Research Article

SonicGest: Ultrasonic Gesture Recognition System Combined with GAN on Smartphones

Xuebing Wang, Junbao Zhang*, and Mengxue Yang

School of Computer Science, Zhongyuan University of Technology, Zhengzhou 450007, China

Correspondence should be addressed to Junbao Zhang; junbao.zhang@zut.edu.cn

Received 4 March 2023; Revised 15 May 2023; Accepted 29 May 2023; Published 15 June 2023

Academic Editor: Lei Chu

Copyright © 2023 Xuebing Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, with the widespread popularity of mobile devices, gesture recognition as a way of human-computer interaction has received more and more attention. However, existing gesture recognition methods have their limitations, such as requiring additional hardware devices, invading user privacy, and causing difficulty in data collection. To address these issues, we propose SonicGest, a recognition system that utilizes acoustic signals to sense in-air gestures. The system only needs the built-in speaker and microphone of the smartphone, without any additional hardware and no privacy disclosure. SonicGest transforms the features of the acoustic Doppler effect caused by gesture movements into a spectrogram, uses spectrogram enhancement techniques to remove noise interference, and then builds a convolutional neural network (CNN) classification model to recognize different gestures. To solve the problem of data collection difficulties, we utilize the Wasserstein distance based on gradient penalty to optimize the loss function of the generative adversarial network (GAN) to generate high-quality spectrograms to expand the dataset. The experimental results show that SonicGest has a recognition accuracy of 98.9% for ten kinds of gestures.

1. Introduction

Nowadays, mobile terminals such as smartwatches and smartphones have been deeply integrated into people’s daily lives. Usually, people touch the screens with their fingers or a voice assistant to implement human-computer interaction (HCI). However, the screens of mobile terminals are usually small, which is not friendly for people with large fingers to interact with mobile terminals. Besides, using the mobile terminal with the user’s wet hands or with gloves on may lead to a lousy user experience. Although people can now overcome these difficulties by utilizing voice, they are affected by dialect or ambient noise, and it is inconvenient to control devices by voice in libraries, study rooms, and other places where silence is required. To solve the above problems, the in-air gesture recognition that enables HCI has gained a lot of attention.

In recent years, many researchers have made a lot of effort in the work of gesture recognition. These include wearable device-based [1], computer vision-based [2], radio frequency (RF) signal-based [3], and acoustic signal-based solutions [4]. However, work based on wearable devices requires users to wear special devices, which greatly reduces the convenience of operation and user experience. Computer vision-based solutions are susceptible to lighting conditions and have the risk of privacy leakage. RF-based signals not only require additional equipment but are also susceptible to device deployment locations. Existing acoustic-based solutions can be divided into active and passive sensing [5]. The passive sensing approach uses a microphone to record the sound generated by the user making gestures or writing numbers and characters on a rough desktop to recognize the gestures and writing. However, the passive sensing approach does not work properly in a loud-noise environment. The active sensing approach senses gestures by converting the mobile terminal into a small transceiver capable of transmitting and receiving ultrasonic signals, which has the advantage of being noise-immune. Although some smartphones (e.g., Huawei and Samsung) have in-air gesture recognition, their gesture recognition function does
not work in dark environments or when wearing dark clothes [6].

Since most smartphones are embedded with speakers and microphones, active sensing has the advantages of being unaffected by ambient light, nonintrusive to user privacy, and working well in a loud noise environment. Therefore, we propose an acoustic-based system, SonicGest, for gesture recognition. The system converts a smartphone into a small transceiver to receive and transmit ultrasonic signals using an embedded microphone and speaker, then it transforms the echo signal into a spectrogram containing Doppler effect features, and finally uses a convolutional neural network (CNN) classification model to extract the Doppler effect features caused by the user’s movements to recognize gestures. However, in the actual process, we will face the following challenges:

(1) Line-of-sight (LOS) signal transmission and device noise can interfere with the Doppler effect features caused by user gestures

(2) Training classification models require rich data, but collecting data is time-consuming and labour-intensive, so it is difficult to collect large amounts of data

To overcome these challenges, we use a series of steps to enhance the spectrograms generated by the echo signals to remove the interference and use a generative adversarial network (GAN) to generate data to overcome the difficulty of data collection. Meanwhile, we have done extensive experiments to prove that the generated data can effectively improve the gesture recognition accuracy of the system.

