye gestures, we design an Orthogonal Frequency Division Multiplexing (OFDM)-based channel impulse response (CIR) estimation schema that allows two speakers to transmit at the same time and in the same frequency band without collision. We implement eye gesture filtering and adversarial-based eye gesture recognition to identify eye gestures for interaction, filtering out daily eye movements. To address the differences in eye size and facial structure among different users, we employ adversarial training to achieve user-independent eye gesture recognition. We evaluate the performance of our system through experiments on data collected from 16 subjects. The experimental result shows that our system can recognize eight eye gestures with an average F1-score of 0.93, and the false alarm rate of our system is 0.03. We develop an interactive real-time audio-video player based on *EyeGesener* and then conduct a user study. The result demonstrates the high usability of the proposed system.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; **Interaction techniques**.

Additional Key Words and Phrases: hands-free interaction, eye gestures, acoustic sensing

**ACM Reference Format:**
Tao Sun, Yankai Zhao, Wentao Xie, Jiao Li, Yongyu Ma, and Jin Zhang. 2024. EyeGesener: Eye Gesture Listener for Smart Glasses Interaction Using Acoustic Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 128 (September 2024), 28 pages. https://doi.org/10.1145/3678541

*Both authors contributed equally to this research.
†The corresponding author is Jin Zhang.

Authors' addresses: Tao Sun, 11810206@mail.sustech.edu.cn; Yankai Zhao, kr4y799@gmail.com, Research Institute of Trustworthy Autonomous Systems, Department of Computer Science and Engineering, Southern University of Science and Technology, China; Wentao Xie, wxieaj@cse.ust.hk, Southern University of Science and Technology, and The Hong Kong University of Science and Technology, China; Jiao Li, lij@sustech.edu.cn; Yongyu Ma, 12232427@mail.sustech.edu.cn; Jin Zhang, zhangj4@sustech.edu.cn, Research Institute of Trustworthy Autonomous Systems, Department of Computer Science and Engineering, Southern University of Science and Technology, China.


2474-9567/2024/9-ART128
https://doi.org/10.1145/3678541

## 1 INTRODUCTION

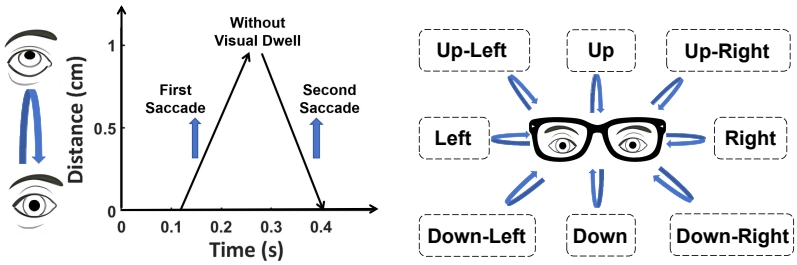
In recent years, there has been a burgeoning popularity of smart glasses. There are optical see-through head-mounted displays (e.g., Google Glass, Epson MOVERIO), AR/XR glasses (e.g., Nreal Air, HoloLens), VR glasses (e.g., Meta Quest 3, Apple Vision Pro), and smart audio glasses (e.g., Ray-Ban Smart Glasses, Bose Frames). The interaction requirements for smart glasses cannot be well met. The interaction with smart glasses is mainly divided into hand-based interaction and hands-free interaction. Hand-based interaction relies on touchpad technologies [23, 24, 89], additional control devices [42, 70] or mid-air gestures [59, 93]. However, frequently raising one's arm for interaction may cause fatigue and is not friendly to elderly and disabled users. Moreover, this is unsuitable for scenarios where both hands are occupied. Hands-free interaction has been proposed to address the aforementioned issues, but there are still some concerns that need to be addressed. Voice input [47] faces constraints in situations like meetings or libraries and may result in privacy issues. Facial expressions [27, 90] and silent speech interfaces [51, 96] may cause physical fatigue and embarrassment when inputting. Consequently, the development of a new interaction system for smart glasses is desirable.

Recent studies have proposed the utilization of eye movement as an input modality [38] due to several benefits: (i) Eye movements are easily captured by the sensors mounted on glasses. (ii) Eye movement directly reflects the user's intention, making it an intriguing input modality for attentive user interfaces [79]. (iii) Performing eye movements is effortless for most individuals.

Active research efforts have been undertaken to recognize eye movements in smart glasses. Camera-based methods [16, 66, 67, 74], involving the mounting of a small camera on eyeglasses, can recognize eye movements with high accuracy. However, these methods are costly, suffer from poor ambient light conditions, and raise concerns about potential privacy issues. Multichannel electrooculography (EOG) electrodes serve as an intrusive method for eye tracking. Prior studies have proposed integrating the EOG sensor into glasses [32, 33], but these approaches are vulnerable to sweat artifacts [39] due to the requirement for physical contact with the skin. Some researchers employ an infrared distance sensor array on eyewear to recognize gaze movement gestures [43], but these works are sensitive to environmental conditions such as sunlight, smoke, etc [19, 77].

Acoustic sensing is more suitable for detecting eye movements on smart glasses for several reasons. Firstly, it is a contact-free method as the sensors do not need to contact with the skin. Secondly, the cost of commercial speakers and microphones is low [82]. Thirdly, acoustic sensing is resilient to varying light conditions. There are some acoustic-based methods to sense eye activities, but they cannot meet the interaction requirements of smart glasses well. Some researchers [36, 60] utilize a pair of speakers and a microphone to accurately detect blink, but these methods cannot distinguish finer-grained eye movements beyond blink. A study [81] enables eye-tracking on glasses based on piezoelectric micromachined ultrasonic transducers. However, this method requires ultrasound in the MHz band, but commercial speakers and microphones only support frequencies up to 24 kHz. Golard et al. [46] conduct a modeling and empirical study to prove that ultrasound can achieve low-power, fast, and light-insensitive eye tracking results, but this method is evaluated on a physical 3D model of a human eye and its performance on a real user is unknown. Li et al. [55] propose an acoustic-based eye tracking system on glasses. It achieves accurate eye tracking results with low power consumption. However, it cannot distinguish between eye movements for interaction and daily eye movements and cannot achieve user-independent results. Currently, there is no existing eye movement-based interaction system for smart glasses that is low-cost, user-independent, and suitable for long-term interaction in everyday environments.

To overcome the limitations of prior works, we propose ***EyeGesener***, *an* ***Eye Ges****ture List****ener*** *for Smart Glasses Interaction using Acoustic Sensing* with minimally obtrusive modifications to a glass frame. Based on the observation that eye movements can be apparent through the skin of the eyelids [43] and are accompanied by skin movements around the eyes [90], we can recognize eye movements by sensing the movements of the eyelids and surrounding skin. We utilize speakers to emit near-ultrasound signals towards the user's eyes, and




(a) Eye gesture is a combination of two saccades.

(b) Eight eye gestures.

Fig. 1. Eye gestures design.

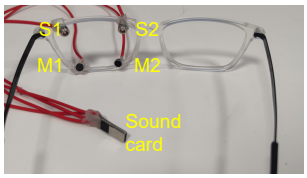



Fig. 2. *EyeGesener* prototype.

the microphone will receive the reflected echoes carrying the eye movement-related information. To enable the sensing ability on the glass frame with minimally obtrusive modification, we add two microphones and speakers at the four corners of the glass frame. We position two microphones on the lower side of the glass frame with the consideration that, when integrated into smart glasses, these microphones are closer to the mouth, enhancing the reception of voice input. The microphone and speakers are positioned towards the eyes to better capture eye movement information. Our design ensures comfort and minimal obtrusiveness, ensuring it does not interfere with the user's daily use of the glasses. It can be further deployed on commercial smart glasses.

Implementing *EyeGesener* in real-life environments presents the following challenges. First, eyelids and surrounding skin movements induced by eye gestures are tiny and non-periodic. Moreover, individuals exhibit variations in eye sizes and facial structures, making it challenging to achieve accurate user-independent eye movement recognition. We overcome the above challenge with the following designs. The position of two microphones and speakers is designed at the four corners of the glass frame to enhance spatial diversity. This setup allows for the sensing of eye movements from various observation angles through multiple paths. To improve time and frequency utilization, we employ Orthogonal Frequency Division Multiplexing (OFDM) de/modulation schemes for Channel Impulse Response (CIR) estimation. This approach enables simultaneous transmission by both speakers in the same frequency band. The CIR measurements have sub-sample resolution, making them suitable for measuring the movements of eyelids and the surrounding skin induced by eye movements. The CIR sequence encompasses user-independent features associated with eye movements, as well as user-specific features related to the individual's eye size and facial structures. We employ an adversarial training strategy, incorporating gradient reversal (GR) layers, to extract user-independent features related to eye movements.

Second, eye movements are frequent and common in daily life. The system must be capable of distinguishing between daily eye movements and those intended for interaction. This challenge is known as the Midas Touch problem [50]. To address this, we meticulously design eye gestures for interaction. Eye gestures are several predefined consecutive movements with specific patterns [33] and these patterns can be designed to be infrequently used in daily life. We have the observation that eye movements in daily life serve the purpose of obtaining information, resulting in visual dwell time. Interaction eye gestures can be designed as two intentional consecutive saccades in a specific direction without visual dwell. Depending on various saccade directions, eight distinct input options exist for interaction: up, down, left, right, up-left, up-right, down-left, and down-right as shown in Fig. 1. The amplitude, time interval, and pattern of eye gestures significantly differ from those of daily eye movements. Consequently, we design eye gesture filtering and adversarial-based eye gesture recognition to identify eye gestures and filter out daily eye movements.

In summary, the main contributions of this work are highlighted as follows:

- To the best of our knowledge, we propose the first acoustic-based, user-independent eye gesture recognition system for hands-free interaction on smart glasses.
- We meticulously design intentional eye gestures for interaction to mitigate the Midas touch problem. We design eye gesture filtering and adversarial-based eye gesture recognition to identify intentional eye gestures, filtering out daily eye movements. We employ adversarial training to achieve user-independent eye gesture recognition.
- We conduct extensive experiments to evaluate the system's robustness. We evaluated the *EyeGesener* with 16 subjects to assess the performance of eye gesture recognition. The results indicate that our system attains high accuracy and robustness with an average F1-score of 0.93 and the false alarm rate of our system is 0.03. We develop an interactive real-time audio-video player and a user study with 8 participants demonstrates the high usability of *EyeGesener*.

The paper is organized as follows: Section 2 gives an overview of our design. Section 3 elaborates on the detailed design of the system. Section 4 introduces the implementation. Section 5 covers the evaluation of the system, followed by a user study in Section 6. Section 7 discusses the limitations and future directions of the system. Section 8 summarizes the related work in this paper, and Section 9 concludes the paper.

## 2 DESIGN OVERVIEW

In this section, we first elaborate on the rationale behind our design of the 8 eye gestures for interaction. Then we provide an overview of the system to accurately recognize interaction eye gestures and distinguish daily eye movements with low computation cost and generalization ability for new users.

The design of eye gestures for interaction should meet the following requirements: the designed eye gestures need to be easy to perform, can be detected through acoustic sensing and can be distinguished from daily eye movements. We meticulously design intentional eye gestures for interaction to mitigate the Midas touch problem. Eye movements in daily life are frequent and common, and there is currently no perfect solution for the Midas Touch problem. Eye gestures that consist of several consecutive movements were proposed to address this problem. To design interaction eye gestures that can be differentiated from daily eye movements, we conducted a week-long intermittent study with three individuals to record their eye movements with a camera in their daily lives under different scenarios. Blink is not suitable as an interaction input because it is frequently used in our daily lives. Solely using a saccade is not suitable for interaction as it is difficult to distinguish from everyday eye movements. We found that eye movements in daily life serve the purpose of acquiring information, and people often have visual dwell time to obtain information, resulting in an interval between two consecutive eye movements. Additionally, people typically use a combination of head turning and eye movements to gather the desired information, resulting in relatively small amplitudes of eye movements. Therefore, we design two intentional consecutive saccades (quickly looking in a specific direction and quickly looking back without visual dwell) as an interaction eye gesture. Depending on various saccade directions, there are naturally eight distinct input options for interaction: up, down, left, right, up-left, up-right, down-left, and down-right, as shown in Fig. 1(b).

The *EyeGesener*'s system overview is shown in Fig. 3. *EyeGesener* comprises three modules. The initial module is acoustic signal preprocessing, utilizing ZC-based signal design for CIR estimation with low computational complexity and high resolution. The second module eye gesture filtering provides preliminary differentiation of interaction eye gestures and daily eye movements. The third module adversarial-based eye gesture recognition performs fine-grained classification to further distinguish between interaction eye gestures and daily eye movements.

