Acoustic-based Upper Facial Action Recognition for Smart Eyewear

WENTAO XIE, The Hong Kong University of Science and Technology and Southern University of Science and Technology
QIAN ZHANG*, The Hong Kong University of Science and Technology
JIN ZHANG*, Southern University of Science and Technology

Smart eyewear (*e.g.*, AR glasses) is considered to be the next big breakthrough for wearable devices. The interaction of state-of-the-art smart eyewear mostly relies on the touchpad which is obtrusive and not user-friendly. In this work, we propose a novel acoustic-based upper facial action (UFA) recognition system that serves as a hands-free interaction mechanism for smart eyewear. The proposed system is a glass-mounted acoustic sensing system with several pairs of commercial speakers and microphones to sense UFAs. There are two main challenges in designing the system. The first challenge is that the system is in a severe multipath environment and the received signal could have large attenuation due to the frequency-selective fading which will degrade the system's performance. To overcome this challenge, we design an Orthogonal Frequency Division Multiplexing (OFDM)-based channel state information (CSI) estimation scheme that is able to measure the phase changes caused by a facial action while mitigating the frequency-selective fading. The second challenge is that because the skin deformation caused by a facial action is tiny, the received signal has very small variations. Thus, it is hard to derive useful information directly from the received signal. To resolve this challenge, we apply a time-frequency analysis to derive the time-frequency domain signal from the CSI. We show that the derived time-frequency domain signal contains distinct patterns for different UFAs. Furthermore, we design a Convolutional Neural Network (CNN) to extract high-level features from the time-frequency patterns and classify the features into six UFAs, namely, *cheek-raiser*, *brow-raiser*, *brow-lower*, *wink*, *blink* and *neutral*. We evaluate the performance of our system through experiments on data collected from 26 subjects. The experimental result shows that our system can recognize the six UFAs with an average F1-score of 0.92.

CCS Concepts: • **Human-centered computing** → **Gestural input**; **Mobile devices**; • **Computing methodologies** → *machine learning*.

Additional Key Words and Phrases: wearables, eyewear, acoustic sensing, facial actions, OFDM

ACM Reference Format:
Wentao Xie, Qian Zhang, and Jin Zhang. 2021. Acoustic-based Upper Facial Action Recognition for Smart Eyewear. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 41 (June 2021), 28 pages. https://doi.org/10.1145/3448105

1 INTRODUCTION

Recent years have witnessed rapid advances in smart wearables. Smart eyewear (*e.g.*, AR glasses) is considered to be the next big breakthrough for smart wearables because they put useful information right in front of our eyes,

*Corresponding authors.

Authors' addresses: Wentao Xie, wxieaj@cse.ust.hk, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, Southern University of Science and Technology, Shenzhen, China; Qian Zhang, qianzh@cse.ust.hk, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong; Jin Zhang, zhang.j4@sustech.edu.cn, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China.

2474-9567/2021/6-ART41 $15.00
https://doi.org/10.1145/3448105

which enables us to enjoy both the real world and the virtual world. However, the interaction of state-of-the-art smart eyewear mostly relies on touchpad or control pad technologies [9, 10, 13, 18, 50]. Unlike smartphones and smartwatches where touch-based interaction requires only finger tapping, to interact with a glass-mounted touchpad, a user needs to raise his/her arm to reach the touchpad. Thus, frequent interaction with smart eyewear may cause fatigue and is not friendly to the elderly and disabled users. Although most smart eyewear is equipped with a voice assistant (*e.g.*, Amazon Alexa [1]), there are situations when voice commands are restricted, such as during a meeting or in a library. Consequently, a new hands-free interaction system for smart eyewear is desirable.

Recent research has proposed that facial actions can be leveraged as a hands-free input modality [2, 22, 23, 41, 49]. The idea of using facial actions as an input mechanism for human-computer interaction is suitable for smart eyewear for the following three reasons. (i) Glass-mounted sensors can easily capture facial actions because eyewear is worn on the human face. (ii) There are more than 40 basic facial actions that one can perform [8]. We can enumerate a set of them as the vocabulary for interaction. (iii) Facial actions are performed with little effort for most people.

The question is: *how to sense facial actions unobtrusively and reliably on eyewear?* Since facial actions are caused by the contraction of certain facial muscles, facial Electromyography (fEMG) sensors and Electrooculography (EOG) sensors are capable of sensing facial actions [16] [7]. However, both fEMG sensors and EOG sensors are required to be put on the user's face. This is too intrusive for normal usage. Rostaminia *et al.* [41] proposes that a commercial EOG-enabled eyewear [32], which hides the electrodes on the nose bridge, can sense facial actions unobtrusively. Different from traditional EOG sensors that use gel-based electrodes to ensure the firm contact between the electrodes and the skin, this eyewear uses dry electrodes and the contact between the electrodes and the skin is ensured by the weight of the eyewear. However, this method is vulnerable to motions. Therefore, additional motion sensors are required to remove any motion artifacts [41]. Also, since EOG electrodes are susceptible to any conductivity changes brought about by the chemicals in sweat [6], sweat artifacts need to be removed as well.

A camera is a straightforward solution for facial action detection. However, there is a practical issue with eye-tracking cameras: the field-of-view (FoV) is usually restricted to the eye area [17]. Therefore, it is challenging to derive facial actions from images with such a shallow FoV. In [17], the authors have tried to infer facial actions only from eye images. The result shows a mean detection accuracy of 74% among five emotive expressions and a mean accuracy of 70% among ten facial action units. Although their work successfully demonstrates the opportunities to infer facial actions just from eye images, the reported performance is not reliable enough to support human-eyewear interaction.

Proximity sensors have been explored to detect facial expressions. Masai *et al.* [31] proposes to embed photo reflective sensors to a normal eyeglass to sense facial expressions. Their system can recognize eight facial expressions by measuring the different facial skin deformations when different expressions are performed. One problem with this system is that photo reflective sensors are highly sensitive to lighting conditions [41]. In [21], the authors mounted an FMCW radar on an eyeglass to detect eye blink. Apart from the fact that an FMCW radar is too large for an eyeglass, one facial action alone is not enough to achieve effective human-eyewear interaction.

Recent works have shown that acoustic signals can be leveraged to perform hand motion tracking and gesture recognition with high accuracy [5, 30, 34, 53, 54, 60]. The principle behind these acoustic sensing systems is described as follows. The acoustic signal transmitted by the speaker is reflected by the hand and received by the microphone. When the hand is moving, the propagation path in which the acoustic signal traverses is time-varying. By processing the received signal, the variation of the signal's propagation path can be derived and therefore the hand motion is obtained. Similarly, when performing different facial actions, the facial skin will have different deformation patterns. The deformation of the skin could also alter the propagation path of acoustic signals. Therefore, if an eyeglass is equipped with the above-mentioned acoustic sensing system, it is possible to

detect facial actions. Also, compared with other on-glass sensing solutions, an acoustic-based system is more suitable for the task of interaction. The reasons are stated as follows: (i) **Contactless**. Acoustic sensors do not need to have physical contact with the skin which is user-friendly. (ii) **Cost-efficient**. Speakers and microphones are cost-efficient as most of the state-of-the-art smart eyewear are already equipped with them. To enable the sensing ability, no additional sensing modules are needed, and rearrangement of the existing speakers and microphones would be the only overhead. (iii) **Noise-resilient**. Compared to photo reflective sensors, ambient light conditions do not affect acoustic sensors. Also, since it is a contact-free system, there are no motion or sweat artifacts as in EOG/EMG sensors. Although the environmental noise could harm the system, previous works have shown that the ambient noise can be effectively removed by proper signal design and filtering.

However, the existing acoustic sensing solutions cannot be directly applied to perform facial action recognition. This is mainly because the acoustic signal's path length changes caused by the deformation of the facial skin is minimal. It is challenging to compute such a short path length variation. Compared with the hand motions whose path length variations are usually decimeter-level, the path length changes caused by a facial action are usually around one centimeter (see Sec. 4.2). That is even less than the wavelength of the acoustic signal that a commercial speaker can send ($1.4cm$ for a $24kHz$ signal). Another challenge faced by the existing solutions is the frequency-selective fading effect [54]. Since the transceivers are close to the human face, the system is in a severe multipath environment and the arriving signals may add up destructively. Therefore, new designs are needed.

In this paper, we propose a novel acoustic-based upper facial action (UFA) recognition system for smart eyewear that can recognize six UFAs, that is, *cheek-raiser*, *brow-raiser*, *brow-lower*, *wink*, *blink* and *neutral*. Our system detects a UFA by identifying the characteristics of the distance variation patterns between the facial skin and the eyewear. We successfully overcome the above two challenges by the following observations and designs. (i) Although frequency-selective fading causes the attenuation of the received signal, we observe that the influence only exists on a limited range of the frequency bands. Thus, we design an Orthogonal Frequency Division Multiplexing (OFDM)-based channel state information (CSI) estimation scheme to measure the distance variation on multiple carrier frequencies. In this way, the attenuation on some frequencies can be averaged out by fusing all the CSI estimations. Besides, it is beneficial to use CSI to measure distance variation. This is because CSI measures the phase changes of a wireless signal and the phase information in a wireless signal is sensitive to small distance changes [60]. (ii) Although it is hard to derive facial action features directly from CSI because the CSI variations caused by a facial action is too little, we observe that there is a clear pattern when the time-domain CSI is transformed into the time-frequency domain. Therefore, our system applies a short-time Fourier transform (STFT) to the CSI to compute the Doppler frequency shift (DFS) profile. The DFS profiles of different UFAs show distinct patterns. We also observe that the DFS profiles of different facial actions have different spatial distributions. Thus, our system uses multiple pairs of speakers and microphones to capture the distinct spatial distribution of the DFS profile. (iii) To further extract features from the DFS profile, we design a Convolutional Neural Network (CNN). The CNN can extract a high-level representation from the DFS profile and classify it into six UFAs. An additional issue we need to consider is that the extracted DFS profile contains the user's information. That means the trained model can be difficult to generalize to new users. Therefore, we use a transfer learning technique to personalize the trained CNN to new users.

We highlight the contributions of this work as follows. First, we design a novel acoustic-based UFA recognition system for smart eyewear that can recognize six UFAs with high accuracy and robustness. To the best of our knowledge, our system is the first acoustic-based UFA recognition system. Secondly, we leverage an OFDM-based CSI estimation scheme to characterize a UFA while successfully overcoming the frequency-selective fading caused by the multipath effect. Thirdly, We implement our system on an eyeglass. Experiments with 26 subjects are conducted to evaluate the performance of the system. The results show that our system achieves high accuracy and robustness with a 0.92 F1-score on average.

The rest of this paper is organized as follows. Section 2 provides the background of the design of our system. Section 3 gives an overview of our system. Section 4 elaborates on the detailed design of the system and in Section 5, we evaluate the performance of our system with experiments. In Section 6, we discuss the limitations and future directions of our system. Section 7 summarizes the related work of this paper and Section 8 concludes the paper.

2 BACKGROUND

In this section, we provide the background of our system design. We first discuss the reason why we select the six target UFAs, that is, *cheek-raiser*, *brow-raiser*, *brow-lower*, *wink*, *blink* and *neutral*, as the interaction vocabulary for our system. Next, since we use an OFDM scheme in our system, we give a brief introduction to the OFDM technique.

2.1 The Target Facial Actions

There are three basic requirements for the selected facial actions. (i) **Easy-to-perform**. Since the purpose of our system is to provide a hands-free interaction mechanism for smart eyewear, it is of vital importance that the facial actions we select can be performed naturally and with little effort. (ii) **Detectable**. The second requirement is that the selected facial actions should be detectable by our system. As the acoustic sensors have limited sensing ability, the selected facial actions can neither be too weak to detect nor out of the sensing range. (iii) **Not frequently performed**. Another requirement is that the target facial actions cannot be the ones that are performed constantly in our daily lives. Otherwise, it is hard for the system to decide whether the user performs the facial action consciously or unconsciously. For example, mouth-open is not an ideal facial action because mouth-open is frequently performed when we talk.

The facial action units listed in the facial action coding system (FACS) [8] are the candidate facial actions. This is because every facial action unit in FACS is a unit of muscle movements that facial expressions can be destructed into. Hence, it only involves a small number of muscle movements such that it can be performed easily by a user. Moreover, we narrow down our choices to the UFA units in FACS to ensure they are detectable by our system since they only involve the movements of the upper facial muscles. Also, UFA units are less likely to be involved in daily activities than the other facial actions, say lower facial actions.

We have surveyed the related literature [14–17, 21, 22, 24, 41, 58] and we conclude that the most studied UFA units are: ***brow-raiser* (AU2)**, ***brow-lower* (AU4)**, ***upper lid-raiser* (AU5)**, *cheek-raiser* (AU6), *nose-wrinkler* (AU9), *eye-close* (AU43), ***squint* (AU44)**, *blink* (AU45) and ***wink* (AU46)** among which the highlighted ones are suitable for human-computer interaction, as reported in [22]. We conduct a preliminary experiment with 9 subjects to examine whether users can perform the above mentioned UFAs naturally. In this experiment, we ask the subjects to repeat the above nine facial actions 20 times and report whether they can easily perform the facial actions or not. The result shows that some subjects have difficulty in performing *nose-wrinkler* and *squint*, while some subjects report that they feel mild eye fatigue after frequently performing *upper lid-raiser*. Therefore, considering these factors, we select *cheek-raiser* (AU6), *brow-raiser* (AU2), *brow-lower* (AU4), *wink* (AU46), *blink* (AU45) and a *neutral* state as our target UFAs.

