HearASL: Your Smartphone Can Hear American Sign Language

Yusen Wang, Fan Li, *Member, IEEE*, Yadong Xie, *Member, IEEE*, Chunhui Duan, *Member, IEEE*, and Yu Wang, *Fellow, IEEE*

***Abstract*—Sign language is expressed by movements of the hands and facial expressions, which is mainly used by the deaf community. Although some gesture recognition methods are put forward, they possess different defects and are not applicable to deal with the sign language recognition (SLR) problem. In this article, we propose an end-to-end American SLR system with built-in speakers and microphones in smartphones, which enables SLR at both word level and sentence level. The high-level idea is to use the inaudible acoustic signal to estimate channel information and capture the sign language in real time. We use channel impulse response to represent each sign language gesture, which can realize finger-level recognition. We also pay attention to conversion movements between two words and treat them as an additional label when training the sentence-level classification model. We implement a prototype system and run a series of experiments that demonstrate the promising performance of our system. Experimental results show that our approach can achieve an accuracy of 97.2% at word-level recognition and word error rate of 0.9% at sentence-level recognition, respectively.**

***Index Terms*—Acoustic sensing, American sign language (ASL), mobile computing.**

I. Introduction

HEARING loss is the third most prevalent chronic health condition in the U.S. Approximately 15% of American adults (37.5 million) report some trouble hearing [1]. Sign language is invented to facilitate communication between the deaf/mute people, and American sign language (ASL) is a language of the deaf community in the U.S. and most of Canada, which is a primary language used by people who are deaf or hard of hearing [2]. However, people with normal hearing normally do not learn sign language. There exists a huge gap between the deaf community and people with normal hearing. Therefore, it is essential to build a system that can help sign language users to be understood anytime and anywhere.

In recent years, researchers propose various novel approaches to deal with this problem. There are some ASL recognition (ASLR) systems using Kinect or Leap Motion-based visual methods [3], [4], [5], [6], [7], [8]. Lang et al. [5] presented an open-source framework based on Hidden Markov models (HMMs) for general gesture recognition using Kinect. Chong and Lee [9] developed an ASLR system using a support vector machine (SVM) and a deep neural network (DNN) based on a new digital sensor called Leap Motion, but they are all sensitive to the locations of sensors. With the improvements in wearable devices, some systems utilize smartwatches to recognize sign language [10], [11]. For instance, Signspeaker [10] uses a long short-term memory (LSTM) for sentence-level pattern recognition based on accelerometers and gyroscopes in smartwatches. But these systems need users to wear additional devices and can not cope with nonmanual markers (e.g., head tilting and shoulder raising) and two-handed signs well. Recently, Wi-Fi and acoustic signals are used for sign language recognition (SLR) [12], [13], but they both have some limitations. SignFi [12] recognizes 276 sign gestures using channel state information (CSI) measured by Wi-Fi packets and a convolutional neural network (CNN) as the classification algorithm. SonicASL [13] leverages the recurrent convolutional networks to recognize subtle pattern changes from the reflected sonic wave. However, Wi-Fi sensing can not be used for outdoor scenes. SonicASL adds an outward-facing speaker next to the microphones of noise canceling headphones additionally, which means the devices are not pervasive enough. And the system relies on the Doppler Effect in received signals reflected by sign gestures, which causes low resolution and limits the performance for finger-level sign gestures.

Motivated by the above limitations, we ask a question: *can we overcome these limitations and design a more ubiquitous, low cost, and accurate ASLR system?* Nowadays, smartphones become more and more powerful with built-in sensors, such as cameras, accelerators, and gyroscopes, which greatly improve the sensing ability. Among numerous sensing methods, acoustic sensing makes tremendous progress in gesture recognition [14], [15], [16], [17], [18]. In this article, we design a novel end-to-end ASLR system named HearASL which can run on a smartphone with a built-in speaker and microphone. We use a speaker to emit an inaudible acoustic signal and simultaneously use the microphone to receive the

Manuscript received 25 July 2022; revised 10 November 2022; accepted 23 December 2022. Date of publication 27 December 2022; date of current version 9 May 2023. The work of Fan Li was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62072040. The work of Chunhui Duan was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61902212, and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars. *(Corresponding author: Fan Li.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board (IRB) at the Beijing Institute of Technology.

Yusen Wang, Fan Li, Yadong Xie, and Chunhui Duan are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100089, China (e-mail: yusenwang@bit.edu.cn; fli@bit.edu.cn; ydxie@bit.edu.cn; duanch@bit.edu.cn).

Yu Wang is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: wangyu@temple.edu).

Digital Object Identifier 10.1109/JIOT.2022.3232337

Fig. 1. Possible scenarios where HearASL can be applied. HearASL can be embeded in a translator software, clicking the button to translate sign language. It can also be used for sign language learning software, people emulate standard sign language word, and get feedbacks if correct of not.

signal reflected by sign language gestures. Besides being a translator to help hearing people understand sign language, such a system can also empower sign language learning. Fig. 1 shows two possible applications of this system. However, several unique challenges need to be addressed when developing such a system.

1) Different from hand gesture recognition for human-computer interaction, ASL is more complicated because it involves in many similar arms, hands, and especially finger movements. It is hard to distinguish these sign language words.
2) Considering that our system may be used in the outdoors, there is a lot of interference when we extract the features of sign language, including environmental noises and different multipath situations. The features are the combinations of all signals reflected from both static and dynamic objects within the sensing range, we need to focus on the reflected signal from sign language gestures.
3) It is tough to realize sentence-level SLR since there are some extra actions between two words. For example, if a person wants to express a sentence like "go home," the sign for "go" is performed in front of the chest, and the sign for "home" is performed by tapping the cheek, there exists a conversion movement from the chest to cheek.

In the design of HearASL, we take several effective measures to resolve the above challenges. First, to distinguish the subtle difference between sign language, we extract channel impulse response (CIR) as the features of sign language, which contains fine-grained movement information, including arm, hand, and finger movements. Then, we analyze the components of CIR magnitudes to eliminate the inference from static objects. Specifically, we use a 1-order difference operation to focus on the channel information of sign language, and also employ a Hampel filter to further purify the CIR image. In this way, the CIR images only contain the component of sign language for further recognition. Third, we analyze the conversion movements between two words and take them as an additional label when training the sign language classification model. Specifically, we add a gate recurrent unit (GRU) after a CNN to further extract sequence features. We also take advantage of the connectionist temporal classification (CTC) loss function to avoid manual alignment during training sentence-level recognition model.

The prototype of HearASL is built using an iPhone 12 Pro with built-in microphones and speakers. To evaluate the performance of HearASL, we recruit 20 volunteers (9 males and 11 females) to do selected ASL several times. The experiments involve 50 words and 30 sentences in ASL. Finally, we collect 20 000 signs in words and 12 000 signs in sentences. Results demonstrate that HearASL can accurately identify ASL words and sentences in various environments.

In a nutshell, our contributions are summarized as follows.

1) We propose an SLR system at both word level and sentence level based on microphones and speakers in the smartphones. To the best of our knowledge, HearASL is the first smartphone-based SLR system using acoustic sensing.
2) We extract fine-grained CIR measurements to recognize finger-level sign language and design effective methods to eliminate the inference from both static and dynamic objects.
3) We address the challenge of the conversion movements between two words by treating them as an additional label and put forward an appropriate deep learning model to realize both word-level and sentence-level SLR.
4) We conduct comprehensive experiments to evaluate our design in various real-world scenarios. Our system achieves a promising performance with 97.2% accuracy for 50 sign language words and a word error rate (WER) of 0.9% for 30 sentences on average.

Outline: The remainder of this article is organized as follows. In Section II, we give an overview of HearASL. In Section III, we give the preliminaries of the CIR measurement. Following that, we give details of system design in Sections IV–VI. We describe the implementation, experiments, and display performance evaluation in Section VII. In Sections VIII and IX, we discuss related works and future works. In Section X, we make the conclusion of this article.

II. System Overview

HearASL consists of three modules: 1) *Data Collection*; 2) *CIR Enhancement*; and 3) SLR, as shown in Fig. 2.