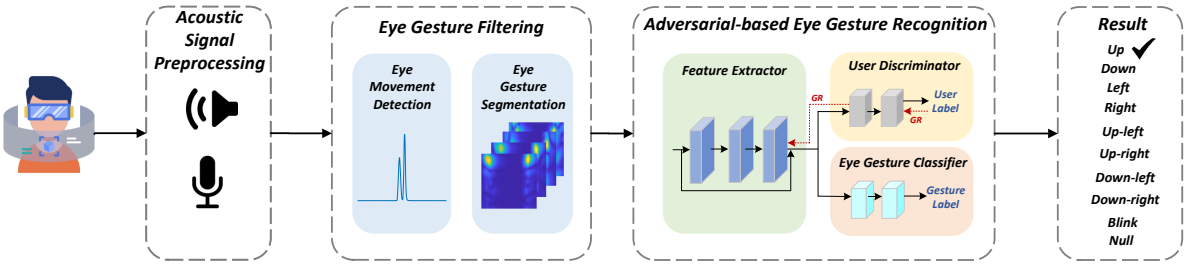



Fig. 3. *EyeGesener*'s system overview.

## 3 SYSTEM DESIGN

In this section, we elaborate on the design of the three modules of our system. The workflow of our system is shown in Fig. 3.

### 3.1 Acoustic Signal Preprocessing

*3.1.1 Transmission Signal Design.* To achieve fine-grained sensing of the CIR profile induced by eye gestures, we meticulously design the transmit signal rather than employing a random sequence, as typically done in traditional OFDM systems [68, 90]. To enhance both time and frequency utilization, we employ the Orthogonal Frequency-Division Multiple Access (OFDMA) scheme, enabling concurrent transmission by both devices within the same frequency band. The key to the design lies in selecting two signals with high auto-correlation and low cross-correlation. We choose the Zadoff-Chu (ZC) sequence [73] as our baseband signal, known for its ideal auto-correlation property, facilitating the separation of paths at different distances [84]. For clarity in representation, we use uppercase letters for frequency domain signals and lowercase letters for time domain signals. The baseband ZC sequence, with a length of $N_{zc}$, is expressed as follows:

$$zc[n] = \exp\left(-j\frac{\pi un(n+1+2q)}{N_{zc}}\right), \tag{1}$$

where $0 \leq n < N_{zc}$, $q$ is a constant integer, and $j$ is the imaginary unit, i.e., $j^2 = -1$. $N_{zc}$ is the length of the sequence, which determines the bandwidth in the final modulated signal. The parameter $u$ determines the correlation property, and it should be coprime to $N_{zc}$, i.e., $gcd(N_{zc}, u) = 1$.

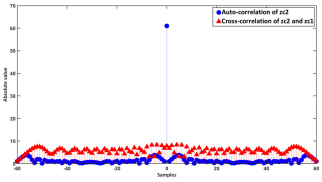
In our OFDM scheme, we choose distinct values of $u$ during the generation of the baseband ZC sequence to prevent collisions between the two speakers. Specifically, the frequency domain baseband signals for the two speakers, denoted by $ZC_1$ and $ZC_2$, are expressed as:

$$ZC_1[n] = FFT\left(zc_1[n]\right) = FFT\left(\exp\left(-j\frac{\pi u_1 n(n+1+2q)}{N_{zc}}\right)\right), \tag{2}$$

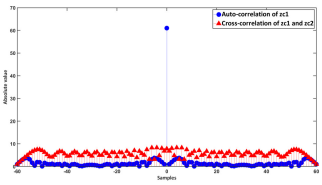
$$ZC_2[n] = FFT\left(zc_2[n]\right) = FFT\left(\exp\left(-j\frac{\pi u_2 n(n+1+2q)}{N_{zc}}\right)\right), \tag{3}$$

where $u_1$ and $u_2$ are two different values and $FFT(\cdot)$ denotes equal-points Fast Fourier Transform (FFT).

We use two orthogonal ZC sequences for two speakers to enable simultaneous transmission utilizing full bandwidth and all subcarriers. We choose the value of $u_1$ and $u_2$ for two $ZC$ sequences based on the following criterion: Firstly, auto-correlation results should only have one peak to reduce ambiguity. Secondly, the results of cross-correlation should be low to reduce interference. We first select $u_1$ based on the first criterion and then select

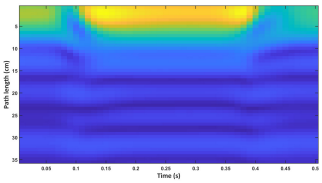



(a) Correlation property of zc1.

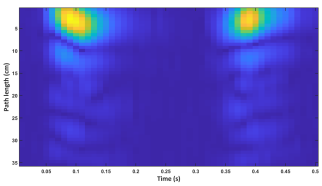



(b) Correlation property of zc2.

Fig. 4. The auto-correlation and cross-correlation properties of zc1 and zc2.




(a) Amplitude of raw CIR.




(b) Amplitude of different CIR.

Fig. 5. CIR comparison of with and without background subtraction.

the pair $u_1$ and $u_2$ based on the second criterion. Finally, we choose of $u_1 = 1$ and $u_2 = 60$. The auto-correlation and cross-correlation of sequences $zc_1$ and $zc_2$ are illustrated in Fig. 4, with $N_{zc} = 61$. It is evident that the auto-correlation of sequence 1 and sequence 2 is favorable, whereas the cross-correlation between sequence 1 and sequence 2 is poor. This characteristic enables two speakers to transmit simultaneously within the same frequency band without encountering collisions.

The direct utilization of the ZC sequence as our transmit signal is precluded for two primary reasons. Firstly, speakers are capable of transmitting only real signals, whereas the ZC sequence constitutes a complex signal. Secondly, the ZC sequence occupies the entire frequency band, making it audible to humans. Consequently, it is imperative to reduce the bandwidth of the generated ZC sequence to ensure its accommodation within a narrow transmission band that remains inaudible to humans. Additionally, it is necessary to convert the complex ZC sequence into real signals suitable for transmission by speakers. To achieve this, we employ frequency domain de/modulation schemes similar to those described in [84, 87]. This approach is favored due to its substantial reduction in computational complexity when compared to the time domain de/modulation schemes employed in [80].

To enable the transmission signal inaudible to most individuals [76], our system employs a narrow frequency band in 17-23 $kHz$ (i.e., $B = 6\ kHz$) with a central frequency of 20 $kHz$ and a sampling rate of 48 $kHz$. The transmission comprises consecutively repeated frames, with a frame length of $N = 480$. Consequently, the frame rate for CIR is established at 100 Hz, enabling the stacking of multiple CIR frames to examine the impact of eye gestures on CIR. The ZC length is configured as 61 based on $N_{zc} = \frac{B \cdot N}{fs} + 1$, ensuring compatibility with the bandwidth of the modulated signal. After generating the ZC sequence, the sequence is subsequently shifted to the carrier frequency $f_c$ through OFDM modulation. To convert the modulated signal into a real signal, the negative frequency components of the signal are configured to be the conjugate counterparts of the corresponding positive frequency components. This modulation procedure is elucidated in Algorithm 1, where $conj(\cdot)$ returns the complex conjugate of each element in the input, and $flip(\cdot)$ returns the reverse of the input. The transmission signals of both speakers employ a similar modulation process, differing only in the value of $u$. To reduce the impact of frequency leakage caused by sudden frequency jumps between successive frames, we use the Hanning

window on the transmit signal. Moreover, we adjust the max amplitude of the transmit signal to 0.2 to further minimize the risk of being heard.

**Algorithm 1** Transmitting signal generation

**Input:** $N_{zc}, N, f_c, u, q$
**Output:** The modulated sequence $x_T[n]$ of length $N$

1: Generate ZC sequence from Eq.( 1) and perform $N_{zc} - point$ FFT to get frequency domain signal $ZC[n]$
2: Generate a all zero sequence $X[n]$ with a length of $N$
3: $X[\frac{f_c L}{fs} - \frac{(N_{zc}-1)}{2} : \frac{f_c L}{fs} + \frac{(N_{zc}-1)}{2}] \Leftarrow ZC[n]$
4: $X[N - \frac{f_c L}{fs} - \frac{(N_{zc}+1)}{2} : N - \frac{f_c L}{fs} + \frac{(N_{zc}+1)}{2}] \Leftarrow flip(conj(ZC[n]))$
5: Perform IFFT on $X[n]$ to get time domain signal $x_T[n]$

*3.1.2 Received Signal Processing.* After the signal is transmitted from the speaker, the microphones record signals originating from both the Line-of-Sight (LOS) path and reflections from the subjects' eye gestures and the environment. Our system incorporates two speakers and two microphones positioned at various locations on the eyewear. This configuration enables the measurement of CIR profiles from diverse observation angles. This capability proves advantageous, as the same eye gesture manifests different CIR patterns when observed from varying angles. We apply a Butterworth band-pass filter with a cut-off frequency range of 17-23 kHz on the received signal to remove the environmental noise. For each pair of speaker/microphone, we can extract a distinct set of CIRs per frame by conducting cross-correlation between the received signal and the known transmitted signal.

The received signal is modeled as:

$$y_R[n] = \sum_{i=1}^{P} A_i e^{-j\phi_i(t)} x_T\left[n - \frac{\tau_i}{fs}\right], \tag{4}$$

where $y_R[n]$, represents the signal received and $P$ signifies the number of paths. Each path $i$ is characterized by an attenuation coefficient $A_i$, a phase shift $\phi_i$ induced by the propagation or reflection of the path, and a time of flight (ToF) denoted as $\tau$. To facilitate further analysis, the received signals $y_R[n]$ are initially partitioned into frames of length $N$. Subsequently, a frequency domain multiplication approach is employed for performing frequency domain correlation. This technique serves to significantly mitigate the computational complexity associated with correlation. The detailed demodulation process is elucidated in Algorithm 2 where $fftshift(\cdot)$ shifts zero-frequency component to center of spectrum.

With the utilization of two speakers and two microphones, we can obtain CIR estimation information for four links (speaker 1 - microphone 1, speaker 1 - microphone 2, speaker 2 - microphone 1 and speaker 2 - microphone 2). For each link, the measurement of the $cir[n]$ is acquired per frame. To capture the dynamic changes in the CIR, the measurements of the CIR across multiple frames are aggregated within an observation slot, resulting in the construction of a 2D CIR map. This map serves as a representation of the CIR variations over time. The time-domain resolution of 10 milliseconds in the CIR map corresponds to a sampling rate of 100 Hz. The range resolution is $c \cdot \frac{1}{fs} \approx 0.007\ m$, where $c$ denotes the speed of sound and $fs$ is the sampling rate. The high time and range resolution enable us to effectively monitor the movements of eyelids and the surrounding skin induced by eye gestures.

We use background subtraction methods to extract user eye gesture information and eliminate the influence of static objects in CIR measurements. The raw CIR corresponding to an eye gesture is depicted in Fig. 5(a). Notably, the raw CIR is dominated by static components, posing a challenge in discerning signals associated with two

**Algorithm 2** Received signal demodulation

**Input:** Received signal sequence $x_R[n]$ of length $N$
**Output:** Channel response sequence $cir_1[n]$ and $cir_2[n]$ of length $N$ for each frame

1: Perform $N$ point FFT on $y[n]$ to get $Y[n]$
2: $CIR_{baseband1} \Leftarrow Y[\frac{f_c L}{fs} - \frac{(Nzc-1)}{2} : \frac{f_c L}{fs} + \frac{(Nzc-1)}{2}] \times conj(ZC_1[n])$
3: $CFR_{baseband2} \Leftarrow Y[\frac{f_c L}{fs} - \frac{(Nzc-1)}{2} : \frac{f_c L}{fs} + \frac{(Nzc-1)}{2}] \times conj(ZC_2[n])$
4: Generate an all-zero sequence $CIR_1$ and $CIR_2$ with length $N$.
5: $CIR_1[\frac{N}{2} - \frac{(Nzc-1)}{2} : \frac{N}{2} + \frac{(Nzc-1)}{2}] \Leftarrow CFR_{baseband1}[n]$.
6: $CIR_2[\frac{N}{2} - \frac{(Nzc-1)}{2} : \frac{N}{2} + \frac{(Nzc-1)}{2}] \Leftarrow CFR_{baseband2}[n]$.
7: $CIR_1[n] \Leftarrow fftshift(CIR_1[n])$.
8: $CIR_2[n] \Leftarrow fftshift(CIR_2[n])$.
9: Perform IFFT on $CIR_1[n]$ and $CIR_2[n]$ to get the time domain $cir_1[n]$ and $cir_2[n]$ respectively.

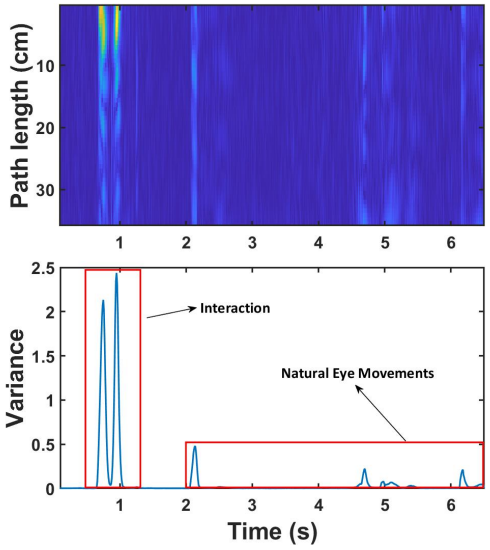



Fig. 6. One eye gesture and daily eye movements.

