IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 4, 2020, accepted June 16, 2020, date of publication June 23, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004349

Adaptive Blowing Interaction Method Based on a Siamese Network

YEQING CHEN[1], YULONG BIAN[2], WEI GAI[3,4], AND CHENGLEI YANG[3,4]

[1]School of Software, Shandong University, Jinan 250100, China
[2]School of Mechanical, Electrical, and Information Engineering, Shandong University, Weihai 264209, China
[3]School of Software, Ministry of Education, Shandong University, Jinan 250100, China
[4]Engineering Research Center of Digital Media Technology, Ministry of Education, Shandong University, Jinan 250100, China

Corresponding authors: Chenglei Yang (chl_yang@sdu.edu.cn) and Wei Gai (gaiwei1987@126.com)

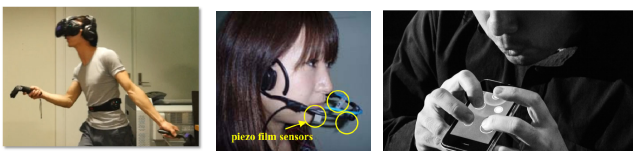
This work was supported in part by the National Key Research and Development Project of China under Grant 2018YFC0831003, and in part by the National Natural Science Foundation of China under Grant 61972233 and Grant 61802232.

ABSTRACT Breathing is a natural and directly controllable human activity. Currently, some works have considered breath as a direct input controlling mechanism. The equipment relied upon in these works is generally complicated, expensive, inconvenient to wear, and sometimes insufficiently controllable. The use of breathing interaction is also limited to a certain scene and is not universal. This paper proposes an adaptive interaction method, which is a natural and directly controllable interaction based on blowing air that only uses headset microphones to obtain the sound waveform of the blowing action without requiring expensive equipment, and that can be used conveniently anytime and anywhere. This blowing interaction uses a Siamese network to achieve ''self-adaptation'' - the first step adapts to noise interference, including environmental noise and the user's own speaking interference, and the second step adapts to different users and equipment, that is, the blowing interaction is used by different people or on different equipment, and the interaction mode can accurately identify the type of blowing. This paper also develops several applications of the blowing interaction method to test the algorithm. During tests, it's proved that this interface not only increases the type of blowing used for interaction but also eliminates interference from speaking in a normal volume effectively and addresses the problem of individual differences.

INDEX TERMS Blowing interaction, domain adaptation, Siamese network, adaptive.

I. INTRODUCTION

From classic interaction modalities, such as keyboard and mouse, to natural interactions such as multitouch, voice, gesture and posture, eye tracking and brain-computer interactions, the field of human-computer interaction technology has made substantial progress. However, these forms of interactions are not necessarily suitable in all scenarios, e.g., when one's hands are unavailable for mouse or touch interaction, in noisy or speechless environments for voice interaction, or for physically challenged people in the case of speech or eye tracking interaction. The convenience and controllable nature of breathing interactions can sometimes make up for the disadvantages of more common interactive modes. Recently, breathing has been considered an alternative control mechanism to influence the physical world and the virtual environment [38]. For example, Sra *et al.* used breathing as a direct control interface in two VR games [38]; see Fig. 1(a). Kuzume presented a hands-free interface based on expiration signals [22]; see Fig. 1(b). Wang *et al.* developed the Ocarina, making blowing into a microphone as a popular input method for smartphone games and music applications [43]; see Fig. 1(c). Ban *et al.* proposed a method to control the rhythm of breathing for relaxation [3].



(a) (b) (c)

FIGURE 1. Some examples that have already adopted breathing as a control mechanism.

To date, some works have studied breathing or blowing air as a direct input modality in proper detail. Depending on whether special detection equipment is needed to obtain

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Hua Chen.

TABLE 1. Collation and comparison of breathing/blowing related research work.

References	Content/Application/Purpose	Breathing/Blowing Data	Equipment	Special Processing	Recognition Methods
[9]	classify breathing activities	depth image data	Kinect depth sensor	fast Fourier transformation (FFT)	SVM
[12]	classify sleep/wake states	ECG, respiration, and body movement signals	a thoracic (VTH) inductive plethysmography record	derived parameters of the three signals	multinomial logistic regression techniques
[14], [24], [28], [35], [36]	experience, entertainment, etc	the rate, volume, rhythm of breathing	chest/abdomen breathing sensors,airflow sensor	–	directly map
[22]	a hands-free man-machine interface using expiration and tooth-touch sound signals	piezoelectric and pyroelectric signal of breathing	three piezo film sensors near user's mouth	–	dyadic wavelet transform
[38]	leveraging breathing as a directly controlled interface	raw waveform obtained by special sensor	special sensor Zephyr	zero-crossing points of the raw waveform's first order difference, not twist torso	dynamic time warping
[17]	detect breathing phase	lung sounds	an electret microphone inserted at the tube of a stethoscope	spectrogram image	CNN
[32]	recognition of breathing activity and medication adherence	breathing signal	a microphone attached on the inhalation device	a slide window processes spectra	RNN-LSTM
[45]	blow to "trigger" event	blowing signal	microphone	set the threshold above the speaking value	check if the threshold is exceeded
[42], [43]	a wind instrument designed for the iPhone	blowing signal	microphone in mobile phones	–	directly map
[30]	identify which computer screen area the blow is aimed at	blowing signal	a single microphone	the location of the microphone and the user must be fixed	KNN
[16]	breathing controlled mobile phone interface: trigger events	blowing signal	microphone in mobile phones	time counter	The time counter is checked by threshold judgment
our method	improves the accuracy of recognition, adapt to environmental noise and different users, equipment	blowing signal	microphone	time domain and frequency domain characteristics, image-frames	Siamese Network

the breathing signal, the methods for detecting breathing can be divided into two categories: methods based on special detection equipment and methods based on common microphones [30], [45]. For the first category, one method based on special detection equipment is to obtain breath signals by detecting chest or abdominal movements [8], [38], and the other is to obtain breathing signals by placing special devices in the mouth that directly detect the airflow of breathing [11], [31]. However, the equipment (e.g., breathing sensors, breathing belts, or other special sensors) used in these works is generally complicated, expensive, inconvenient to wear, and on occasion insufficiently controllable, while also lacking universal portability in daily life. The second category uses common equipment but does not consider noise interference and does not address the difference problem.

In this paper, the relevant research work on breathing/blowing is summarized and compared, as shown in Table 1. By collating and comparing the research work related to breathing/blowing, it can be found that the current research work on breathing/blowing may depend on special equipment used in a particular field or may use simple equipment but have some deficiencies in dealing with interference or individual differences and are not adaptive. In this paper, we propose an adaptive interaction method based on blowing air that can provide interaction operations to applications by transferring blowing into sound using common headset microphones and then identifying and classifying the sounds into different blowing types. This blowing interaction uses a deep learning model, the Siamese network, to achieve ''self-adaptation''. To verify the advantages of our method, after excluding those jobs that use special equipment, we chose to reproduce the work of Jackson *et al.* [16] for user comparison experiments.

In this paper, we assembled a sensor setup lighter than in previous work with a different sensing algorithm, and we performed our evaluation while embracing the environmental noise in different scenarios with different devices; this is the main difference between our study and previous literature (see Table 1). The contributions of our paper include the follwing: **a.** Our blowing interaction method is simple, convenient, and controllable. The interaction is simple (blowing), and the device (common microphone) is simple to use. **b.** Our approach is more effective at dealing with the interference of user speech than existing methods and can better adapt to environmental noise, different users and equipment, to improve the accuracy of recognition. **c.** Our interaction method can accurately identify the type of blowing, and it

is suitable not only for ordinary scenes but also for special environments (e.g., noise, unavailability of hands) or for special groups (e.g., deaf mutes). We designed three application examples (i.e., playing video on a PC, manipulating Amap on a mobile phone, and playing a VR game) to assess the merits of our method.

The remainder of this paper is organized as follows: We begin by introducing related work and then describe in detail our proposed approach, followed by our experimental results and conclusion.

II. RELATED WORK

This section summarizes some of the most directly related work on interactions based on breathing or blowing air.

A. BREATHING OR BLOWING INTERACTION

Breathing is a human instinct. Because breathing can be consciously controlled, some works have studied it as a directly interactive method. Depending on whether special detection equipment is needed to obtain the breathing signal, the methods for detecting breathing can be divided into two categories: methods based on special detection equipment and methods based on common microphones.

For the first category, some researchers have used wearable sensors or custom sensors to obtain physiological signals (e.g., the amount and speed of the exhaled air, the piezoelectric signal and the temperature) of breathing to achieve direct mapping operations or interactive control. Some studies [14], [24], [28], [35], [36], [38] have used wearable breathing sensors, which are inconvenient and expensive to use and not common in everyday life. The specially customized devices in these studies [1], [2], [18], [22], [29], [31], [34], [37], [39] were also inconvenient and expensive to use and impose considerable limitations in terms of usage scenarios and usage methods. All of the above works generally relied on custom devices such as breathing sensors, breathing belts, or other special sensors, which are intricate, expensive, inconvenient to wear, lack universal portability in daily life, and may be less controllable, placing considerable limitations on the usage modes and scenarios that have not been further explored.