Specifically, the main contributions of this paper are as follows:

(1) We designed a network architecture consisting of convolutional and transposed convolutional layers as the generator and discriminator and modified the loss function of the GAN using a gradient-based penalty Wasserstein distance, which makes our model more stable in training. We also prove that the model can generate high-quality data to overcome the difficulties of data collection and that the enhancement method of generating data has advantages than traditional data enhancement methods by experiments

(2) We designed a gesture segmentation algorithm based on the spectrogram, which can detect the starting and ending points of each gesture and segment multiple consecutive gestures efficiently

(3) We propose a spectrogram enhancement technique. The technique can remove the interference of line-of-sight signal transmission and device noise to obtain a spectrogram that contains only Doppler shift features. In addition, we also designed a CNN framework based on residual blocks to recognize gestures

The rest of the paper is organized as follows: in Section 2, we introduce the related work. We introduce the implementation details of the system in Section 3. We validate the approach proposed in this paper in Section 4. Finally, we summarize the paper in Section 5.

2. Related Work

According to differences in the devices used and the signals acquired, existing gesture recognition works can be divided into the following four categories: wearable device-based, computer vision-based, RF-based, and acoustic-based.

2.1. Wearable Device-Based. There are some traditional gesture recognition works based on wearable devices, and such methods generally require some customized additional devices (e.g., bracelets, watches, gloves, and armbands with sensors). L-Sign [7] used a bracelet with built-in sensors to recognize Chinese universal sign language and achieved a 90% recognition rate. Serendicity [8] used the accelerometer and gyroscope embedded in the smartwatch to recognize five-finger gestures. WearSign [9] utilized a smartwatch and an EMG sensor armband with 8 channels to capture the user’s sign language gesture data, and then input the captured data into a neural network to recognize sign language actions. Fabriccio [10] recognized 11 touchless gestures and 1 touch gesture with 92.8% accuracy, which contains gestures with small movements such as thumb drawing circles and thumb swiping, but it requires the use of a pair of antennas sewn on a fabric substrate by a conductive thread to sense gesture movements. Piskozub et al. [11] design a data glove equipped with 10 voltage sensors to recognize 16 static gestures. However, wearing such bulky physical devices places an additional physical burden on the user, thereby degrading the user experience. Compared to such approaches that use wearable devices to recognize gestures, SonicGest can accommodate user-device interaction without wearing any physical device.

2.2. Computer Vision-Based. The method based on computer vision generally collects the image information of the gesture through the camera and then uses image processing technology to recognize the gesture. Wang et al. [12] used a Kinect depth camera to capture the user’s hands and extract their shapes to achieve recognition of 10 static gestures with 99% accuracy. Sulyman et al. [13] recognized gestures by a computer camera, then converted the computer-stored RGB images to YCbCr color space, set a threshold to separate the gesture images from the background images, and finally, for five static gestures, the recognition accuracy is 98%. The advantage of the vision-based gesture recognition approach is that it can take advantage of the powerful characterization ability of CNN for image data and input the image or video data of gestures into CNN so that it can obtain high recognition accuracy. However, the disadvantages of such vision-based approaches are also significant, as they are easily affected by cluttered backgrounds and a lack of illumination [14], and the recognition process requires the user to turn on the camera, which poses the risk of user privacy leakage.
2.3. RF-Based. The RF signal recognition method has received a lot of attention because it is not affected by ambient light, does not require the wearing of physical devices, and enables contactless HCI. AirDraw [15] used three commodity WiFi devices to build a two-dimensional geometric model, implemented tracking of user gestures using channel state information (CSI), and experimentally showed that the median error of tracking user gestures was less than 2.2 cm. WiG [16] collected CSI based on commercial shelf-product WiFi devices only and then used support vector machines to recognize the four interaction gestures with 91% accuracy. RFree-GR [17] recognized 16 commonly used American gestures.
Sign Language gestures by an RFID tag array and used an adversarial model to remove domain-specific information to overcome the effects of a new environment, and the experimental results showed that RFree-GR had a recognition accuracy of 88.38% at a new location. Although the recognition method based on RF signals can achieve contactless sensing of gestures, this method requires the deployment of additional transceiver devices, and the location of the transceiver cannot be changed at will.