In *Data Collection*, users can just press a button and let their smartphones face the sign language users when recognizing ASL gestures. We aim to estimate the CIR through a single-carrier communication channel. First, the transmitter which is a speaker of the smartphone in our system emits an inaudible acoustic signal modulated by the Barker code with the frequency of 18–22 kHz. The microphone records the received signal with the sampling rate of 48 kHz, then the

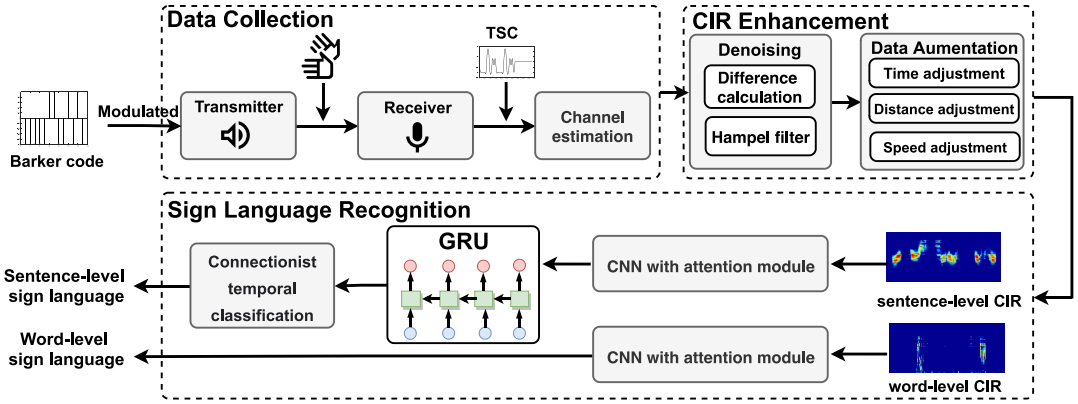


Fig. 2. Workflow of HearASL.

least square (LS) method is conducted to do channel estimation based on our training sequence code (TSC). After that, we get the raw CIR magnitudes.

In *CIR Enhancement*, CIR magnitudes are converted to an image, and then HearASL utilizes a series of methods to enhance the CIR image, including difference calculation of CIR and a Hampel filter. The difference operation can retain the dynamic components of measured CIR, eliminate the inference of multipath reflected by static objects, and the Hampel filter can filter out the outliers that are caused by other dynamic components besides sign gestures. Then, in order to increase the robustness, several data augmentation techniques are performed to handle user diversity, including time adjustment, distance adjustment, and speed adjustment.

In SLR, we design two deep learning modules for word-level and sentence-level sign language, respectively. We adopt a CNN-based architecture with an attention module, which focuses on the main component of CIR images and recognizes word-level sign language. And the sentence-level CIR images go through a series of GRU units and CTC loss function after CNN, which performs sentence-level SLR. In Sections IV to VI, we will introduce the technical details of the above three modules.

III. Preliminary

In this section, we give the limitations of the Doppler shift to recognize ASL gestures first and introduce the CIR measurement in our system.

A. Limitations of Doppler Shift

Existed acoustic sensing-based ASLR system detects the finger/hand movements by Doppler shift [13]. In order to extract the Doppler shift, a short-time Fourier transform (STFT) is usually performed to calculate a spectrogram of the reflected signal which can reflect the time-frequency relation. However, the resolution of STFT is limited by fundamental constraints such as segmented frame length and an overlap length. We can calculate the minimal frequency resolution of STFT: $\Delta f = (F_s/W)$, where F_s is the sampling rate which is 48 kHz and W means the STFT window length which is 8192. Then, we can further get the minimal speed resolution

$$\Delta v = \frac{F_s \cdot c}{W \cdot F_c} \tag{1}$$

where F_c is the center frequency which is set to be 20 kHz. Therefore, the speed resolution is about 10 cm/s in this case. It means that users have to express sign language at a higher speed than that. Note that there is no speed requirement for ASL, it is possible to do sign language at a low speed about 5 cm/s which may not be detected. The speed limitations can deeply decrease the user experience. Therefore, we investigate CIR measurement and find that the resolution of speed and frequency can satisfy the need of recognizing ASL.

B. CIR Measurement

When the transmitter and receiver keep stationary in our system, the attenuations and propagation delays do not depend on time. Therefore, our system can be treated as a linear time-invariant (LTI) system. Suppose the channel between the transmitter and the receiver has several paths M and amplitude of the received signal a_i, we have the usual LTI channel with an impulse response

$$h[n] = \sum_{i=1}^{M} a_i e^{-j2\pi f' \tau_i} sinc(n - \tau_i W) \tag{2}$$

where $sinc(t) = ([\sin(\pi t)]/\pi t)$. τ_i and f' are propagation delay of each path and frequency of reflected signal, respectively. n denotes nth channel tap of CIR, which reflects length information of prorogation path and corresponding reflected signal strength. We aim to estimate these channel taps during hand/finger movements in front of the smartphones.

Usually CIR is measured based on a known TSC, we can utilize the TSC and the corresponding received signal after processing for estimating CIR. There are two main approaches of channel estimation, i.e., the LS and the linear minimum mean squared error (LMMSE) method. We choose the LS method for reducing time complexity.

IV. Data Collection

In this section, we mainly introduce the procedure of channel estimation, including the design of transmission signals

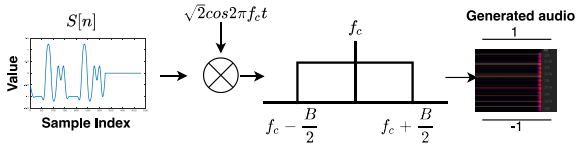


Fig. 3. Transmitter design.

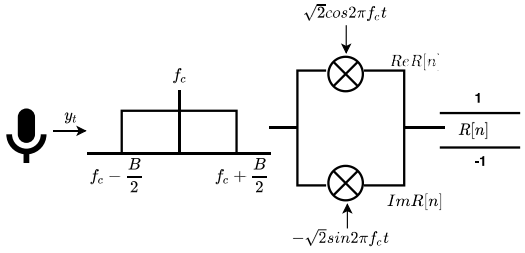


Fig. 4. Receiver design.

and signal reception. We also give the details of calculating CIR magnitudes of each sign language gesture.

Transmitter Design: A transmitter sends a known TSC for channel estimation, which is denoted by $m = [m_0\ m_1\ \ldots\ m_{P+L-1}]$, where $P + L - 1$ is the length of TSC, P is the length of the data section and L is the length of the guard period. L determines how many channel taps we can estimate which are related to the sensing distance. We choose 13-bit Barker Code to generate the TSC, which is a finite sequence of digital values with the ideal autocorrelation property. It is usually used for pulse compression of radar signals [19]. To achieve enough length and avoid frequency leakage, we increase the length of TSC by copying it twice and up-interpolate the sequence by 12 times. The whole length of the final TSC is 480, which means we can get 100 columns of CIR every second as the sampling rate of the received signal is 48 kHz. The length of the data section is set to be 350, and the remaining 130 samples are empty sections to avoid intersymbol interference (ISI). Fig. 3 depicts the process of signal transmission and passband conversion. At the transmitter end, let $S[n]$ denote the final TSC and f_c denotes the center frequency of the passband, then we multiply $\sqrt{2}\cos(2\pi f_c t)$ to up-convert the signal to the passband. To further remove the noises outside the ultrasonic band (i.e., 18–22 kHz), we perform band-pass filtering of $[f_c - (B/2), f_c + (B/2)]$Hz on the passband signals, and B is set to be 4 kHz which represents the channel bandwidth. Then, the generated passband acoustic signal is normalized and saved as a format of 16-bit pulse coded modulation (PCM) in a waveform audio (WAV) file, which can be played on any smartphone through a speaker.

Receiver Design: The microphone records signal reflected by sign gestures. Fig. 4 depicts the process of signal reception and baseband conversion. At the receiver end, the received signal first go through a band-pass filtering from 18 to 22 kHz to remove environmental noises. Then a down-conversion process is performed to convert passband signal $y(t)$ to baseband signal $R[n]$. Specifically, $y(t)$ is multiplied by $\sqrt{2}\cos(2\pi f_c t)$ and $-\sqrt{2}\sin(2\pi f_c t)$ to acquire real and imaginary parts of the baseband signal, respectively. Finally, we normalize the complex baseband signals to the range of -1 to 1 to reduce calculation costs.