For the second category, there is a low-cost way of interacting with breathing: using a microphone to obtain the sound of breathing for interactive control. Blowing into the microphone has been a popular input method for smartphone games and music applications since the Ocarina by Wang [42], [43]. Misra *et al.* explored the use of microphones as a generic sensor in MobileSTK to drive sound synthesis algorithms in expressive ways [25]. Igarashi uses nonverbal features (such as ahhh and tatata) in speech to directly control interactives [15]. Patel and Abowd presented a coarse-grained system, called BLUI, that enables blowing at a laptop or computer screen to directly control interactive applications [30]. They classified the air pressure signatures of the signals recorded by a fixed-positioned microphone and assigned them to 1 of 9 cells on the screen. The disadvantage of this approach is that it requires a fixed placement of the microphone and it has considerable limitations in terms of usage scenarios and usage methods. Zielasko *et al.* presented an alternative trigger approach for hands-free interaction scenarios to precisely trigger events by blowing into a microphone [45]. When the blowing value exceeds a given threshold, the event is triggered; otherwise, the event is not triggered. However, to avoid triggers caused by speaking in a normal volume, they set the threshold above the speaking value, which largely limits the range of blowing. Filho *et al.* proposed a hands-free and silent interaction with a mobile phone interface by exploring the processing of the audio from the microphone in mobile phones to trigger and launch software events [16]. They started a time counter that would wait for silence. When the sound level was below the threshold, the time counter was checked to identify the type of breathing. Because mobile phones only have simpler computing processes due to their limited processing power, compared to laptops and desktops, they exhibit difficulty handling complex types of blowing operations such as identifying blowing sounds and speaking voices.

Our research in this paper instead only needs regular microphones to obtain data pertaining to the exhaled airflow and relies on these sound data to identify different interactive operations, which is effective, controllable, and convenient. This paper classifies blowing sound into different categories as directly controlled interactions using a machine learning method, which not only increases the types of blowing used for interaction, but also effectively avoids triggers caused by speaking in a normal volume.

B. RECOGNITION ALGORITHM

In addition to simply detecting sound intensity to correspond to simple interactions [16], [45], there are other works that identified different breathing actions through more accurate and complex recognition algorithms.

Maksym presented an algorithm (multinomial logistic regression techniques) for the nonobtrusive recognition of sleep/wake states using signals derived from ECGs, respirations, and body movements captured while lying in a bed [12]. Xinyue Lu presented a new algorithm to process tracheal sounds has been developed that combines breathing detection in both temporal and frequency domains [23]. Mera described a new approach using a noncontact capturing method of breathing activities using a Kinect depth sensor [9]. They detected the morphological changes of the participant's chest area in real time to obtain depth data, then used FFT to convert the signal from its original domain into a representation in the frequency domain, and finally used a SVM to classify and identify the respiratory activity. Jácome applied deep learning to create an algorithm (Faster R-CNN) for breathing phase detection in lung sound recordings [17]. Pettas employed recurrent neural networks with long short term memory (LSTM) units, to monitor pressurized metered dose inhaler medication adherence [32]. Hamke detected breathing rates and the depth of breath using LPCs and restricted Boltzmann machines to classify the respirator sounds [13].

Emoto proposed a new ANN-based method to effectively detect low-intensity SBEs (non-SBE and SBE classes) from sleep sound recordings, which is useful in effectively and automatically detecting the existence of apnea (silence) segments in sleep sounds [10]. The above machine learning algorithms provide good results in recognizing breathing, but they use a large amount of training data and either target a specific field or require specific equipment.

For the classifier used in this paper, we focus on the domain adaptive works that can solve the equipment difference problem encountered by our blowing interaction. Rita used a domain adaptive approach for SEMG signal classification across multiple subjects [4], [5]. In domain-adaptive subspace learning algorithms, Siamese networks [7] can perform different tasks well [6], [21], [27], [41]. The supervised domain adaptation (SDA) requires labeled target data. Supervised domain adaptation [26], [27] can significantly improve the accuracy of the classifier in identifying target domain samples by learning a small number of labeled samples. In a supervised domain adaptation approach [40], unlabeled and sparsely labeled target domain data were used to optimize domain invariance to facilitate domain transfer, while using label distribution to match loss. There are two network processing streams used for domain adaptation. Koniusz *et al.* [20] presented an approach to domain adaptation by partial alignment of the within-class scatters to discover the commonality, using two CNN streams: the source and target networks fused at the classifier level. Rozantsev *et al.* [33] introduced a two-stream architecture, where one steam operates in the source domain and the other steam operates in the target domain. Chopra introduced a Siamese architecture to minimize the distance between source and target sample pairs [7]. The CNN parameters would be shared as in a Siamese architecture [27]. In addition, the source stream would continue with additional fully connected layers for modeling. In this approach, every class only requires an extremely low number of labeled target training samples, and even one per category can be effective.

In this paper, a domain adaptive approach is adopted to identify the type of air blowing interaction. The specific implementation uses a Siamese network sharing CNN parameters to find a shared subspace for the source and target data. The Siamese network provides excellent classification performance for cases where few samples of training are available in each class. This is in line with the classification training in this paper.

III. SYSTEM ARCHITECTURE

Fig. 2 provides an overview of the system architecture. A web server works as a back-end for processing data, training, and running models. When a person uses the interaction interface, the prediction model running on the server obtains that person's breath data to recognize air blowing actions, and sends them back to the client (e.g., PC, mobile phone, or HTC VIVE). Once the clients receive the action information, the running application performs the corresponding operations.

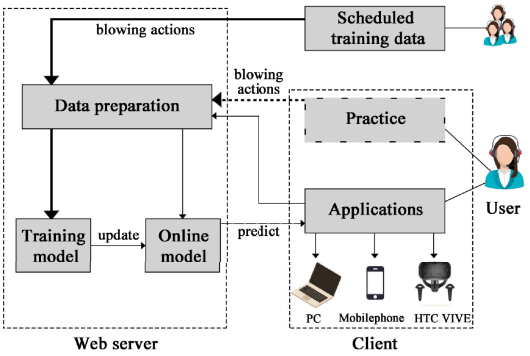


FIGURE 2. System architecture.

To improve the recognition performance, we ordinarily update the model by collecting new training data in two ways. We schedule a time slot to invite a group of people to collect accurate training data, which is called scheduled training data acquisition. Additionally, every day, whenever people come in to use our interaction interfaces, such as microphones, they can choose the practice module to calibrate the system to their particular forms of blowing air. This is not compulsory, and users can use our interface directly without first practicing. The experiments rely on a standard headset microphone, which is common and easy to carry in everyday life, to transform the blowing actions to sound waveforms. The typical sound signals for four different types of blowing are shown in Fig. 3. Subsequently, the interface transmits the sound data to the server. Once the server has collected enough new training data, it will train a new model to update the online model.

(a) *gale* (b) *gust* (c) *breeze* (d) *2gusts* (e) *noise*

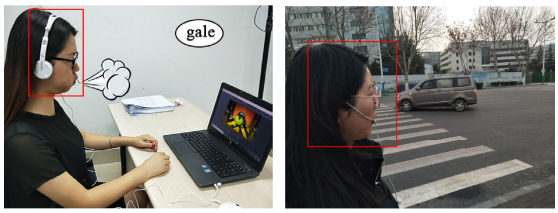
FIGURE 3. Sound waveforms obtained by the microphone for four types of blowing air and an example of a speaking voice.

Air Blowing Interaction Design Scheme. Referring to the work of Sra *et al.* [38], we distinguish five forms of blowing air with regard to duration, intensity and frequency. Among them, four are used for possible interactions (see Figs. 3) and one is used to eliminate interference from a speaking voice (an example in the Fig. 3(e) shows the category of speaking voice):

gale : very strong exhaling sustained for 2 seconds (see Fig. 3(a)).
gust : strong jet but transient for a short duration of less than 1 second (see Fig. 3(b)).
breeze : slow and gentle but transient for a duration of approximately 2 seconds (see Fig. 3(c)).

2gust : strong jet but twice, less than 1 second each (see Fig. 3(d)).
nosie : speaking voice in a normal volume (see Fig. 3(e)).

The low-cost equipment we need to obtain blowing data is a common microphone used in our daily life such as headset microphones. One important consideration is the placement of the microphone. An effective position is near the mouth and pointed towards the mouth (see Fig. 4). Fig. 3 shows that the waveforms of blowing interactions and speaking voices are very different from each other, so our blowing interaction will still be effective even if the user speaks. This assumption is verified in our experiments.