2.2 OFDM

OFDM is a data transmission scheme that is used in many modern wireless communication systems such as 4G/5G and Wi-Fi. OFDM splits the wide-band channel into many narrow-band orthogonal subcarriers and the transmitted data is modulated to the subcarriers. One of the biggest advantages of OFDM is its immunity to frequency-selective fading. Frequency-selective fading often exists in a multipath environment where the received signal partially cancels itself out. This happens because, in a multipath environment, the signal received by the

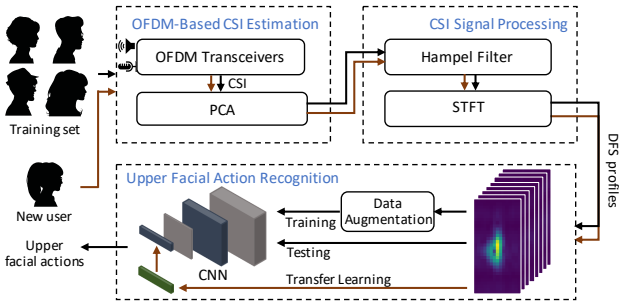


Fig. 1. The workflow of our system.

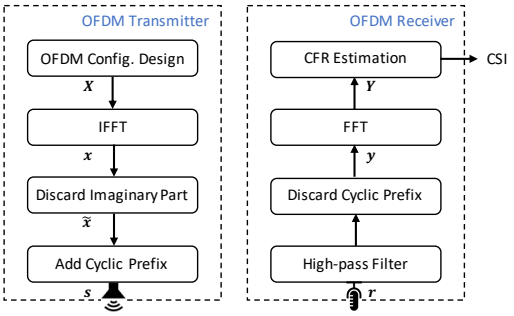


Fig. 2. OFDM transceivers.

receiver is the superposition of signals that propagate through different paths and these signals may add up destructively. The reason why OFDM signals are more resilient to frequency-selective fading than single-carrier systems is that since OFDM divides the overall channel into multiple narrow-band signals, only some subcarriers are affected by the selective fading. Thus, even in a frequency-selective environment, the data modulated by OFDM can still be decoded with good quality. See [43] for more extended discussions about OFDM.

3 DESIGN OVERVIEW

In this section, we give an overview of our design. We mount two speakers and four microphones on the frame of a normal eyeglass as shown in Fig. 13. As explained in Sec. 1, our system recognizes a UFA by identifying the CSI variation patterns caused by the skin deformation. To fulfill this objective, we design three modules for our system. (i) OFDM-based CSI estimation. This module uses an OFDM scheme to achieve a stable CSI estimation while mitigating the frequency selection effect. (ii) CSI signal processing. This module applies a time-frequency analysis to derive the CSI variation patterns in the time-frequency domain. (iii) UFA recognition. This module extracts UFA-related features from the time-frequency pattern using a CNN model and classifies the extracted features into six UFAs. The workflow of our system is shown in Fig. 1. The followings briefly describe the working process of each module.

(1) **OFDM-based CSI estimation**. The speakers transmit a specially designed OFDM signal. The microphones receive the reflected signals. The system first removes the environmental noise by a high-pass filter. Then, the system estimates CSI on multiple carrier frequencies using the received OFDM signals and the transmitted OFDM signals. Next, the estimated CSI with different carrier frequencies are fused into a single-channel CSI stream using principal component analysis (PCA). The detailed design of this module is presented in Sec. 4.1.
(2) **CSI signal processing**. In this module, the system first applies a Hampel filter to remove outliers from the CSI. Then, the system applies a short-time Fourier transform (STFT) to extract the CSI variation patterns in the time-frequency domain. The obtained time-frequency signal is the DFS profile. The details of this module are discussed in Sec. 4.2.
(3) **UFA recognition**. In this module, a CNN model is used to extract high-level features from the DFS profile and classify the features into UFAs. To train the CNN model, we apply several data augmentation techniques including time-shifting and energy-rearrangement to enhance the training data. To predict UFAs from new users, our system leverages transfer learning to personalize the trained CNN model to new users. The design details of this module are presented in Sec. 4.3.

Table 1. Parameters in OFDM-based CSI estimation module.

Symbol	Meaning	Value
N	The total number of subcarriers	128
N_1	The number of functional subcarriers	16
N_0	The number of nulled subcarriers	112
N_{CP}	The length of the cyclic prefix	20
N_{lag}	The transmission lag between the two speakers	296

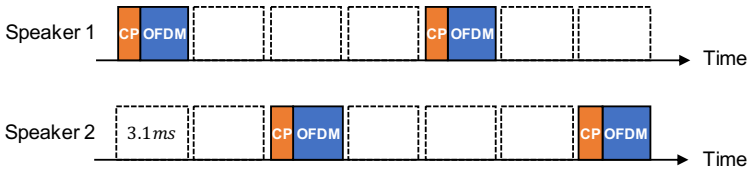


Fig. 3. The transmission scheme of two speakers.

4 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. 1.

4.1 OFDM-based CSI Estimation

In this module, our system estimates the CSI. We start the discussion by pointing out the challenge in designing this module. Since an eyeglass is placed close to the user's face, the face cannot be viewed as a single reflector. Thus, our system is in a multipath environment with the frequency-selective fading effect. To overcome this challenge, we design an OFDM-based CSI estimation scheme to combat the frequency-selective fading. This module works as follows. The speakers transmit a specially designed OFDM signal and the microphones receive the reflected signal. Then, the system estimates the CSI by computing the channel frequency response (CFR) using the received OFDM signal and the transmitted OFDM signal. Next, the system fuses the multi-channel CSI streams into one stable CSI stream using principal component analysis (PCA) to average out the frequency-selective fading. In the following subsections, we discuss the design of each component in this module. The system's parameters of this module are summarized in Tab. 1.

4.1.1 OFDM Transmitter. There are several issues we need to consider in designing the OFDM signal. (i) On one hand, the number of subcarriers in the OFDM signal should be large enough. This is because we rely on these subcarriers to provide multiple channels of CSI estimation to average out the attenuation caused by the frequency-selective fading. The more subcarriers we have in the OFDM signal, the more reliable the CSI estimation can achieve. (ii) On the other hand, too many subcarriers can negatively affect the system. Since more subcarriers lead to a longer OFDM signal, if an overlong OFDM signal is used, it takes more time to transmit an OFDM frame. In this way, the time resolution of the estimated CSI could be too low to capture the high-frequency features of a UFA. Besides, an excessive number of subcarriers will bring large overhead to the system because, as we discuss later in this section, fast Fourier transform (FFT) and PCA will be applied with more data points. (iii) Since our system aims at providing a non-invasive interaction experience for smart eyewear users, the transmitted signal should be inaudible to humans. In other words, the designed OFDM signal should work at a frequency band above $18kHz$. The following presents our design.

The sampling frequency of common audio devices is up to $48kHz$. Inspired by [34], we split the entire $48kHz$ frequency band into 128 subcarriers, each of which spans $375Hz$. The total number of subcarriers is denoted by N. We randomly set the subcarriers with frequency from 18 to $24kHz$ to +1 or −1. We call these subcarriers *functional subcarriers* and the number of functional subcarriers is denoted by N_1. We set the rest of the subcarriers to zero and call them *nulled subcarriers*. The number of nulled subcarriers is denoted by N_0. Then, $N = 128$, $N_1 = 16$ and $N_0 = 112$. We believe this configuration resolves the above three issues as follows. (i) There are 16 functional subcarriers that can be leveraged to estimate CSI. Our evaluation shows that 16 subcarriers can effectively average out the frequency-selective fading (Sec. 5.2.2). (ii) As discussed later in this section, the time resolution of the estimated CSI under this configuration is $12.4ms$ ($80Hz$) which is enough to preserve the high-frequency characteristics of a UFA since, normally, a UFA lasts for 1 to 2 seconds. (iii) Our evaluation in Sec. 5.2.1 shows that it takes our system $87.1ms$ to process a $1.5s$ segment of CSI signal. Considering the transmission frame rate is $80Hz$, processing a frame of OFDM signal only takes $0.73ms$ which is an acceptable delay since an OFDM frame is transmitted every $12.4ms$. (iv) The operational carrier frequencies of the designed signal are all above $18kHz$, thus it is inaudible to humans. Next, we continue to discuss how we generate the OFDM signal under this configuration.

Let $(\cdot)^{null}$ and $(\cdot)^{func}$ denote the nulled and functional carrier frequencies in our OFDM system. Let X_k ($k = 0$ to $N-1$) be the sequence of symbols of all the N subcarriers. Then, X_k^{func} denotes the functional subcarriers and X_k^{null} denotes the nulled subcarriers. First of all, an inverse fast Fourier transform (IFFT) is applied to X_k to compute the time domain signal x_n ($n = 0$ to $N-1$). Next, we keep the real part of x_n and discard the imaginary part because a speaker can only transmit real signals. The resulting signal is denoted as $\widetilde{x}_n$. Finally, the last N_{CP} samples of $\widetilde{x}_n$ are appended in the front as a cyclic prefix (CP). The resulting signal is a frame of OFDM signal denoted by s_n. Note that we set N_{CP} to 20 samples because a 20-sample CP is enough to cover the coherence time since the eyewear is worn only a few centimeters away from the human face. Let N_{frame} be the length of s_n. Then $N_{frame} = N + N_{CP}$ which is 148 samples that span $3.1ms$. The two speakers in our system concurrently transmit s_n in time division multiplexing (TDM) with a N_{lag}-sample lag where $N_{lag} = 2N_{frame}$. With the above settings, a speaker transmits an OFDM frame every $12.4ms$ ($80Hz$). The TDM scheme of the two speakers is shown in Fig. 3. Note that N_{lag} can be as short as N_{frame} because the CP ensures there is no inter-frame-interference. We manually increase the lag just to maintain a moderate transmission rate. The transmitter's pipeline is shown in Fig. 2.

4.1.2 OFDM Receiver. Let r_n denote a frame of the received signal. We first apply a Butterworth high-pass filter to r_n to remove the environmental noise and we discard the first N_{CP} samples to remove the CP. Let y_n be the resulting signal. We first apply an FFT to y_n to derive the frequency domain signal, *i.e.*, Y_k. Next, we compute the CFR of the functional subcarriers as follow:

$$H_k^{func} = 2 \cdot \frac{Y_k^{func}}{X_k^{func}}, \tag{1}$$

where k is from 0 to $N_1 - 1$. Then, by continuously collecting CFRs, our system obtains the CSI signal, *i.e.*, $H_k^{func}(t)$. The processing pipeline of the receiver is shown in Fig. 2. In the remaining if this subsection, we first proof the correctness of Eqn. 1, then we briefly discuss why we can infer UFAs from the CSI signal. Note that the correctness of Eqn. 1 can be derived by some basic Fourier transform properties. We provide this proof just for the reader's reference.

As has been discussed in the previous subsection, we first apply an IFFT on the designed symbol X_k to derive the time-domain signal x_n, *i.e.*, $x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi/Nkn}$. Next, the system discards the imaginary part of x_n, *i.e.*, $\widetilde{x}_n = \Re\{x_n\}$ where $\Re(\cdot)$ represents the real part of a signal. Equivalently, $\widetilde{x}_n = \frac{1}{N}\sum_{k=0}^{N-1} \Re\{X_k e^{j2\pi/Nkn}\}$. Since

X_k is a real number, $\Re\{X_k e^{j2\pi/Nkn}\} = X_k \Re\{e^{j2\pi/Nkn}\}$. Then, we have

$$\widetilde{x}_n = \frac{1}{2N} \sum_{k=0}^{N-1} X_k (e^{j\frac{2\pi}{N}kn} + e^{-j\frac{2\pi}{N}kn}). \tag{2}$$

Next, $\widetilde{x}_n$ is sent by the speaker and received by the microphone. The received signal y_n can be modeled as $y_n = \widetilde{x}_n * h_n$, where h_n is the channel impulse response. In the frequency domain, we have $Y_k = \widetilde{X}_k \cdot H_k$ where $\widetilde{X}_k$ is the Fourier transform of $\widetilde{x}_n$ and H_k is the CFR. Therefore, we can compute the CFR as

$$H_k = \frac{Y_k}{\widetilde{X}_k}. \tag{3}$$

That is to say, to derive the CFR, we need to compute $\widetilde{X}_k$. By definition, $\widetilde{X}_k = \sum_{n=0}^{N-1} \widetilde{x}_n e^{-j2\pi/Nkn}$. Substitute $\widetilde{x}_n$ in this equation with Eqn. 2, we have

$$\widetilde{X}_k = \frac{1}{2N} \sum_{n=0}^{N-1} \sum_{r=0}^{N-1} X_r (e^{j\frac{2\pi}{N}rn} + e^{-j\frac{2\pi}{N}rn}) e^{-j\frac{2\pi}{N}kn}. \tag{4}$$

After swamping the order of the two summations, we obtain

$$\widetilde{X}_k = \frac{1}{2N} \sum_{r=0}^{N-1} X_r \left(\sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(r-k)n} + \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(-r-k)n} \right). \tag{5}$$

Since $\{e^{j\frac{2\pi}{N}kn} | n = 0, 1, ..., N-1\}$ forms an orthogonal basis over N-dimension complex space for any $k \neq 0$, Eqn. 5 can be simplified as

$$\widetilde{X}_k = \frac{1}{2} \sum_{r=0}^{N-1} X_r \left(\delta_{r,k} + \delta_{-r,k} \right), \tag{6}$$

where $\delta_{p,q}$ is the Kronecker delta. Since both r and k are non-negative integers, $\delta_{-r,k} = 0$ for all r and k. Then,

$$\widetilde{X}_k = \frac{1}{2} X_k. \tag{7}$$

Therefore, by plugging Eqn. 7 into Eqn. 3 and selecting only the functional subcarriers, Eqn. 1 is derived.