Now we have the received baseband signals and the known TSC. To calculate $h[n]$, we use LS estimation which only involves matrix calculation to reduce the time complexity [20]. In LS channel estimation, the received signal y can be expressed as follows:

$$y = M * h \tag{3}$$

where the complex CIR h of the received signal is expressed as follows:

$$h = [h_0\ h_1\ \ldots\ h_L]^T. \tag{4}$$

A circulant training matrix M is formed as follows:

$$M = \begin{bmatrix} m_L & \cdots & m_1 & m_0 \\ m_{L+1} & \cdots & m_2 & m_1 \\ \vdots & \ddots & \vdots & \vdots \\ m_{L+P-1} & \cdots & m_P & m_{P-1} \end{bmatrix}. \tag{5}$$

We have the received baseband signal $y = \{y_1, y_2, \ldots, y_{L+P}\}$, the channel is estimated as follows:

$$\hat{h}_{LS} = \left(M^H M\right)^{-1} M^H y_L \tag{6}$$

where $y_L = \{y_{L+1}, y_{L+2}, \ldots, y_{L+P}\}$. The sound speed c is 340 m/s and L is the maximum CIR channel taps, thus maximum sensing distance is $d = (L/f_s * 2) * c$ in our system. We set L to be 150 which corresponds to the sensing distance of 0.5 m. In this way, we can get a $[150 \times 100]$ CIR complex matrix every second.

V. CIR Enhancement

In this section, we introduce the methods of the denoising process. After data collection we get the CIR magnitudes of the sign language gestures. We perform a 1-order difference operation on CIR magnitudes to eliminate the inference of static component. And, we convert CIR magnitudes to images and utilize the Hampel filter to purify the whole image for classification. After that, we introduce three data augmentation methods used in the system, which can increase the system robustness and accuracy greatly.

A. Denoising

Note that the collected CIR matrix not only contain signals reflected by sign gestures, but also other multipath information which can interfere our recognition accuracy as shown in Fig. 5(a). According to [21], CIR measurement can be classified into static component and dynamic component. The static component consists of the direct transimission from the speaker to the microphone and the static background reflection from the environment, which is unnecessary for the recognition. The dynamic component consists of the user's sign gestures in front of the microphone or other movements such as a pedestrian walking by the signer. To cancel the influence of the static component and make the sign language gesture highlighted in the image, we calculate the 1-order CIR difference between two consecutive complex samples at time t

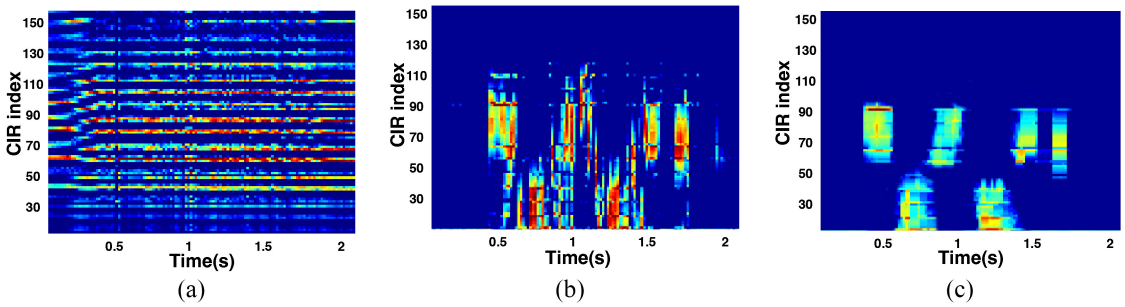


Fig. 5. Procedure of CIR enhancement. (a) CIR image of church. (b) CIR image of church after differential operation. (c) CIR image of church after hampel filter.

and $t-1$. Take the sign "church" as an example, the processed result after calculating the CIR difference is shown in Fig. 5(b). The x-axis denotes time, while y-axis denotes CIR channel taps, and the brightness represents the CIR magnitude. We can see that the dynamic component is extracted, and we can only focus on moving objects to recognize each sign language gesture.

After the difference operation, we find that the CIR image still has some noise, which may be caused by other dynamic components besides sign gestures. Thus, the CIR image is processed by a Hampel filter to further clarify the whole image. The function of the Hampel filter is to identify and replace outliers in a given series. Because the CIR image is a matrix, the filter treats each column of the matrix as an independent channel. And, every value of a channel is treated as a sample. For every sample in the CIR image, the filter first calculates the median of the sample and the surrounding six samples, then uses the absolute value of the median to estimate the standard deviation of the median for each sample. Finally, it uses a sliding window to go over every channel. A sample is replaced with the median of the window if it differs from the median by more than three times of standard deviations. Through the Hampel filter, the outliers which are other dynamics components in the whole image are eliminated and the main brightness area is more highlighted as shown in Fig. 5(c).

B. Data Augmentation

Before we train the deep learning model using processed CIR images, we also employ some data augmentation techniques to increase the amounts of training data. The deep learning network is a data-driven method, which requires a huge amount of training data to achieve high accuracy and high robustness. Moreover, we aim to recognize 50 kinds of sign language words, which is tough for us to collect a large number of gesture samples for every sign language. In our system, we take into account the fact that people express sign language in different times and distances, even in various speeds. By emulating these cases, we can also get over the problem of user diversity to make our system more robust.

Different Time: There has an activation operation in our system, that is the user clicks the button to start translation, then clicks again to stop. Therefore, the whole sensing period is called detection time. We assume that users express sign language at different times during detection time. So, we manually shift the brightness area along the time axis within detection time.

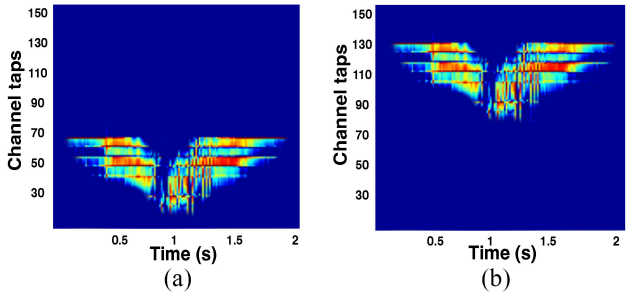


Fig. 6. CIR magnitude in different distances to the receiver. (a) Push and pull at 20 to 30 cm. (b) Push and pull at 10 to 20 cm.

Different Distance: According to (2), CIR magnitudes are related to propagation delay, which can show different distances by channel taps (a larger channel tap index corresponds to a further distance to the receiver). In order to make the impact of distance more obvious, we perform a push-pull gesture from 20 to 30 cm, and then 10 to 20 cm in front of the receiver. We can find that the brightness area is near the top of image when the distance is far as shown in Fig. 6(a). Otherwise, it is close to the bottom as shown in Fig. 6(b). Based on the above observation, we emulate different distances to the receiver by vertical drifts in tap indexes within maximum channel taps.

Different Speed: People may have different speeds when they express sign language. We perform a push-pull gesture at two different speeds. As shown in Fig. 7(a) and (b), the brightness area of two CIR images have a similar trend, but the image in a slower speed changes smoothly as shown in Fig. 7(a). Thus, we emulate different speeds by horizontally expanding or contracting an CIR image.

We consider the above three factors that may impact CIR image and design three strategies to perform data augmentation. Finally, enough images of every sign language are generated to train the deep learning model.

VI. Sign Language Recognition

In this section, we introduce two deep learning models against word-level and sentence-level sign language. For the word-level SLR, we treat this task as an image multiclassification problem. Considering that CNN is great at extracting

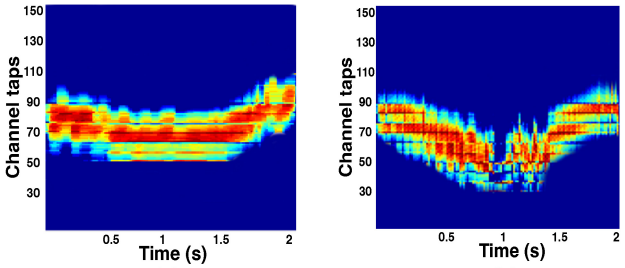


Fig. 7. CIR magnitude in different speeds. (a) Push and pull in slower speed. (b) Push and pull in normal speed.

abstract features, we design a CNN network with an attention mechanism, which focuses on the brightness area of each image and enhances the performance of the CNN. For sentence-level sign language, we get inspiration from optical character recognition and design a CNN + GRU + CTC structure.