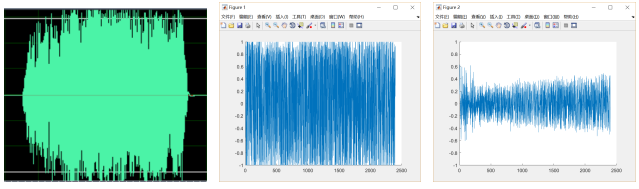


(a) (b)

FIGURE 4. The most efficient place to place a microphone on an ordinary headset.

IV. MODEL DESIGN

The proposed interface acquires the sound signals resulting from blowing air in real time and then determines the particular form of blowing air as interactions for different applications. The signals of the same type of blowing obtained by different equipment used by different people may not be consistent because of individual differences. For example, when one person performs the same type of blowing on a PC and a mobile phone, the blowing data obtained are different, resulting in different test accuracies, as shown in Figs. 5(a), 5(b) and 5(c); when one person uses different mobile phones, the blowing data will also be different, as shown in Figs. 5(b) and 5(c). Because of this, when we use the classifier of a single training dataset to predict new blowing data, the recognition result is insufficient, and the accuracy is not high because of these individual differences. To overcome these differences and achieve accurate interaction, this paper uses the Siamese network of domain adaption concept to deal with differences between training samples (source data) and new test samples (target data).



(a) *PC* (b) *oppo r7* (c) *oppo r11s*

FIGURE 5. The sound waveform representations of a gale obtained by the three devices.

A. TRAINING DATA ACQUISITION

The training data can be collected in two ways:

1) SCHEDULED DATA TRAINING

In the experiments, we relied on two participants for training data collection. To collect a sufficiently large-scale dataset for training onset, we rely on audio recording with a sampling frequency of 8192 Hz. To make the initial training set more standardized, participants are required to blow air every 3 seconds. Through the test, we found that when the sampling rate of the sound signal was 2048 Hz, the data redundancy could be reduced while ensuring the accuracy.

Specifically, we used an Android phone (source data acquisition device: *Source Device A*) to obtain 1043 data samples (after removing the abnormal samples). The training set consists of data X_s and label Y_s. Then a different Android phone (target data acquisition device: *Target Device B*) was used to obtain 10 data samples (data X_t and label Y_t) for model training. Among them, 1043 samples were used as source domain data and 10 samples were used as target domain data. The 10 target domain samples are composed of 2 samples from each of the four blowing types and 2 samples from the speaking voice. *Target Device B* is the device used by a new user. In addition, 200 target domain data points were collected for testing and verification.

2) USER PRACTICE

Due to the different intensity levels when blowing air, users are able to practice before using the interface to obtain better results. There are two purposes of practice: one is with the goal of better familiarizing novices with the available interaction forms, and the other is to collect different new data of different devices from different individuals for the training of the new model.

During practice, every user is instructed to perform the correct kind of exhalation in accordance with the indicated form of blowing air. Each category is repeated several times. If the sound volume is greater than a given threshold, the system collects the sound data. Every user's personal data is uploaded into the server's training set and is treated as new target data, which can be used to train the new domain adaptive model.

B. DATA PREPARATION

1) SIGNAL PREPROCESSING

The sound signal is first normalized to a standardized range of $[-1, 1]$. Then a sliding window is applied for sampling discretization. The continuous signal of a speaking voice in this normalized data is then divided into discrete segments using a sliding window of 8192 Hz (2048 Hz*4) with 67% overlap between segments [19].

2) FEATURE EXTRACTION

In this paper, we propose a data processing method that converts the blowing signal into image-like frames [44] through feature extraction. First, we extracted the relevant features of the blowing sound signal, and then compressed the multiple features into the image-like frames. A CNN is good at learning task-related features and mining the correlation between

different features from the image frames through designed convolutional filters. In this way, we can make full use of a CNN to identify the blow sequence images.

The classifier operates based on the following signals extracted as features from the discrete sound data.

a: TIME DOMAIN FEATURES

In this paper, five types of blowing are distinguished from the intensity and duration of blowing according to the characteristics of the time domain.

a. These features were chosen for their ability to be discriminative in distinguishing the five categories of blowing air. Among them, the mean value (given by (1)), variance (given by (2)), and first-order difference (given by (3)) are all well-known statistical features.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{1}$$

$$s^2 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n-1} \tag{2}$$

$$f'(X_i) = \frac{|\{f'(X_{i+1}) - f'(X_i)\}|}{X_{i+1} - X_i} \tag{3}$$

where X_i is the i-th element in a discrete sample, and n is the length of each sample.

b. In reference to a short-time zero crossing rate, a value-crossing rate means the proportion of the number of normalized values greater than a particular value a (given by (4)).

$$p = \frac{|\{X_i > a\}|}{n} \tag{4}$$

where X_i is the i-th element in a discrete sample, and n is the length of each sample, a is a certain threshold. In this paper, we find the optimal parameter value of a based on a grid search. According to the results of the grid search (see Fig. 6), we finally selected 0.1 and 0.2 values as parameters of the feature ''value-crossing rate''. The feature p is chosen to account for both the strength and duration characteristics of the exhalation.

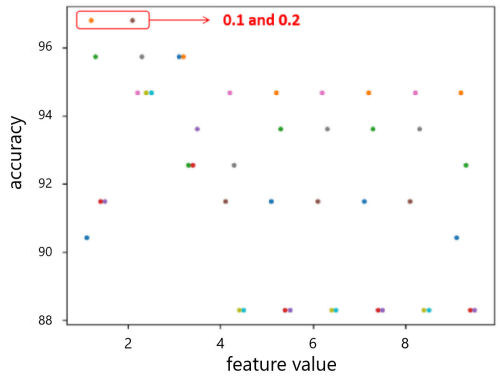


FIGURE 6. The result of the grid search.

c. To access the degree of fluctuation, the data are specifically divided into four equal intervals. If the maximum value of each segment exceeds a certain threshold, it is marked as true (marked as 1); otherwise, it is marked as false (marked as 0), and a true and false table is obtained. Then according to this table, we calculate the sum of the true numbers. In addition, if the true or false value of the current segment is different from that of the previous segment, we add 1 to the previous sum. Finally, the obtained sum is converted into a proportional value p' (given by (5)), which is not greater than 1, to judge the overall fluctuation range.

$$p' = \frac{\sum_{i=1}^{4} |\{f(C_i)\}| + \sum_{j=1}^{3} |\{TF(j)\}|}{5} \tag{5}$$

where C_i is the data of segment i. $f(C_i)$ and $TF(j)$ are calculated using (6) and (7):

$$[f(C_i)] = \begin{cases} 1, & maxC_i > \theta, \\ 0, & otherwise. \end{cases} \tag{6}$$

$$[TF(j)] = \begin{cases} 1, & |\{TF(j+1) - TF(j)\}| > 0, \\ 0, & otherwise. \end{cases} \tag{7}$$

where $\theta = 0.6$, i is 1,2,3,4, j is 1,2,3, and $TF(j+1) - TF(j)$ is the difference between segment $j+1$ and segment j.

b: FREQUENCY DOMAIN FEATURES

To solve the interference problem of the speaking voice to the blowing interaction, we also identify the speaking voice as a category, and the classifier will distinguish the user's speaking voice from the situation of using the blowing interaction. To better identify the interference of the speaking voice on the blowing interaction, we carried out frequency domain analysis on the acquired sound signal.

We start with the spectrum analysis of the data to obtain the spectral data. Specifically, we conduct FFT on the original data and intercept the positive frequency interval of the signal, that is, the frequency corresponding to the first half of the signal ([0, fs/2]). After obtaining the spectral data, we calculate the features of the spectral data, such as the mean square, frequency variance and crossing rate, as the composition of the image-like frames.

In addition, we calculate the peak and extract the characteristics related to the peak. Based on the analyzed spectrum waveform, we find that the frequency band of the sine wave with a larger influence in the speech waveform is different from several other ways. X_{peak} is the peak value of the signal. We use the peak counting method to find n peaks $\{X_{p1}, X_{p2}, \ldots, X_{pn}\}$ (n<N) from the N values of $\{X_1, X_2, \ldots, X_N\}$. The peak index of $\{X_1, X_2, \ldots, X_N\}$ is: $X_{peak} = 1 / n * X_{pj}$. Combining the peak and sliding window processing methods, we deform the spectrum data. Specifically, we set the sliding window length to 10 Hz, calculate the difference between the maximum and minimum values record the sum of these differences across all windows, and normalize the result. We call this the peak difference X'_{peak-T}. According to the characteristics of our data, we slightly deform X'_{peak}, and obtain the peak difference X'_{peak-T} formula

using (8):

$$X'_{peak\_T} = \frac{X'_{peak}}{n * maxX'_{peak}} \tag{8}$$

where the calculation of the X'_{peak} is used by (9):

$$X'_{peak} = \sum_{i=1,i=i+m}^{n} [max_{1\le i\le m}X_i - min_{1\le i\le m}X_i] \tag{9}$$

where n is the number of segments and m is the size of the sliding window. maxX_i is the maximum value in $\{X_i,X_{i+1},\ldots,X_{i+m}\}$, min$X_i$ is the minimum value in $\{X_i,X_{i+1},\ldots,X_{i+m}\}$, max X'_{peak} refers to selecting the maximum value of X'_{peak} from the existing test data, and its role is to perform simple normalization on the data feature X'_{peak-T}.