2.4. Acoustic-Based. Compared with the previous gesture recognition methods, the use of acoustic sensing gestures does not infringe on the privacy of users, and existing commercial devices, such as smartphones and smartwatches, are embedded with microphones and speakers without the need to deploy additional devices. Therefore, gesture recognition and hand motion tracking based on acoustic waves have received extensive attention. Ipanel [18] transformed the sound signal features generated by sliding fingers on the table surface into images and then used a CNN classification model to recognize gestures. WordRecorder [19] first extracted the sound signal generated by the pen rubbing on the paper to recognize English letters and then used a word segmentation algorithm to segment the words to finally realize the recognition of English words. SoundWave [20] used the loudspeaker on the notebook to send ultrasonic signals and used the microphone to capture the Doppler effect features generated by the user’s gestures to recognize 5 predefined gestures.

Cai et al. [21] used wearable A-mode ultrasonic transducers to get signals for dynamic gestures and achieved the recognition of five dynamic gestures. AudioGest [22] extracted the Doppler effect features generated by gesture movements and recognized 6 gestures with 96% accuracy. UltraGesture [23] recognized 12 gestures with 97% accuracy using the channel impulse response, but the system requires additional sensors.

In hand motion tracking, LLAP [24] tracked the user’s movement with an accuracy of 3.5 mm by measuring the phase information of the ultrasound signal reflected from the user’s finger or palm by existing commercial devices. Strata [25] tracked the hand motion using channel impulse response (CIR) features with an error of 1.0 cm. FingerIO [26] used the orthogonal frequency division multiplexing (OFDM) modulation technique to achieve 2D tracking of the finger with an average accuracy of 8 mm.

Although the abovementioned passive gesture sensing approach using acoustic waves can recognize gestures of written numbers or letters, this type of approach does not work in high-noise environments. The active perception method based on ultrasonic signals, on the other hand, can work in high-noise environments by separating noise interference through filters. SonicGest uses an active sensing method that not only overcomes noise interference but also generates fake data through GAN to avoid the high cost of collecting data and improve the accuracy of gesture recognition.

\[
\begin{align*}
1. & \textbf{Input:} T_1, T_2, T_3, \text{fame}; y_i \in \mathbb{R}^N, i = 1, 2, 3 \cdots, n. \\
2. & \textbf{Output:} \text{List of start and end points.} \\
3. & \textbf{Define} \text{sum()} \text{to calculate the sum.} \\
4. & \textbf{Define} \text{power()} \text{to calculate fame power} \\
5. & \text{for } i = 1: n \text{ do} \\
6. & \quad \text{if flag == False then} \\
7. & \quad \quad \text{if sum(power(y_i)) > T2) >= T1 then} \\
8. & \quad \quad \quad \text{Add the first possible start frame} \\
9. & \quad \quad \text{startCount} \leftarrow \text{startCount} + 1 \\
10. & \quad \quad \text{else} \\
11. & \quad \quad \quad \text{startCount} \leftarrow 0 \\
12. & \quad \quad \text{end if} \\
13. & \quad \quad \text{if startCount >= T3 then} \\
14. & \quad \quad \quad \text{flag} \leftarrow \text{True} \\
15. & \quad \quad \text{end if} \\
16. & \quad \text{else if flag == True then} \\
17. & \quad \quad \text{if sum(power(y_i)) > T2) < T1 then} \\
18. & \quad \quad \quad \text{Add the first possible end frame} \\
19. & \quad \quad \text{endCount} \leftarrow \text{endCount} + 1 \\
20. & \quad \quad \text{else} \\
21. & \quad \quad \quad \text{endCount} \leftarrow 0 \\
22. & \quad \quad \text{end if} \\
23. & \quad \quad \text{if endCount >= T3 then} \\
24. & \quad \quad \quad \text{Add the first possible end frame and start frame to the start-end list} \\
25. & \quad \quad \text{end if} \\
26. & \quad \text{end if} \\
27. & \text{end for}
\end{align*}
\]

\textbf{Algorithm 1:} Start and end point detection.
3. System Design

3.1. System Overview. The overall design of SonicGest is shown in Figure 1. SonicGest is mainly composed of five parts: transceiver, data preprocessing, spectrogram enhancement, data augmentation, and gesture recognition. First, the smartphone speaker is used to send ultrasonic signals, and the microphone is used to receive the echoes. In the preprocessing module, we use a band-pass filter to remove low-frequency noise and transform the denoised signal into a spectrogram using the short-time Fourier transform (STFT). After that, we detect the starting and ending points of the valid gestures in the spectrogram to segment the valid gesture features. After obtaining the valid gesture features, we use the spectrogram enhancement technique to remove the interference of signal transmission along the line-of-sight path and device noise. Finally, we use the fake data generated by GAN and the enhanced spectrogram as data to train CNN for gesture recognition.