Next, we discuss why this CSI model can infer UFAs. Readers can refer to [48] for more extended discussion about CSI in wireless communication systems. The received signal y_n can be modeled as the superposition of the transmitted signal $\widetilde{x}_n$ with different delays and attenuation. *i.e.*,

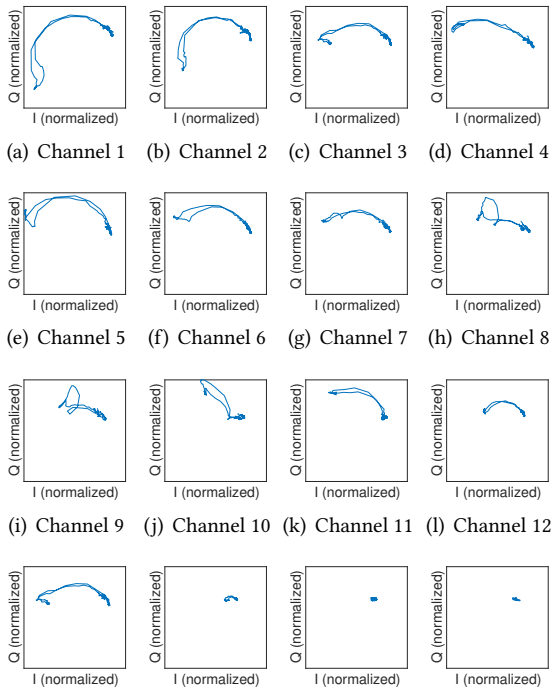
$$y_n = \widetilde{x}_n * h_n = \sum_{i=1}^{L} \alpha_i \widetilde{x}_{n-m_i}, \tag{8}$$

where L is the total number of propagation paths, α_i and m_i are the attenuation and the delayed samples of the ith path. According to the linear property and the time-shifting property of discrete Fourier transform [36], we have

$$Y_k = \sum_{i=1}^{L} \alpha_i \widetilde{X}_k e^{-j\frac{2\pi}{N}km_i}. \tag{9}$$

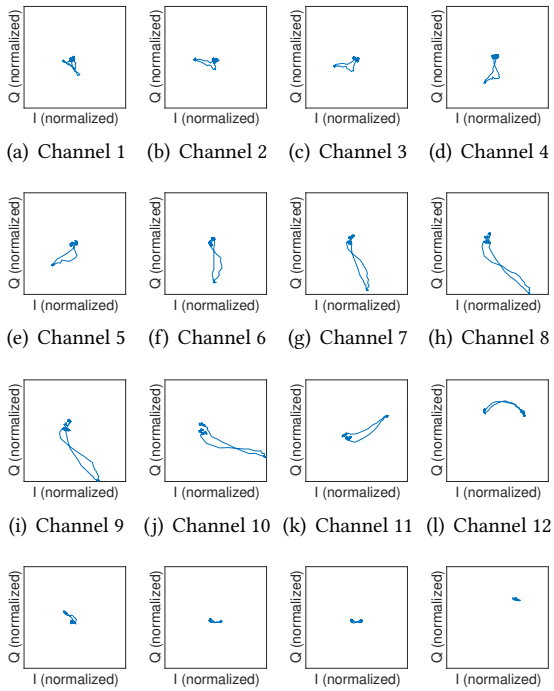
Since we use Eqn. 3 to compute the CFR and we discard the nulled frequencies, we have $H_k^{func} = \sum_{i=1}^{L} \alpha_i e^{-j\frac{2\pi}{N}km_i}$. Consequently, the derived CSI signal can be interpreted as

$$H_k^{func}(t) = \sum_{i=1}^{L} \alpha_i(t) e^{-j\frac{2\pi}{N}km_i(t)}. \tag{10}$$



(a) Channel 1 (b) Channel 2 (c) Channel 3 (d) Channel 4
(e) Channel 5 (f) Channel 6 (g) Channel 7 (h) Channel 8
(i) Channel 9 (j) Channel 10 (k) Channel 11 (l) Channel 12
(m) Channel 13 (n) Channel 14 (o) Channel 15 (p) Channel 16

Fig. 4. A 1.5*s* CSI segment of a subject performing a *brow-raiser*.

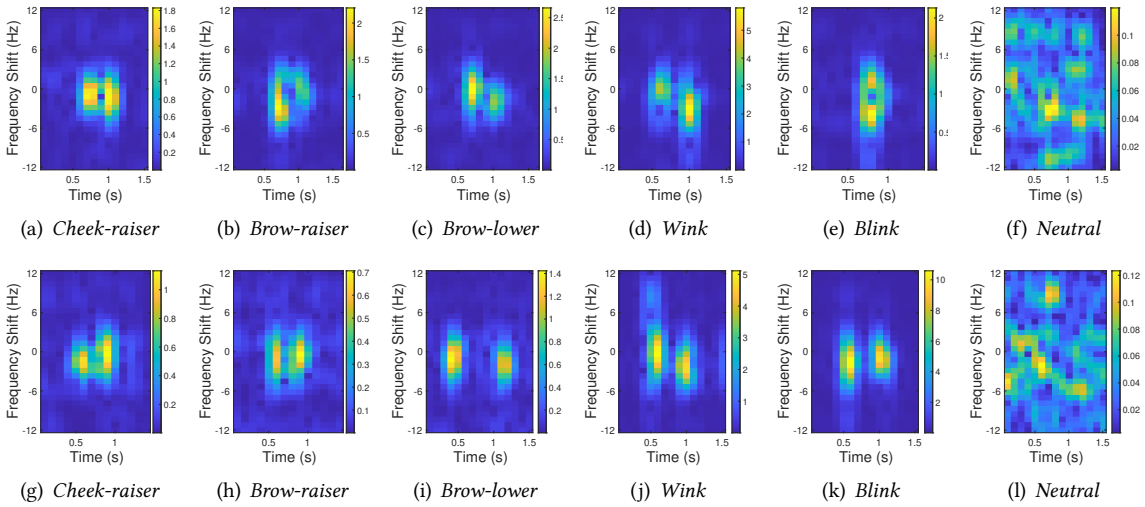


(a) Channel 1 (b) Channel 2 (c) Channel 3 (d) Channel 4
(e) Channel 5 (f) Channel 6 (g) Channel 7 (h) Channel 8
(i) Channel 9 (j) Channel 10 (k) Channel 11 (l) Channel 12
(m) Channel 13 (n) Channel 14 (o) Channel 15 (p) Channel 16

Fig. 5. A 1.5*s* CSI segment of a subject (different from that of Fig. 4) performing a *brow-raiser*.

Eqn. 10 indicates that the amplitude of CSI is related to the attenuation of the received signal which implies the reflective property of the surrounding reflectors and the phase of CSI indicates the delay of the received signal which implies the distance of the reflectors. Hence, by analyzing the variation patterns of the CSI signal, we can infer the motion patterns of the facial skin and further predict the performed UFAs. Examples of the estimated CSI are shown in Fig. 4 and Fig. 5.

4.1.3 CSI Fusion. The raw CSI suffers from the frequency-selective fading. Moreover, the frequency-selection effect is hard to predict because it is affected by many factors such as the user. For example, Fig. 4 and Fig. 5 show the 16-channel CSI of two users performing a *brow-raiser*. The data collection process of these two examples is described in Sec. 5.1. In Fig. 4, channels 9-12 suffer more severe attenuation than channels 1-4 while in Fig. 5, channels 1-4 suffer more serious attenuation than channels 9-12. Note that channels 14-16 have large attenuation in both examples. This is because the working frequency of channels 14-16 is close to the Nyquist frequency. Thus, they suffer from hardware-constrained fading. Since the frequency-selective fading only causes attenuation on a limited number of channels, by fusing the 16-channel CSI into one channel, we are able to average out the attenuation on some channels. To this end, the system applies a principal component analysis (PCA) to compress the 16-channel signals into one channel. Each of the 16 channels is input as an observation of the PCA algorithm. In PCA, a linear transformation is performed to transform all of the observations into a new orthogonal coordinate system. These new coordinates are called principal components (PCs) and there is little mutual information among different PCs. The PC with the highest eigenvalue carries the most significant information, that is, it



(a) *Cheek-raiser* (b) *Brow-raiser* (c) *Brow-lower* (d) *Wink* (e) *Blink* (f) *Neutral*

(g) *Cheek-raiser* (h) *Brow-raiser* (i) *Brow-lower* (j) *Wink* (k) *Blink* (l) *Neutral*

Fig. 6. The DFS profiles when different UFAs are performed. (a)-(f) and (g)-(l) are from two different subjects. All data are collected from the speaker 1 - microphone 1 link (see Fig. 13).

Table 2. Parameters in CSI signal processing module.

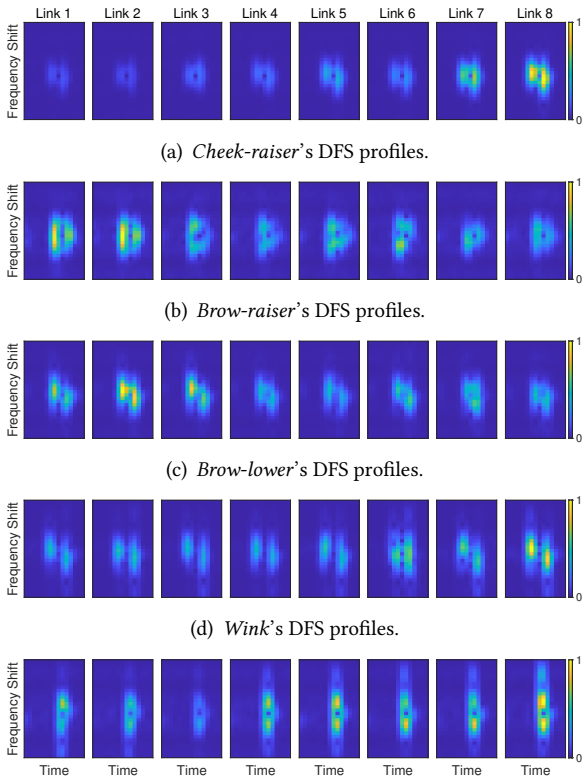
Symbol	Meaning	Value
T_{detect}	The detection window length	$1.5s$
N_{win}	The FFT window size	32
N_{fft}	The number of FFT points	64
ρ	The overlapping size of adjacent FFT windows	75%

summarizes the 16 channels of CSI. Therefore, we take the PC with the largest eigenvalue as the output CSI signal of this module.

4.2 CSI Signal Processing

After the CSI is derived, our system extracts the CSI variation patterns in order to characterize a UFA. However, as shown in Fig. 4 and Fig. 5, the signal path length changes caused by a UFA is so small that it is even less than the wavelength of the transmitted signal. Thus, the main challenge of this module arises: how to extract UFA patterns from such a tiny CSI variation. To tackle this challenge, instead of processing on the time-domain CSI signal, we extract features from the time-frequency domain. This module works in the following two steps. The system first passes the CSI stream to a Hampel filter to filter out the outliers caused by noise. Then, the system applies a time-frequency analysis to extract time-frequency patterns from the CSI signal. The following discusses the time-frequency analysis technique used in our system.

We use short-time Fourier transform (STFT) to extract time-frequency features from CSI because recent works [27, 29, 42, 59, 63] have shown that STFT has a satisfactory performance in analyzing acoustic signals. An STFT divides the input signal into short segments of the same length and computes the FFT coefficient separately on each segment. Thus, there is a trade-off between the time resolution and frequency resolution. Therefore,



(a) *Cheek-raiser*'s DFS profiles.

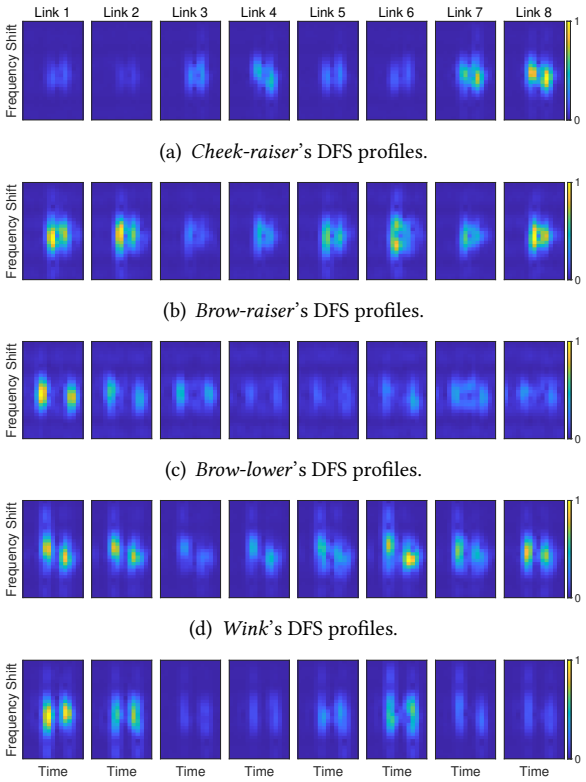
(b) *Brow-raiser*'s DFS profiles.