C. TRAINING AND PREDICTION

1) CLASSIFIER

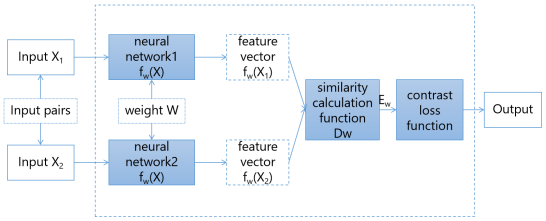


FIGURE 7. The basic structure of the Siamese network.

In our system,we rely on the Siamese architecture which has two streams-one for the source and the other for target samples-to find a suitable shared feature space for both. The basic Siamese network consists of two (symmetric) neural networks, a similarity calculation function and a contrast loss function (see Fig. 7). Different from the general model that simply uses two CNNs to train the source and target samples, the CNN parameters are shared in the Siamese architecture [15]. In this way, samples with the same label can be mapped to each other as much as possible, even if they come from different data domains. The representation of the source domain and target domain in this article is shown in Fig. 8. The classifier in this paper minimizes the distance between samples of the same label (i.e., the minimum distance between samples connected by the blue line in the same ellipse) and maximizes the separation of samples of different classes in different domains (i.e., the maximum distance between samples connected by the green line).

The specific process of domain adaptation based on the Siamese architecture in this paper is shown in Fig. 8. First, we use the feature extraction methods mentioned to initially process the blowing data into ''image-like frames''. Then, we provide these ''image pairs'' formed by the source domain data and the target domain data to the Siamese architecture. The Siamese architecture can be divided into a first half and a second half. The first half is modeled by a CNN (i.e., the embedding function g), which is a convolutional neural

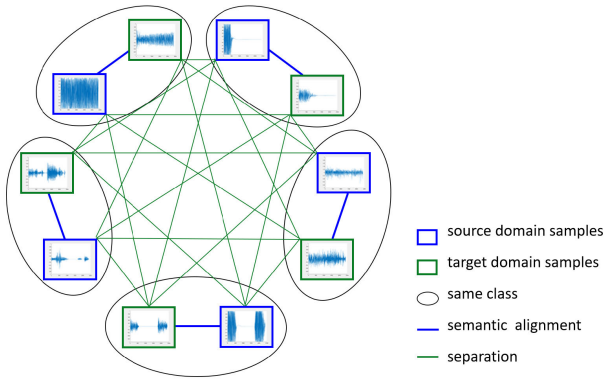


FIGURE 8. Relationship diagram of the source domain and target domain.

network for feature extraction. The training model structure has two processing flows: one for the source domain samples and another for the target domain samples. We provide the ''image pairs'' to the CNN feature extraction network and obtain two feature vectors. Then, we provide these feature vectors to the second half. The second half constructs a distance measure of two feature vectors as a similarity calculation function h of the two ''image-like frames''. The convolutional network training structure used in this paper is as follows: input → convolution (ReLU) → convolution (ReLU) → pooling → flatten → fully connected → fully connected → softmax → output.

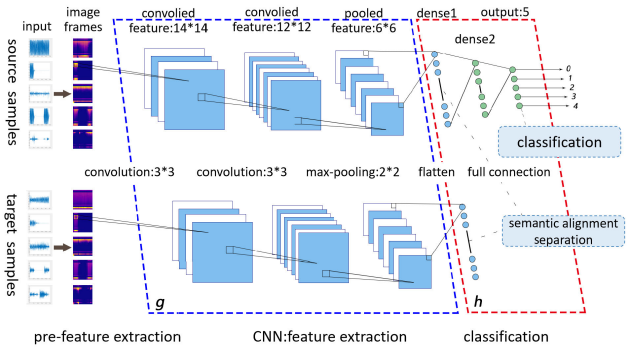


FIGURE 9. Domain adaptation based on the Siamese architecture.

We refer to the method presented in articles [20], [27], which can address supervised domain adaptation by learning a deep model. Analogously, in training, our classification model consists of three loss functions. The semantic alignment loss minimizes the distance between samples from different domains but of the same category of labels.The separation loss maximizes the distance between samples from different domains and the class labels. The classification loss guarantees high classification accuracy. In addition, the source domain processing stream will continue to model h using an additional fully connected layer, which is the modeling of the classification recognition part. The Siamese

network in this paper can be implemented by (10):

$$\begin{aligned} &L(h*g) \\ &= E[loss(h*g(X^s), Y)] \\ &\quad + \sum_{l=1}^{T=5}\{\frac{1}{N_l^S * N_l^T}\sum_{i,j} dis[g(X_i^S), g(X_j^T)]|y_i^S = y_j^T = l\} \\ &\quad + \sum_{l,l'|l\neq l'}^{5,5}\{\frac{1}{N_l^S * N_{l'}^T}\sum_{i,j} sim[g(X_i^S), g(X_j^T)]|y_i^S = l \neq l' = y_j^T\} \end{aligned} \tag{10}$$

In the first part, E denotes statistical expectation and loss is a categorical cross-entropy error function. The function g and the function h represent two functions for constructing the depth model of this paper, and $h * g$ represents the combined form of the two.

In the second part, T is the number of class labels, T=5 in this paper. l denotes the current label. N_l^S represents the number of samples in the source domain data and N_l^T represents the number of samples in the target domain when label is l. $X_i^S = X^S \mid Y = l, X_j^T = X^T \mid Y = l$ refers to random samples in the specified domain when the label is l. *dis* refers to a suitable distance measure of the distribution of X_i^S and X_j^T in the embedded space, expressed as $dis = \| g(X_i^S) - g(X_i^S) \|2$, and $\|\cdot\|$ represents the Frobenius norm.

In the third part, l denotes the current label in the source domain, l' is the label in the target domain, and $l \neq l'$. The definition of N_l^S and $N_{l'}^T$ is similar to the previous definition. $X_i^S = X^S \mid Y = l$ refers to random samples in source domain when the label is l. $X_i^S = X^T \mid Y = l'$ refers to random samples in the target domain when the label is l'. *sim* is a similarity measure, which refers to a suitable similarity grid distributed by X_i^S and X_i^S in the embedding space, expressed as $sim = \max(0, m - \|g(X_i^S) - g(X_i^S)\|^2)$, where m is the margin that specifies the separability in the embedding space.

2) TRAINING

Given the training data acquired using the two procurement schemes described above, the classifier is trained following Algorithm 1 using preprocessed training data as described above. The trained model is periodically updated as new data come in.

Algorithm 1 Training Model

Require: training data d with the corresponding actions a;
Ensure: model m.
1: d'=preprocess(d);
2: m = train(d', a);
3: update online model with m;
4: **return** m

3) PREDICTION

Algorithm 2 provides the details of blowing type recognition. Similar to the practice step, only data with a sound volume greater than a threshold θ are collected (cf. Line IV-C3). Then, the data are prepared (cf. Line IV-C3) as introduced earlier and sent to the online model for recognition (cf. Line IV-C3). Finally, the system performs corresponding operations according to the recognized actions. It can be applied in a wide range of applications, some of which will be introduced in the next section.

Algorithm 2 Air Blow Category Recognition

Require: θ is the threshold;
1: **while** our interface is being used **do**
2: get real-time volume v of blowing sound;
3: **if** $v > \theta$ **then**
4: get sound data d;
5: d'=prepare(d);
6: a = recognize(d');
7: client performs interactive operations according to a;
8: **end if**
9: **end while**
10: **return**

D. EXPERIMENTS

Because we use a domain-adaptive method based on the Siamese architecture to process sound signals, there is no related literature to verify its correctness and recognition accuracy. To evaluate the accuracy of the data collection method, feature extraction processing method, and our classification model independently of its applied use as a direct input controlling method, we performed three comparative verifications: **a.** between different features; **b.** raw data vs. feature extraction data; **c.** SVM vs. Siamese network.

a.Different Features

In feature extraction, this paper mainly carried out time domain features and frequency domain features. In this part, without considering the impact of individual differences, we rely on support vector machines (SVMs) to identify the category based on the features. Although other learning algorithms are applicable as well, we need many test cycles to verify the correctness of the feature selection. We chose a SVM in consideration of the efficiency and availability of SVMs in processing small samples, nonlinear relationships and multiple classification problems. The results of the different features are shown in Fig. 10.

feature1-4 represents the (1)-(4) indicating the time domain features.

feature1-5 represents the (1)-(5) indicating the time domain features.

feature-all represents all features, including time-domain features and frequency-domain features.