3.2. Transceiver Design. To estimate the Doppler effect caused by the gesture motion, we choose a sine wave as the waveform at the transmitting end; this waveform can be used to estimate the Doppler effect and is less computationally intensive [27]. Since most of the hearing range of the human ear is below 18 kHz, we need to generate ultrasonic signals with frequencies of 18 kHz or higher to avoid disturbing people’s normal activities.

In this paper, we utilize the Doppler effect features caused by the gesture motion sensed by the microphone and speaker embedded in the smartphone to recognize the gesture. Thus, the smartphone can be approximated as a transceiver device with the receiver and transmitter bound in one piece [28]. As the speed of sound is much greater than the speed of hand movement, the amount of frequency shift caused by hand movement calculation is as follows:

$$\Delta f = \left( \pm \frac{2v_h}{c + v_h} \right) \times f_s \approx \left( \pm \frac{2v_h}{c} \right) \times f_s, \quad (1)$$

Figure 3: Spectrogram enhancement. (a) Raw spectrogram. (b) Spectrogram before setting a threshold. (c) Spectrogram after enhancement.
where \( c \) is the speed of sound, \( v_h \) is the speed of hand movement, and \( f_t \) is the frequency of the transmitted signal.

According to formula (1), theoretically, the higher the frequency of the transmitted signal, the greater the change in the received frequency shift. To better extract features, the transmission frequency should be as large as possible. However, the maximum transmitting frequency of existing smartphones is slightly higher than 20 kHz [29], and the energy of acoustic signals higher than 20 kHz decays rapidly. For the above reasons, we choose to send a 20 kHz sine wave at the transmitter end. Finally, we save the generated sine wave on the smartphone and transmit this sine wave using a speaker of the smartphone. At the same time, the microphone records the echoes reflected by the gesture. At the receiving end, a sampling rate of 44.1 kHz is used to sample the received signal, which is supported by most smartphones. In this way, the smartphone is transformed into a small transceiver that senses gestural movements.

3.3. Data Preprocessing

3.3.1. Remove out of Band Noise. After receiving the echo signal in the way described in Section 3.2, we need to filter out the background noise generated by people’s speech, activity, etc. to reduce the subsequent computational burden as well as to highlight the Doppler effect features caused by gesture. To achieve the above goal, we first estimate the range of frequency shift caused by the hand motion at around 20 kHz. Since the maximum speed of the hand motion is 4 m/s [20], according to equation (1), the maximum frequency shift caused by the hand motion is about 470 Hz, but in the experiment, we found that the frequency shift caused by the hand motion does not exceed 300 Hz. Therefore, to further reduce the computational complexity, we set the valid frequency shift range to 19700-20300 Hz and filter out the frequencies outside the valid band using a 6th-order Butterworth band-pass filter. From Figure 2(b), we can see that the noise concentrated below 10 kHz has been filtered.

3.3.2. Gesture Segmentation. After removing the noise, we need to segment the signal fragments that contain the gesture motion. Compared to previous work that segmented only a single gesture [30], our segmentation algorithm can segment multiple gestures. Specifically, we first frame the echo signal by STFT and convert it into a spectrogram. For better frequency domain resolution, we set the frame length and step size to 8192 points and 1024 points, respectively.

After that, we detect the different frequency bins on each frame of the spectrogram. When the total of frequency bins over T1 has a power value over T2, the frame is a valid gesture frame; otherwise, it is an invalid gesture frame. To filter out the frequency shifts generated by random motions, we empirically set T1 and T2 to 5 and -10, respectively. As the frequency shifts caused between different gestures are intermittent, we can check whether there is a threshold number of consecutive valid gesture frames (T3) to determine the starting and ending points of each gesture. We can further segment the gestures by determining the starting and ending points of valid gestures. Due to the minimum time to execute the gesture being about 0.5 s [23] (about 3 frames), we set T3 to 3. The details are described in Algorithm 1.

Figure 2(c) shows the segmentation results for the four gestures of finger clockwise rotation, finger anticlockwise rotation, palm down press, and palm up raise, where the
solid line indicates the starting point of the gesture and the dotted line is the end point.