(c) *Brow-lower*'s DFS profiles.

(d) *Wink*'s DFS profiles.

(e) *Blink*'s DFS profiles.

Fig. 7. The eight-link DFS profiles of a subject performing different UFAs.



(a) *Cheek-raiser*'s DFS profiles.

(b) *Brow-raiser*'s DFS profiles.

(c) *Brow-lower*'s DFS profiles.

(d) *Wink*'s DFS profiles.

(e) *Blink*'s DFS profiles.

Fig. 8. The eight-link DFS profiles of a subject (different from that of Fig. 7) performing different UFAs.

carefully selecting the STFT parameters is of vital importance. Next, we discuss the STFT in our system. The parameters used in our STFT are summarized in Tab. 2.

First of all, we set the detection window to be 1.5s. That is to say, we apply STFT on 1.5s of the CSI signal and later we infer a UFA using the extracted time-frequency features within this 1.5s interval. The window size is chosen as such because our collected data shows that most of the UFAs span less than 1.5s. Next, we discuss the other STFT parameters. Considering the frame rate of the CSI signal is 80Hz and a UFA typically lasts one to two seconds, we select the FFT window size to 32 samples which take around 0.4s of the CSI signal. We believe this FFT window size is short enough to capture the temporal features of a UFA. We apply a 64-point FFT with a Gaussian window to the 32-sample segment. Note that the FFT is applied with 64 points, instead of 32, to increase the frequency resolution. The overlap size between adjacent FFT windows is set to be 75% for a higher time resolution. The resulting time-frequency signal is a DFS profile with 64 frequency bins and 15 time bins. The frequency resolution of the derived DFS profile is 0.79Hz and the time resolution is 0.1s. We eliminate the static frequency components by subtracting a DFS profile from its previous one. We ignore the frequency components out of $\pm 12Hz$ in the DFS profile because we find that the frequency shift caused by our target UFAs is mostly within this range. The size of the resulting DFS profile is 30×15.

Fig. 6 shows examples of the DFS profile when different UFAs are performed. It is clear that different UFAs show different patterns in the time-frequency domain. Besides, since our system has two speakers and four microphones (eight microphone-speaker links) placing at different locations on the eyewear, we can measure the DFS profile from different observation angles. This is beneficial because the same UFA shows different DFS patterns when observed from different angles. This is because a UFA is the result of the deformation of different parts of the facial skin. Thus, the multiple links in our system can capture the spatial distribution patterns of the DFS profiles. Figs. 7 and 8 present the eight-link DFS profiles of two subjects performing different UFAs. The data collection process of these data is the same as described in Sec. 5.1. We observe that some UFAs have distinct energy distributions among the eight links. For example, a *cheek-raiser* shows high energy in the last two links (Fig. 7(a) and Fig. 8(a)). This is because a *cheek-raiser* mostly involves the contraction of the muscles under the eye. Since links 7 and 8 (formed by speaker 2 with microphones 3 and 4 as shown in Fig. 13) are close to the skin deformation area, these two links show high energy. Similarly, a *brow-raiser* (Fig. 7(b) and Fig. 8(b)) and a *brow-lower* (Fig. 7(c) and Fig. 8(c)) show high energy in the first two links. This is because links 1 and 2 (formed by speaker 1 with microphones 1 and 2) are close to the eyebrow. We also observe that the DFS profile shows different characteristics when the same UFA is performed by different subjects. For example, the subject of Fig. 8 tends to perform *brow-lower* and *blink* slower than the subject of Fig. 7, and the energy distributions of *wink* and *blink* among the eight links between the two subjects are hugely different. This happens because different people have different face shapes, plus the way the same facial action is performed is different among people (*e.g.*, speed, muscle contraction level, etc.). Thus, how to recognize UFAs based on the DFS profiles with personal characteristics is challenging. Next, we present our design to tackle this challenge.

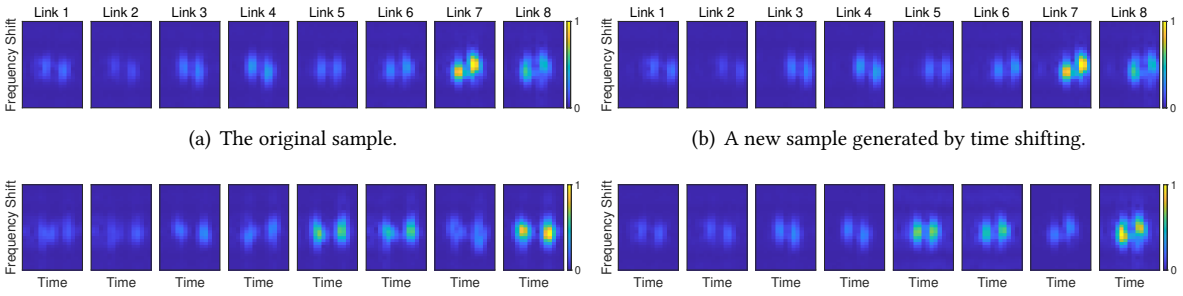
4.3 Upper Facial Action Recognition

After the DFS profile is derived, we design a deep learning model to extract high-level features from the DFS profiles and classify them into six UFAs. There are two challenges in designing this module. First, since the derived DFS profiles contain subject-related information, the trained model is easily correlated with the training data. Therefore, training a model that can generalize to new users is challenging. Secondly, training a neural network requires a huge amount of training data. However, since we collect data manually, the amount of training data is insufficient and how to make full use of the collected data is of vital importance. To tackle the first challenge, we leverage several techniques to alleviate overfitting when training the model and we use transfer learning to personalize the trained model to new users. To resolve the second issue, we use several data augmentation techniques to enhance the dataset.

4.3.1 Data Augmentation. We apply two techniques to augment our dataset. (i) **Time shifting**. As explained in Sec. 4.2, we set our detection window to 1.5s. However, a UFA normally takes less than 1.5s. Thus, we manually shift a DFS profile along the time axis until the main component (the high energy part as shown in Fig. 6) is about to reach the two ends of the window. (ii) **Energy rearrangement**. This technique is based on an observation that the energy distribution of the DFS profiles among the eight links varies depending on real-world factors such as the user and the position in which the eyewear is worn on the user's face. Thus, we generate a new virtual sample from a real sample (we call it an original sample) by assigning the energy distribution of another randomly selected sample (we call this sample a template sample) from the same class to the original sample, *i.e.*,

$$P_{new} = \beta P_{orig} + (1 - \beta) P_{temp},$$

where P_{new}, P_{orig} and P_{temp} are the energy distribution of the DFS profiles of the newly generated sample, the original sample, and the template sample, β is the preserving factor. Examples of newly generated samples are shown in Fig. 9.



(a) The original sample.

(b) A new sample generated by time shifting.

(c) A template sample.

(d) A new sample generated by energy rearrangement.

Fig. 9. Examples of data augmentation. (a) The original sample. (b) A new sample generated by time shifting. (c) A template sample. (d) A new sample generated from the original sample by energy rearrangement that combines the energy distribution of the template sample.

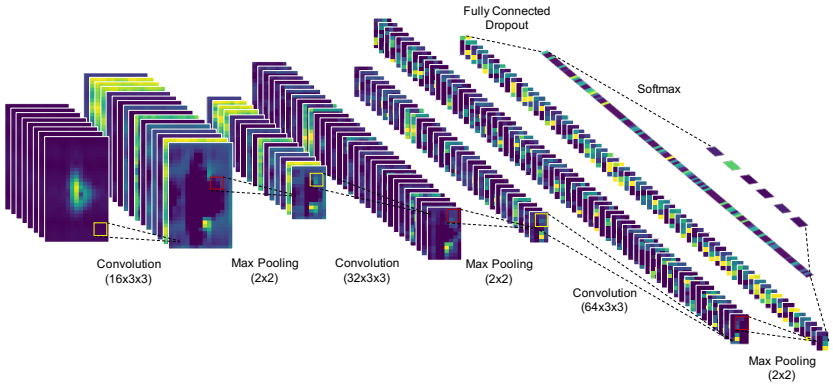


Fig. 10. The architecture of our CNN.

Table 3. CNN model specs.

Parameter	Default
Input size	$8 \times 30 \times 15$
Conv. kernel size 1.	$16 \times 3 \times 3$
Conv. kernel size 2	$32 \times 3 \times 3$
Conv. kernel size 3	$64 \times 3 \times 3$
Pooling kernel size	2×2
FC layer size	128
Output size	6
Learning rate	0.0001
Dropout rate	0.5
α	0.1
λ	0.01

4.3.2 Deep Learning Model Design. We design a Convolutional Neural Network (CNN) to predict the UFAs. Note that we choose to use CNN because previous works have shown that CNN has good performance in extracting features from image-like data. We also compare the performance of CNN with other deep learning and machine learning models in Sec. 5.2.8. According to our experiments, CNN achieves the best performance. Our CNN takes the 8-channel DFS profiles as the input. Followed by the input layer, we apply three convolutional layers. These three convolutional layers are used to extract deep features from the input data. The kernel size of each convolutional layer is 3×3 which is a widely used kernel size in image classification. The numbers of kernels of the three convolutional layers are 16, 32, and 64, respectively. These parameters are chosen empirically. We apply batch normalization, ReLU activation, and a max-pooling layer after each convolutional layer. We use a ReLU activated fully connected (FC) layer with 128 units to summarize the features extracted by the previous convolutional layers and we use a Softmax layer to classify the features into the six UFAs. The size of the FC layer is chosen experimentally. The architecture of our CNN is shown in Fig. 10.

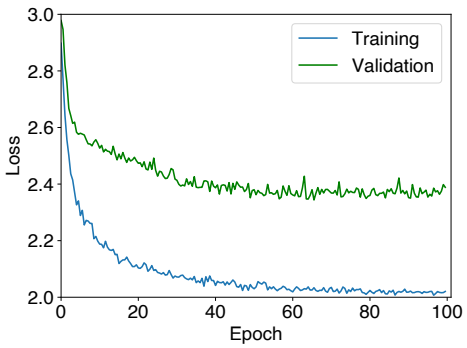


Fig. 11. Loss changes while training.

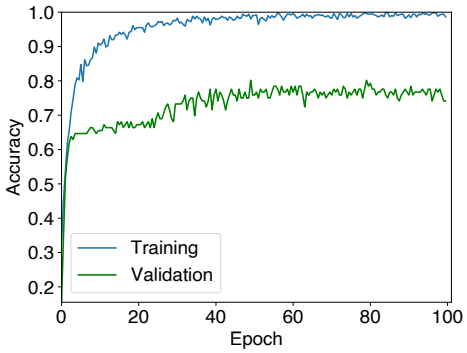


Fig. 12. Accuracy changes while training.

Since the manually collected data is insufficient, it is hard to train a model that can generalize to new users. Therefore, we use the collected data to train a base model and personalize the base model for new users (see Sec. 4.3.3). Even so, we still want the trained CNN to mainly focus on the facial action-related features rather than the subject-related features such that the base model requires only a few samples to adapt to a new subject [12]. To this end, we apply three techniques to ensure the generalizability of the trained base model. (i) **Confidence control constraint**. Proposed in [20], confidence control constraint is a penalty function that penalizes the loss function when the model classifies a training sample into a class with overlarge confidence. The rationale behind this design is that placing all the probability into one class while training is a symptom of overfitting [46]. If that happens, it is likely that the model has co-adapted to the training data. (ii) **Dropout**. The Dropout technique is used to prevent the co-adaptation of the model to the training data. We use the dropout technique with a probability of 50% after the fully connected layer. (iii) **Weight decay**. Weight decay is an L2 regularization technique that penalizes the network parameters with overlarge values. By using weight decay, the model is forced to learn the most significant information from the training data. The final loss function is:

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} log(P(y_i|\mathbf{x_i}, \mathbf{w})) - \alpha\frac{1}{N}\sum_{i=1}^{N}(log(P(y_i|\mathbf{x_i}, \mathbf{w})) + log(1 - P(y_i|\mathbf{x_i}, \mathbf{w}))) + \lambda||\mathbf{w}||_2^2, \tag{11}$$

where the first term is the cross-entropy loss, the second term is the confidence control constraint, the third term is the L2 regularization term, N is the batch size, $\mathbf{x_i}$ is the ith training sample, y_i is the label of the training sample, $\mathbf{w}$ is the network parameters, α is the confidence constraint coefficient and λ is the weight decay factor. We set α to 0.1 and λ to 0.01 in our implementation. Note that the dropout rate, weight decay factor we use are standard values, and the confidence control constraint is chosen experimentally. An example of the model training process is shown in Fig. 11 and Fig. 12. This example shows the loss function and accuracy changes during the training process. We use leave-one-subject-out validation on a subject (subject 7 in Fig. 14(b)) to show this example.