The horizontal axis represents the proportion of source domain data used for testing in the 1043 samples. As seen from the Fig. 10, in different test proportions, the accuracy of using the feature extraction method proposed in this paper

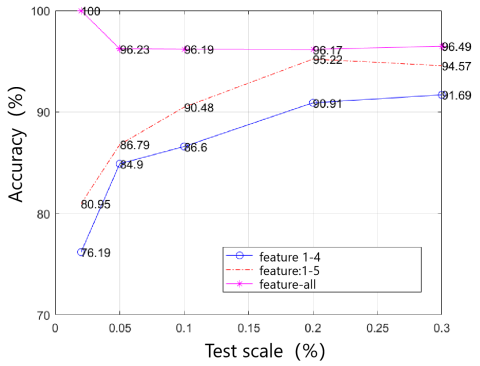


FIGURE 10. The results of comparative test 1: different features.

will generally be better than other features. This shows that the feature method we adopt is desirable.

b.Raw Data vs. Feature Extraction Data

This paper uses the Siamese network to compare and analyze the accuracy of two types of input data: one is to directly use the obtained raw data as the input to the model, and the other is to input the data after feature extraction processing to the model training. That is, according to the feature extraction method described above, we first perform preliminary feature processing on the raw data.

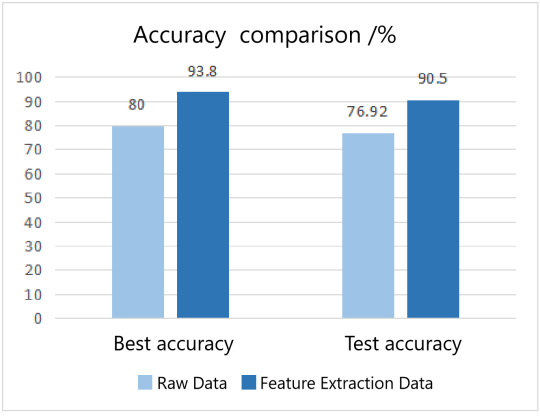


FIGURE 11. The results of comparative test 2: Raw Data vs. Feature Extraction Data.

As seen in Fig. 11, compared with the raw data as the model input, the prediction accuracy of the model using the data after feature extraction is much better. When the raw data are directly used as the model input, the test accuracy is 76.92%, and the best accuracy is only 80%. When using the data processing method proposed in this paper, the average accuracy is 90.5%, and the optimal accuracy is 93.8%. Although a CNN can offer good functionality for processing image features, if we first perform special data processing on the sound signal before inputting the data into the model, this will greatly help improve the prediction accuracy.

c.SVM vs. Siamese network

We use the previously used SVM and the Siamese network used in this paper to predict the accuracy of the same target domain data. Among them, SVM1 has only source domain data for training, and SVM2 has active domain data and 10 target domain data for training; the Siamese network training data are consistent with the SVM2 training data.

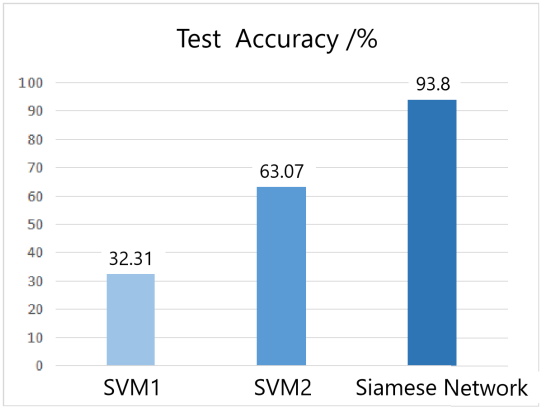


FIGURE 12. The results of comparative test 3: SVM vs. Siamese Network.

As shown in Fig. 12, when processing the same target domain data with a small amount of training data, the optimal accuracy of the Siamese network is 93.8%, which is more powerful than the SVM. In a preliminary study using only the scheduled training data, the different air blowing actions were easily identified and the average accuracy rate reached a good result, indicating that the features we extracted and the depth model we chose were wellselected.

V. USER STUDY

To assess the usability and effectiveness of our interaction method leveraging exhalation actions, we conducted a user study. The user study consisted of four parts: an algorithm performance test, a usability test, a contrast test and a user experience study.

A. PARTICIPANTS

A total of 16 student volunteers (11 females, 5 males) were enrolled to participate in the algorithm performance test, usability test and user experience study. The age of the sample ranged from 15 to 25 years (M = 17.94 years, SD = 3.83 years).[1] The number of volunteers involved in the contrast test will be described separately in "*D. CONTRAST TEST*". Before this test, all participants did not have any prior experience of blowing air as an interaction method.

B. ALGORITHM PERFORMANCE TEST

1) DESIGN

We adopted a within-subjects design for the algorithm performance test. To test the performance of our Siamese Network in dealing with device differences, we have developed two android applications for testing: one using the SVM model for identification and one using our Siamese network. We tested

[1]M refers to mean age. SD refers to the Standard Deviation, which is a measure of how spread out the age distribution is.

these two applications on two mobile phones (*Source Device A* and *Target Device B* mentioned in 4.1.1) to test the accuracy of interactions using different algorithms. To avoid the impact of interactive proficiency on the test results, the experience order of the devices and the algorithm are counterbalanced, that is, the test order of *Source Device A* and *Target Device B*, the SVM and Siamese Network are counterbalanced.

2) PROCEDURE

First, we explained to the participants the relevant interaction tasks relevant to the test. To finish these tasks, the participants needed to rely on the four different forms of blowing air as interactions. Then, the participants experienced the application in one test condition. We designed a practice module, and the participant practiced the four types of blowing following a simple instruction phase before starting the formal test. Then, the participant used the interaction method to finish some simple tasks in the designated test application.

The algorithm performance test was divided into four parts (two devices * two algorithms, named here Source_SVM / Source_SN / Target_SVM / Target_SN). Each part consisted of ten rounds and each round included 10 interaction tasks (two times for each type of the four blowing types and two times for participant speaking, ''Target'' and ''Source'' refer to whether the device is a source domain device or a target domain device. ''SVM'' refers to the test program implemented by SVM, and ''SN'' refers to the program implemented by the Siamese network). Among the 10 rounds, 5 rounds simply performed 10 interaction tasks each round to record the accuracy. The other 5 rounds required the participants to perform 10 correct interactions each round and record the time taken. To avoid the participants from developing a certain regularity, the sequence of six breathing tests in each group was random.

3) MEASURES AND RESULTS

The interaction accuracy was used to evaluate different algorithms in dealing with the interaction recognition problem with differences. At the same time, for recording accuracy, we also recorded the interaction time. From the interaction response time, we evaluated the feasibility of blowing interaction as a multitouch and voice-assisted interaction. The results of the algorithm test for the ten participants are given in Fig. 13 and Fig. 14.

From the test results of the algorithm test, we can observe that the accuracy of the Siamese network algorithm (91.33% and 94%) is higher than that of the SVM classifier (85.67% and 53.33%), both in the identification of the source domain device and the target domain device. By analyzing the variance in Fig. 14, the Siamese network is more stable than the SVM in interaction accuracy and response time. Therefore, it can be concluded that the Siamese network used in this paper can effectively cope with the problem of device differences. In terms of the time to complete the specified interactive instructions, the Siamese network is also faster and more stable than the SVM. It is important to note that in the Target_SVM part, participants cannot complete the interactive task within 120 seconds because the SVM algorithm cannot solve the problem of domain adaptation.

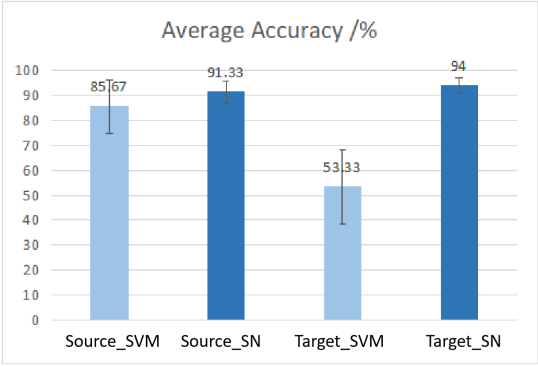


FIGURE 13. The average accuracy of participants in the algorithm test.

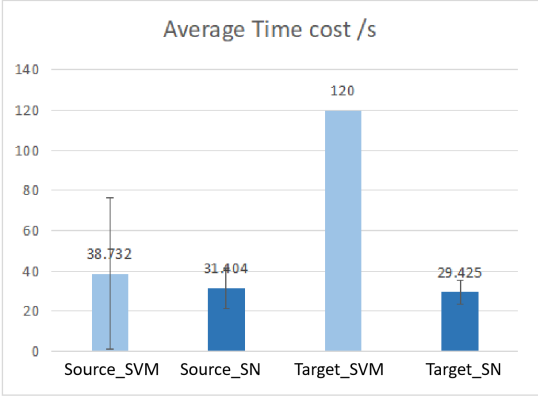


FIGURE 14. The average time of participants in the algorithm test.