3.4. Spectrogram Enhancement. Since the speaker is omnidirectional, the received echo signal encounters interference from the signal propagating along the line-of-sight path. In addition, we can also see in Figure 3(a) the noise generated by equipment defects.

Previous methods for removing the abovementioned noise required periodic activation of the microphone to record ambient reflections even when no one was in motion [31], resulting in extra power consumption from the smartphone. However, our method eliminates the need for periodic activation of the microphone, thereby reducing energy consumption.

Specifically, we first use a band-stop filter to remove the 19980-20020 Hz frequency bin. Since the spectrogram can be considered an image formed by a two-dimensional matrix, a 3×3 median filter can be used to remove the random noise in the image while smoothing the image. To avoid the effect of absolute signal intensity due to volume, we normalize the spectrogram matrix by

\[ P_N = \frac{P - \text{min} \,(P)}{\text{max} \,(P) - \text{min} \,(P)}, \]

where \( P_N \) is the result of normalization, \( P \) is the spectrogram matrix, and \( \text{max} \,(P) \) and \( \text{min} \,(P) \) are the maximum and minimum values of the spectrogram matrix, respectively.

After that, we calculate the spectrogram matrix squared to amplify the difference between noise and features and set an empirical threshold \( N_t \) (set to 0.4) to eliminate the sporadic noise, yielding:

\[ P'_N = (P_N)^2, \]

\[ P'_N(f, t) = \begin{cases} P'_N(f, t) & \text{if } P'_N(f, t) \geq N_t, \\ 0 & \text{if } P'_N(f, t) < N_t, \end{cases} \]

where \( P'_N(f, t) \) is the power at frequency \( f \) and time \( t \) on the spectrogram matrix. Finally, we use a two-dimensional Gaussian filter to smooth the image. Its template coefficient can be calculated as follows:

\[ f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \]

where \( x \) is the distance from the origin in the horizontal axis, \( y \) is the distance from the origin in the vertical axis, and \( \sigma \) is the standard deviation of the Gaussian distribution. The enhanced result is shown in Figure 3(c).

3.5. Data Augmentation. Previous work used only traditional data augmentation methods based on affine transformations, such as rotation, panning, and scaling, to expand the dataset. Although such methods can improve the generalization ability of the model to a certain extent, but for data with strict spatial structure or temporal structure limitation, we need to choose the enhancement methods carefully; otherwise, it may destroy the data features and reduce the accuracy of the model. In recent years, generative adversarial network (GAN) [32] has shown powerful capabilities in the field of data generation. GAN-generated data is more diverse and can be applied to various types of data generation.

As shown in Figure 4, GAN is generally composed of two network architectures: a generator \( G \) and a discriminator \( D \), where the generator generates fake data from the input noise \( z \) to deceive the discriminator, and the discriminator recognizes the fake data generated by the generator and the real data. The goals of generators and discriminators are contradictory. The discriminator needs to learn toward less loss to recognize false data and real data, while the generator needs to generate data that is more like the real data to trick the discriminator. This process can be judged using a loss function, as shown in the following:

\[ \min \max \mathbb{E}_{D,G} \mathbb{E}_{x \sim P_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim P_{z}(z)} \left[ \log (1 - D(G(z))) \right], \]

where \( x \) is the real data input to the discriminator, \( P_{data}(x) \) is the distribution of the real data, \( z \) is the noise vector obeying
Gaussian or uniform distribution, and $P_z(z)$ is the distribution of the noise vector $z$.

Although GAN can generate diverse data, the loss function of GAN essentially uses JSD (Jensen-Shannon divergence) to measure the distance between the true data distribution and the false data distribution. However, JSD is not a good measure of the distance between the two distributions when there is little or no overlap between the real data distribution and the fake data distribution. In this case, optimizing the loss function in equation (5) can easily lead to an unstable training process, which can cause problems such as gradient disappearance, gradient explosion, and pattern collapse. DCGAN [33] uses a convolutional layer and a transposed convolutional layer as the basic structure of the discriminator and generator, respectively, and uses a convolutional kernel to extract image features for data generation, but it does not fundamentally solve the problem of training instability and requires a careful balance between generator and discriminator during training.