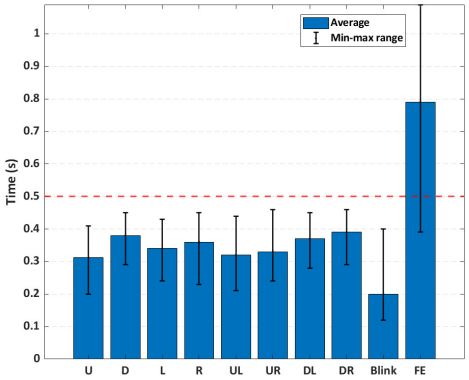



Fig. 7. Interval of two eye movements. (U: up; D: down; L: left; R: right; UL: up-left; UR: up-right; DL: down-left; DR: down-right ; FE: the interval between two fast eye movements with visual dwell in daily life).

saccades. To mitigate the influence of static components, differential operation is applied to the complex-valued CIR sequence, as opposed to a direct operation on the intuitive CIR amplitude. This choice is motivated by the observation that amplitude changes over a short duration are minimal, and a direct differential operation would eliminate both movement-related signals and static components. The resulting differential CIR is illustrated in Fig. 5(b). Furthermore, CIR measurements encompass information from various surrounding objects. The arrangement of these objects during training and testing phases may exhibit significant disparities, thereby potentially diminishing recognition performance. In contrast, with differential CIR images (dCIR), gesture data remains unaffected by surrounding objects, provided they remain static during the gesture process. The static elimination enhances stability in eye gesture recognition.

## 3.2 Eye Gesture Filtering

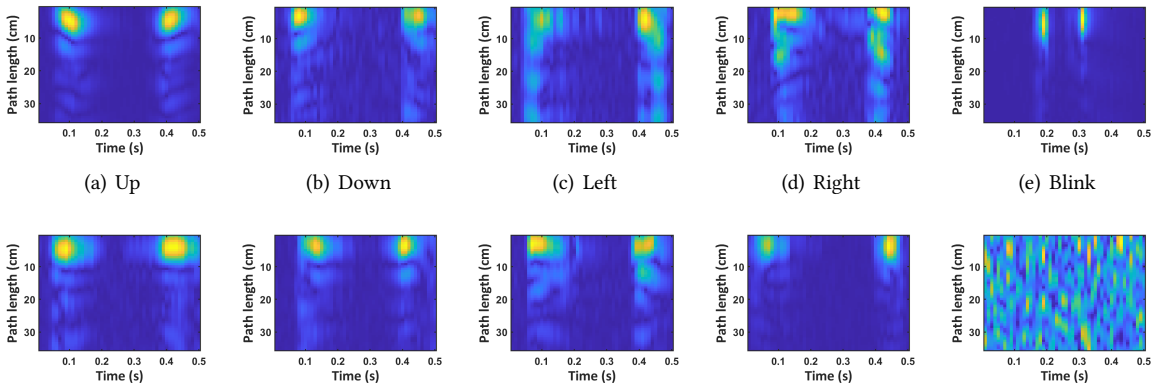
The purpose of this module is to detect and segment all interaction eye gestures, allowing for some segmentation of non-interaction eye movements. The interaction eye gestures and non-interaction eye movements have different dCIR patterns and can be distinguished through subsequent eye gesture recognition module.

*3.2.1 Eye Movement Detection.* We use the variance of dCIR to detect the occurrence of eye movements. Eye movement can cause changes in dCIR, so we can determine the occurrence of eye movement by calculating the variance of dcir. The change of dCIR induced by eye gestures typically occurs within a small range. Therefore, we only consider the first 50 points of the dCIR to reduce computational complexity, corresponding to the information within a distance of $c \cdot \frac{50}{fs} \approx 0.35\ m$ where $c$ denotes the speed of sound and $fs$ is the sampling rate. We calculate the dCIR variance on each frame and obtain a sequence of dCIR over time. Each eye movement leads to a peak in dCIR, and the amplitude of the peak in dCIR is related to the amplitude of eye movement. We detect eye movements by finding peaks on the time sequence of variance of dcir, with a minimum peak interval of 0.1 seconds, as changes within 0.1 seconds are considered as one eye movement. The detected eye movements will be inputted into eye gesture segmentation for subsequent differentiation between interaction eye gestures and daily eye movements.

*3.2.2 Eye Gesture Segmentation.* The amplitude and interval of interaction eye gestures and daily eye movements are different, so we can distinguish them through amplitude and interval. After detecting eye movements, we segment interaction eye gestures and ignore non-interaction eye movements. The algorithm filters out non-interaction eye movements by the amplitude of the peak and the interval between the peaks. The goal of this segmentation algorithm is to guarantee the recognition of all interaction eye gestures while concurrently permitting the identification of certain rapid non-interaction eye movements. Consequently, we seek a balance between attaining a high detection rate and accommodating some false alarm instances. The non-interaction eye movements will be discerned through fine-grained eye gesture recognition.

The rationale of the design is described as follows. In daily activities, daily eye movements have small amplitudes and long intervals, which differ greatly from our interaction gestures as shown in Fig. 6. Even in a few cases, we may simply want to look in an extreme direction and then look back. Since we want to obtain visual information in that direction, there must be visual dwell. As shown in Fig. 7, we have calculated the average intervals of various eye movements. This average interval of two saccades lasted about 0.3 seconds. Although the speed may vary among individuals, it was always less than 0.5 seconds. There is a significant interval difference between interaction and non-interaction eye movements. Therefore, we set 0.5 seconds as the maximum threshold for interactions because a saccade lasts around 200 ms on average [41], and all interactions must be completed within 0.5 seconds. Two consecutive eye saccades (quickly looking in a specific direction and quickly looking back) are considered as a complete interaction. Since each eye movement generates a peak on the dCIR data, the time interval between two eye movements can be determined by the time interval between adjacent peaks. Although the time interval between two adjacent dCIR peaks may not be exactly the same as the actual eye movement interval, they are highly correlated, serving as an indicator of the interval of eye movements.

It is unlikely that we can accurately filter out all interactions at this time due to the variability in the speed of eye movements influenced by different states, such as fatigue, where sleepiness can alter the timing of eye movements. Additionally, certain combinations of two actions may be detected, like the sequence of looking down followed by looking right. The short time interval between these actions enables their detection. However, their trajectories differ from those of interaction eye gestures and subsequent recognition algorithms aid in differentiating between these actions. The segmentation window length of 0.5 seconds is adequate for our design as eye gestures are quick. After completing the segmentation operation, the data dimensions used for subsequent analysis are configured as $4 \times 50 \times 50$. Here, 4 represents the number of links, and $50 \times 50$ denotes the dimensions




(a) Up (b) Down (c) Left (d) Right (e) Blink

(f) Up-left (g) Up-right (h) Down-left (i) Down-right (j) Neural

Fig. 8. The dCIR profiles of a subject performing different eye gestures.

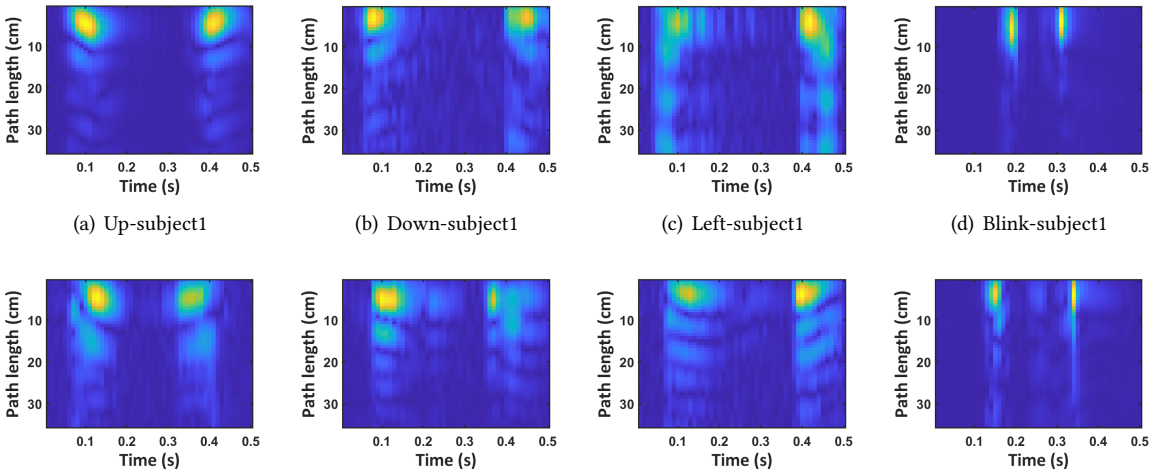
of the dCIR plot. The subsequent eye gesture recognition process relies on the 4-linked dCIR spectrogram. The dCIR profiles of a subject performing different eye gestures is shown in Fig. 8. There will be significant differences between different eye gestures, so we can recognize them through the following network design.

### 3.3 Adversarial-based Eye Gesture Recognition

The system should operate as a plug-and-play version, accommodating new users without the need for training. The eye gesture recognition module should have generalization ability and address variations among different users. Acoustic signals, collected from diverse users during the system's training phase, inherently encompass user-specific features including eye size and facial structures. The dCIR profiles of different eye gesture performed from two different subjects are shown in Fig. 9. There are differences in the dCIR profile when different users perform the same eye gestures because the dCIR profile contains user-related features. This may lead to a decrease in the accuracy of the model's predictions of new users. To address this, we employ adversarial training to achieve user-independent eye gesture recognition. Adversarial training with GR layers eliminates user-specific features and retains user-independent features. We only need to use adversarial training during the training phase, and in the test phase, we use trained models for inferences. Using adversarial training makes *EyeGesener* a plug-and-play version for new users without additional data collection and training efforts.

As depicted in Fig. 3, *EyeGesener* consists of three fundamental neural network components: a feature extractor, an eye gesture classifier, and a domain classifier. These components are denoted by their respective parameter sets $\theta_f$, $\theta_e$, and $\theta_d$. Regarding the eye gesture classifier, the optimization concentrates solely on the parameters of the feature extractor and the eye gesture classifier. This optimization is achieved by minimizing the loss function $L_c(\theta_f, \theta_e)$, ensuring accurate eye gesture classification. The domain classifier serves as the third neural network, utilizing the same set of feature outputs from the feature extractor to discern different users within the training dataset. Its optimization involves minimizing its own loss $L_d(\theta_f, \theta_d)$. Essentially, the feature extractor engages in a min-max game against the user-specific discriminator, aiming to prevent the discriminator from distinguishing between users based on the feature output. To effectively eliminate user-specific features, the comprehensive loss function for the entire system is constructed as follows:

$$L_{\text{loss}}\left(\theta_f, \theta_e, \theta_d\right) = L_c\left(\theta_f, \theta_e\right) - \lambda \times L_d\left(\theta_f, \theta_d\right), \tag{5}$$




(a) Up-subject1 (b) Down-subject1 (c) Left-subject1 (d) Blink-subject1

(e) Up-subject2 (f) Down-subject2 (g) Left-subject2 (h) Blink-subject2

Fig. 9. The dCIR profiles when different eye gestures are performed. (a)-(d) and (e)-(f) are from two different subjects.

To facilitate the training of this network, we integrate GR layers to address this challenge, following the proposal in [44]. The update process for the BP estimator, domain classifier, and feature extractor is executed through back propagation, as outlined below:

$$\theta_c = \theta_c - \alpha \cdot \frac{\partial L_c}{\partial \theta_c}, \tag{6}$$

$$\theta_d = \theta_d - \alpha \cdot \lambda \cdot \frac{\partial L_d}{\partial \theta_d}, \tag{7}$$

$$\theta_f = \theta_f - \alpha \cdot \left( -\lambda \cdot \frac{\partial L_d}{\partial \theta_f} + \frac{\partial L_c}{\partial \theta_f} \right), \tag{8}$$

where $L_d$ is the cross-entropy loss for domain classification, and $L_c$ is the cross-entropy loss for eye gesture classification. The parameters $\alpha$ and $\lambda$ correspond to the learning rate and loss weight, respectively. The loss weight, $\lambda$, is crucial in balancing the contributions of the domain classifier and eye gesture classifier in the overall optimization process. Standard stochastic gradient descendent (SGD) algorithm is applied to train the entire network directly.

In the feature extraction layer, pre-trained ResNet-18 [48] is utilized as the feature extractor. However, as the original ResNet-18 input has a dimension count of 3, an adaptation layer is introduced before ResNet-18 to ensure consistent dimensionality. For the user discriminator network and eye gesture classifier network, two linear layers are used, and a dropout rate of 0.5 is applied to the first linear layer to mitigate overfitting. The incorporation of the Dropout technique, with a 50% probability, aims to prevent the co-adaptation of the model to the training data, enhancing the generalization capabilities of the model. The output of the linear layer is processed through the soft-max layer as the network output.