C. USABILITY TEST

1) DESIGN

We adopted a within-subjects design for the usability test. The SN application and *Target Device B* introduced earlier in detail were used for this test. Since the input that we actually obtain is the sound wave resulting from any blowing of air, the test results might be affected by environmental noise. Therefore, to assess the usability of the interaction technology adequately, we tested the interaction accuracy under two conditions: a quiet environment and a noisy environment. The experience order was counterbalanced. The noisy environment condition is divided into indoor and outdoor to better simulate the actual use scene. In the noisy environment of the room, music was played, serving as noise for the test. The music was set to approximately 65 decibels, and the music source was less than 0.5 meters away from the participants. Additionally, 3 extra people were arranged around the participant and requested to chat casually, but they were not allowed to communicate with the main participants. For outdoor testing, we chose to conduct it next to the road with more vehicles. In the quiet environment condition, there was

no artificial noise in the lab, i.e., neither music nor chatting was allowed.

2) PROCEDURE

The usability test was divided into three parts: a quiet environment, an indoor noisy environment and an outdoor noisy environment. Each part consisted of 10 rounds. The interaction tasks of each round were the same as those in the algorithm test. The participants experienced the SN application using *Target Device B* in three predetermined environmental conditions. The accuracy and time of the actual interaction were recorded.

3) MEASURES AND RESULTS

Similar to the algorithm test, the interaction accuracy was used to evaluate the usability of this interaction technology in different environments. At the same time, for recording accuracy, we also recorded the interaction time.

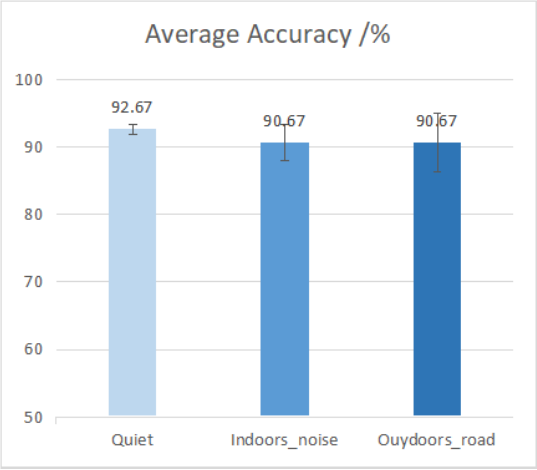


FIGURE 15. The average accuracy of participants in the usability test.

From the usability test results(see Fig. 15), we can observe that the average accuracy (92.67% in quiet environment) in the usability test is basically consistent with the accuracy (93.8%) of the training verification in Chapter 4. A recent study used breathing as a direct control interface in two VR games [38]. They proposed four intuitive active breathing control actions corresponding to different special effects in games, and the average recognition accuracy in that study was 88.3% with two authors. By comparison, our best accuracy is 93.8% in validation and 91.34% in the usability test, which is still comparable to previous research that required custom hardware. It can be seen from Fig. 16 that it takes approximately 30 seconds for the participants to perform 10 correct blowing interaction instructions. Moreover, compared with the quiet environment, the accuracy rate and average time spent did not decrease considerably in the noisy environment, showing that our technology can cope well with potential interference stemming from background noise.

As far as response time is concerned, this blowing interaction is acceptable as an important auxiliary form

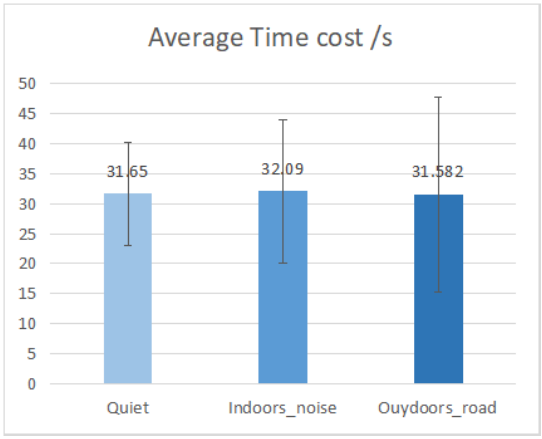


FIGURE 16. The average time of participants in the usability test.

of interaction. The algorithm technology in this paper has certain universality when dealing with different devices used by different users. In terms of accuracy, response time and usage scenarios, the interaction technique proposed in this study is deemed acceptable.

D. CONTRAST TEST

1) DESIGN

Unlike the other tests in this paper, 8 volunteers (3 females, 5 males, aged between 16 and 49 years old (M = 22.625 years, SD = 10.83 years)) were enrolled to participate in this test. We choose to compare our method with the method in [16] (calling it the time counter method). The reproduction principle of the time counter method is as follows: when the volume is higher than the threshold, the system starts the time counter until the volume is lower than the threshold, and the time counter will be checked to determine the type of blowing(*single short puff*, *double short puff* and *long puff*). These three types correspond to *gust*, *two gust* and *gale*.

The contrast test mainly tests the blowing interactive accuracy, not only to test the impact of device differences, but also to test the impact of speech interference. To test the differences, we first implement the time counter method in Unity3D on the PC (source domain device); the same program is packaged in. apk format and ported to the mobile phone (target domain device) for testing. To test the speech interference, when testing the time counter method, we will calculate the accuracy under the two conditions of normal use and prohibition of speech. In this way, we analyze the influence of speech interference on the time counter method. Because our method has already considered speech interference, the accuracy of normal use is directly calculated for comparison.

2) PROCEDURE

The test procedure of our method refers to the previous methods.

The test procedure of the time counter method: The participants conducted 10 rounds of tests on the PC and mobile phone respectively to record the accuracy. To test the performance of the time counter method in dealing with speech interference, each round has 8 interactive tasks (two times for each type of the three blowing types and two times for participant speaking). To avoid the participants from developing a certain regularity, the sequence of six breathing tests in each group was random.

3) MEASURES AND RESULTS

As shown in Fig. 17, the accuracy of our method based on the Siamese network in identifying source devices and target devices (91.67% and 92.8%) is higher than that of the time counter method (75% and 60.52%). When used on the target domain device, the time counter method cannot handle the problem of device differences, so the accuracy is diminished. However, our adaptive method can better solve the problem of device differences. The accuracy of the time counter method to prohibit speech is 80.41%, which is higher than the 75% accuracy under normal conditions, indicating that the time counter method cannot deal with speech interference well. The results also illustrate that our adaptive method has obvious advantages in dealing with device differences and noise interference, such as speaking.

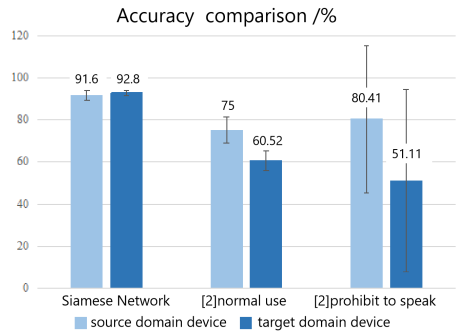


FIGURE 17. The average accuracy of participants in the contrast test.

E. USER EXPERIENCE STUDY

1) DESIGN

To avoid participants becoming fatigued during the test, each participant chooses one of the algorithm performance tests and usability tests and then experiences the user study. In this part, we use the three applications introduced in Section VI - a video player application on a PC, Amap on a mobile device and the a VR game application - to explore the participant's experience of using the interaction method in three different applications. The participant experienced the three applications in a certain order. After the participant experienced the applications, we conducted a follow-up interview.

2) PROCEDURE

After completion of the previous test, the participants continued to participate in the application experience test. Then, the participants experienced the video player application on a PC, Amap on a mobile device and a VR game application. The participants were able to take breaks during the test. After experiencing the three applications, we requested a follow-up interview with the participants.

3) MEASURES AND RESULTS

A structured interview was conducted to understand the participants' experiences and suggestions regarding this kind of interaction. The interview included four structural elements: (1) interest; (2) technology acceptance; (3) applicability; and (4) generalization. Moreover, we added an open topic item regarding blowing air as a natural and directly controllable interaction method.

a: INTEREST

Through experience with the web application and VR game application, the participants felt that this interaction method was very interesting, and most of them remarked how they had never tried to control applications by way of blowing air before, e.g., "*I have interacted with the mouse, touch, and voice, but not breath, and it was funny*".

b: TECHNOLOGY ACCEPTANCE

Participants agreed that this was a convenient way to interact, i.e., that it is usable and easy to use: "*You can release your hands, and this advantage is important when your hands are not available*". "*For example, when you're learning to cook by following a video, you may be busy preparing ingredients with both hands or your both hands are covered with oil. Here, blowing is a useful way to control the cell phone or other video-playing devices*". "*It's very easy to use, just move your mouth and blow out*".

c: APPLICABILITY

According to the interview results, the potential of this interaction method is very wide. Although it cannot be used a mainstream form of interaction, it may solve problems under certain special circumstances for people with particular needs. "*In some special dangerous situations or in special emergencies, such as when a hostage needs to call for help, he can't shout, call or send messages. By using this interaction, he can send a secret signal for help, which is not easily detected by criminals*". "*Deaf-mutes are a potential user group. They may not be able to speak, but everyone can blow air and breathe*".

d: GENERALIZATION

Although it cannot be regarded as a primary means of interaction, it can be used as an auxiliary interaction modality. In some cases, it may replace touching and other interaction forms. "*In some cases it can replace voice and touch, for example, when both hands are inconvenient to use, you can blow to answer the phone*".