A. Word-Level Recognition

For word-level sign language, we adopt a CNN model to automatically classify a CIR image to a specific word. We observe that extracted CIR image of every sign language is distinguishable due to the different shapes of the brightness area. However, the brightness area just occupies a small section rather than the whole image. To make the classification network focus on these areas of the CIR image, we introduce the attention mechanism in computer vision [22]. In our system, we utilize the convolutional block attention module which consists of a channel attention module and spatial attention module to make important features more weighted in the training phase [23]. We describe the detailed operation as follows.

First, we apply 2-D convolutions for feature extraction. After that we get a $[C * H * W]$ feature map F, where C is the number of channels, W and H denote width and height of the resized image after the convolution operation, respectively. Then, we put it into the attention module. The module has two sequential submodules: 1) channel attention and 2) spatial attention. The channel attention module is used for exploiting the interchannel relationship of features. It focuses on the meaningful area in an input image. It aggregates spatial information of a feature map by using both average-pooling and max-pooling operations, generating two descriptors which denote average-pooled features and max-pooled features, respectively. Both descriptors are then forwarded to a shared network which is composed of multilayer perceptron (MLP) with one hidden layer to produce a 1-D channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$. The spatial attention module is used for extracting interspatial relationship of features. It focuses on the location of the informative part, which is the brightness area in the CIR image. Average-pooling and max-pooling operations are performed along the channel axis and concatenated to generate a feature descriptor. Then, a convolution operation is applied on it to generate a spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ which represents where to emphasize or suppress. In this way, we get the refined features after these two attention modules. The overall attention process can be summarized as follows:

$$F' = M_c(F) \otimes F$$
$$F'' = M_s(F') \otimes F' \tag{7}$$

where $\otimes$ denotes element-wise multiplication and F'' is the final refined output. Finally, we put F'' into a dense layer with softmax activation functions after a flatten layer to generate the probability prediction of every possible sign language word.

B. Sentence-Level Recognition

1) End-to-End Sign Language Sentence Recognition Model: For sentence-level SLR, one possible method is to segment a whole sentence into individual words and recognize them, respectively. But the recognition accuracy mainly depends on the accuracy of the segmentation algorithm. And, it is difficult to segment a sign language sentence into individual words due to the unpredictable conversion movements between two words. Based on these observations, we utilize an end-to-end sequence learning model, which is generally used for optical character recognition [24]. The whole model consists of three parts: 1) CNN; 2) GRU; and 3) CTC. Given a sentence-level sign language CIR sequence $X = [x_1, x_2, \ldots, x_M]$ and the corresponding words $Y = [y_1, y_2, \ldots, y_M]$, the aim in end-to-end sequence learning is to build a deep learning model to map X to Y. Through the CTC loss function, we do not need to label the specific location of every word manually. For instance, for the sentence "I need some help," we can only give this training data labels [I, NEED, SOME, HELP] instead of labeling individual words in the input sequence. Different from previous work in SLR [13], we take account of conversion movements between two words, we label these movements before the training process and delete them in the final results. The function of these labels is similar to "blank" (used for labeling nonsign inputs) in CTC. In this way, the trained deep learning model can also recognize these conversion movements in the inferring process, which realize a more precise implicit segmentation and recognition accuracy.

The structure of the model is shown in Fig. 8. We use the aforementioned CNN structure as the front-end to extract features of the whole image. We first resize the image to $[224 * 50 * 3]$, which can hold the information of sign language sentences. We get a feature map after convolutional layers. Then, we perform reshape operation for input of GRU. GRU can further extract sequence features of the image from several time steps. As a form of the recurrent neural network, it can achieve the same function and greatly improve training efficiency compared to the LSTM network. In the training process, we do not only label complete semantic information to the input sign language sentence but also need to give the model basic sign language words that involve in every sentence as the dictionary. The output of GRU is a posterior probability matrix, which presents the probability prediction of every word in the dictionary at every time step. We design a CTC layer at the end for avoiding manual alignment of X and Y.

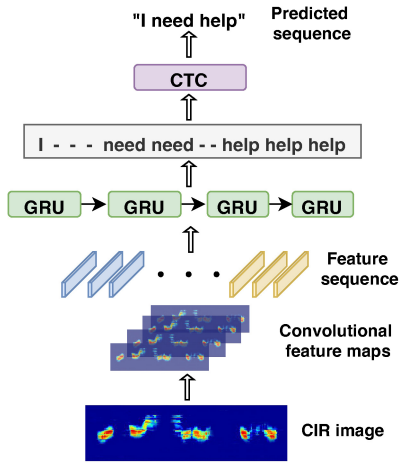


Fig. 8. Architecture of the sentence-level model.

Given an input sequence X, CTC can give all possible outputs. Finally, the network calculates the most possible result by maximuming posterior probability.

2) Labeling Conversion Movements Between Sign Language Words: The blank label in CTC is used for labeling the boundaries of different sign gestures. Therefore, the CTC loss function can segment sentences automatically and provide a way to learn without prealignment. However, there have unpredictable conversion movements between two words that may affect the training procedure. For example, the sentence go home consists of two words, which are preformed by pointing foward using two hands in the neutral space and taping your face two times, respectively, an extra movement from neutral space to the human face is also captured in a CIR image. These extra movements cause interference for the model. Note that we get a feature map after CNN, due to the convolutional layer and the maximum pooling layer are conducted on partial areas of the whole image, each feature vector extracted from the feature map corresponds to a receptive field of the original image. As shown in Fig. 9, every sign language word gets multiple rectangular areas, which corresponds to the receptive fields of CNN. And, the conversion movements in the image also get multiple receptive fields. They interfere the network during the training process and affect the overall performance in long sentences specially, because the network treats these conversion movements as part of normal sign language words.

We investigate 100 kinds of common sign language words covering the topic of family, places, colors, etc. We find that these words can be classified into two categories by area, one is the human's face A, and the other is the neutral space B in front of the human's chest. Therefore, there are four kinds of conversion situations between two words: 1) $A \rightarrow B$; 2) $B \rightarrow A$; 3) $A \rightarrow A$; and 4) $B \rightarrow B$, we focus on the first two conversion movements that are named as $\mathcal{C}_1, \mathcal{C}_2$ because they have more impact on the CIR magnitudes. We treat these two movements as additional labels during the training process. Specifically, we have the example sentence "Apple, green, you like eat," there have two conversion movements in it. We label it as [APPLE, $\mathcal{C}_1$, YOU, LIKE, $\mathcal{C}_2$, EAT]. In this way, the network can recognize $\mathcal{C}_1, \mathcal{C}_2$ in the inferring phase. We delete them and get the final results at the end.

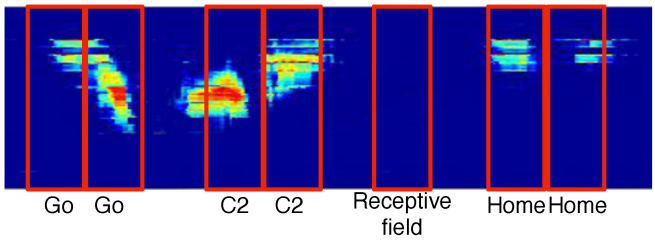


Fig. 9. CNN receptive fields in a CIR image.

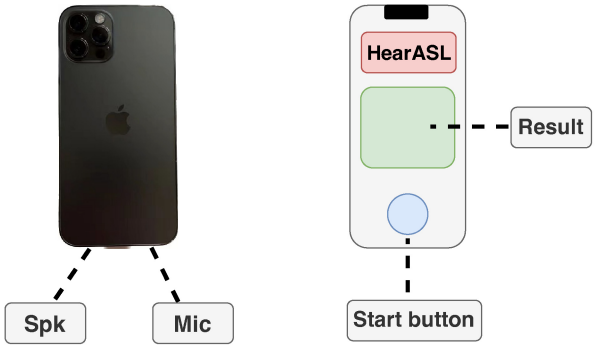


Fig. 10. Hardware used in the experiment and UI design.