When conducting eye gesture recognition, we classify eye gestures into 10 classes: eight interaction classes, blink, and null (non-interaction class). The amplitude and interval of a blink may reach the conditions of our filter module, and it has its own unique pattern, so we recognize it and consider it as a non-interaction class.

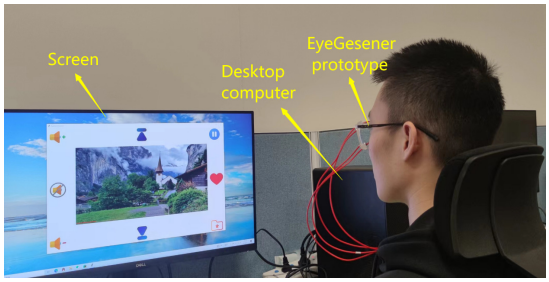
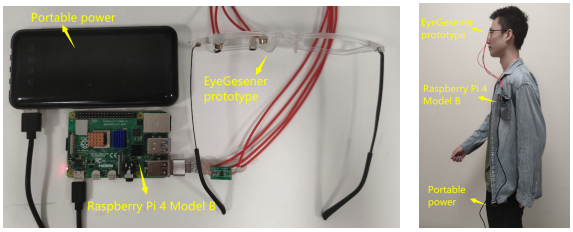



Fig. 10. The setup for the data collection using the desktop version of *EyeGesener*.




(a) (b)

Fig. 11. Portable version of *EyeGesener*. (a) Prototype details. (b) Data collection scenarios with device movement.

Eye gestures that are not similar to the eight interaction eye gestures and blink will be classified into the non-interaction class. We noticed that the eight interaction eye gestures and blinking have fixed patterns, but some patterns of eye movements in daily life (not filtered out by the eye gesture filtering) are not fixed. Therefore, in the prediction stage, we use a similarity-based strategy to filter out unseen non-interaction eye movements. We judge based on the output value of soft-max in the eye gesture classifier, if the output result of each class is less than a threshold. We will classify this eye gesture into a null class. Based on experience, we set this threshold to 0.5.

## 4 IMPLEMENTATION

The hardware prototype of *EyeGesener* is depicted in Fig. 2. We utilize a glass frame to construct the prototype. Commercial microphones [20] and speakers [17], each priced at approximately 3 RMB (0.42 USD), are acquired from online retail platforms. Two microphones and two speakers are connected to the Thitronix T256 3 ADC USB Audio sound card [25] via wired connections. The sampling rate for audio playback and recording is set at 48 kHz. We implement two versions of *EyeGesener* for different scenarios. The audio interface card is connected to an HP desktop computer equipped with a 2.9 GHz Intel(R) Core(TM) i5-10400F CPU for software processing, and this is the desktop version of *EyeGesener*. We also implement a portable version of *EyeGesener*, as shown in Fig. 11. We connect the audio interface card to a Raspberry Pi 4 Model B [5] equipped with a 1.5 GHz Quad-core Cortex-A72 CPU for software processing. We use a mobile power source for power supply, and the recorded audio data is stored on the SD card. Signal processing and network design are implemented using Python 3.8 and PyTorch 2.1. Model training is performed on an Ubuntu 22.04 server equipped with an NVIDIA GeForce RTX 3090. The model's training involves an SGD optimizer, a batch size of 36 samples, and a learning rate of 0.001.

## 5 EVALUATION

### 5.1 Data Collection

This study includes 16 participants, comprising 8 females and 8 males. The age range of the subjects was 18-35 years old. Among the 16 participants, six individuals did not wear glasses in their daily lives. The experiments take place in a laboratory and participants sit comfortably in front of the computer screen while wearing the prototype of our glasses as illustrated in Fig. 2. The distance between the user's face and the computer screen is approximately 0.8 m. Cameras are utilized to simultaneously record participants' eye movements as ground truth. Experiments are conducted following the ethical policies of our institutions.

Each participant completes two data collection sessions. In the first session, a visual indicator appears on the computer screen to guide participants in performing eye movements for interaction, covering eight directions:

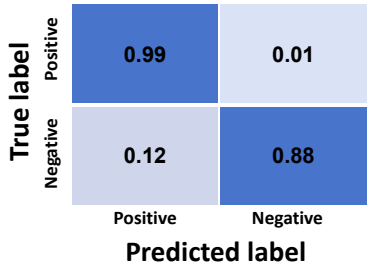



Fig. 12. Confusion matrix of interaction and non-interaction class after eye gesture filtering.

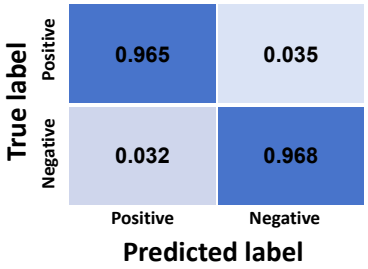



Fig. 13. End-to-end confusion matrix of interaction and non-interaction class after eye gesture recognition.

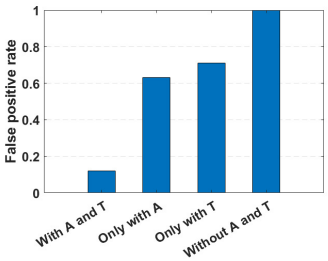



Fig. 14. The impact of thresholds. A: Amplitude threshold, T: Time threshold.

up, down, left, right, up-left, up-right, down-left, down-right. The participant performs each eye gesture 20 times. The first stage is repeated twice with a 10-minute time interval between them. During this period, they can take off their glasses and rest. In the second session, we collect participants' daily non-interaction eye movements in real-world scenarios. We set the ground truth to non-interaction for the eye movements that can be observed in the camera. Participants are instructed to wear the glasses to watch videos or other applications for 10 minutes, and data is collected during this period. They are allowed to move their bodies and engage in free conversation with others in the laboratory.

In total, 11680 samples are collected, comprising 5120 eye gesture samples for interaction and 6560 non-interaction samples (including 2400 blinks and 4160 daily eye movements). The process ensures that we collect a sufficient number of interaction eye gesture samples and non-interaction samples.

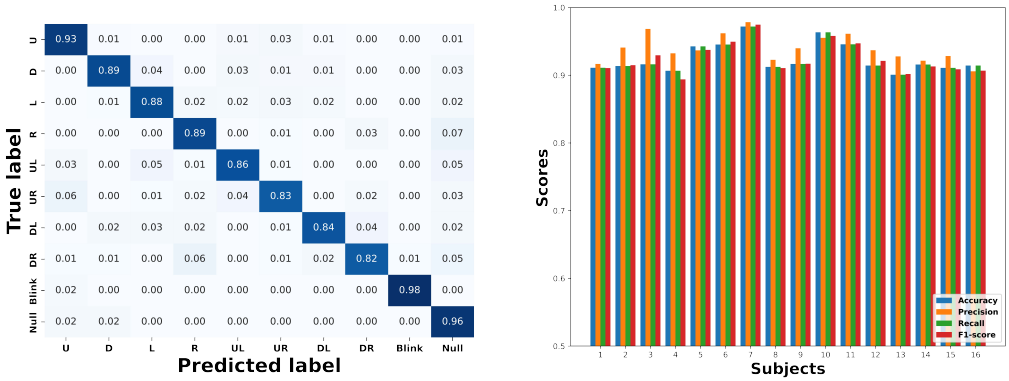
## 5.2 Performance of Eye Gesture Filtering

In this section, we evaluate the performance of eye gesture filtering composed of eye movement detection and eye gesture segmentation. All collected samples are inputted into our filter, which divides these samples into interaction and non-interaction eye gestures and then segments the interaction eye gestures. Subsequently, we compare the predicted outcomes with the ground truth.

*5.2.1 Overall Performance.* We evaluate the eye gesture filter through these four indicators: true positive (TP) rate, false positive (FP) rate, true negative (TN) rate and false negative (FN) rate. Fig. 12 depicts the confusion matrix, showcasing that TP rate is 0.99, the FN rate is 0.01, FP rate is 0.12, and TN rate is 0.88. The high TP rate implies that our system excellently detects nearly all eye gestures for interaction. Meanwhile, 0.12 of the samples are judged as false positives during this process, our eye gesture recognition will further classify these samples in subsequent classification steps. In summary, the eye gesture filter recognizes almost all eye gestures for interaction and effectively filtering out the majority of non-interaction eye movements.

*5.2.2 Impact of Eye Gesture Segmentation.* We evaluate the importance of eye gesture segmentation for the filter. Eye gesture segmentation is composed of threshold A (amplitude threshold) and threshold T (time threshold), so we present the performance of the filter without these thresholds. The data used for this analysis is all from the collection in Section 5.1. The results are then compared to the ground truth.

As shown in Fig. 14, only with the threshold A, The false positive rate has risen to 0.63, because even if there is a long interval, the combination of two eye movements exceeding the threshold A will still be retained by the filter. Only with threshold T, the false positive rate has risen to 0.71, because any two eye movements with an interval smaller than the threshold T will be retained by the filter regardless of amplitude. Obviously, after




(a) Confusion matrix.

(b) Performance on each subject.

Fig. 15. Overall performance.

removing both thresholds simultaneously, all combinations are retained with a false positive rate of 1, so setting two thresholds is very important as they enable our filter to reduce the false positive rate.

## 5.3 Performance of Eye Gesture Recognition

In this section, we demonstrate the system's performance in eye gesture recognition using data comprising samples obtained from the above section after filtering. The samples are categorized into ten classes. Subsequently, we compare the system's predicted results with the ground truth to evaluate the performance. We utilize leave-one-subject-out (LOSO) validation, where we use data from one participant for testing and data from the remaining participants for training.

*5.3.1 Overall performance.* The end-to-end confusion matrix of interaction and non-interaction class after eye gesture recognition illustrated in Fig. 13 depicts that TP rate is 0.965, the FN rate is 0.035, FP rate is 0.032, and TN rate is 0.968. The TP rate decreased from 0.99 to 0.965 compared to the results after eye gesture filtering, as some interactions were classified as non-interactions. The end-to-end FP rate decreases from 0.12 to 0.035, showing that eye movements and blink in our daily lives seldom result in false positive samples with the system.

The confusion matrix of each eye gesture is shown in Fig. 15(a) and Fig. 15(b) depicts the performance of each subject. The average accuracy, precision, recall, and F1 scores for all subjects were 0.93, 0.93, 0.93 and 0.93. As in Fig. 15(a), some interaction eye gestures exhibit a degree of confusion, and the direction of wrong classification aligns with our intuition. For example, at times, the upper right is recognized as right and upper. Considering their similarity, this is an expected outcome. Nonetheless, our system demonstrates good classification performance.

*5.3.2 The Impact of Adversarial Training.* To evaluate the impact of adversarial training, we compare the system's performance with and without the user-specific discriminator component. Without adversarial training, the system architecture undergoes modification through the removal of the user discriminator neural network. The modified training process aims to evaluate the system's eye gesture classification performance independently of the adversarial constraint imposed by the user discriminator. We utilize the data collected in Section 5.1 along with LOSO validation. The average results for accuracy, precision, recall, and F1 score with and without adversarial training are shown in Fig. 16. The model without adversarial training is prone to extract user-specific features leading to poor generalization ability for the new user. The result shows the effectiveness of adversarial training

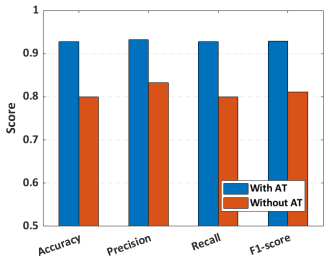



Fig. 16. Impact of adversarial training.

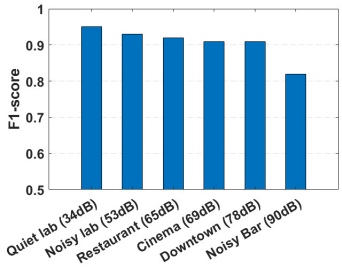



Fig. 17. Results of scenarios with different noise levels.

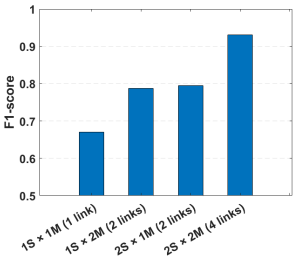



Fig. 18. Results of different sensor numbers.

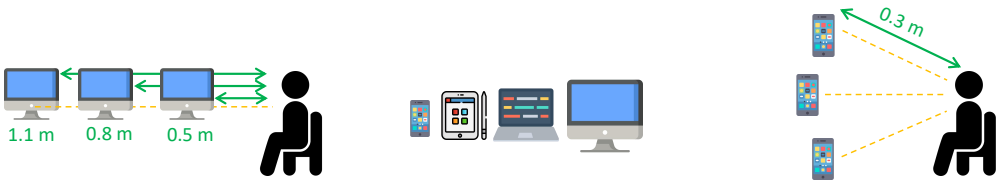
in improving the robustness and generalization capability of eye gesture recognition. Adversarial training aids in achieving user-independent eye gesture recognition.