We use an Adam optimizer to train the CNN model. The other network hyper-parameters such as batch size, learning rate, etc are chosen empirically. The specifications of the CNN is summarized in Tab. 3. Our CNN model is implemented with Pytorch [38].

4.3.3 Personalization. Since the face shape and the way that a facial action is performed vary among people, it is hard to train a network that generalizes to every user. In our system, we train a base model with the collected data and adapt the base model to new users using transfer learning [37]. To perform transfer learning, we first obtain a small number of labeled samples of each UFA from the new user. Then, we freeze the network parameters

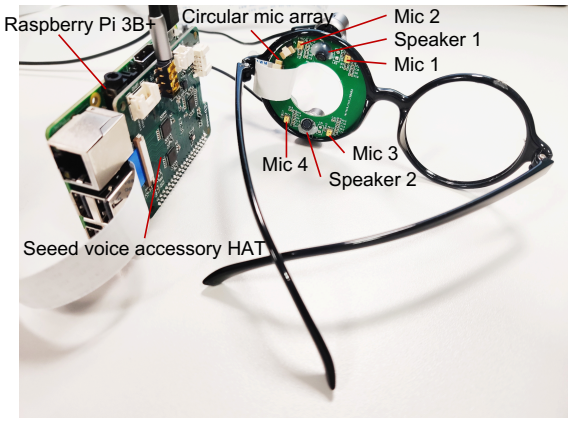


Fig. 13. Our prototype.

except for the last layer (the Softmax layer) and retrain the last layer using the obtained data. In this way, the model is adapted to the new user. Specifically, we retrain the network for 500 epochs with four labeled samples of each UFA. During training, we use the same learning rate and we set $\lambda = 0.1$, $\alpha = 0$. Note that these parameters are also chosen empirically.

5 EVALUATION

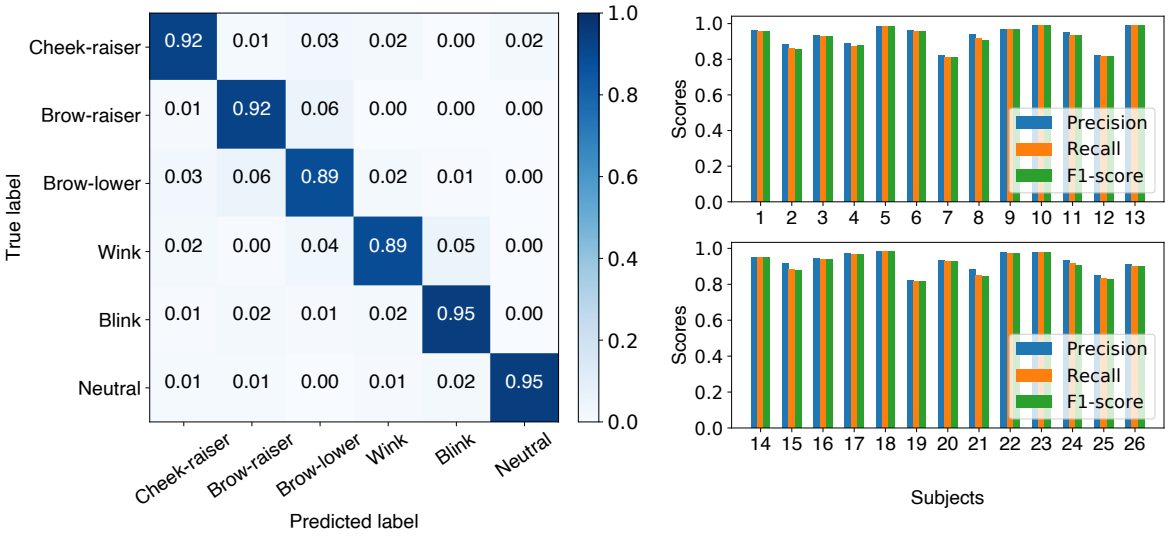
To demonstrate the feasibility of our system, we design the following experiments to evaluate the system.

5.1 Experimental Setting

We implement our system on a normal eyeglass with a Raspberry Pi 3 Model B+ development board [39], a Seeed ReSpeaker 4-Mic Linear Array Kit [44] and a pair of Mi In-Ear Headphones Pro [55]. Note that the original microphone array is linear, we customize it to a circular array to fit in the eyeglass frame. Fig. 13 shows the prototype. Note that in commercial eyeglasses, the speakers and microphones are usually not placed this way [2, 22, 23, 41, 49]. To enable the sensing ability of commercial eyewear, rearranging the existing speakers and microphones is required. In the current implementation, we only focus on detecting UFAs on the left face. Apart from Raspberry Pi 3 Model B+, we also run our system on a Raspberry Pi 4 Model B [40] with a 4GB memory and a high-end server with an Intel Xeon Platinum 8000 8-core processor and a 32GB memory for performance comparison. Note that in the current implementation, our system does not support real-time inference. Instead, we store the recorded data into audio files on the hard disk and our system reads audio data from these files for further processing. However, as discussed in Secs. 4.1.1 and 5.2.1, the processing pipeline of our system only causes mild delay. Therefore, we believe supporting real-time is an engineering problem and we don't include it in this work.

We select six UFAs from the Facial Action Coding System (FACS) [8] as our facial action set, that is *cheek-raiser* (AU6), *brow-raiser* (AU2), *brow-lower* (AU4), *wink* (AU46), *blink* (AU45) and a *neutral* class where no facial action is performed. We recruit 26 volunteers (15 males and 11 females with age from 20 to 29) to participate in our experiments[1]. Before each experiment, we show a visual guide book of the FACS [11] to help the volunteer get familiar with the six UFAs and we give each volunteer 2-5 minutes to practice the facial actions. In each experiment, the volunteer performs the above six UFAs while wearing our eyeglasses. There are two data collection sessions

[1]Experiments were conducted following the ethical policies of our institutions.



(a) Confusion matrix.

(b) Performance on each subject.

Fig. 14. Overall classification result.

Table 4. The average execution time for a UFA inference. (CFR: CFR estimation. R Pi: Raspberry Pi)

	OFDM-based CSI estimation				CSI signal processing	UFA recognition	Total
R Pi 3 Model B+	87.1*ms*				3.9*ms*	56.3*ms*	147.3*ms*
	FFT	CFR	PCA	Others			
	42.1*ms*	16.2*ms*	8.8*ms*	20.0*ms*			
R Pi 4 Model B	9.8*ms*				0.7*ms*	50.2*ms*	60.7*ms*
	FFT	CFR	PCA	Others			
	5.1*ms*	1.9*ms*	1.4*ms*	1.4*ms*			
Server	11.4*ms*				0.4*ms*	1.4*ms*	13.2*ms*
	FFT	CFR	PCA	Others			
	5.7*ms*	2.4*ms*	0.7*ms*	2.6*ms*			

for each subject. In each session, a program instructs the subject to perform each facial action 20 times. The program iteratively generates the label of a UFA and the start time and ending time of the performed UFA are input by the subject by pressing a key on the keyboard. Between two data collection sessions, the volunteer takes off the eyeglass and has a 2-5 minutes break. Unless otherwise noted, all the experiments are conducted in a quiet lab room and the subjects sit still during the data collection sessions.

In total, we collect around 6000 samples of the six UFAs. With the data augmentation techniques mentioned in Sec. 4.3.1, the final dataset has around 27000 samples. In the following evaluations, we use the data from the second data collection session to test the performance of our system. The data from the first session is used for personalization.

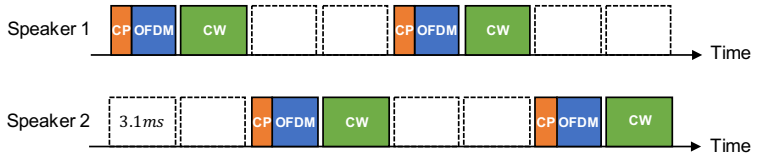


Fig. 15. The signal transmission scheme in Sec. 5.2.2.

5.2 Experimental Result

5.2.1 Overall Performance. In this section, we present the overall performance of our system. We conduct leave-one-subject-out validation on each subject. Fig. 14(a) shows the confusion matrix and Fig. 14(b) shows the performance on each subject. The average F1-score is 0.92 while the worst performance is from the 7th subject with 0.81 F1-score. Overall, the result shows that our system is able to achieve UFA recognition with high accuracy and it is robust to different users. We also evaluate the execution time of our system. We run 500 times of UFA inference and compute the average running time for each component of our system. The result is shown in Tab. 4. This result indicates that the time it takes for our system to predict a UFA is less than 150ms and this delay can be further reduced with more powerful devices.

5.2.2 Benefits of OFDM. In this section, we evaluate the benefits of the OFDM scheme in our system. As discussed in Sec. 4.1, our system uses an OFDM-based CSI estimation scheme to sense the skin deformation patterns of a UFA while mitigating the frequency-selective fading. We estimate CSI with 16 carrier frequencies and fuse the 16 estimations into a one-dimension CSI signal. In general, the more channels we design for our OFDM scheme, the more accurate the estimated CSI will be. This is because a large number of channels can reliably average out the fading in some frequencies. However, a larger number of channels bring more computational overhead to our system as the FFT, CFR estimation, and PCA in the CSI estimation module are operated with more signal samples. In this evaluation, we examine the system's accuracy and execution time with and without the OFDM scheme. We also evaluate the performance of the OFDM scheme when different numbers of subcarriers are used.

We collect data from six subjects (subjects 21-26 in Fig. 14(b)). The data collection process is the same as described in Sec. 5.1 except that the speakers transmit both the OFDM signal and an 18kHz continuous wave (CW) signal. As illustrated in Fig. 15, the OFDM frame and the CW frame are transmitted in TDM. In this way, we can compare the performance when these two signals are used respectively. We also compare the performance when our OFDM scheme has different numbers of subcarriers by selecting subsets of the 16 channels of the OFDM signal. Specifically, we select 8 subcarriers, 4 subcarriers, and 2 subcarriers from the 16-subcarrier OFDM signal. We use the selected subcarriers to estimate the CSI and to further train new CNN models as discussed in the previous sections. Note that there are many combinations when selecting a subset of subcarriers. We randomly select five of the combinations and compute the average accuracy. We use leave-one-subject-out validation among the six selected subjects to evaluate the performance.

The result is shown in Fig. 16. Overall, the accuracy decreases as the number of subcarriers decreases. Specifically, the performance decreases slightly when the number of subcarriers is decreased from 16 to 4 and when there are less than 4 subcarriers, the performance drops sharply. Especially, when the transmitted signal is a CW signal, the F1-score drops to 0.77 as there is no mechanism to combat the frequency-selective fading. In terms of execution time, as the number of subcarriers decreases, the average execution time decreases accordingly. This is because a shorter OFDM signal leads to faster FFT, CFR estimation, and PCA.

5.2.3 Benefits of Multiple Links. As discussed in Sec. 4.2, our system uses 8 links (2 speakers with 4 microphones) to capture DFS profiles from different observation angles. In this section, we evaluate the system's performance

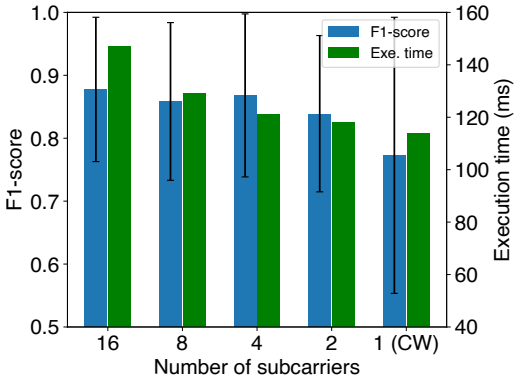


Fig. 16. The system's performance with different numbers of subcarriers.

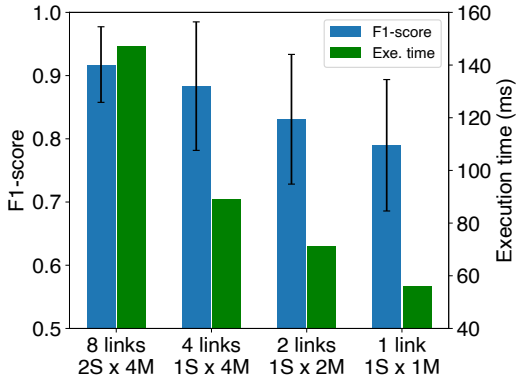


Fig. 17. The system's performance with different numbers of links. (S: speaker, M: microphone)

with different numbers of links. We select subsets of links, that is, 4 links (1 speaker with 4 microphones), 2 links (1 speaker with 2 microphones), and 1 link (1 speaker with 1 microphone) from the original 8 links to train new CNN models as discussed in Sec. 4.3. We use the new models to predict UFAs. Note that we modify the input size of the CNN architecture to match the number of links. Also, there are many combinations when selecting the microphone-speaker pairs. In this evaluation, we select every possible microphone-speaker pair and compute the average performance. We also evaluate the average execution time when different numbers of links are used. We train new models using data from subjects 6-26 and we test the models on subjects 1-5.