VII. Implementation and Experiments

In this section, we first introduce the experimental setup, including system implementation. Then, we evaluate the performance of HearASL under different environments, sensing distances, and angles to validate its effectiveness and robustness.

A. System Implementation

Hardware Configuration: In this experiment, we use iPhone 12 Pro (with IOS 15.0 OS, a A14 CPU, and 6-GB RAM) to collect data whose sampling rate can reach 48 kHz, and we implement the CNN and CNN + GRU + CTC network for training on the PC, which is equipped with an Intel Core i5-10400F, 16-GB ROM, and an Nvidia GeForce GTX 1660Ti graphics card.

Software Implementation: Our sensing data collection application is implemented in swift for the IOS platform over the smartphone. The application transmits an inaudible acoustic signal after pressing the start button, and receives the reflected echo signals from the microphone and transmits the data to a remote server. After that, we employ our trained model on the server and provide a real-time prediction for the IOS mobile platform. The hardware used in the experiment and the design of UI are shown in Fig. 10.

B. Sign Language Selection and Collection Procedure

We implement a data collection prototype on iPhone 12 Pro. We use the speaker of the smartphone to play the generated WAV file in Section IV and simultaneously record the reflected signal by its microphone. We collect two data sets: a word-level data set and a sentence-level data set. For word

TABLE I
SELECTED ASL WORDS (TWO-HANDED WORDS ARE IN BOLD)

Category	Words	Amount
n.	**name**,**time**,**family**,**space**,dad,mom,**church**,**brother**,**hurry**,mirror,night,**house**,apple	13
adj.	hot,happy,blue,green,yellow,red,hungry,sorry,**dark**,**nice**	10
v.	**love**,**aid**,**pick**,need,thank you,**meet**,**wash**,sleep,drink,will,**stop**	11
pron.	who,**what**,**when**,where,he/she/it,I,**how**,you	8
adv.	please,and,as well,but,before,yes,now	7
int.	hello	1

TABLE II
SELECTED SENTENCE EXAMPLES

Sentence Length	Sentence examples
2	HE HUNGRY, HUNGRY YOU, SORRY CAN'T, HELP YOU
3	I NEED SLEEP, I LOVE CHURCH, COME HERE PLEASE
4	MY SISTER LIKE(s) SCHOOL, HOT WEATHER YOU LIKE
5	APPLE, GREEN, YOU LIKE EAT?

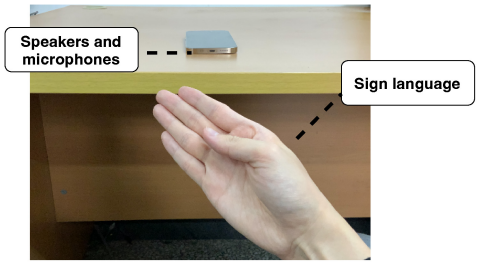


Fig. 11. Experiment setup for data collection.

selection, we refer to a widespread ASL learning website—Lifeprint [25], and finally select 50 common sign words as shown in Table I. Users need both hand and arm movements to represent these words. These words are concluded into six categories: 1) noun; 2) adjective; 3) verb; 4) pronoun; 5) adverb; and 6) interjection. Nineteen two-handed sign languages are also included in them and the others are performed by the dominant hand. For sentence-level sign language, we use these selected words to constitute 50 sentences for recognition, which follow the ASL grammar and have various transition information caused by conversion movements between two words. Some examples are shown in Table II. The length of these sentences is ranged from 2 to 5. We recruit 20 participants (9 males and 11 females aging between 23 and 50). We ask participants to perform 50 sign language words and 30 sentence-level gestures ten times in front of the smartphone in two separate sessions. The experiment setup of the data collection is shown in Fig. 11. Specifically, participants are instructed to perform the gestures for a certain period followed by a 1-s audio cue (i.e., 6 s for a word and 20 s for a sentence). The distance between the hand and the microphone ranges from 20 to 80 cm. And participants place the bottom of their smartphones in front of themselves, the angle between the hand and the smartphone ranges from 0° to 60°. We train two deep-learning models against word-level and sentence-level gestures using collected data. Take the word-level training as an example, we randomly select ten participants as new users. Thus, $[(50\,\text{gestures} \times 10\,\text{times} \times 10\,\text{participants}) \times 2\,\text{sessions}]$ words are used for user-independent evaluation, while the rest $(50 \times 10 \times 10 \times 2)$ constitute the training set and test set. In the training stage, we utilize three data augmentation strategies on both word-level and sentence-level data sets with a rate $= 30\times$, which means that the number of samples increases 30 times compared to the original one. In practice, we shift the brightness area along the time axis within detection time to emulate different times. And we also find that the duration of a sign language gesture is usually from 2 to 4 s, therefore, we expand and contract the whole image at maximum two times to emulate different speeds. Considering that users often conduct sign language from 20 to 50 cm, which indicates that CIR index L ranges from 60 to 150. In this case, we vertically shift the CIR image according to the target range. For every sign language gesture, we emulate different distances, speeds, and time to increase the data set with a rate $= 10\times$, respectively. Then, we put three increased data set together to achieve a rate of $30\times$.

C. Evaluation Protocol

We evaluate HearASL by two strategies.

1) *User-Dependent Test:* User-dependent test means that each user appears in both the training and testing test. And, the data set also consists of the images generated by the data augmentation.
2) *User-Independent Test:* In the user-independent test, we utilize data from ten participants to train the model and the rest ten participants to test. And the data set from them is not trained in our model. We evaluate the ten participants' average performance for one-hand gestures and two-hand gestures. As these ten participants perform the sign language in different environments, the result can also validate the capacity of our system to deal with the environment independent.

At the word-level sign language, we apply intuitive accuracy in machine learning as the evaluation metric. At the sentence-level sign language, we adopt the WER to measure the accuracy, which is a commonly used evaluation metric for speech recognition systems [26].

D. Overall Performance

Word-Level Performance: We first evaluate the user-dependent recognition accuracy for word-level sign gestures.

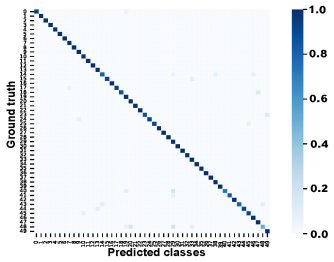


Fig. 12. Overall performance of HearASL.

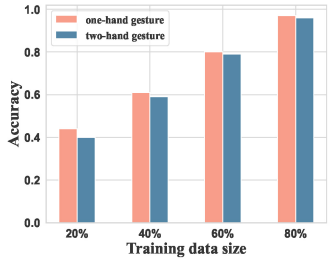


Fig. 13. User-dependent accuracy of word-level with different data size.

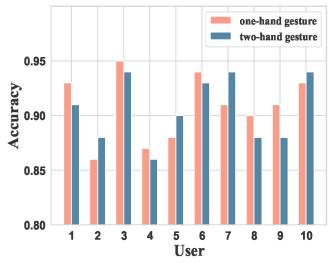


Fig. 14. User-independent accuracy of word level.

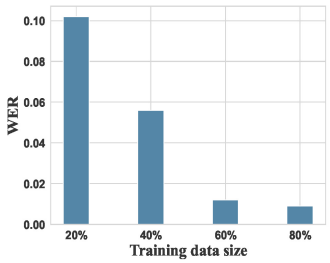


Fig. 15. User-dependent accuracy of sentence level with different data size.

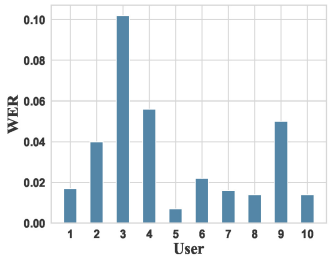


Fig. 16. User-independent accuracy across different volunteers.