## 5.4 Impact Factors

*5.4.1 Different Noise Levels.* We evaluate the system performance under different noise environments: (1) **Quiet lab**. The noise level of this scene is 34 dB. (2) **Noisy lab**. In this scenario, the people in the laboratory are communicating, with a noise level of 53 dB. (3) **Restaurant**. In this scenario, experiments are conducted in a student restaurant with a noise level of 65 dB. (4) **Cinema**. Experiments are conducted at the cinema with a noise level of 69 dB. (5) **Downtown**. Experiments are conducted at a city center intersection with noise, including conversations, pedestrian footsteps, vehicle noise, horn sounds, and advertisements. The noise level is 78 dB. (6) **Noisy Bar**. In this scenario, participants are invited to a noisy bar with loud music played. The noise level is 90 dB. We invite five people for this experiment, three of whom have participated in the data collection in Section 5.1. In the first two scenarios, data is collected using the desktop version of *EyeGesener*, while in the other four scenarios, data is collected using the portable version of *EyeGesener*. In each scenario, we collect 8 interaction eye gestures with each gesture repeated 20 times, and then collect non-interaction samples for 10 minutes. The first three participants use their own LOSO models, while the two new participants use the model trained using data from the previous 16 individuals in Section 5.1. The result is shown in Fig. 17. In the first five noise scenarios, the performance of *EyeGesener* is hardly affected because our system works in the inaudible frequency band, and the band-pass filter removes the audible noises. Although *EyeGesener* 's performance is slightly impacted by high-frequency noise in the noisy bar, it can still perform well due to the ideal auto-correlation property of the ZC sequence.

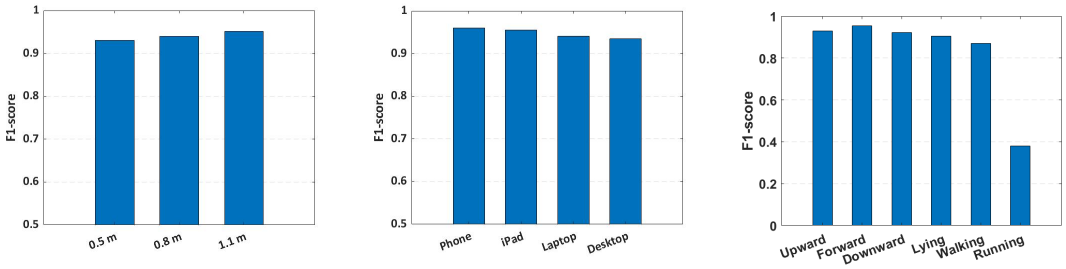
*5.4.2 Different Amounts of Links.* We evaluate the system performance using different amounts of links. We use the data collected in Section 5.1. We utilize LOSO validation and choose the combination with the best performance given the number of microphones and speakers. As shown in Fig. 18, utilizing more sensors enhances performance. This increasing number of sensors allows for sensing eye gestures from various observation angles through multiple paths. Moreover, increasing the number of sensors will lead to higher dimensional input data, thus providing more information for sensing eye gestures.

*5.4.3 Different Distances.* We evaluate whether the relative distance between the screen and the person will impact the system's performance. The visual screen distance of VR glasses is adjustable, and the distance between the screen and the person in daily life varies. It is worth noting that the interaction eye gestures are not affected by distance, but smaller distance from the device will lead to greater amplitude of non-interaction eye movements, which may result in more false positive samples. Therefore, we conduct a performance evaluation of the system




(a) Experiment under different distance (b) Experiment with different screen size (c) Experiment under different posture

Fig. 19. Experiment scenarios.



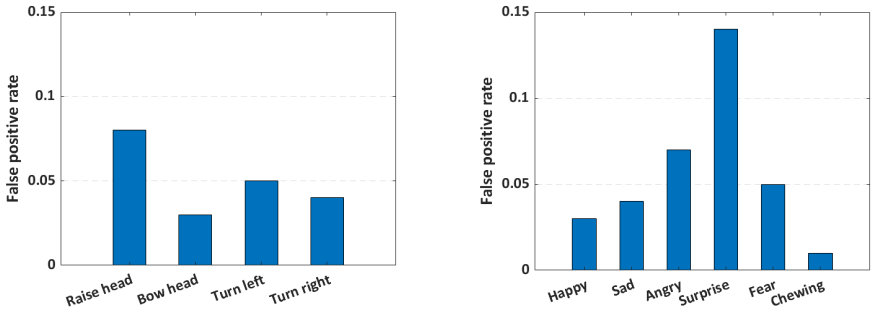

(a) Performance under different distance. (b) Performance with different screen size. (c) Performance under different posture.

Fig. 20. Experiment results.

by changing the distance between the screen and the person. As shown in Fig. 19(a), we change the distance between the glasses and the computer screen, ranging from 0.5 to 1.1 m, with a movement step of 0.3 m. Four participants who have participated in the data collection in Section 5.1 participate in this evaluation and the data collection process is the same as Section 5.1. Each participant uses their own LOSO model. The results depicted in Fig. 20(a) indicate that the system maintains good performance within normal visual distance.

*5.4.4 Different Screen Sizes.* We evaluate whether the screen size would impact the system's performance. In daily life, users will use devices with different screen sizes, and different brands of AR glasses have different virtual screen sizes [18]. The interaction eye gestures is not affected by screen size, but larger screen size will lead to greater amplitude of non-interaction eye movements, which may result in more false positive samples. As shown in Fig. 19(b), we evaluate the performance on devices with different screen sizes (mobile phone, tablet, laptop computer, and desktop computer) while maintaining a fixed distance of 1.1 m. The data collection process was the same as Section 5.1 and five subjects participated in this evaluation. The five subjects have participated in the data collection in Section 5.1 and they use their own LOSO model for evaluation. The results are depicted in Fig. 20(b). It is worth noting that as the screen increases, the amplitude of non-interaction eye movements increases, resulting in a slight decrease in performance. Nonetheless, the system can maintain good performance on devices with different screen sizes.

*5.4.5 Different Postures.* Users may adopt different postures while using glasses as shown in Fig. 19(c). In this section, we evaluate the system performance under the following positions: (i) **Looking Upward**. This is a commonly used posture during our office and leisure time. (ii) **Looking Forward**. This is also a commonly used




(a) Head movements. (b) Facial activities.

Fig. 21. False positives rate caused by head movements and facial activities.

posture. (iii) **Looking Downward**. Eye movement interaction data was collected in this posture. (iv) **Lying**. Eye movement interaction data was collected in this posture to detect whether changes in facial muscles during lying down will impact the accuracy of interaction recognition. (v) **Walking**. In this scenario, subjects are required to walk on the ground while interaction data is collected, and the step frequency can be freely adjusted by the subjects. (vi) **Running**. In this scenario, subjects are required to run on the ground and participants are free to adjust their stride frequency. Four participants who have participated in the data collection in Section 5.1 participate in this evaluation and use their own LOSO model for evaluation. In the first four scenarios, the data collection process remains consistent with Section 5.1, except that the screen changes from computer to phone, and in the latter two scenarios, participants use the portable version of *EyeGesener* and make interaction eye gestures during the movement process.

The results depicted in Fig. 20(c) show that the out system works well under the first four scenarios because the eye gesture patterns do not change under different postures. The system remains accurate in various stationary postures and can be applied to any stationary position in daily life. The performance is slightly affected while walking, while our system cannot work while running. Due to device displacement, the facial area is no longer relatively stationary to the eyewear. Device displacement is a challenge for wireless sensing, as the detected movement information originates from both the target and the device.

*5.4.6 Head Movements and Facial Activities.* We evaluate whether the system is susceptible to head movements and facial activities. To explore this concern, we collect data on four common head movements:(i) Raising the head and back, (ii) bowing the head and back, (iii) turning the head to the left and back, (iv) turning the head to the right and back and six common facial activities: (i) happy, (ii)sad, (iii) angry, (iv) surprised, (v) fear and (vi) chewing. Facial expressions and head movements are instantaneous actions of the participants and will cause skin movements. Similar to non-interaction eye movements, we focus on whether these actions will produce false positive samples. We think it is unnatural to perform interaction eye gestures and facial expressions simultaneously. Therefore, we only collect samples of these actions in this section. This section involves five participants who have participated in the data collection in Section 5.1. Instructions will appear on the screen to prompt participants to take corresponding actions. We utilize cameras to capture user actions, collecting 20 instances for each activity, resulting in a total of 1000 samples. Subsequently, we input these 1000 samples into the system to obtain prediction results. Each participant uses their own LOSO model for evaluation.

As shown in Fig. 21(a), there is an average false positive rate of around 0.05 for head movements. We observe the video and find that this is because some participants instinctively move their eyes in that direction while

(a) Glasses a (b) Glasses b (c) Glasses c (d) Glasses d

Fig. 22. Four different structures of glasses.

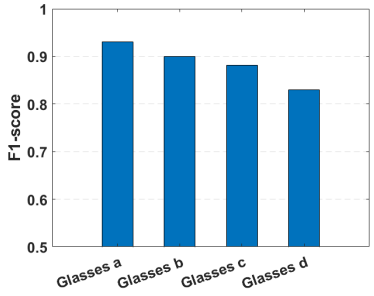



Fig. 23. Experiment results under different glasses with fine-tuning.

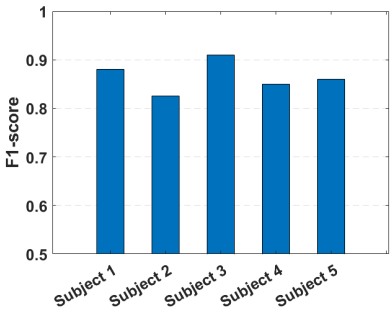



Fig. 24. Performance in real-life scenarios.

performing large head movements, resulting in a small number of false positive samples. As shown in Fig. 21(b), compared to other facial activities, the surprise expression has a higher false positive rate. We notice that the eyes undergo an up and back movement during the surprise activity, resembling the eye gesture for interactions. The false positive rates of the other five facial activities are very low. Overall, the system performs well, demonstrating its ability to handle common head movements and facial activities.

*5.4.7 Different Glasses.* We evaluate the system performance results using different glasses. We implement the system on four different commercial glasses, as shown in Fig. 22. We collect data from five subjects who have participated in the data collection in Section 5.1 and the data collection process is consistent with Section 5.1. Fig. 22(a) shows our system prototype. Glasses b are very similar to our standard glass structure. Glasses c appear as a circular frame, and the change in glass structure leads to a change in the CIR measurements. As shown in Fig. 22(d), the distance between the microphone and speaker of the VR glasses is different, so the CIR of eye movements has undergone significant changes. We cannot directly use their LOSO model on the new glasses, as there is a domain gap between them. We use a small amount of data collected on the new glasses to fine-tune the previously trained model to achieve predictive performance on the new glasses. The performance after fine-tuning is shown in Fig. 23, which proves that our system can be further deployed on the other glasses.

## 5.5 Performance in Real-life Scenarios

In this section, we evaluate the performance of *EyeGesener* in real-life scenarios. We invite 5 participants to conduct a 3-day, 1-hour daily, uncontrolled real-life experiment. This experiment involves 2 participants who have participated in the data collection in Section 5.1 and 3 new participants. They all receive training on eye gestures for interaction before the experiment. We use the portable version of *EyeGesener* to collect data and store the data on the SD card, which allows participants to move around with our glasses freely. They can wear our glasses to walk, have meetings, play games, and engage in any activity. They perform eye gestures when they

Table 1. The average latency of *EyeGesener*.

| | CIR estimation | | Eye gesture filter | Eye gesture recognition | Total |
|---|---|---|---|---|---|
| **HP desktop computer** | 10.8 ms | | 0.1 ms | 7.6 ms | 18.5 ms |
| | Demodulation | Subtraction | | | |
| | 10.3 ms | 0.5 ms | | | |
| **Raspberry Pi 4 Model B** | 17.7 ms | | 0.1 ms | 23.5 ms | 41.3 ms |
| | Demodulation | Subtraction | | | |
| | 16.9 ms | 0.8 ms | | | |

want to interact to ensure that we can collect enough interaction samples. A camera records their eye movements simultaneously as the ground truth. The first two participants use their own LOSO model, while the other three participants use the model trained using data from the previous 16 individuals collected in Section 5.1.