To overcome the limitation of JSD to measure two distributions and to solve the training instability problem of GAN, we utilize the Wasserstein distance based on the gradient penalty proposed by Gulrajani et al. [34] to modify the loss function of the GAN. The modified loss function is shown in the following:

$$L = E_{x \sim P_{data}(x)}[D(x)] - E_{z \sim P_z(z)}[D(z)] + \lambda \ E_{\tilde{x} \sim P_{\tilde{x}}(\tilde{x})} \left( \| \nabla_{\tilde{x}} D(\tilde{x}) \|_2 - 1 \right)^2,$$

(6)
where \( \tilde{x} = G(z) \) denotes the generated data, \( \tilde{x} \) is the uniform sampling on the line between the real and generated data sampling points, \( P_z \) is the distribution of \( \tilde{x} \), and \( \lambda \) is the penalty coefficient. We optimize equation (6) to make the generator and discriminator confront each other for extracting the essential features of the real data and finally make the generator produce fake data with a similar distribution to the real spectrogram.

3.6. Gesture Recognition. The movement trajectory, speed, range, and time of different gestures are different. Therefore, the spectrograms of different gestures converted by STFT will show different frequency shift features, so we can recognize the gestures by distinguishing the spectrograms of different gestures.

The convolution operator of CNN extracts the pixel information around each pixel and between its different channels on the image by sliding between them with a two-dimensional convolution kernel [35]. Compared to existing work in gesture recognition that tends to build distance models or manually select signal features as the basis for classification, more essential gesture features can be automatically extracted by training CNN as a classification model. Therefore, in this paper, we use a CNN as a classifier for gesture classification.

By deepening the network depth, the network classification accuracy can be improved [36]. However, repetitive deepening of the network can lead to the phenomenon of network degradation, i.e., as the network depth increases, the accuracy of the network not only does not increase but also tends to level off or even decrease. To cope with the above network degradation, ResNet [37] proposed the residual block structure to effectively solve the network degradation problem, and the basic structure of the residual block is shown in Figure 5.

To ensure the classification accuracy of the network and reduce the number of model parameters while avoiding the phenomenon of network degradation, we design a convolutional neural network SonicGest-Net with a combination of multiple convolutions and 2 residual blocks. The network architecture is shown in Figure 6. After each convolutional layer, we add the ReLU activation function and the BN layer. To reduce the number of parameters, we use global averaging pooling at the end of the network to downsample the final feature map to a size of 1*1 and feed it to the output layer to predict the gesture. In which, to prevent the model from overfitting, we introduce an early stopping technique (i.e., we set the network training to stop early when the loss does not decrease in 5 consecutive test sets).

### 4. Experimental Validation

#### 4.1. Experimental Setup. The SonicGest transceiver is implemented in JAVA and deployed as an app on a Samsung Galaxy C8 smartphone. In addition to that, no other installation on the phone is required. As shown in Figure 7(a), when the user clicks on “start,” SonicGest sends the ultrasound signal through two threads executing in parallel inside the program and simultaneously formats the received echo signal into WAV audio file format for storage on the disk. Moreover, the GAN and CNN models are written using PyTorch and trained on a personal laptop with an NVIDIA GeForce GTX 950 M graphics card.

In the experiments, the GAN model uses Adam as the optimizer. The learning rate of the model is 10^{-4}, the BatchSize is 64, and the penalty factor \( \lambda \) is 10. We set the spectrogram dimension of the input to the classifier model to 112×112 and use the RMSprop optimizer for optimization, where the learning rate of the classification model is 10^{-5} and the BatchSize is 64. The experimental process is shown in Figure 7(b).

#### 4.2. Experimental Data. We selected 10 commonly used interactive gestures, whose descriptions are shown in Table 1. The gestures collected experimentally, and their spectrograms are shown in Figure 8.

For the gesture data collection, we randomly invited six volunteers. The data collection environment contained noise from the activities of other people and all had people walking around. During the gesture collection, volunteers were asked to perform 10 common interaction gestures at 5-10 cm from the phone. The collection process was entirely based on the user’s habits, except for the distance limitation. For this, a distance limit was made after two considerations:

1. The energy of the acoustic signal decreases as the propagation distance increases, and to collect effective data containing gesture features, the user cannot perform gestures too far away from the smartphone.
2. During data collection, we indicated to the volunteers the purpose of the experiment, and most of them thought that collecting at 5-10 cm was more appropriate to the operating habits.

For 10 common gestures, 6 volunteers were asked to repeat each gesture 50 times, and the collection process lasted for one month. Afterwards, we used a GAN model to generate 900 fake data for each gesture. In this case, all the data constituted a training set and a test set in the ratio of 8:2. The training set consists of 9000 generated data.