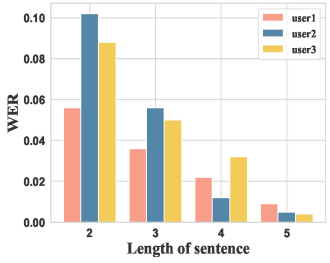


Fig. 17. User-independent accuracy of sentence level with different lengths.

Fig. 12 shows the overall confusion matrix of HearASL for 50 sign language words. The matrix shows that all 50 signs can be classified correctly with a greater probability of about 95.3%. As the training data set size can affect the recognition result, we evaluate the recognition accuracy by using 20%, 40%, 60%, and 80% of the whole collected samples and the remaining part is used for the test. The result is shown in Fig. 13. At the same time, we divide the whole data set into one-hand and two-hand gestures to evaluate the effect of two-hand gestures. We can also see that one-hand gestures and two-hand gestures have similar high recognition accuracy which means our system can handle two-hand situations. When 80% of the collected samples are used for training, the average accuracy of ten participants is 97.2%. This result shows that HearASL can precisely recognize individual ASL words. In a user-independent test, the recognition accuracy across volunteers is shown in Fig. 14. The average accuracy is 90.8% on the one-hand gesture data set and 90.6% on the two-hand gesture data set. We can see that the lowest recognition accuracy is higher than 85% (the lowest is 86%). We owe the good generalization of the system to the CIR enhancement and attention module in our CNN structure, which supports the diversities of different individuals.

Sentence-Level Performance: We first perform the user-dependent test on the sentence-level sign language, we adopt a K-fold cross-validation strategy, where $K = 10$ in our implementation. It means that $K - 1$ of the folds are used to train the model and the remaining part of the data set is used to evaluate the performance. The average WER is only 0.9%. And we also evaluate the effect of training data size, the WER is depicted in Fig. 15. As we utilize GRU after CNN to further extract features from time-series data in our model, context information is also captured for sentence recognition. Thus, HearASL performs well in sentence-level recognition when we put 60% of data into the training process. In a user-independent test, the average WER is 3.4%. Fig. 16 shows the WER across volunteers (the highest is 10.2% and the lowest is 0.7%). To investigate the influence of the length of the selected sentence, we access the WER by using different lengths as the training data which is generated by seven participants in the user-independent test. Fig. 17 shows that the increase in sentence length leads to a decrease in WER. The WER is 12% when there are just two words in a sentence, it decreases to 0.2% when there have five words in a sentence since the longer sign language sentence has more context information.

Parameters Configuration: We evaluate the performance of different parameters configuration in our model which impacts the recognition accuracy. Specifically, we test the CNN + GRU + CTC model on the sentence-level data set by using

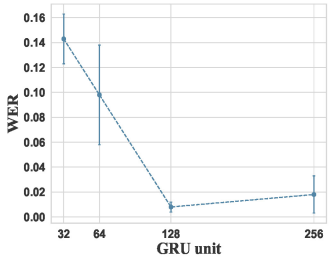


Fig. 18. Accuracy of sentence level with different GRU units.

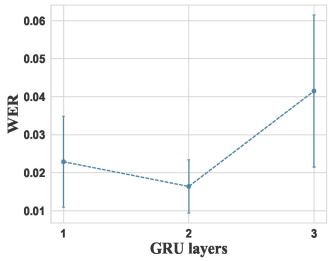


Fig. 19. Accuracy of sentence level with different GRU layers.

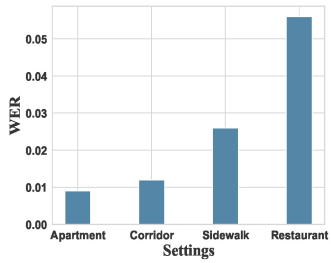


Fig. 20. Accuracy of sentence level with different settings.

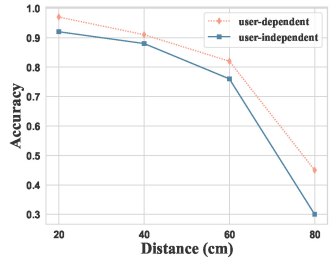


Fig. 21. Accuracy of word level with different distances to the receiver.

the different numbers of units and layers in GRU, we evaluate the WER of the sentence-level data set. Figs. 18 and 19 indicate that increasing units and layers can decrease WER. We have 0.8% WER at 128 GRU units and 1.64% WER at two GRU layers, respectively. However, WER increases when there have three layers and 256 GRU units, which may be due to the overfitting problem. Furthermore, more units and layers mean there has more parameters which increase the training time and a high memory loss. We train our model with a two-layer GRU with 64 units to balance the accuracy and the cost, and the average WER is 0.9% on the test data set.

E. In-the-Wild Evaluation

The evaluation in the wild is essential because our system for translation may be used outdoors more frequently. We utilize 18–22 kHz signal to capture reflected signal on people's sign language, which is far beyond the bandwidth of urban noises which ranges from 1–4 kHz and human speaking voice which usually ranges from 0.5–3 kHz [27]. Therefore, our system is resistant to direct ambient noise and speaking noise from surrounding people. However, the different multipath interference generated by different environments may affect the performance of our system (e.g., the collected CIR images may be affected by surrounding walking pedestrians due to the multipath interference).

In order to prove this, we provide the recognition accuracy in different settings including an apartment with an area of 8 m $\times$ 6 m, a corridor in the apartment, a sidewalk with several pedestrians, and a noisy restaurant. All four settings have the demand for sign language translation. We recruit two participants to act as an ASL signer and an ASL viewer using the smartphone to translate. The ASL signer performs 30 selected sign language sentences of different lengths with a repetition of 20 times in every setting. The data set from the apartment is used for training and the other is used for evaluating the performance. In a sidewalk setting, the ASL signer and viewer stand face to face, while one pedestrian walks during the translating process. The restaurant has more interference than the sidewalk (noise averaged at 60–80 dB) and more surrounding people. The experiment result is shown in Fig. 20, the performance in four settings has slight differences. The WER in the noisy restaurant is the highest and reaches 5.6%. And, it is reasonable that performance in the apartment is the best because there are less interference and the training samples are collected in it.

F. Robustness Quantification

In this part, we investigate the robustness of HearASL in different scenarios including different distances and angles between the microphone and hands. We also evaluate the performance of the data augmentation method.

Impact of Sensing Distances: As mentioned in Section V-B, the CIR images under different distances are different. We evaluate the performance of HearASL with four distances from hand to the receiver, including 20, 40, 60, and 80 cm. As shown in Fig. 21, our system performs well in the distance of 40 cm in either user-dependent or user-independent tests. However, when the distance increases to 80 cm, the accuracy decreases greatly. This is mainly because the sound intensity decreases due to the increasing distance. Another reason is that in order to get the CIR image in a larger sensing distance we need to increase L in (5), which impacts the accuracy of channel estimation. Considering our application scenarios, 50 cm between hands and the receiver is acceptable to cover the need of daily life.

Impact of Angles Between Signs and Smartphones: When people hold the smartphone to recognize the sign language, there may have different angles between hands and the receiver. To investigate the impact of different angles, we perform the selected ten sign language words within 40 cm

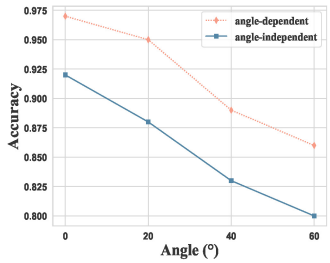


Fig. 22. Accuracy of word level with different angles.

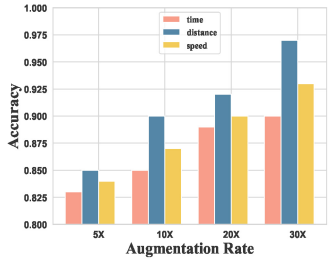


Fig. 23. Accuracy with different data augmentation strategies.

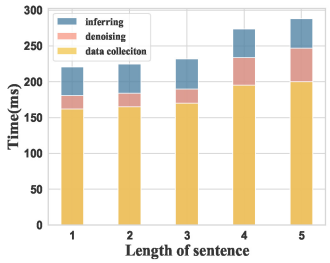


Fig. 24. Time consumption in different sentence lengths.