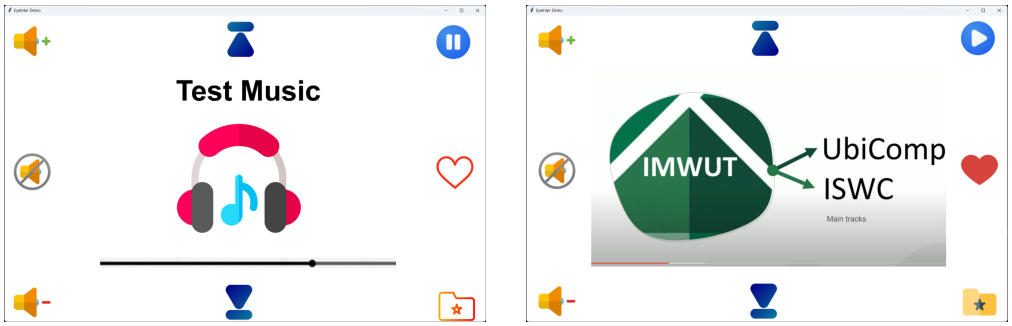
We calculate the F1 score for each participant, and the results are depicted in Fig. 24. It is worth noting that subject 3 has better results than subject 2. By observing the video, we find that subject 3 uses our system more in a stationary state, such as playing games and eating in a noisy restaurant. Subject 2 uses our system more in movement scenarios, such as walking up and down the stairs with body movement, which causes some relative displacement between the glasses and the user's eyes. Nonetheless, our system still performs well for subject 2 with an F1 score of 0.82. The average F1 score of 5 people is 0.865, demonstrating the system's robustness in real-life scenarios.

## 5.6 Latency

In this section, we evaluate the latency of the *EyeGesener* software system. In the current implementation, the software operates on an HP desktop computer and a Raspberry Pi 4 Model B. We calculate latency on two versions of *EyeGesener*, respectively. We perform eye gesture inference 1000 times and compute the average runtime for each component of our system, including CIR estimation, eye gesture filtering, and eye gesture recognition. The results are presented in Tab. 1. This result shows that the average time required for our system to predict an eye gesture is 18.5 milliseconds and 41.3 milliseconds for the desktop version and portable version, respectively, and this delay can be further reduced with more powerful devices.

## 5.7 Power Consumption

Power Consumption is an essential issue for smart glasses. We measure the power consumption of our system implemented on the Raspberry Pi 4 Model B using a USB power meter [4]. We implement a real-time processing pipeline on the Raspberry Pi 4 Model B with an FPS of 20 Hz and measure the power consumption while all components are operating. It is worth noting that we only need model inference for the adversarial-based eye gesture recognition component in the real-time pipeline. We measure the difference between the power consumption while running the real-time processing pipeline and the power consumption in the idle state as *EyeGesener*'s power consumption. The measurements show that the current flowing through our system was 0.36 A at the voltage of 5.09 V. Therefore, the power consumption of *EyeGesener* is 1.83 W with a refreshing rate of 20 Hz. The power consumption of two speakers and two microphones is only 0.23 W. This is reasonable because low energy is not the top priority for the Raspberry Pi 4 Model B. This power consumption should allow *EyeGesener* to work on current smart glasses for a reasonable period. For instance, the battery capacity of Google Glass, Epson Moverio, and Microsoft HoloLens are 570 mAh [21], 3400 mAh [10], and 16500 mAh [11], guaranteeing around 1.6, 9.4, and 45.8 hours of battery life in theory if *EyeGesener* is used alone. We further discussed methods for reducing system power consumption in Section 7.




(a) Screenshot of listening to music.

(b) Screenshot of watching videos.

Fig. 25. Screenshot of the interactive video-music player.

## 6 USER STUDY

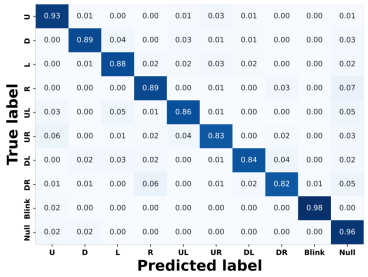
In addition to the offline performance evaluation, we perform user studies to evaluate the usability of *EyeGesener*. We implement a real-time version of *EyeGesener* with an FPS of 20 Hz to enable user interaction with *EyeGesener*. The evaluation focuses on real-time and real-world eye movement recognition performance, with a specific emphasis on assessing the usability of *EyeGesener* through user feedback. The study simulates the experience with AR glasses using a combination of glasses and a computer monitor. The glasses listen to users' eye gestures, and the monitor simulates the projection of AR glasses.

### 6.1 App Design

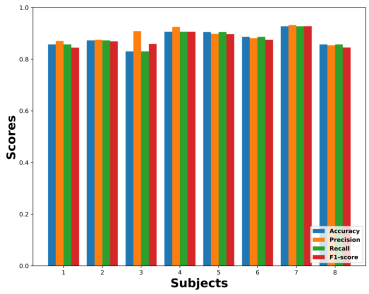
An interactive real-time audio-video player is developed based on *EyeGesener*. The design of our app is rooted in the following two principles. First, the positions of different function buttons should mirror those commonly found in everyday apps, allowing users to learn and use them with reduced learning costs. Second, the function buttons should be positioned in the 8 directions of the app, allowing an eye gesture for interaction to be considered as pressing the corresponding button when the user looks at the button direction. Screenshots of music playback are shown in Fig. 25(a), and screenshots of video playback with this app are displayed in Fig. 25(b). Similar to watching short video apps in our daily lives, the interaction action of an upward eye gesture is a slide up, representing the next video. Conversely, the interaction action of a downward eye gesture is a slide down, indicating the previous video. Liking a video involves looking at the like button, typically to the right. Similarly, the bookmark button is in the down-right corner. To pause, users can look to the up-right corner. The second time to perform the same operation is to cancel it. The up-left, left, and down-left directions on the left represent increasing volume, muting, and decreasing volume, respectively. The music player operates in the same way as the video player.

### 6.2 Study Process

This study enrolls eight participants, consisting of 4 females and 4 males. The deep learning model used in this user study is trained using the data collected in Section 5.1, which occurs approximately two weeks before the user study. It is worth noting that the four participants who have participated in the data collection in Section 5.1 use their own LOSO model. The remaining participants use models trained on all data in the dataset. The study commences by providing participants with a guide on how to use the application, with a specific emphasis on the eye gesture instructions. Participants are permitted to practice eye movement exercises on the prototype




(a) Confusion matrix.




(b) Performance on each subject.

Fig. 26. Performance in the user study.

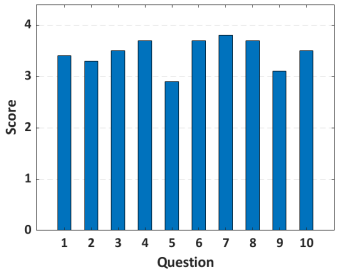



Fig. 27. The SUS questionnaire results.

to familiarize themselves with the interaction method. The introduction and practice phase typically takes five minutes. The settings for the user research are similar as Fig. 10.

Once participants feel sufficiently prepared, the experiment commences. In each experiment, participants are verbally instructed to complete five tasks before being allowed to navigate the app freely. Instructions include tasks such as "Switch down to the third video; if you find it appealing, perform the like operation. If you also like the previous video, switch back and bookmark it. If someone is talking to you, mute and pause the video." After completing the five instructed tasks, users are then free to use our system. Each participant's actual application experience lasts approximately 10 minutes. Cameras record the user's eye movements in this process, providing the ground truth data.

Following the experiment, participants complete a System Usability Scale (SUS) questionnaire and provide feedback on *EyeGesener*'s performance through the questionnaire [30]. In addition, individual interviews are conducted to gain insights into participants' subjective evaluations and user experiences of *EyeGesener*. The discussions cover participants' overall impressions, strengths, and weaknesses, as well as provide suggestions for system improvement. Participants are encouraged to pose questions throughout the research process, and each interview lasts approximately 10 minutes.

## 6.3 Results

*6.3.1 Eye Gesture Recognition Accuracy.* In this section, we assess the real-time eye gesture recognition accuracy of *EyeGesener* in real-time applications. This complements the offline evaluations in Section 5.1, which are conducted under controlled conditions where a specific type of eye movement is repeated multiple times, and similar eye movements are grouped together. The purpose of this real-time evaluation is to determine whether *EyeGesener* can perform well in real-world app usage when users can navigate the app freely.

The confusion matrix for each eye gesture is depicted in Fig. 26(a). The average accuracy, precision, recall, and F1 score across the 10 classes are 0.89, 0.90, 0.89, and 0.89, respectively. The performance for each subject is illustrated in Fig. 26(b), where subject 1 exhibits the lowest F1 score of 0.84, while subject 7 attains the highest F1 score of 0.93. A comparison with the results from the offline study reveals a slight decrease in the average F1 score, amounting to 0.04. The decline in accuracy may be attributed to inaccuracies in performing eye gestures for interaction. Users prioritize interaction over the accurate completion of eye gestures during the usage of the real-time app.

*6.3.2 Questionnaire Results.* The average score for each question in the standard SUS questionnaire is depicted in Fig. 27. Using the scoring calculation method introduced in [30] (4 is the highest score), we compute the total

system score: 86.5 ± 5.08 out of 100, obtaining an overall usability value. The results indicate that the usability of *EyeGesener* is exceptionally high. According to the standard, Q4 and Q10 pertain to ease of learning, while the rest focus on usability. After considering these two classifications separately, the score for ease of learning is 90.5 (±3.70), and the score for usability is 85.6 (±8.03). This outcome demonstrates that *EyeGesener* is easy to learn and practical to use.

*6.3.3 Subjective Feedback.* Almost everyone mentions **Enjoyment** and praises this system, "It's really interesting. I just need to lie comfortably in a chair to control electronic devices at will." (P1). "This is the system I've been looking forward to using before. I think it's too convenient." (P7). Several participants mention the issue of **Rationality**, "The system shares the same manual interaction method as existing mainstream video applications. I completely replaced my hands with my eyes." (P1). "I only needed to look in the direction of the button and quickly look back to complete the interaction." (P2). "I easily learned this interaction method and freed my hands." (P4). Most users think the system has good **Accuracy**, "I think it's accurate, and the system always follows my eye movements to execute corresponding instructions." (P3). "Before using it, I was worried that eye movements were easy to accidentally touch, but to my surprise, it showed good accuracy." (P5). However, one user expresses dissatisfaction. "I accidentally touched it during use, and I don't know why it suddenly turned silent." (P6). The ability to fully distinguish between eye movements for interaction and daily eye movements is an area in which our system needs further improvement. A participant mentions **Distraction**, "My eyes are always moving, and this eye movement interaction distracts me." (P6). However, when we ask all the other users, they do not think so, "I also need to slide with my hands; it's better to just slide with my eyes like this." (P8). "The interaction was completed in less than 0.5 seconds. This doesn't affect my viewing experience." (P3). Three participants (P1, P6, and P8) mention **Vision Blockage**, "My line of sight is obstructed by these microphones, speakers, and wires." (P8). However, this is considered a solvable engineering problem, and we discuss it in Section 7, where the sensors can be embedded into the mirror frame without appearing abrupt. Three participants (P1, P4, and P7) express their mention of **Social Concerns**, saying, "I think using this eye movement method may be perceived as strange by others and cause unnecessary social attention." (P4). We attribute this to inherent concerns related to the appearance of smart glasses, while in many dark or VR glasses scenes, our eyes are not seen by others. Compared to facial expressions and silent speech, our approach avoids social concerns. For the **Frequency Leakage**, all participants express that they do not hear any noise from the glasses. We also measure the signal level while the system was operating by separately placing a digital sound level meter [6] close enough to the speakers and near users' ears. The decibel meter gives an average signal level of 44.6 dB(A) and 37.3 dB(A), respectively. Comparing the result with noise exposure limits recommendation from Centers for Disease Control and Prevention, which is 85 dB(A) over eight hours in the work space [1], *EyeGesener* is safe to wear with little concern of hearing damage.

## 7 DISCUSSION

In this section, we discuss the limitations of our system and potential future work.

**The number of interaction gestures**. Our system supports 8 interaction eye gestures: up, down, left, right, up-left, up-right, down-left, and down-right. Performing these gestures in these eight directions is natural and easy. Recent study [28] demonstrated that approximately five types of command recognition are necessary and sufficient for simple hands-free input. For instance, five commands (play, stop, forward, back, and like) are necessary and sufficient for operating a media player such as music, video, and images. Therefore, the provided number of interactions can meet the requirements of most interactive applications in daily life. In the future, we will explore recognizing additional eye gestures to expand the interaction space of the system, such as the recognition of combinations of existing eye gestures similar to [36].

**Power Consumption.** Power consumption is an important issue for smart glasses. There are some methods to further reduce the power consumption of *EyeGesener*. First, replacing the speaker with more power-efficient

ones while maintaining the same sound pressure level can reduce the power consumption [56]. Second, since power consumption is not the top priority for Raspberry Pi 4 Model B, we can reduce the power consumption by implementing *EyeGesener* on a low-energy chip such as GAP 8 low-power neural accelerator [3]. Moreover, we can further decrease the power consumption of the ResNet module by using knowledge distillation techniques to transfer the large model to a lightweight one [91].

**Device displacement**. Although *EyeGesener* can currently perform well in various stationary scenarios and even while walking, it is not suitable for fast-moving scenarios (such as running) and bumpy scenarios (sitting in a car on a bumpy road). Due to device displacement, the facial area is no longer relatively stationary to the eyewear. Device displacement is a challenge for wireless sensing, as the detected movement information originates from both the target and the device. A potential approach to addressing this challenge is to employ the method that enables acoustic sensing under device motion [61].