This method of interaction can be generalized to special groups. "*This technology can be extended to deaf-mutes, helping them communicate with people by combining the interaction method with digital dictionary in some way*".

e: DISADVANTAGES

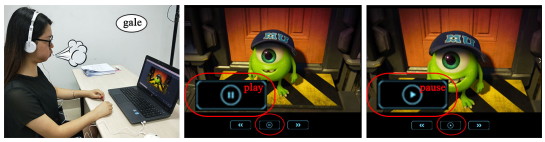
The participants also mentioned some shortcomings of this interaction modality. First, the number of interaction types that can be achieved is limited. "*One disadvantage is that this way cannot achieve too much control, otherwise it is easy to confuse. Unlike touch and voice, which can achieve many kinds of control*". Second, it easily leads to fatigue. "*Although it is an interesting interaction method, the user will be tired after using many times*".

VI. APPLICATION DESIGN

Our novel blowing-based interface can be applied in a range of different applications.

A. PLAYING VIDEO ON A PC

The client relies on Unity 3D (version 5.6.0) as the platform for showing video. It obtains the exhalation sound waves in real time and monitors which blowing operation the user has blown in realtime. When monitoring any blowing actions, it performs operations corresponding to the identification and classification result: *gale* controls the play vs. pause state of the video playing interface, *gust* is invoked to transition to playing the next video, while *breeze* is used to return to the previous video (cf. Fig. 18).



(a) *gale* controls the play or pause action

(b) *gust* for the next video

(c) *breeze* for the previous video

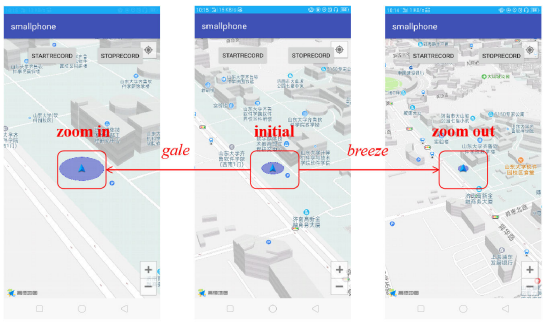
FIGURE 18. Blowing actions and corresponding effects on a PC.

In Fig. 18(a), the participant pauses while watching the motion picture Monsters University, via a *gale* operation, and the Play button turns into a Pause button. In Fig. 18(b), the participant moving to the next video, the motion picture Flipped, using the *gust* operation, the Next button turning blue. Fig. 18(c) shows that the participant navigates to the previous video, the motion picture Paddington 2, using the *breeze* action, the previous button turning blue.

B. MANIPULATING AMAP ON A MOBILE PHONE

On the mobile phone, we created a simple prototype of our blowing interactive interface on Amap, which can be used in driving situations. Specifically, the action of enlarging the map, as can be done via ZoomIn, is mapped to the gale action (see Fig. 19(a)), while shrinking the map, as can be achieved with ZoomOut, is mapped to the breeze action (see Fig. 19(b)).

Fig. 19 shows the process of using blowing actions to manipulate the map. In particular, Fig. 19(b) shows the initial state, Fig. 19(a) shows the ZoomIn result in the Amap using gale action, and Fig. 19(c) shows the ZoomOut result in the Amap using breeze action.

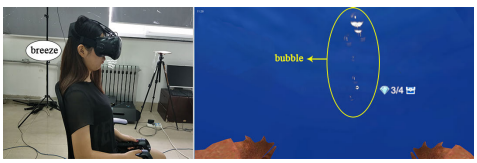


(a) (b) (c)

FIGURE 19. Blowing actions and corresponding effects on Amap.

(a) *gale*-water jet

(b) *gust*-blister



(c) *breeze*-bubble

FIGURE 20. Blowing actions and corresponding effects in a VR game.

C. PLAYING A VR GAME

We integrate the blowing actions into a VR game called Undersea Treasure Hunt. It is developed using Unity 3D (version 5.6.0) and played using an HTC VIVE headset. In this game, we define different game effects for every kind of blowing action. The three game effects are associated with blowing actions (i.e., *gale*, *gust*, *breeze*) and are also associated with certain phenomena in the real world. *Gale* sprays a water jet (cf. Fig. 20(a)), *Gust* will open a treasure box

(cf. Fig. 20(b)), and *Breeze* triggers the bubbling operation of the crab (cf. Fig. 20(c)). These correspondences can further impress upon the users the reasonableness of cause and effect within the VR environment.

VII. CONCLUSION

This paper presented a simple method to implement a natural and directly controllable interaction based on blowing out air. We designed and implemented four blowing actions and three applications based on them. We implemented this by relying on headset microphones to record the audio signals and then classify them to distinguish different blowing categories.

We used a deep learning model, the Siamese network to address domain adaptations to improve the robustness of this blowing interaction. The test results show that individual differences, speaking noise and environmental noise had little influence on the interaction and that this interaction modality is a good way to improve user interest and experience in applications, specifically in VR applications. Moreover, it is highly available in a multitude of environments and particularly usable in special environments (e.g., noise, unavailability of hands) or for special groups (e.g.,deaf-mutes).

REFERENCES

[1] I. Alakärppä, E. Jaakkola, A. Colley, and J. Häkkilä, "BreathScreen: Design and evaluation of an ephemeral UI," in *Proc. CHI*, 2017, pp. 4424–4429.

[2] K. L. Amon and A. Campbell, "Can children with ad/hd learn relaxation and breathing techniques through biofeedback video games?" *Austral. J. Edu. Develop. Psychol.*, vol. 8, pp. 72–84, Jan. 2008.

[3] Y. Ban, H. Karasawa, R. Fukui, and S. I. Warisawa, "Relaxushion: Controlling the rhythm of breathing for relaxation by overwriting somatic sensation," in *Proc. SIGGRAPH Asia Emerg. Technol. (SA)*, New York, NY, USA, 2018, doi: 10.1145/3275476.3275492.

[4] R. Chattopadhyay, N. C. Krishnan, and S. Panchanathan, "Hierarchical domain adaptation for SEMG signal classification across multiple subjects," in *Proc. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2011, pp. 7853–7856.

[5] R. Chattopadhyay, N. C. Krishnan, and S. Panchanathan, "Topology preserving domain adaptation for addressing subject based variability in SEMG signal," Stanford, CA, USA, Tech. Rep. SS-11-04, Mar. 2011.

[6] Q. Chen, J. Huang, R. S. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5315–5324, doi: 10.1109/CVPR.2015.7299169.

[7] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR05)*, San Diego, CA, USA, pp. 539–546, doi: 10.1109/CVPR.2005.202.

[8] S. Dar, V. Lush, and U. Bernardet, "The virtual human breathing relaxation system," in *Proc. 5th Exp. Int. Conf. (exp.at)*, Jun. 2019, pp. 276–277, doi: 10.1109/EXPAT.2019.8876478.

[9] M. K. Delimayanti, B. Purnama, N. G. Nguyen, K. R. Mahmudah, M. Kubo, M. Kakikawa, Y. Yamada, and K. Satou, "Clustering and classification of breathing activities by depth image from Kinect," in *Proc. 12th Int. Joint Conf. Biomed. Eng. Syst. Technol. (BIOSTEC)*, Prague, Czech Republic, vol. 3, E. D. Maria, A. L. N. Fred, and H. Gamboa, Eds. SciTePress, pp. 264–269, doi: 10.5220/0007567502640269.

[10] T. Emoto, U. R. Abeyratne, K. Kawano, T. Okada, O. Jinnouchi, and I. Kawata, "Detection of sleep breathing sound based on artificial neural network analysis," *Biomed. Signal Process. Control*, vol. 41, pp. 81–89, Mar. 2018, doi: 10.1016/j.bspc.2017.11.005.

[11] M. Fukumoto, "Silentvoice: Unnoticeable voice input by ingressive speech," in *Proc. 31st Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, Berlin, Germany, P. Baudisch, A. Schmidt, and A. Wilson, Eds., 2018, pp. 237–246, doi: 10.1145/3242587.3242603.

[12] M. Gaiduk, R. Seepold, T. Penzel, J. A. Ortega, M. Glos, and N. M. Madrid, "Recognition of Sleep/Wake states analyzing heart rate, breathing and movement signals," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 5712–5715, doi: 10.1109/EMBC.2019.8857596.