The result is shown in Fig. 17. As the number of links decreases, the average accuracy decreases accordingly. This result indicates that with multiple speakers and microphones, the system can capture more comprehensive features of UFAs, thus achieving higher accuracy and robustness. In terms of execution time, the result shows that every time the number of links is cut by half, the execution time is reduced by half. This is because, in the CSI estimation module and the CSI signal processing module, each link computes the DFS profile independently. Therefore the computational overhead in these two modules is proportional to the number of links.

5.2.4 Benefits of Data Augmentation. In this subsection, we evaluate the system's performance with and without data augmentation. So far, the CNN model is trained with a 4× data augmentation rate. We train new models without data augmentation and with 2×, 3×, 5× data augmentation rates. We use the newly trained models to predict the six UFAs. We train the new models using data from subjects 6-26 and we test the models on subjects 1-5. The result is shown in Fig. 18. This result indicates that the accuracy is increased as the data augmentation rate increases. However, when the augmentation reaches 5×, the accuracy decreases instead. This result demonstrates the data augmentation techniques we use can truly provide good quality and sufficient data to our CNN model.

5.2.5 Benefits of the Confident Control Constraint. In this subsection, we evaluate the benefits of introducing the confidence control constraint. By default, the confidence control constraint coefficient (*i.e.*, the α in Eq. 11) is set to 0.1. In this evaluation, we train CNN models with different α values and we evaluate the system's performance using the newly trained models. We collect data from subjects 6-26 as the training set and we test the models on subjects 1-5. The result is shown in Fig. 19. Note that $\alpha = 0$ means the model is trained without the confidence control constraint. This result indicates that with a proper confidence control constraint, the model is forced to learn more general features, and thus, it is easier to adapt to new users. However, when the constraint is too

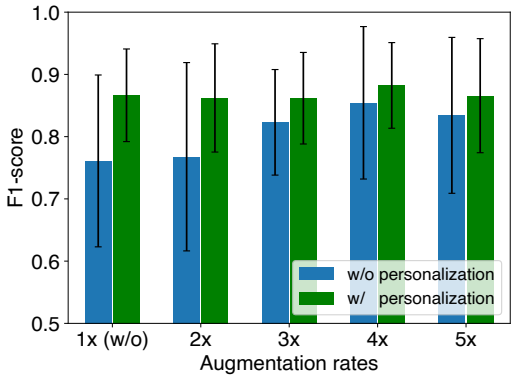


Fig. 18. The system's performance with different data augmentation rates.

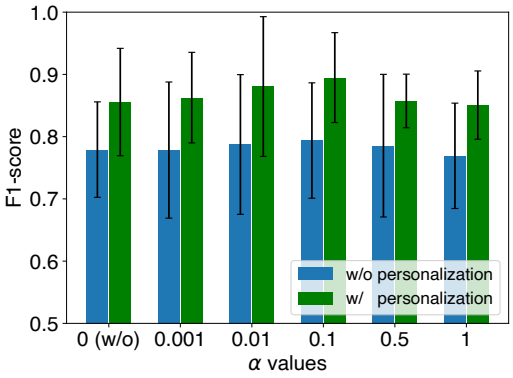


Fig. 19. The system's performance with different confidence control constraint coefficients.

strong, the performance degrades instead. We believe that in this case, the loss function is dominated by the confidence control constraint, and therefore, the model parameters are not optimized.

5.2.6 Benefits of Personalization. In this subsection, we evaluate the performance with and without personalization. As explained in Sec. 4.3.3, different users have different DFS profiles when performing the same facial action. Thus, our system uses transfer learning to adapt the base model to a new user. In this subsection, we explore how the number of samples used in transfer learning can influence performance. We use leave-one-subject-out validation to train a model for each user. We randomly select 2, 4, 6, 8, and 10 samples from the first data collection session of each user (see Sec. 5.1) and we use the sampled data to personalize the model. As shown in Fig. 20, generally, the F1-score increases as more samples are used in transfer learning. Note that when four samples are used, the F1-score increases to 0.92, and with more samples are used, the F1-score doesn't change too much. Thus, in our system, we use four samples of each facial action to personalize the base model to a new user. We also evaluate the time overhead for our system to complete the personalization process. We compute the average personalization time on both the Raspberry Pi 3 and the server. As shown in Fig. 20, on the Raspberry Pi 3, personalizing a model with four samples of each UFA takes around three minutes and it takes six minutes to personalize the model with ten samples of each UFA. When personalization is conducted on a server, the time overhead is within 30 seconds.

5.2.7 Impact of Elapsed Time after Model Training. In the above evaluations, the second data collection session is conducted less than 5 minutes after the first data collection session (see Sec. 5.1). One concern is that does the time interval between model training and testing affects the system's performance? In this evaluation, we evaluate the system's performance with different elapsed times between model training and testing without tuning the network parameters. Four subjects (subjects 1, 5, 9, and 18) are involved in this experiment. In addition to the two data collection sessions described in Sec. 5.1, we introduce a third data collection session that is conducted one day, one week, and one month after the first two data collection sessions. We use the previously trained model to predict the data from the third session. The result is shown in Fig. 21. This result demonstrates that even after one month, the system is still accurate with little performance degradation.

5.2.8 Performance with Different Learning Models. In this subsection, we compare the performance of our CNN model with other deep learning and machine learning models. We first introduce the models we use in this evaluation and then we present the experimental result. The followings are the three deep learning models used

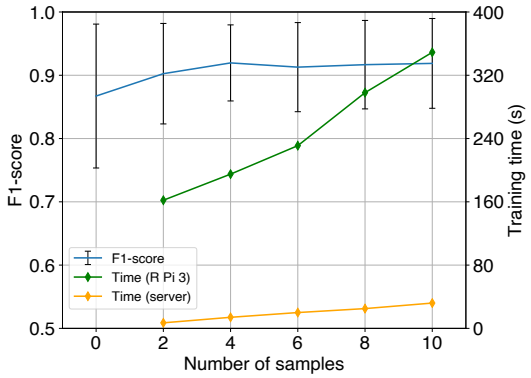


Fig. 20. The system's performance with different numbers of samples used in personalization. (R Pi: Raspberry Pi)

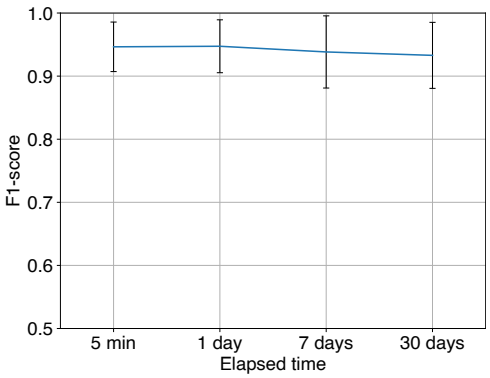


Fig. 21. The system's performance with different elapsed times after model training.

in this evaluation. (i) **CNN-GRU**. This is a hybrid architecture with a CNN module and a Gated Recurrent Unit (GRU) module. The CNN module contains two consecutive convolutional layers that are identical to the first two convolutional layers in our CNN architecture. The CNN module generates 32 channels of 8×4 feature maps. The CNN module is followed by the GRU module. The GRU module contains 4 GRU units corresponding to the 4 columns of each of the 32 feature maps. The hidden layer of this GRU module has 128 units. The GRU module is followed by a 128-unit FC layer and a Softmax layer. (ii) **GRU**. The GRU architecture we use has 15 GRU units corresponding to the 15 columns of the 8 DFS profiles. The GRU model has 128 hidden units and the output of this GRU is fed to a 128-unit FC layer and a Softmax layer. (iii) **FNN**. This is a simple feed-forward network with 4 hidden layers and each of which contains 1024 units and is followed by a ReLU activation. The hidden layers are followed by a 128-unit FC layer with a Softmax output. Note that we use dropout at the last FC layers in the above deep learning models with the same dropout rate as our CNN model and we use the same transfer learning strategy as described in Sec. 4.3.3 to personalize the above models. We also include the following machine learning model in this comparison. (iv) **SVM**. We use a Gaussian kernel SVM where the input features are the flattened 3600-sample ($8 \times 30 \times 15$) DFS profiles. We use data from subjects 6-26 to train the above-mentioned models and we test the models on subjects 1-5. The result is shown in Fig. 22. Among all the models, the CNN model achieves the best performance.

5.2.9 Performance in Real-world Scenarios. In the above evaluations, all the experiments are conducted in a quiet lab environment where the subjects sit still. In this subsection, we evaluate the system's performance in some real-world scenarios. Two subjects are involved in this evaluation. We design the following nine scenarios. (i) **Quiet lab**. This is the standard scenario where the above evaluations are conducted. The noise level in this scenario is $42dB$. (ii) **Noisy lab**. In this scenario, we play music in the lab room to increase the environmental noise to $65dB$. (iii) **Theater**. In this scenario, we simulate the situation when the user uses our eyewear in a theater by playing a movie clip with a laptop connected to a pair of stereo loudspeakers. The noise level in this scenario is $71dB$. (iv) **Canteen**. We conduct experiments in a canteen on our campus at lunchtime. The noise level in the canteen is $64dB$. (v) **Downtown**. We simulate the situation when the user is in noisy downtown areas by playing a video clip of a downtown street tour. The noise includes conversations, pedestrian footsteps, vehicle noise, horn honking, advertisements, etc. The noise level in this scenario is $78dB$. We also conduct experiments where the subject has different postures and body activities. (vi) **Lying down**. In this experiment, we ask the subjects to lie down while performing UFAs. We design this experiment to examine whether the facial muscle

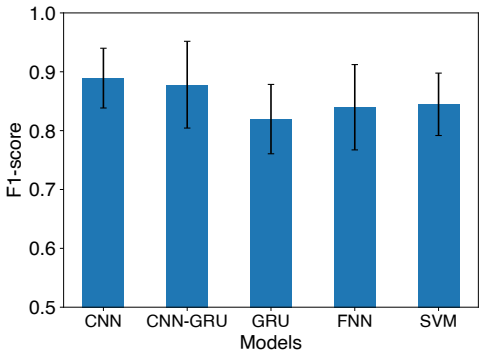


Fig. 22. The system's performance with different learning models.

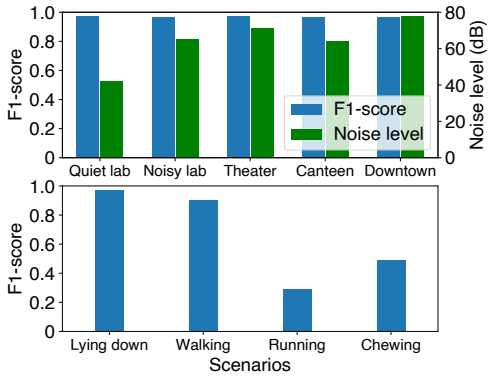


Fig. 23. The system's performance in real-world scenarios.

shape changes caused by different postures affect UFA detection accuracy. (vii) **Walking**. In this scenario, we ask the subjects to perform marching on the ground while collecting the UFA data. The subjects can adjust the stride frequency according to their ordinary practice. (viii) **Running**. In this experiment, we ask the subjects to perform running on the ground. Same as the previous scenario, the subjects can adjust the stride frequency freely. (ix) **Chewing**. In this scenario, the subjects chew gum while performing the UFAs. We design this experiment to examine whether other facial activities would interfere with UFA detection.

The result is shown in Fig. 23. In the first five scenarios, since the background noise is the only interference to our system, the performance is hardly affected because our system works in the inaudible frequency band and the audible noises are removed by low-pass filtering. The result also shows that lying down does not affect the system's performance. This is because although the facial skin may have a different shape when lying down, the skin deformation patterns (speed, spatial distribution) of different UFAs do not change. In the walking scenario, the result indicates that the performance is slightly affected while in the running scenario, our system fails to work. This is because when the user's head is in fast motion, the facial area is no longer relative stationary to the eyewear. Thus, the CSI variation is mainly caused by the user's body motions rather than UFAs. The result also indicates that our system fails to detect UFAs when the user is chewing food. This is because when chewing, a large portion of facial skin deforms accordingly. Therefore, the measured CSI contains the information of both UFAs and chewing patterns which are interfered with each other.