**Vision blockage**. Although we have made minimally obtrusive modifications to the glass frame to reduce visual interference to users, the presence of microphones, speakers, and wires may still cause vision blockage. This is because the microphone and speaker are glued to the four corners of the glass frame rather than embedded inside it. Addressing this issue is considered a solvable engineering problem. In the future, it will be possible to embed the sensors into the mirror frame without appearing abruptly. We can fully integrate the sensors into the glasses frame and retain some openings for audio playback and recording, following the practices of existing commercial glasses [22].

## 8 RELATED WORK

In this section, we present an overview of the related work of this study. We categorize the relevant works into eye movement-based interaction systems and acoustic sensing for interaction.

### 8.1 Eye Movement-based Interaction Systems

*8.1.1 Camera-based Interaction.* Camera-based methods can accurately recognize eye movements using computer vision techniques. Webcam-based eye-tracking platforms provide online eye-tracking solutions [7, 9, 13], but their performance can be affected by lighting conditions, occlusions, and camera orientations due to their fixed location, and relatively low resolution. Cameras with a higher resolution than webcams can provide more reliable eye tracking results. Tobii Pro Fusion [15] and Tobii Pro Glasses 3 [16] are widely used commercial eye trackers and can provide accurate eye tracking for new users with a calibration process. Many other commercial eye trackers can also continuously track eye movements, such as Dikablis Glasses 3 [8], Pupil Labs [12], and SMI Eye Tracking Glasses [14]. Recent studies have focused on improving eye-tracking performance from different perspectives. Ahlstrom et al. provide larger eye tracking coverage with additional cameras [26]. Hennessey and Fiset enable long range eye tracking into the living room and allow for freedom of user motion [49]. Mahanama and Bhanuka propose a multi-user eye tracking system with commodity hardware [62]. Ryan et al. present a low-cost wearable eye tracker under variable lighting conditions [78]. Mayberry et al. present camera-based eye-tracking solutions with accurate tracking performance while maintaining a low-power consumption [66, 67] with performance validation in outdoor settings. Various eye movement-based interaction systems have been enabled with the advancement of these technologies. Researchers have concentrated on directly manipulating user interfaces through video-based gaze trackers [74, 75] for human-computer interaction (HCI). Pfeiffer et al. [72] propose a method for analyzing eye tracking data employing computer cameras and augmented reality technology. Matsubara et al. achieve the extraction of read text using a wearable eye tracker [65]. Kytö et al. investigate precise selection techniques using head motion and eye gaze [54]. Paletta et al. enable accurate gaze recovery on mobile displays [71]. Other studies employ eye tracking for controlling computer functions [52, 63]. However, camera-based methods require expensive hardware manufacturing costs [69], and these methods

are susceptible to poor ambient light conditions [83]. Furthermore, privacy concerns also pose a drawback in camera-based interaction.

*8.1.2 EOG-based Interaction.* Multichannel EOG electrodes serve as an intrusive method for eye tracking. Barea et al. [31] introduce an EOG model for eye-based computer interaction incorporating wavelet transform and neural network techniques. Bulling et al. [33] present a wearable EOG goggle designed to enhance situational awareness and facilitate eye-based human-computer interaction. Xiao et al. [88] propose a human-computer interface based on a single-channel EOG signal, enabling real-time interactions within the VR environment. Yamagishi et al. [92] develop a communication support interface for individuals with motor disabilities who cannot speak, controlled by eye movement and voluntary blink. Bulling et al. implement a novel eye tracker for context-awareness and mobile HCI applications using wearable EOG goggles [32]. Kunze et al. use smart glasses with integrated electrodes to detect eye movements in application cases from reading detection to talking recognition for social interaction tracking [53]. However, EOG-based interaction methods require skin contact, making it intrusive and less user-friendly. Moreover, these methods are vulnerable to sweat artifacts [39] as they necessitate physical contact with the skin.

*8.1.3 Infrared-based Interaction.* Infrared-based interaction methods utilize infrared sensors integrated into glasses to detect eye movements by sensing eyelid skin movements. Google Glass [2] recognizes intentional blink gestures, while Dual Blink [40] identifies natural blinks. Cho et al. propose a gaze estimation method using wearable near infrared devices [37]. Li et al. [58] propose to use near infrared emitters and receivers on glasses for continuous eye tracking, but this method can be impacted by glasses movement and direct sunlight. Researchers [43] utilized glasses equipped with infrared distance sensors to detect eyelid skin movements induced by gaze motions. They applied machine learning to analyze the time-series sensor data for identifying gaze motions. Wang et al. [85] introduce a method for detecting pupils and blinks in a gaze tracking system, utilizing wearable camera sensors and a near-infrared LED array. Masai et al. [64] employ 16 sensors to identify seven distinct gestures, including blinking and gaze shifts at 90-degree intervals in motion direction. However, infrared-based interaction methods may not be suitable for prolonged interaction in everyday environments due to the sensitivity of infrared sensors to environmental conditions such as sunlight, smoke, etc. [19, 77].

## 8.2 Acoustic Sensing for Interaction

Due to the ubiquity and low-cost of speakers and microphones [82] and fine-grained sensing ability [34], acoustic sensing has also been widely studied. Acoustic sensing has been widely applied in HCI systems, such as hand gesture recognition [29, 80, 87, 89], facial expression recognition [56, 57, 90, 97], silent speech recognition [45, 51, 96, 98], handwriting recognition [35, 86, 94, 95], and so on.

Recently, some researchers have utilized acoustic sensing to detect eye activities. BlinkListener [60] detects eye blink motion using acoustic signals in a contact-free manner with a pair of microphone and a speaker. TwinkleTwinkle [36] proposes an interacting method with smart devices through eye blink, leveraging ultrasound signals on commercial devices. However, these methods cannot distinguish finer-grained eye movements beyond blinking. Therefore, the interaction space is very limited. A study [81] enables eye-tracking on glasses utilizing piezoelectric micromachined ultrasonic transducers. However, this method requires ultrasound in the MHz band, but commercial speakers and microphones support frequencies up to 24 kHz under the sampling rate of 48 kHz. Golard et al. [46] conduct a modeling and empirical study to prove that ultrasound can achieve low-power, fast, and light-insensitive eye tracking results. However, this method is evaluated on a physical 3D model of a human eye, and its performance on a real user is unknown. Li et al. propose GazeTrak [55], an acoustic-based eye-tracking system on glasses. It achieved accurate eye-tracking results with low power consumption. However, it does not address the Midas Touch problem, which requires distinguishing between eye movements for interaction and

daily eye movement. Moreover, GazeTack cannot achieve user-independent results and needs additional data collection and model training efforts to achieve accurate eye-tracking results for new users.

To address the limitations of previous works, we propose an acoustic-based, user-independent eye gesture recognition system for hands-free interaction on smart glasses. We carefully design eye gestures for interaction and propose techniques to mitigate the Midas touch problem. With minimally obtrusive modifications to a glasses frame. *EyeGesener* is low-cost, contact-free, and suitable for long-term interaction in everyday environments.

## 9 CONCLUSION

In this paper, we propose the first acoustic-based, user-independent eye gesture recognition system for hands-free interaction on smart glasses. We employ OFDM de/modulation schemes that enable both speakers to transmit simultaneously simultaneously and in the same frequency band for CIR estimation. CIR measures the movements of the eyelids and surrounding skin elicited by eye gestures. We meticulously design eye gestures for interaction to mitigate the Midas touch problem. We design eye gesture filtering and adversarial-based eye gesture recognition to identify intentional eye gestures for interaction and filter out daily eye movements. We employ an adversarial training strategy, incorporating GR layers, to extract user-dependent features related to eye movements. We evaluated *EyeGesener* with 16 subjects to assess the performance of eye gesture recognition. The results indicate that our proposed system attains high accuracy and robustness with an average F1-score of 0.93. A study with 8 participants demonstrates the high usability and learnability of *EyeGesener*. The evaluation results demonstrate that *EyeGesener* can be further deployed on commercial smart glasses.

## REFERENCES


[1] 1998. *Criteria for a recommended standard: occupational noise exposure.* https://www.cdc.gov/niosh/docs/98-126/
[2] 2013. *The Google Glass Wink Feature Is Real|TechCrunch.* https://techcrunch.com/2013/05/09/the-google-glass-wink-feature-is-real/
[3] 2020. *GAP8 IoT Application Processor.* https://greenwaves-technologies.com/wp-content/uploads/2021/04/Product-Brief-GAP8-V1_9.pdf
[4] 2020. *POWER-Z KM001.* https://www.chargerlab.com/power-z-km001-usb-power-tester-voltage-current-ripple-dual-type-c-meter/
[5] 2020. *raspberry-pi-4-model-b.* https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
[6] 2020. *SanLiang Sound Level Meter.* https://www.mtmpre.com.my/showproducts/productid/3939952/cid/0/sanliang-sound-level-meter/
[7] 2021. *WebGazer.js: Democratizing Webcam Eye Tracking on the Browser.* https://webgazer.cs.brown.edu/#publication
[8] 2022. *Dikablis Glasses 3.* https://autonomoustuff.com/products/ergoneers-eye-tracking
[9] 2022. *GazeRecorder Webcam Eye Tracking.* https://gazerecorder.com/webcam-eye-tracking-accuracy/
[10] 2022. *Microsoft HoloLens.* https://epson.com/moverio-augmented-reality?utm_source=marketing&utm_medium=van&utm_campaign=us-moverio
[11] 2022. *Microsoft HoloLens.* https://www.niora.net/en/p/microsoft_hololens
[12] 2022. *Pupil Labs.* https://pupil-labs.com/products
[13] 2022. *RealEye Webcam Eye-Tracking.* https://www.realeye.io/
[14] 2022. *SMI Eye Tracking Glasses.* https://imotions.com/products/hardware/smi-eye-tracking-glasses/
[15] 2022. *Tobii Pro Fusion.* https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-fusion
[16] 2022. *Tobii Pro Glasses 3.* https://www.oppo.com/en/newsroom/press/oppo-air-glass/
[17] 2023. *0613-D65.* https://m.tb.cn/h.5LI0uVM9lBHwxJD?tk=8ukQWh3UTbe
[18] 2023. *The Best AR Smart Glasses of 2024.* https://www.popularmechanics.com/technology/gadgets/a44067373/best-ar-smart-glasses/
[19] 2023. *ECSTUFF4U for Electronics Engineer.* https://www.ecstuff4u.com/2019/08/infrared-sensor-advantage-disadvantage.html
[20] 2023. *EO6022G-42P.* https://m.tb.cn/h.5MOf9lV?tk=ciRyWTOMs9G
[21] 2023. *Google Glass.* https://en.wikipedia.org/wiki/Google_Glass
[22] 2023. *huawei eyewear 2.* https://consumer.huawei.com/en/audio/huawei-eyewear-2/
[23] 2023. *Iristick - Smart glasses built for every industry.* https://iristick.com/
[24] 2023. *Rokid Glass 2 | Everyday AR Glasses Built for Enterprises.* https://rokid.ai/products/rokid-glass-2/
[25] 2023. *T256 3 ADC USB Audio sound card.* https://www.thitronix.com/Products/Index-142.html
[26] Christer Ahlstrom and Tania Dukic. 2010. Comparison of eye tracking systems with one and three cameras. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research.* 1–4.

[27] Takashi Amesaka, H. Watanabe, and Masanori Sugimoto. 2019. Facial expression recognition using ear canal transfer function. (Jan 2019).

[28] Takashi Amesaka, Hiroki Watanabe, and Masanori Sugimoto. 2019. Facial expression recognition using ear canal transfer function. In *Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 1–9.

[29] Takashi Amesaka, Hiroki Watanabe, Masanori Sugimoto, and Buntarou Shizuki. 2022. Gesture recognition method using acoustic sensing on usual garment. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–27.

[30] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[31] Rafael Barea, Luciano Boquete, Sergio Ortega, Elena López, and JM Rodríguez-Ascariz. 2012. EOG-based eye movements codification for human computer interaction. *Expert Systems with Applications* 39, 3 (2012), 2677–2683.

[32] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2008. It's in your eyes: towards context-awareness and mobile HCI using wearable EOG goggles. In *Proceedings of the 10th international conference on Ubiquitous computing*. 84–93.

[33] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2009. Wearable EOG goggles: eye-based interaction in everyday environments. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 3259–3264.

[34] Chao Cai, Rong Zheng, and Jun Luo. 2022. Ubiquitous acoustic sensing on commodity iot devices: A survey. *IEEE Communications Surveys & Tutorials* 24, 1 (2022), 432–454.

[35] Mingshi Chen, Panlong Yang, Jie Xiong, Maotian Zhang, Youngki Lee, Chaocan Xiang, and Chang Tian. 2019. Your table can be an input panel: Acoustic-based device-free interaction recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–21.

[36] Haiming Cheng, Wei Lou, Yanni Yang, Yi-pu Chen, and Xinyu Zhang. 2023. TwinkleTwinkle: Interacting with Your Smart Devices by Eye Blink. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 2 (2023), 1–30.