[13] E. E. Hamke, M. Martínez-Ramón, A. R. Nafchi, and R. Jordan, "Detecting breathing rates and depth of breath using LPCs and restricted Boltzmann machines," *Biomed. Signal Process. Control*, vol. 48, pp. 1–11, Feb. 2019, doi: 10.1016/j.bspc.2018.09.009.

[14] K. Höök, M. P. Jonsson, A. Stahl, and J. Mercurio, "Somaesthetic appreciation design," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2016, pp. 3131–3142.

[15] T. Igarashi and J. F. Hughes, "Voice as sound: Using non-verbal voice input for interactive control," in *Proc. 14th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, New York, NY, USA, 2001, pp. 155–156, doi: 10.1145/502348.502372.

[16] J. Feijó Filho, W. Prata, and T. Valle, "Breath mobile: A low-cost software-based breathing controlled mobile phone interface," in *Proc. 14th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services Companion (MobileHCI)*, San Francisco, CA, USA, 2012, pp. 157–160.

[17] C. Jácome, J. Ravn, E. Holsbø, J. C. Aviles-Solis, H. Melbye, and L. A. Bongo, "Convolutional neural network for breathing phase detection in lung sounds," *Sensors*, vol. 19, no. 8, p. 1798, Apr. 2019, doi: 10.3390/s19081798.

[18] D. Jain, M. Sra, J. Guo, R. Marques, R. Wu, J. Chiu, and C. Schmandt, "Immersive terrestrial scuba diving using virtual reality," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst. (CHI EA)*, New York, NY, USA, 2016, pp. 1563–1569, doi: 10.1145/2851581.2892503.

[19] M. M. Javaid, M. A. Yousaf, Q. Z. Sheikh, M. M. Awais, S. Saleem, and M. Khalid, "Real-time eeg-based human emotion recognition," in *Neural Information Processing*, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds. Cham: Springer, 2015, pp. 182–190.

[20] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7139–7148, doi: 10.1109/CVPR.2017.755.

[21] B. G. V. Kumar, G. Carneiro, and I. D. Reid, "Learning local image descriptors with deep Siamese and triplet convolutional networks by minimising global loss functions," 2016, pp. 5385–5394, doi: 10.1109/CVPR.2016.581.

[22] K. Kuzume, "Input device for disabled persons using expiration and tooth-touch sound signals," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1159–1164.

[23] X. Lu, D. Guiraud, S. Renaux, T. Similowski, and C. Azevedo, "Breathing detection from tracheal sounds in both temporal and frequency domains in the context of phrenic nerve stimulation," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 5473–5476, doi: 10.1109/EMBC.2019.8856440.

[24] J. Marshall, D. Rowland, S. Rennick Egglestone, S. Benford, B. Walker, and D. McAuley, "Breath control of amusement rides," in *Proc. Annu. Conf. Hum. Factors Comput. Syst. (CHI)*, 2011, pp. 73–82.

[25] A. Misra, G. Essl, and M. Rohs, "Microphone as sensor in mobile phone performance," in *Proc. New Int. Musical Expression (NIME)*, Genova, Italy, 2008, pp. 185–188.

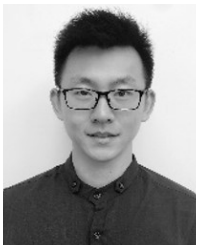
[26] S. Motiian and G. Doretto, "Information bottleneck domain adaptation with privileged information for visual recognition," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9911, Amsterdam, The Netherlands, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 630–647, doi: 10.1007/978-3-319-46478-7_39.

[27] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5716–5726, doi: 10.1109/ICCV.2017.609.

[28] L. E. Nacke, M. Kalyn, C. Lough, and R. L. Mandryk, "Biofeedback game design: Using direct and indirect physiological control to enhance game interaction," in *Proc. Annu. Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, 2011, pp. 103–112, doi: 10.1145/1978942.1978958.

[29] Y. Okuno, H. Kakuta, T. Takayama, and K. Asai, "Jellyfish party: Blowing soap bubbles in mixed reality space," in *Proc. 2nd IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Washington, DC, USA, 2001, pp. 358–359.
[30] S. N. Patel and G. D. Abowd, "BLUI: Low-cost localized blowable user interfaces," in *Proc. 20th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, Newport, RI, USA, 2007, pp. 217–220.
[31] R. Patibanda, F. Mueller, M. Leskovsek, and J. Duckworth, "Life tree: Understanding the design of breathing exercise games," in *Proc. Annu. Symp. Comput.-Hum. Interact. Play CHI PLAY*, Oct. 2017, pp. 19–31.
[32] D. Pettas, S. Nousias, E. I. Zacharaki, and K. Moustakas, "Recognition of breathing activity and medication adherence using LSTM neural networks," in *Proc. IEEE 19th Int. Conf. Bioinf. Bioeng. (BIBE)*, Athens, Greece, Oct. 2019, pp. 941–946, doi: 10.1109/BIBE.2019.00176.
[33] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019, doi: 10.1109/TPAMI.2018.2814042.
[34] E. Sawada, S. Ida, T. Awaji, K. Morishita, T. Aruga, R. Takeichi, T. Fujii, H. Kimura, T. Nakamura, M. Furukawa, N. Shimizu, T. Tokiwa, H. Nii, M. Sugimoto, and M. Inami, "BYU-BYU-view: A wind communication interface," in *Proc. ACM SIGGRAPH Emerg. Technol. (SIGGRAPH)*, 2007, p. 1-es.
[35] H. Schnädelbach, K. Glover, and A. A. Irune, "ExoBuilding: Breathing life into architecture," in *Proc. 6th Nordic Conf. Hum.-Comput. Interact. Extending Boundaries (NordiCHI)*, Reykjavik, Iceland, 2010, pp. 442–451.
[36] H. Schnädelbach, A. A. Irune, D. S. Kirk, K. Glover, and P. Brundell, "Exobuilding: Physiologically driven adaptive architecture," *ACM Trans. Comput.-Hum. Interact.*, vol. 19, no. 4, pp. 25:1–25:22, 2012.
[37] T. Sonne and M. M. Jensen, "ChillFish: A respiration game for children with ADHD," in *Proc. 10th Int. Conf. Tangible, Embedded, Embodied Interact. (TEI)*, 2016, pp. 271–278.
[38] M. Sra, X. Xu, and P. Maes, "BreathVR: Leveraging breathing as a directly controlled interface for virtual reality games," in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, pp. 340:1–340:12, doi: 10.1145/3173574.3173914.
[39] P. Tennent, D. Rowland, J. Marshall, S. R. Egglestone, A. Harrison, Z. Jaime, B. Walker, and S. Benford, "Breathalising games: Understanding the potential of breath control in game interfaces," in *Proc. 8th Int. Conf. Adv. Comput. Entertainment Technol. (ACE)*, 2011, pp. 58:1–58:8.
[40] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4068–4076, doi: 10.1109/ICCV.2015.463.
[41] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, "A siamese long short-term memory architecture for human re-identification," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9911, Amsterdam, The Netherlands, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2011, pp. 135–153, doi: 10.1007/978-3-319-46478-7_9.
[42] G. Wang, "Designing Smule's Ocarina: The iPhone's magic flute," in *Proc. New Interfaces Musical Expression (NIME)*, Pittsburgh, PA, USA, 2009, pp. 303–307.
[43] G. Wang, "Ocarina: Designing the iPhone's magic flute," *Comput. Music J.*, vol. 38, no. 2, pp. 8–21, 2014, doi: 10.1162/COMJ_a_00236.
[44] Y. Yang, Q. Wu, M. Qiu, Y. Wang, and X. Chen, "Emotion recognition from multi-channel EEG through parallel convolutional recurrent neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–7, doi: 10.1109/IJCNN.2018.8489331.
[45] D. Zielasko, S. Freitag, and D. Rausch, "BlowClick: A non-verbal vocal input metaphor for clicking," in *Proc. ACM Symp. Spatial User Interact.*, Los Angeles, CA, USA, 2015, pp. 20–23.

YEQING CHEN was born in Shandong, China. She received the bachelor's degree in digital media technology from Shandong University, in 2017, where she is currently pursuing the master's degree in computer science and technology. She studies human–computer interaction and virtual reality.



YULONG BIAN was born in Binzhou, Shandong, China, in 1988. He received the Ph.D. degree in basic psychology, in 2016. He is currently an Associate Research Fellow of human-computer interaction with Shandong University, where he finished the Postdoctoral Research Project. His research interests include human–computer interaction, VR aided training, and user experience study.

WEI GAI was born in Shandong, China. She received the Ph.D. degree from the School of Computer Science and Technology, Shandong University, in 2017. She is currently a Postdoctoral Researcher with the School of Software, Shandong University. Her current research interests include human–computer interaction and virtual reality.

CHENGLEI YANG was born in Shandong, China, in 1972. He received the Ph.D. degree from Shandong University, in 2004.

He is currently a Professor with the School of Software, Shandong University. His research interests include human–computer interaction, virtual reality, and computational geometry.