5.2.10 Performance on Non-upper Facial Actions. So far, we only focus on upper facial actions. In this subsection, we discuss whether our system can also detect non-upper facial actions. Again, we choose the target non-upper facial actions from the FACS [8]. We select *lip corner puller* (AU12), *lip corner depressor* (AU15), *mouth stretch* (AU27) and *lip puckerer* (AU18). We conduct two additional data collection sessions to collect the above-mentioned non-upper facial actions from subjects 1-6 and 16. The data collection process is the same as described in Sec. 5.1. We join the newly collected dataset with the original UFA dataset to train a model to classify all the ten facial actions. Note that we increase the size of the Softmax layer in our CNN architecture to ten units in order to support the new classification task. We conduct leave-one-subject-out validation on each of the six subjects and Fig. 24 shows the classification result. This result indicates that although our eyewear can detect the selected UFAs, it has poor performance in detecting non-upper facial actions. We believe this is because the skin deformation areas of these non-upper facial actions are far away from eyewear thus making them hard to detect. Another reason is that the skin deformation areas of all these non-upper facial actions locate at lower facial areas which

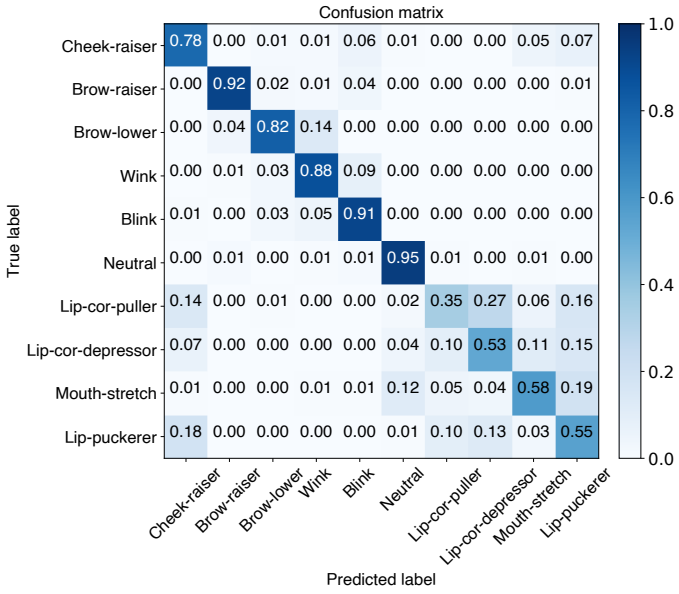
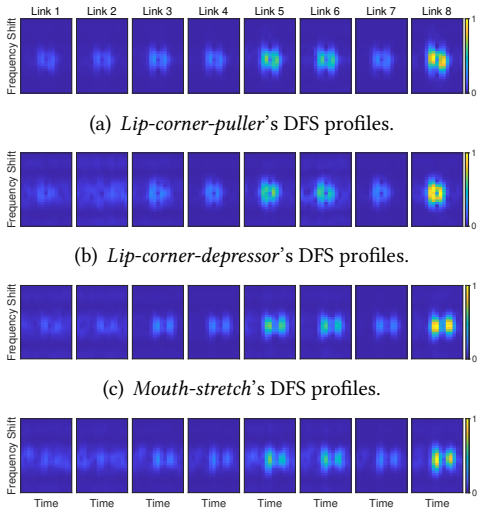


Fig. 24. Confusion matrix of ten facial actions.



(a) *Lip-corner-puller*'s DFS profiles.

(b) *Lip-corner-depressor*'s DFS profiles.

(c) *Mouth-stretch*'s DFS profiles.

(d) *Lip-puckerer*'s DFS profiles.

Fig. 25. The DFS profiles of the non-upper facial actions in Sec. 5.2.10.

means the eight links shall have similar DFS energy distributions. For example, Fig. 25 shows the DFS profiles of the four non-upper facial actions from one subject and all of them show similar energy distributions among the eight links. It is worth noting that a large number of these non-upper facial actions are misclassified into *cheek-raiser* and some of the *cheek-raiser*s are misclassified into the non-upper facial actions. This is because the involved skin deformation areas of both *cheek-raiser* and the four non-upper facial actions are similar.

6 DISCUSSION

Despite the high accuracy and the hands-free interaction mechanism that our system can provide, our system still has limitations. In this section, we discuss these limitations and point out the directions to improve the system in the future.

(1) **Energy consumption**. Energy consumption is an important issue since our system runs on eyewear. We measure the energy consumption of our system on the Raspberry Pi 3 Model B+ using a USB power meter [4]. We run experiments separately for the three modules of our system and the result shows that the energy consumed by the three modules are $41mJ$ ($2.2W$), $2.8mJ$ ($1.75W$) and $370mJ$ ($3.7W$), respectively and the total energy consumption of a UFA inference is $413.8mJ$. It is worth noting that when the device is in the idle state, the energy consumption is $379mJ$ ($2.58W$) within the same amount of time. This result implies that when our system is running, most of the energy is consumed by the tasks that are unrelated to UFA detection (*e.g.*, OS). This is reasonable because Raspberry Pi 3 B+ is a desktop computer that low energy is not its top priority. In the future, our system can be implemented on a low-energy chip (*e.g.*, GAP 8 low-power neural accelerator [47]). Also, we can further decrease the energy consumption of the CNN module by using knowledge distillation techniques to transfer the classification abilities of our large CNN model to a light-weight one [56].

(2) **Personalization overhead**. We use transfer learning to personalize the recognition model to new users (Sec. 4.3.3). However, as shown in Sec. 5.2.6, it takes a Raspberry Pi 3 several minutes to complete the

personalization process. This may degrade the user's experience. One possible solution to handle the overhead is to offload the personalization process to a cloud server since our evaluation shows that personalization only takes less than 30 seconds if conducted on a server. Recent work has shown that the DFS profiles collected from different links can be leveraged to generate a domain-independent body-coordinate velocity profile (BVP) [62]. We can borrow this idea to combine the DFS's from all links to compute a user-independent feature. In this way, the personalization process could be omitted. Also, adversarial learning can be used to force our model to generate user-independent features [20]. Therefore, this technique could also be used to replace personalization.

(3) **Fast body motions**. Although our system can work with high accuracy even when the user is walking (Sec. 5.2.9), the system's performance is poor when the user moves very fast (*e.g.*, running). This is because when the user's head is in fast motion, the facial area is no longer relatively stationary to the eyewear. Thus, the signal path variation is mainly caused by the user's body motion rather than UFAs. One possible direction to solve this problem is to leverage IMUs to track body motions and use the body motion data to filter out the motion-induced noises [41].

(4) **Non-upper facial actions**. As discussed in Sec. 5.2.10, our system cannot detect non-upper facial actions. This is mainly because non-upper facial actions cannot create significant path length variations to the transceivers and they have similar skin deformation patterns. However, with modifications to the orientation of speakers and microphones and with an improved signal processing pipeline and a larger dataset, deriving non-upper facial actions from glass-mounted acoustic sensors is still feasible. This is worth exploring because it could enable some important applications for healthcare such as emotion recognition and fatigue assessment.

7 RELATED WORK

In this section, we review the existing research works that are related to this paper. Since our system is an acoustic-based UFA recognition system, the works we review focus on the following two topics: acoustic sensing systems and facial activity detection systems.

7.1 Acoustic Sensing Systems

In recent years, there has been a vast body of works on acoustic sensing. We organize the state-of-the-art systems in terms of their applications. (i) **Gesture recognition**. AudioGest [42] leverages a pair of built-in speaker and microphone to detect the Doppler shift caused by hand gestures. They transform the Doppler shift to velocity and use the velocity characteristics to distinguish six hand gestures without training. UltraGesture [28] is able to recognize 12 hand gestures with high accuracy. The system measures the channel impulse response (CIR) magnitude and uses a CNN to classify the CIR tensor into hand gestures. Similar to UltraGesture, RobuCIR [54] detects a hand gesture by measuring the CIR. Different from the previous work, RobuCIR measures the phase and magnitude of CIR. Moreover, a frequency-hopping technique is used to mitigate frequency-selective fading. (ii) **Motion tracking**. LLAP [53] is a hand tracking system that is able to measure hand motions by extracting the phase changes from a single-frequency continuous wave (CW) signal. FingerIO [34] is a finger tracking system. Finger motions are tracked by detecting the variations of the cross-correlation profile of the transmitted and received OFDM signal. Covertband [35] is a full-body tracking system that works in a similar way to FingerIO. The difference is that FingerIO uses the built-in speaker and microphone of a mobile device to track a finger while Covertband leverages the speaker and microphone in a home audio system to track human motions. EchoTrack [5] is a tracking system that tracks finger motions by measuring the time of flight (ToF) information of the transmitted signal. Strata [60] is a finger tracking system that achieves higher accuracy than the previous works. Strata estimates the CIR and measures the phase changes of each channel tap to detect finger motions. In [30],

Table 5. A comparison of our work with other facial activity detection systems.

Work	Facial activities	Sensor / contactness	Performance	# subjects
Eyemotion [17]	brow lower, upper lid raiser, cheek raiser, eyes closed, left/right brow raiser, left/right wink and neutral	camera / no	F1-score: 0.68	23
Li *et al.* [26]	gaze fixation, blink and saccade	optical sensor / no	F1-score: 0.93, mean error: 0.8*mm* (pupil tracking)	22
Suzuki *et al.* [45]	happy, angry, sad, surprised and neutral expressions	optical sensor / no	accuracy: 0.88	10
Hamedi *et al.* [16]	right/left cheek raiser, brow lower, brow raiser, clenching molar teeth, mouth stretch and neutral	EMG / yes	accuracy: 0.87	10
Gruebler *et al.* [15]	smile, brow lower	EMG / yes	F1-score: 0.90	10
W!NCE [41]	cheek raiser, brow lower, brow raiser, nose wrinkler, blink and neutral	EOG / yes	F1-score: 0.95 (blink), 0.88 (others)	17
This work	cheek raiser, brow raiser, brow lower, wink, blink and neutral	acoustic sensor / no	F1-score: 0.92	26

the authors propose a hand motion tracking system that leverages a microphone array and an RNN-enhanced algorithm to track hand motions on a home scale. (iii) **Vital sign detection**. ApneaApp [33] is a respiration monitoring system that is able to detect apnea when the user is asleep. The breathing pattern is estimated by measuring the ToF changes caused by the user's chest motions. In [52], the authors propose a C-FMCW ranging method with higher distance resolution that can detect breathing with minor chest displacement. BreathListener [57] is a breathing monitoring system for drivers. They extract breathing patterns using the acoustic signal's energy spectrum density (ESD) and a Generative Adversarial Network (GAN) to enhance the ESD in order to achieve fine-grained breathing pattern detection. BreathJunior [51] is a white noise-based breathing monitoring system for infants. The speakers in BreathJunior send pseudo-randomly generated Gaussian white noise, and the receivers transform the white noise into chirps to perform ranging.

7.2 Facial Activity Detection Systems

In this section, we summarize the state-of-the-art facial activity detection techniques. A comparison of these works with our work is present in Tab. 5. In the following, we organize these works by their sensing modalities. (i) **Camera**. In [17], the authors use an eye-facing camera in a VR headset to detect UFAs. This work tries to infer facial expressions with a gaze-tracking camera where the field of view is restricted to the eye area. Kwon *et al.* [24] uses a customized eyeglass that is equipped with an in-ward camera to detect facial expressions. The camera in their eyeglass is mounted on the side of the frame so that it can capture a larger field of view. Ku *et al.* [22] explores the feasibility of using eye expressions as a hands-free input mechanism for AR/VR devices. They use the wide-angle camera in eye trackers to detect 12 eye expressions which, according to their studies, are suitable for human-computer interaction. (ii) **Optical sensors**. Masai *et al.* [31] develops an eyeglass that is equipped with 17 photo reflective sensors to detect eight facial expressions by measuring the facial skin deformation around

the eye. Similarly, the authors in [45] mount 15 photo reflective sensors in a VR headset to detect five facial expressions. LiGaze [25] is a pupil tracking system on a VR headset. It leverages 16 photodiodes to capture the pupil's light absorption of the VR display's illumination to infer the pupil's position. Another pupil tracking system [26] uses a similar approach to LiGaze. The difference is that [26] is built on a normal eyeglass and a series of NIR LEDs are used to illuminate the eye area. (iii) **RF sensors**. Kim *et al.* [21] proposes a blink detection system using Doppler radar. Their system can classify a blink into a conscious or unconscious blink based on the different spectrogram characteristics. Yamamoto *et al.* [58] uses a Doppler sensor to detect a blink and to estimate the blink duration. The methodology in this work is similar to that of [21]. (iv) **Electromyography (EMG)**. In [14] and [15], the authors develop a wearable device that can detect three facial expressions from the facial EMG signal. They use the readings from the three electrodes, which are attached to human face, to infer a facial expression. Hamedi *et al.* [16] proposes a facial gesture recognition system that uses 3 pairs of EMG electrodes to detect ten facial gestures. (v) **Electrooculography (EOG)**. Zhang *et al.* [61] uses forehead EOG sensors to detect eye activities such as blink, saccade, and fixation in order to assess driving fatigue. In [3], the authors propose an activity recognition system that uses EOG sensors to detect eye movement and use eye movement patterns to infer the user's activities. Ishimaru *et al.* [19] uses a commercial EOG eyeglass to detect user's activities. The methodology of this work is similar to that of [3]. The difference is that the EOG sensor used in this work can sense activities unobtrusively. W!INCE [41] uses the same commercial EOG glasses as in [19] to sense the user's UFAs. W!INCE is able to recognize six UFAs including one neutral state by identifying the distinct patterns of the EOG readings. Besides, W!INCE can remove the motion artifact that is brought about by body motions.