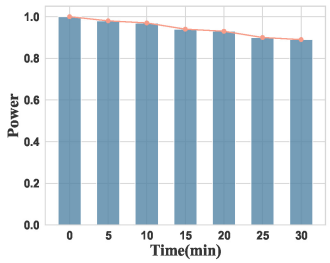


Fig. 25. Power consumption when running the application.

and change the angle from 0° to 60°. We collect data from all four kinds of angles, including 0°, 20°, 40°, and 60°. In the angle-dependent test, we apply the whole data set of the ten selected sign language words for training and testing. For angle-independent experiment, we also utilize any three out of four angle data sets for training and the remaining one for testing. The result is depicted in Fig. 22. It shows that the accuracy is similar in both angle-dependent and angle-independent test, which is mainly because the CIR images exhibit similar patterns when we perform the same gesture from different angles.

Impact of Data Augmentation: In our system, we apply three data augmentation strategies to increase the data for training. To validate the performance of single data augmentation strategy, we vary the data augmentation rate under different augmentation data set include different time, distances, and speeds. The result is depicted in Fig. 23. The results show that the performance improves when increasing the augmentation rate. And, different speeds and distances cause higher performance since the data covers more variations of the sign language.

G. System Running Performance

In this part, we evaluate the system delay of HearASL. We implement a client-server model in our system, the application in the smartphone is used for collecting audio samples with built-in smartphones and microphones and displaying the final recognition results. And all of processes run on the server, which is used for channel estimation, denoising, and inferring.

Running Time of Each Part: The time consumption of different parts in HearASL is shown in Fig. 24. The average time of data collection mainly depends on the length of the whole sentence. A sentence with less than three words usually last about 5 s. The average time consumption in data collection is about 165 ms. A sentence with five words usually lasts within 10 s which needs 190 ms to get a CIR image. The denoising procedure takes about 39 ms for a CIR image with four or five words and 19 ms for a CIR image with less than three words. Finally, all different length of words takes the network about 42 ms to give a final result. The average running time in one translation is 248 ms which indicates that HearASL can recognize ASL in real time.

Energy Consumption: To obtain the energy consumption of HearASL, we shut down all other applications in the smartphone and leave HearASL. At the same time, we monitor the power consumption of the smartphone which emits the designed WAV file and records the reflected signal continuously. As shown in Fig. 25, the power level decreases from 100% to 89% after 30 min. Since users can control sensing time and our system does not run all the time, the power consumption is acceptable for smartphones.

H. Comparison With Typical Related Works

Table III shows the comparison results between HearASL and other typical hand gesture systems based on the wireless signal in terms of signal type, number of sign language words, number of sign language sentences, gesture type, devices, portable, recognition algorithm, and recognition accuracy.

The former gesture recognition system RobuCIR [21] uses acoustic signals to classify self-defined 15 gestures, which achieves state-of-the-art recognition accuracy of gesture recognition systems. However, the number of gestures is limited and the system is not aimed for sign language. Meanwhile, the latter works based on mmWave [28] and WiFi lack portability, which decreases the user experience deeply. And, they do not give the solution of sign language sentence recognition. In contrast, HearASL can achieve portable SLR at the word level and sentence level. Among these works, SonicASL [13] also uses acoustic signals to recognize sign language gestures. However, the system needs people wearing a pair of earphones that

TABLE III
COMPARISON WITH OTHER WIRELESS-BASED HAND GESTURE RECOGNITION TECHNOLOGIES

Technologies	**HearASL**	SonicASL [13]	mmASL [28]	SignFi [12]	RobuCIR [21]
Signal	Acousitc	Acoustic	60GHz mmWave	5GHz WiFi	Acousitc
No. Words	50	42	50	276	15
No. Sentences	30	30	N/A	N/A	N/A
Gesture Type	Sign Language	Sign Language	Sign Language	Sign Language	Self-defined
Devices	Smartphone	Modified Earphone	Radio Platform	WiFi AP	Smartphone
Portable	Yes	Yes	No	No	Yes
Algorithm	CNN+GRU+CTC	CNN+LSTM+CTC	CNN	CNN	CNN+LSTM
Accuracy	97.2 %	93.8 %	87%	94%	98.4%

consist of an outward speaker to sense the opposite people's sign gestures. Compared with existed sign language systems via wireless signals, HearASL just needs a smartphone and achieves higher accuracy.

VIII. RELATED WORKS

We divide the existing SLR into three categories: 1) computer-vision-based; 2) wearable sensor-based; and 3) wireless signal-based.

Computer-Vision-Based SLR: Kuznetsova et al. [29] used a consumer depth camera and recognize static gestures under a multilayered random forest model. Huang et al. [30] recognized continuous sign language by a hierarchical attention network with latent space, which eliminates the preprocessing of temporal segmentation. There also have some works using commercial devices, such as Kinect [3], [4], [5], [6] and Leap Motion [7], [8]. For instance, Zafrulla et al. [6] proposed a new multimodal system that uses Kinect to recognize ASL. Chong and Lee [9] used the Leap Motion Controller to extract features from finger and hand motions to differentiate between the static and dynamic gestures. However, these computer-vision-based SLR systems are prone to be affected by illumination conditions and exist privacy leakage issues.

Wearable Sensor-Based SLR: Some researchers utilize wearable sensors to recognize sign language. Glove-based SLR systems implement multiple sensors on a glove and capture the sign language features [31], [32], [33], [34]. Unlike vision-based systems, it can perform recognition anytime and anywhere without mounting a camera around users. Kau et al. [32] proposed a hand gesture recognition glove for real-time translation of the Twaiwanese sign language. Seymour and Tšoeu [34] recognized the manual alphabet and manual numeric digits that have static gestures using an instrumented glove. These glove-based systems are not convenient for use in daily life due to the high cost. There also have some SLR systems using wearable accelerometers [35] or smartwatches [11] that cost less than glove-based systems. Wu et al. [36] proposed a real-time American SLR system by fusing inertial sensor modalities and the sEMG modality at the feature level. Hou et al. [10] used smartwatches to recognize isolated signs and continuous sentence-level sign language. However, the built-in inertial measurement units (IMUs) can usually recognize only hand/arm gestures, which is insufficient for fine-grained finger-level and two-handed signs.

Wireless Signal-Based SLR: Wireless sensing techniques (e.g., Wi-Fi [12], [37], mmWave [28], and acoustic signal [13], [14], [18], [21]) are less intrusive and avoid privacy issues for SLR. However, these systems are not suitable for outdoor application scenarios besides acoustic sensing-based methods. Sun et al. [14] realized fine-grained gesture-sensing on the back of mobile devices by measuring both the structure-borne and the air-borne signals. Ruan et al. [18] recognized six hand gestures by decoding the Doppler Shift into the hand-waving speed and range. Wang et al. [21] proposed a robust contact-free gesture recognition system based on acoustic signals transmitted by the smartphone. The most relevant work in acoustic sensing is [13], which is an ASLR system based on the acoustic signal. It leverages outward-facing microphones and speakers added to commodity earphones, which means the device is not pervasive enough. Moreover, the doppler shift extracted from sign gestures is associated with speed which means users should move fast to be detected. Compared to previous works in SLR, we do not need any dedicated devices and can recognize finger-level and two-handed gestures well.

IX. DISCUSSION AND FUTURE WORK

Sign Language Set: We recognize 50 sign language words and 30 sentences totally in our system. To make our system cover more words, the users of our system can help with the data collection process. By utilizing unsupervised learning technologies, the words that are commonly used but not in our sign language set can be clustered together. The sign set can be expanded by labeling them manually. In the future, we will expand our sign language set continuously and facilitate sign language learning and communications between deaf people and people with normal hearing.

Fingerspelling: The fingerspelling alphabet is used in sign language to spell out names of people and places for which there is no sign. To solve this problem, Hou et al. [10] employed word segmentation by a jitter between two alphabets. As most of the alphabet signs are static, the extracted features are mainly from conversion movements. The features may be different for the same alphabet sign in various locations of a word. Based on this limitation, in the future, we will ask signers to do a start sign gesture such as five fingers together every time before they do an alphabet sign. In this way, the features of every alphabet are the same as they both start from a specific sign. And the start sign gesture does not add too much burden on the signers because fingerspelling is not so long most of the time.