#### Table 1: Ten interactive gestures.

<table>
<thead>
<tr>
<th>Gesture category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>Palm down press</td>
</tr>
<tr>
<td>G2</td>
<td>Palm up raise</td>
</tr>
<tr>
<td>G3</td>
<td>Slide left</td>
</tr>
<tr>
<td>G4</td>
<td>Slide left and then right</td>
</tr>
<tr>
<td>G5</td>
<td>Double (slide left and then right)</td>
</tr>
<tr>
<td>G6</td>
<td>Press the palm down and then raise it up</td>
</tr>
<tr>
<td>G7</td>
<td>Finger rotation clockwise</td>
</tr>
<tr>
<td>G8</td>
<td>Finger rotation anticlockwise</td>
</tr>
<tr>
<td>G9</td>
<td>Open and close with both hands</td>
</tr>
<tr>
<td>G10</td>
<td>Double (press the palm down and then raise it up)</td>
</tr>
</tbody>
</table>
and 600 real data. The test set data consists entirely of 2400 real data in total.

4.3. System Performance. After 35 rounds of training, the loss and accuracy of the model tend to be stable, and the final accuracy of the recognition of 10 gestures is 98.9%. The confusion matrix of 10 gestures is shown in Figure 9(a); we can observe from the figure that gesture G7 is easily misidentified as gesture G2 and gesture G6, and gesture G2 is easily misidentified as gesture G1 and gesture G9. Accuracy, recall, and F1-score are widely used to evaluate the effectiveness of classification models. The definitions are as follows:

\[
\text{precision} = \frac{TP}{TP + FP}, \\
\text{recall} = \frac{TP}{TP + FN}, \\
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},
\]

where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. We can see in Figure 9(b) that the precision, recall, and F1-score of each gesture are higher than 96%, which proves the high efficiency of classification.

4.4. Classification Model Evaluation. To evaluate the impact of classification models on the classification results, we use multiple classification models to predict the data collected from the experiments. In our experiments, we selected support vector machine (SVM), a representative model in the field of machine learning, and VGG16 and ResNet18, representative models in deep learning. Moreover, considering the influence of the number of parameters, we also chose ShuffleNet and MobileNet, which are the representatives of lightweight models. As seen in Table 2, ResNet18, VGG16, and our models all achieve high accuracy rates. The accuracy of the two lightweight models is relatively low because the lightweight models use many group convolutions to reduce the number of parameters. Although the use of grouped convolution reduces the number of parameters, it inevitably weakens the learning ability of the models at the same time. Among them, the SVM’s performance is poor, and the recognition accuracy is only 72%.

In addition to accuracy, we also consider the effects of the number of model parameters and computational complexity. In our experiments, we selected two common metrics, FLOPs and the number of model parameters, that show the complexity of the model. It can be observed that the FLOPs and number of parameters of Vgg16 and ResNet18 are higher than our model, while the number of model parameters of the two lightweight models is lower than in our model. By considering the accuracy and complexity of the model computation, our proposed classification model achieves a better balance between the two.

4.5. GAN Model Evaluation

4.5.1. Evaluation of Data Enhancement Methods. To evaluate the impact of different data enhancement methods on the recognition performance of the system, we conducted tests through five sets of experiments. In the first group, we use real data from the training set to train the classification model. In the second to fourth groups of experiments, we utilize three traditional augmentation methods, i.e., panning, scaling, and rotation, to expand the training set with 600 pieces of real data by 100 times. In the last group, we used the generator of the GAN model in this paper to generate 900 pieces of generated data for each class of samples and add them to the training set to augment the data.

The experimental results are shown in Table 3; we can see that the accuracy, precision, and recall average values of the way without data enhancement are around 91%. However, the accuracy, precision, and recall of the random rotation and scaling data enhancement method decreased significantly because the random rotation and scaling changed the features of the spectrogram in the frequency domain. The panning enhancement method improves the recognition performance of the system slightly, which is due to
the small magnitude of panning weakening the influence of different gesture execution times to some extent. Finally, from the last group of experiments, we can see that using GAN for data enhancement can have a large improvement in the recognition performance of the system.