8 CONCLUSION

In this paper, we present an acoustic-based upper facial action (UFA) recognition system for smart eyewear. We mount several pairs of acoustic sensors on an eyeglass to sense the skin deformation patterns in order to detect facial actions. Our proposed system is able to recognize six upper facial actions with high accuracy and robustness. We leverage several novel techniques to resolve the challenges in designing the system. We adopt an OFDM design to mitigate the frequency-selective fading. We use a time-frequency analysis and a CNN to extract upper facial action features. When training the CNN model, we apply several data augmentation techniques to enhance our dataset and we adopt some deep learning techniques to alleviate overfitting. We also use transfer learning to personalize the trained model to new users. We conduct comprehensive experiments to evaluate the proposed system. The experimental results show that our system achieves high recognition accuracy and robustness with an average 0.92 F1-score among 26 subjects.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous editors and reviewers for their valuable comments and helpful suggestions. This work was supported in part by the Hong Kong RGC under Contract CERG 16204418, R8015, in part by the National Natural Science Foundation of China under Grant No. 61701216, in part by the Guangdong Natural Science Foundation No. 2017A030312008, in part by the Guangdong Innovative and Entrepreneurial Research Team Program under Grant No. 2016ZT06G587, in part by the Guangdong Provincial Key Laboratory under Grant No. 2020B121201001, in part by the Shenzhen Science, Technology and Innovation Commission Basic Research Project under Grant No. JCYJ20180507181527806 and in part by the Shenzhen Sci-Tech Fund under Grant No. KYTDPT20181011104007.

REFERENCES

[1] Amazon. 2020. Official Site: What is Alexa? https://developer.amazon.com/en-US/alexa

[2] Behrooz Ashtiani and I. Scott MacKenzie. 2010. BlinkWrite2: An Improved Text Entry Method Using Eye Blinks *(ETRA '10)*. ACM, New York, NY, USA, 339–345. https://doi.org/10.1145/1743666.1743742

[3] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. 2011. Eye movement analysis for activity recognition using electrooculography. *IEEE transactions on pattern analysis and machine intelligence* 33, 4 (2011), 741–753. https://doi.org/10.1109/TPAMI.2010.86

[4] CHargerLAB. 2020. POWER-Z KM001 USB Power Tester Voltage Current Ripple Dual Type-C Meter. http://www.chargerlab.com/power-z-km001-usb-power-tester-voltage-current-ripple-dual-type-c-meter/

[5] Huijie Chen, Fan Li, and Yu Wang. 2017. EchoTrack: Acoustic device-free hand tracking on smart phones. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8057101

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

[7] D. Denney and C. Denney. 1984. The eye blink electro-oculogram. *The British journal of ophthalmology* 68, 4 (Apr 1984), 225–228. https://doi.org/10.1136/bjo.68.4.225

[8] Paul Ekman, Wallace V Friesen, and Joseph C Hager. 2002. Facial action coding system: The manual on CD ROM. *A Human Face, Salt Lake City* (2002), 77–254.

[9] Epson. 2020. Moverio BT-300 Smart Glasses. https://www.epson.com.hk/For-Home/Wearables/Smart-Glasses/Moverio-BT-300-Smart-Glasses/p/V11H756060

[10] Everysight. 2020. About Raptor - Everysight. https://everysight.com/about-raptor/

[11] Bryn Farnsworth. 2019. Facial Action Coding System (FACS) – A Visual Guidebook. https://imotions.com/blog/facial-action-coding-system/

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, International Convention Centre, Sydney, Australia, 1126–1135. http://proceedings.mlr.press/v70/finn17a.html

[13] Google. 2020. Tech Specs - GLASS. https://www.google.com/glass/tech-specs/

[14] Anna Gruebler and Kenji Suzuki. 2010. Measurement of distal EMG signals using a wearable device for reading facial expressions. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. 4594–4597. https://doi.org/10.1109/IEMBS.2010.5626504

[15] Anna Gruebler and Kenji Suzuki. 2014. Design of a wearable device for reading positive expressions from facial emg signals. *IEEE Transactions on Affective Computing* 5, 3 (2014), 227–237. https://doi.org/10.1109/TAFFC.2014.2313557

[16] Mahyar Hamedi, Sh-Hussain Salleh, Mehdi Astaraki, and Alias Mohd Noor. 2013. EMG-based facial gesture recognition through versatile elliptic basis function neural network. *Biomedical engineering online* 12, 1 (2013), 73. https://doi.org/10.1186/1475-925X-12-73

[17] S. Hickson, N. Dufour, A. Sud, V. Kwatra, and I. Essa. 2019. Eyemotion: Classifying Facial Expressions in VR Using Eye-Tracking Cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1626–1635. https://doi.org/10.1109/WACV.2019.00178

[18] Huawei. 2020. HUAWEI X GENTLE MONSTER Eyewear II. https://consumer.huawei.com/en/wearables/gentle-monster-eyewear2/

[19] Shoya Ishimaru, Kai Kunze, Yuji Uema, Koichi Kise, Masahiko Inami, and Katsuma Tanaka. 2014. Smarter Eyewear: Using Commercial EOG Glasses for Activity Recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Seattle, Washington) *(UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 239–242. https://doi.org/10.1145/2638728.2638795

[20] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. ACM, New York, NY, USA, 289–304. https://doi.org/10.1145/3241539.3241548

[21] Youngwook Kim. 2015. Detection of Eye Blinking Using Doppler Sensor With Principal Component Analysis. *IEEE Antennas and Wireless Propagation Letters* 14 (2015), 123–126. https://doi.org/10.1109/LAWP.2014.2357340

[22] Pin-Sung Ku, Te-Yan Wu, and Mike Y. Chen. 2017. EyeExpression: Exploring the Use of Eye Expressions as Hands-Free Input for Virtual and Augmented Reality Devices. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* (Gothenburg, Sweden) *(VRST '17)*. ACM, New York, NY, USA, Article 60, 2 pages. https://doi.org/10.1145/3139131.3141206

[23] Pin-Sung Ku, Te-Yen Wu, and Mike Y. Chen. 2018. EyeExpress: Expanding Hands-Free Input Vocabulary Using Eye Expressions. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings* (Berlin, Germany) *(UIST '18 Adjunct)*. ACM, New York, NY, USA, 126–127. https://doi.org/10.1145/3266037.3266123

[24] Jangho Kwon, Da-Hye Kim, Wanjoo Park, and Laehyun Kim. 2016. A wearable device for emotional recognition using facial expression and physiological response. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 5765–5768. https://doi.org/10.1109/EMBC.2016.7592037

[25] Tianxing Li, Qiang Liu, and Xia Zhou. 2017. Ultra-Low Power Gaze Tracking for Virtual Reality. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (Delft, Netherlands) *(SenSys '17)*. ACM, New York, NY, USA, Article 25, 14 pages. https://doi.org/10.1145/3131672.3131682

[26] 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* (New Delhi, India) *(MobiCom '18)*. ACM, New York, NY, USA, 67–82. https://doi.org/10.1145/3241539.3241578

[27] Qiongzheng Lin, Zhenlin An, and Lei Yang. 2019. Rebooting Ultrasonic Positioning Systems for Ultrasound-Incapable Smart Devices. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. ACM, New York, NY, USA, Article 2, 16 pages. https://doi.org/10.1145/3300061.3300139

[28] Kang Ling, Haipeng Dai, Yuntang Liu, and Alex X Liu. 2018. UltraGesture: Fine-Grained Gesture Sensing and Recognition. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2018.8397099

[29] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: High-Precision Acoustic Motion Tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (New York City, New York) *(MobiCom '16)*. ACM, New York, NY, USA, 69–81. https://doi.org/10.1145/2973750.2973755

[30] Wenguang Mao, Mei Wang, Wei Sun, Lili Qiu, Swadhin Pradhan, and Yi-Chao Chen. 2019. RNN-Based Room Scale Hand Motion Tracking. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. ACM, New York, NY, USA, Article 38, 16 pages. https://doi.org/10.1145/3300061.3345439

[31] Katsutoshi Masai, Yuta Sugiura, Masa Ogata, Kai Kunze, Masahiko Inami, and Maki Sugimoto. 2016. Facial Expression Recognition in Daily Life by Embedded Photo Reflective Sensors on Smart Eyewear. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. ACM, New York, NY, USA, 317–326. https://doi.org/10.1145/2856767.2856770

[32] J!NS MEME. 2020. J!NS MEME: The world's first wearable eyewear that lets you see yourself. https://jins-meme.com/en/

[33] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. 2015. Contactless Sleep Apnea Detection on Smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (Florence, Italy) *(MobiSys '15)*. ACM, New York, NY, USA, 45–57. https://doi.org/10.1145/2742647.2742674

[34] 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*. ACM, New York, NY, USA, 1515–1525. https://doi.org/10.1145/2858036.2858580

[35] Rajalakshmi Nandakumar, Alex Takakuwa, Tadayoshi Kohno, and Shyamnath Gollakota. 2017. CovertBand: Activity Information Leakage Using Music. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 87 (Sept. 2017), 24 pages. https://doi.org/10.1145/3131897

[36] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. 1996. *Signals and Systems (2nd Edition)*. Pearson.

[37] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., 8026–8037. https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[39] Raspberry Pi. 2020. Raspberry Pi 3 Model B+. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[40] Raspberry Pi. 2020. Raspberry Pi 4 Model B. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[41] 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. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 23 (March 2019), 26 pages. https://doi.org/10.1145/3314410

[42] Wenjie Ruan, Quan Z. Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: Enabling Fine-Grained Hand Gesture Detection by Decoding Echo Signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 474–485. https://doi.org/10.1145/2971648.2971736

[43] M Salehi and J Proakis. 2007. Digital communications. *McGraw-Hill Education* 31 (2007), 32.

[44] Seeed. 2020. ReSpeaker 4-Mic Linear Array Kit for Raspberry Pi. https://wiki.seeedstudio.com/ReSpeaker_4-Mic_Linear_Array_Kit_for_Raspberry_Pi/

[45] Katsuhiro Suzuki, Fumihiko Nakamura, Jiu Otsuka, Katsutoshi Masai, Yuta Itoh, Yuta Sugiura, and Maki Sugimoto. 2017. Recognition and mapping of facial expressions to avatar by embedded photo reflective sensors in head mounted display. In *2017 IEEE Virtual Reality (VR)*. 177–185. https://doi.org/10.1109/VR.2017.7892245

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826. https://doi.org/10.1109/CVPR.2016.308

[47] GreenWaves Technologies. 2020. Developer Kits for GAP 8. https://greenwaves-technologies.com/developer-kits/

[48] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.

[49] Outi Tuisku, Ville Rantanen, and Veikko Surakka. 2016. Longitudinal Study on Text Entry by Gazing and Smiling. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (Charleston, South Carolina) *(ETRA '16)*. ACM, New York, NY, USA, 253–256. https://doi.org/10.1145/2857491.2857501

[50] Vuzix. 2020. Vuzix Blade - Powering Solutions. https://www.vuzix.com/products/blade-smart-glasses

[51] Anran Wang, Jacob E. Sunshine, and Shyamnath Gollakota. 2019. Contactless Infant Monitoring Using White Noise. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. ACM, New York, NY, USA, Article 52, 16 pages. https://doi.org/10.1145/3300061.3345453

[52] Tianben Wang, Daqing Zhang, Yuanqing Zheng, Tao Gu, Xingshe Zhou, and Bernadette Dorizzi. 2018. C-FMCW Based Contactless Respiration Detection Using Acoustic Signal. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 170 (Jan. 2018), 20 pages. https://doi.org/10.1145/3161188

[53] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (New York City, New York) *(MobiCom '16)*. ACM, New York, NY, USA, 82–94. https://doi.org/10.1145/2973750.2973764

[54] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the Limit of Acoustic Gesture Recognition. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. 566–575. https://doi.org/10.1109/INFOCOM41043.2020.9155402

[55] Xiaomi. 2020. Mi In-Ear Headphones Pro. https://www.mi.com/my/headphonespro/

[56] 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* (London, United Kingdom) *(MobiCom '20)*. ACM, New York, NY, USA, Article 24, 13 pages. https://doi.org/10.1145/3372224.3380901

[57] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. 2019. BreathListener: Fine-Grained Breathing Monitoring in Driving Environments Utilizing Acoustic Signals. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. ACM, New York, NY, USA, 54–66. https://doi.org/10.1145/3307334.3326074

[58] Kohei Yamamoto, Kentaroh Toyoda, and Tomoaki Ohtsuki. 2019. Doppler sensor-based blink duration estimation by analysis of eyelids closing and opening behavior on spectrogram. *IEEE Access* 7 (2019), 42726–42734. https://doi.org/10.1109/ACCESS.2019.2907697

[59] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a Mobile Device into a Mouse in the Air. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (Florence, Italy) *(MobiSys '15)*. ACM, New York, NY, USA, 15–29. https://doi.org/10.1145/2742647.2742662

[60] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-Grained Acoustic-Based Device-Free Tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara Falls, New York, USA) *(MobiSys '17)*. ACM, New York, NY, USA, 15–28. https://doi.org/10.1145/3081333.3081356

[61] Yu-Fei Zhang, Xiang-Yu Gao, Jia-Yi Zhu, Wei-Long Zheng, and Bao-Liang Lu. 2015. A novel approach to driving fatigue detection using forehead EOG. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*. 707–710. https://doi.org/10.1109/NER.2015.7146721

[62] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. ACM, New York, NY, USA, 313–325. https://doi.org/10.1145/3307334.3326081

[63] Bing Zhou, Mohammed Elbadry, Ruipeng Gao, and Fan Ye. 2017. BatTracker: High Precision Infrastructure-Free Mobile Device Tracking in Indoor Environments. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (Delft, Netherlands) *(SenSys '17)*. ACM, New York, NY, USA, Article 13, 14 pages. https://doi.org/10.1145/3131672.3131689