[37] Chul Woo Cho, Ji Woo Lee, Kwang Yong Shin, Eui Chul Lee, Kang Ryoung Park, Heekyung Lee, and Jihun Cha. 2012. Gaze Detection by Wearable Eye-Tracking and NIR LED-Based Head-Tracking Device Based on SVR. *Etri Journal* 34, 4 (2012), 542–552.

[38] Brendan David-John, Candace Peacock, Ting Zhang, T Scott Murdison, Hrvoje Benko, and Tanya R Jonker. 2021. Towards gaze-based prediction of the intent to interact in virtual reality. In *ACM Symposium on Eye Tracking Research and Applications*. 1–7.

[39] Lourdes DelRosso, Richard B Berry, Suzanne E Beck, Mary H Wagner, and Carole L Marcus. 2016. *Pediatric Sleep Pearls E-Book*. Elsevier Health Sciences.

[40] Artem Dementyev and Christian Holz. 2017. DualBlink: a wearable device to continuously detect, track, and actuate blinking for alleviating dry eyes and computer vision syndrome. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 1 (2017), 1–19.

[41] Burkhart Fischer and E Ramsperger. 1984. Human express saccades: extremely short reaction times of goal directed eye movements. *Experimental brain research* 57 (1984), 191–195.

[42] Kyosuke Futami, Kohei Oyama, and Kazuya Murao. 2022. Augmenting Ear Accessories for Facial Gesture Input Using Infrared Distance Sensor Array. *Electronics* 11, 9 (May 2022), 1480. https://doi.org/10.3390/electronics11091480

[43] Kyosuke Futami, Yuki Tabuchi, Kazuya Murao, and Tsutomu Terada. 2022. Exploring Gaze Movement Gesture Recognition Method for Eye-Based Interaction Using Eyewear with Infrared Distance Sensor Array. *Electronics* 11, 10 (2022), 1637.

[44] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research* 17, 59 (2016), 1–35.

[45] Yang Gao, Yincheng Jin, Jiyang Li, Seokmin Choi, and Zhanpeng Jin. 2020. Echowhisper: Exploring an acoustic-based silent speech interface for smartphone users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–27.

[46] Andre Golard and Sachin S Talathi. 2021. Ultrasound for Gaze Estimation—A Modeling and Empirical Study. *Sensors* 21, 13 (2021), 4502.

[47] Jibo He, Alex Chaparro, Bobby Nguyen, Rondell Burge, Joseph Crandall, Barbara Chaparro, Rui Ni, and Shi Cao. 2013. Texting while driving. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. https://doi.org/10.1145/2516540.2516560

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

[49] Craig Hennessey and Jacob Fiset. 2012. Long range eye tracking: bringing eye tracking into the living room. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 249–252.

[50] Robert JK Jacob. 1993. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in human-computer interaction* 4 (1993), 151–190.

[51] Yincheng Jin, Yang Gao, Xuhai Xu, Seokmin Choi, Jiyang Li, Feng Liu, Zhengxiong Li, and Zhanpeng Jin. 2022. EarCommand: " Hearing" Your Silent Speech Commands In Ear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–28.

[52] Chairat Kraichan and Suree Pumrin. 2014. Face and eye tracking for controlling computer functions. In *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 1–6.

[53] Kai Kunze, Katsuma Tanaka, Shoya Ishimaru, Yuji Uema, Koichi Kise, and Masahiko Inami. 2015. MEME: eye wear computing to explore human behavior. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. 361–363.
[54] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A Lee, and Mark Billinghurst. 2018. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.
[55] Ke Li, Ruidong Zhang, Boao Chen, Siyuan Chen, Sicheng Yin, Saif Mahmud, Qikang Liang, François Guimbretière, and Cheng Zhang. 2024. GazeTrak: Exploring Acoustic-based Eye Tracking on a Glass Frame. *arXiv preprint arXiv:2402.14634* (2024).
[56] Ke Li, Ruidong Zhang, Siyuan Chen, Boao Chen, Mose Sakashita, François Guimbretière, and Cheng Zhang. 2024. EyeEcho: Continuous and Low-power Facial Expression Tracking on Glasses. *arXiv preprint arXiv:2402.12388* (2024).
[57] Ke Li, Ruidong Zhang, Bo Liang, François Guimbretière, and Cheng Zhang. 2022. Eario: A low-power acoustic sensing earable for continuously tracking detailed facial movements. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–24.
[58] Tianxing Li and Xia Zhou. 2018. Battery-free eye tracker on glasses. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 67–82.
[59] Sikun Lin, Hao Fei Cheng, Weikai Li, Zhanpeng Huang, Pan Hui, and Christoph Peylo. 2016. Ubii: Physical world interaction through augmented reality. *IEEE Transactions on Mobile Computing* 16, 3 (2016), 872–885.
[60] Jialin Liu, Dong Li, Lei Wang, and Jie Xiong. 2021. BlinkListener: " Listen" to Your Eye Blink Using Your Smartphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–27.
[61] Jialin Liu, Dong Li, Lei Wang, Fusang Zhang, and Jie Xiong. 2022. Enabling contact-free acoustic sensing under device motion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–27.
[62] Bhanuka Mahanama. 2022. Multi-user eye-tracking. In *2022 Symposium on Eye Tracking Research and Applications*. 1–3.
[63] Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*. Springer, 39–65.
[64] Katsutoshi Masai, Kai Kunze, and Maki Sugimoto. 2020. Eye-based Interaction Using Embedded Optical Sensors on an Eyewear Device for Facial Expression Recognition. In *Proceedings of the Augmented Humans International Conference*. https://doi.org/10.1145/3384657.3384787
[65] Mizuki Matsubara, Joachim Folz, Takumi Toyama, Marcus Liwicki, Andreas Dengel, and Koichi Kise. 2015. Extraction of read text using a wearable eye tracker for automatic video annotation. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. 849–854.
[66] Addison Mayberry, Pan Hu, Benjamin Marlin, Christopher Salthouse, and Deepak Ganesan. 2014. iShadow: design of a wearable, real-time mobile gaze tracker. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 82–94.
[67] Addison Mayberry, Yamin Tun, Pan Hu, Duncan Smith-Freedman, Deepak Ganesan, Benjamin M Marlin, and Christopher Salthouse. 2015. CIDER: Enabling robustness-power tradeoffs on a computational eyeglass. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 400–412.
[68] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1515–1525.
[69] Diederick C. Niehorster, Roy S. Hessels, and Jeroen S. Benjamins. 2020. GlassesViewer: Open-source software for viewing and analyzing data from the Tobii Pro Glasses 2 eye tracker. *Behavior Research Methods* 52, 3 (Jun 2020), 1244–1253. https://doi.org/10.3758/s13428-019-01314-1
[70] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 131–136.
[71] Lucas Paletta, Helmut Neuschmied, Michael Schwarz, Gerald Lodron, Martin Pszeida, Stefan Ladstätter, and Patrick Luley. 2014. Smartphone eye tracking toolbox: accurate gaze recovery on mobile displays. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 367–68.
[72] Thies Pfeiffer and Patrick Renner. 2014. EyeSee3D: a low-cost approach for analyzing mobile 3D eye tracking data using computer vision and augmented reality technology. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 195–202.
[73] Branislav M Popovic. 1992. Generalized chirp-like polyphase sequences with optimum correlation properties. *IEEE Transactions on Information Theory* 38, 4 (1992), 1406–1409.
[74] Marco Porta and Matteo Turina. 2008. Eye-S. In *Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA '08*. https://doi.org/10.1145/1344471.1344477
[75] Pernilla Qvarfordt and Shumin Zhai. 2005. Conversing with the user based on eye-gaze patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/1054972.1055004
[76] A Rodríguez Valiente, A Trinidad, JR García Berrocal, C Górriz, and R Ramírez Camacho. 2014. Extended high-frequency (9–20 kHz) audiometry reference thresholds in 645 healthy subjects. *International journal of audiology* 53, 8 (2014), 531–545.

[77] Soha Rostaminia, Alexander Lamson, Subhransu Maji, Tauhidur Rahman, and Deepak Ganesan. 2019. W! nce: Unobtrusive sensing of upper facial action units with eog-based eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–26.

[78] Wayne J Ryan, Andrew T Duchowski, and Stan T Birchfield. 2008. Limbus/pupil switching for wearable eye tracking under variable lighting conditions. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. 61–64.

[79] Jeffrey S Shell, Roel Vertegaal, and Alexander W Skaburskis. 2003. EyePliances: attention-seeking devices that respond to visual attention. In *CHI'03 extended abstracts on Human factors in computing systems*. 770–771.

[80] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. VSkin: Sensing Touch Gestures on Surfaces of Mobile Devices Using Acoustic Signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 591–605. https://doi.org/10.1145/3241539.3241568

[81] Sheng Sun, Jianyuan Wang, Menglun Zhang, Yi Yuan, Yuan Ning, Dong Ma, Pengfei Niu, Yi Gong, Xiaopeng Yang, and Wei Pang. 2021. Eye-tracking monitoring based on PMUT arrays. *Journal of Microelectromechanical Systems* 31, 1 (2021), 45–53.

[82] Xue Sun, Jie Xiong, Chao Feng, Wenwen Deng, Xudong Wei, Dingyi Fang, and Xiaojiang Chen. 2023. Earmonitor: In-ear Motion-resilient Acoustic Sensing Using Commodity Earphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–22.

[83] Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. (Mar 2016).

[84] Haoran Wan, Shuyu Shi, Wenyu Cao, Wei Wang, and Guihai Chen. 2021. RespTracker: Multi-user room-scale respiration tracking with commercial acoustic devices. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[85] Jianzhong Wang, Guangyue Zhang, and Jiadong Shi. 2015. Pupil and glint detection using wearable camera sensor and near-infrared LED array. *Sensors* 15, 12 (2015), 30126–30141.

[86] Lin Wang, Junbao Zhang, Yue Li, and Haoyu Wang. 2023. Audiowrite: a handwriting recognition system using acoustic signals. In *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 81–88.

[87] Lei Wang, Xiang Zhang, Yuanshuang Jiang, Yong Zhang, Chenren Xu, Ruiyang Gao, and Daqing Zhang. 2021. Watching Your Phone's Back: Gesture Recognition by Sensing Acoustical Structure-Borne Propagation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 82 (jun 2021), 26 pages. https://doi.org/10.1145/3463522

[88] Jing Xiao, Jun Qu, and Yuanqing Li. 2019. An electrooculogram-based interaction method and its music-on-demand application in a virtual reality environment. *IEEE Access* 7 (2019), 22059–22070.

[89] Wentao Xie, Huangxun Chen, Jing Wei, Jin Zhang, and Qian Zhang. 2024. RimSense: Enabling Touch-based Interaction on Eyeglass Rim Using Piezoelectric Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 191 (jan 2024), 24 pages. https://doi.org/10.1145/3631456

[90] Wentao Xie, Qian Zhang, and Jin Zhang. 2021. Acoustic-based upper facial action recognition for smart eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–28.

[91] Xiangyu Xu, Jiadi Yu, Yingying Chen, Qin Hua, Yanmin Zhu, Yi-Chao Chen, and Minglu Li. 2020. TouchPass: Towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–13.

[92] Kenji Yamagishi, Junichi Hori, and Michio Miyakawa. 2006. Development of EOG-based communication system controlled by eight-directional eye movements. In *2006 international conference of the IEEE engineering in medicine and biology society*. IEEE, 2574–2577.

[93] Hui-Shyong Yeo, Juyoung Lee, Woontack Woo, Hideki Koike, Aaron J Quigley, and Kai Kunze. 2021. JINSense: Repurposing Electrooculography Sensors on Smart Glass for Midair Gesture and Context Sensing. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[94] Huanpu Yin, Anfu Zhou, Guangyuan Su, Bo Chen, Liang Liu, and Huadong Ma. 2020. Learning to recognize handwriting input with acoustic features. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–26.

[95] Qiuyang Zeng, Fan Li, Zhiyuan Zhao, Youqi Li, and Yu Wang. 2024. AcouWrite: Acoustic-based Handwriting Recognition on Smartphones. *IEEE Transactions on Mobile Computing* (2024).

[96] Ruidong Zhang, Ke Li, Yihong Hao, Yufan Wang, Zhengnan Lai, François Guimbretière, and Cheng Zhang. 2023. EchoSpeech: Continuous Silent Speech Recognition on Minimally-obtrusive Eyewear Powered by Acoustic Sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[97] Shijia Zhang, Taiting Lu, Hao Zhou, Yilin Liu, Runze Liu, and Mahanth Gowda. 2023. I Am an Earphone and I Can Hear My User's Face: Facial Landmark Tracking Using Smart Earphones. *ACM Transactions on Internet of Things* 5, 1 (2023), 1–29.

[98] Yongzhao Zhang, Wei-Hsiang Huang, Chih-Yun Yang, Wen-Ping Wang, Yi-Chao Chen, Chuang-Wen You, Da-Yuan Huang, Guangtao Xue, and Jiadi Yu. 2020. Endophasia: Utilizing acoustic-based imaging for issuing contact-free silent speech commands. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–26.