4.5.2. Data Generation Quality Evaluation. To verify the quality of the data generated by our GAN model, we compare the real data of the four gestures, clockwise, anticlockwise, palm down press, and palm up raise, with the fake data generated by DCGAN and our model under the same conditions, respectively. From Figure 10, we can see that our model generates enhanced spectrograms that are more like the original images. The quality of the data generated by DCGAN is lower than that of our model and differs significantly from the original images. During training, the
DCGAN experienced gradient disappearance and pattern collapse, which led to poor results in the spectrogram generation of gestures.

4.5.3. Network Stability Evaluation of GAN. In this paper, we utilize the Wasserstein distance based on the gradient penalty to replace the loss function of the GAN. To test its effect on the stability of the network, we compare the loss curve of the GAN model in this paper with that of DCGAN. The results are shown in Figure 11.

We can see from Figure 11(a) that since DCGAN uses JSD as the loss function, there is large volatile on both the discriminant loss and the generator loss in the middle and late stages of training, which leads to a difficult learning process. As can be seen from Figure 11(b), since our GAN model uses the Wasserstein distance based on gradient penalty as the loss function, the training process is more stable and the loss curve is less volatile.

4.6. Impact of Different Factors

4.6.1. Impact of Environmental Noise. The noise level when users perform gestures is different. Therefore, we need to test the robustness of the system under different noise levels. In this paper, we simulate three noise environments with 50-60 db, 60-70 db, and 70-80 db levels by playing music around the volunteers. The noise levels were measured using an existing commercial APP. The experimental results are shown in Figure 12. The average recognition accuracies of the 10 gestures were 99.5%, 98%, and 96.5% under the three noise levels, respectively. Although with the increase in noise level, the recognition accuracy of the system decreases; it remains stable under the high noise environment of 70-80 db. Among them, we can also see that the recognition accuracy in the dormitory environment is higher than in the laboratory, which is because the laboratory is distributed with more tables, chairs, and equipment, which leads to a more serious multipath effect and thus affects the recognition accuracy of the system.

4.6.2. Impact of Distance. We set five intervals of 5 cm to evaluate the sensing boundary of the system for gestures, and the average recognition accuracy for the five ranges of 0-5 cm, 5-10 cm, 10-15 cm, 15-20 cm, and 20-25 cm was 99%, 99%, 97%, 94.5%, and 86.5%, respectively. From Figure 13, we can find that the recognition accuracy of the system tends to decrease with the increase in distance. When
the gesture action is more than 20 cm away from the smartphone, the recognition performance of the system decreases significantly, which is because the ultrasonic signal has a large loss of energy as the propagation distance increases. The experimental results show that SonicGest can effectively recognize gestures within 20 cm.

4.6.3. Impact of Angle. Considering that different users have different habits of using smartphones, we evaluated the effect of angle on recognition accuracy when smartphones are placed vertically or horizontally. The experimental results show that SonicGest can effectively recognize gestures within 20 cm.

Figure 11: Discriminators and generators loss curve. (a) DCGAN loss curve. (b) Loss curve of the GAN model in this paper.

Figure 12: Recognition accuracy under three noise levels.

Figure 13: Recognition accuracy at different distances.

shown in Figure 14(b); we can see that the accuracy rate of vertical placement is slightly higher than that of horizontal placement. The average accuracy rates at 0°, 30°, and 60° are 97%, 93%, and 91%, respectively. When the angle of the smartphone increases, the recognition accuracy starts to decrease. This is because the Doppler shift is closely related to the relative angle. For the decrease in recognition accuracy due to angle changes, we will add data from different angles to the training dataset in future work to enhance the robustness of the system to angle changes. Overall, SonicGest can maintain high recognition accuracy in all directions and angles.
5. Conclusions

In this paper, we propose a design based on an acoustic signal recognition gesture system. The system requires only the built-in speaker and microphone of existing mobile terminals and does not violate user privacy. To satisfy the recognition of multiple consecutive gestures, we propose an algorithm for the segmentation of multiple gestures in the spectrogram. Moreover, for recognizing fine-grained gestures, we remove noise interference by spectrogram enhancement techniques and utilize a CNN based on residual blocks to extract features. To further improve the robustness of the system and overcome the problem of difficult data collection, we modify the loss function of GAN using the Wasserstein distance based on gradient penalty to make its training more stable and generate diverse data. The experimental results show that the system has high robustness, and the average accuracy of recognition for ten interactive gestures is 98.9%.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would also like to express their special appreciation to the volunteers for participating in their experiments.

References