Nonmanual Makers Recognition: Nonmanual markers consist of various facial expressions, hand tilting, or mouthing that we add to "signs" to create or influence meaning [38]. In recent works, Xie et al. [39] designed an acoustic-based upper

facial action (UFA) recognition system using smart eyewear. In the future, we will explore and optimize our algorithms to recognize these nonmanual markers to build a more complete ASLR system.

X. Conclusion

Motivated by limitations of existing methods for ASL, we propose a novel end-to-end acoustic-based word-level and sentence-level ASLR system named HearASL, which only needs a smartphone to provide sign language translation anytime and anywhere. HearASL relies on the built-in speakers and microphones in smartphones to estimate channel information in different sign gestures. We take the extracted CIR as a image and put it into a CNN + GRU + CTC structure to recognize each word and sentence. To evaluate performance of HearASL, we implement our system on a smartphone. Results show that our approach enables users to recognize sign language in a promising recognition accuracy even in the wild and achieves real-time ability.

References

[1] "33 eye opening hearing loss facts and statistics." Hearsoundly. 2021. [Online]. Available: https://www.hearsoundly.com/guides/hearing-loss-statistics

[2] "What is American sign language (ASL)?" NIDCD. 2021. [Online]. Available: https://www.nidcd.nih.gov/health/american-sign-language

[3] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Proc. ECCV*, 2014, pp. 572–578.

[4] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," *J. Mach. Learn. Res.*, vol. 13, pp. 2205–2231, Jan. 2012.

[5] S. Lang, M. Block, and R. Rojas, "Sign language recognition using kinect," in *Proc. ICAISC*, 2012, pp. 394–402.

[6] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in *Proc. ICMI*, 2011, pp. 279–286.

[7] A. S. Elons, M. Ahmed, H. Shedid, and M. F. Tolba, "Arabic sign language recognition using leap motion sensor," in *Proc. IEEE ICCES*, 2014, pp. 368–373.

[8] C.-H. Chuan, E. Regina, and C. Guardino, "American sign language recognition using leap motion sensor," in *Proc. IEEE ICMLA*, 2014, pp. 541–544.

[9] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, p. 3554, 2018.

[10] J. Hou et al., "SignSpeaker: A real-time, high-precision smartwatch-based sign language translator," in *Proc. ACM MobiCom*, 2019, pp. 1–15.

[11] D. Ekiz et al., "Sign sentence recognition with smart watches," in *Proc. IEEE SIU*, 2017, pp. 1–4.

[12] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, 2018.

[13] Y. Jin et al., "SonicASL: An acoustic-based sign language gesture recognizer using earphones," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–30, 2021.

[14] K. Sun, T. Zhao, W. Wang, and L. Xie, "VSkin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proc. ACM Mobicom*, 2018, pp. 591–605.

[15] Y. Xie, F. Li, Y. Wu, and Y. Wang, "HearFit: Fitness monitoring on smart speakers via active acoustic sensing," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[16] Y. Xie, F. Li, Y. Wu, S. Yang, and Y. Wang, "D^3-guard: Acoustic-based drowsy driving detection using smartphones," in *Proc. IEEE INFOCOM*, 2019, pp. 1225–1233.

[17] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the doppler effect to sense gestures," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 1911–1914.

[18] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, "AudioGest: Enabling fine-grained hand gesture detection by decoding echo signal," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 474–485.

[19] "Barker code." Wolfram MathWorld. 2021. [Online]. Available: https://mathworld.wolfram.com/BarkerCode.html

[20] M. Pukkila, *Channel Estimation Modeling*, vol. 17, Nokia Res. Center, Espoo, Finland, 2000, p. 66.

[21] Y. Wang, J. Shen, and Y. Zheng, "Push the limit of acoustic gesture recognition," *IEEE Trans. Mobile Comput.*, vol. 21, no. 5, pp. 1798–1811, May 2022.

[22] Y. Sun and R. Fisher, "Object-based visual attention for computer vision," *Artif. Intell.*, vol. 146, no. 1, pp. 77–123, 2003.

[23] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, 2018, pp. 3–19.

[24] B. Su and S. Lu, "Accurate recognition of words in scenes without character segmentation using recurrent neural network," *Pattern Recognit.*, vol. 63, pp. 397–405, Mar. 2017.

[25] W. Vicars, "ASL American sign language." 2017. [Online]. Available: https://lifeprint.com/

[26] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[27] H. Hellberg. "Frequency, hertz & more: All about audiograms | miracle ear." 2021. [Online]. Available: https://www.miracle-ear.com

[28] P. S. Santhalingam, A. A. Hosain, D. Zhang, P. Pathak, H. Rangwala, and R. Kushalnagar, "mmASL: Environment-independent ASL gesture recognition using 60 GHz millimeter-wave signals," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–30, 2020.

[29] A. Kuznetsova, L. Leal-Taixé, and B. Rosenhahn, "Real-time sign language recognition using a consumer depth camera," in *Proc. ICCVW*, 2013, pp. 83–90.

[30] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proc. AAAI*, 2018, pp. 2257–2264.

[31] S. A. Mehdi and Y. N. Khan, "Sign language recognition using sensor gloves," in *Proc. IEEE ICONIP*, vol. 5, 2002, pp. 2204–2206.

[32] L.-J. Kau, W.-L. Su, P.-J. Yu, and S.-J. Wei, "A real-time portable sign language translation system," in *Proc. IEEE MWSCAS*, 2015, pp. 1–4.

[33] C. Preetham, G. Ramakrishnan, S. Kumar, A. Tamse, and N. Krishnapura, "Hand talk-implementation of a gesture recognizing glove," in *Proc. IEEE TIIEC*, 2013, pp. 328–331.

[34] M. Seymour and M. Tšoeu, "A mobile application for South African sign language (SASL) recognition," in *Proc. AFRICON*, 2015, pp. 1–5.

[35] Y. Li, X. Chen, X. Zhang, K. Wang, and Z. J. Wang, "A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2695–2704, Oct. 2012.

[36] J. Wu, Z. Tian, L. Sun, L. Estevez, and R. Jafari, "Real-time american sign language recognition using wrist-worn motion and surface EMG sensors," in *Proc. IEEE BSN*, 2015, pp. 1–6.

[37] L. Zhang, Y. Zhang, and X. Zheng, "WiSign: Ubiquitous american sign language recognition using commercial Wi-Fi devices," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–24, 2020.

[38] W. Vicars. "Non-manual markers in ASL (NMM's)." 2021. [Online]. Available: https://www.lifeprint.com/asl101/pages-layout/nonmanualmarkers.htm

[39] W. Xie, Q. Zhang, and J. Zhang, "Acoustic-based upper facial action recognition for smart eyewear," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–28, 2021.

Yusen Wang received the B.E. degree in computer science and technology from Jilin University, Changchun, China, in 2020. He is currently pursuing the master's degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His research interests include mobile computing, human–computer interaction, and acoustic sensing.

Fan Li (Member, IEEE) received the B.Eng. and first M.Eng. degrees in communications and information system from the Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, the second M.Eng. degree in electrical engineering from the University of Delaware, Newark, DE, USA, in 2004, and the Ph.D. degree in computer science from the University of North Carolina at Charlotte, Charlotte, NC, USA, in 2008.

She is currently a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. Her current research focuses on wireless networks, *ad hoc* and sensor networks, and mobile computing.

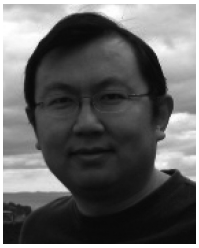
Dr. Li's papers won Best Paper Awards from IEEE MASS in 2013, IEEE IPCCC in 2013, ACM MobiHoc in 2014, and Tsinghua Science and Technology in 2015. She is a member of ACM.

Yadong Xie (Member, IEEE) received the B.E. degree in network engineering from Hebei University, Baoding, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His research interests include mobile computing, mobile health, human–computer interaction, and deep learning.

Chunhui Duan (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Software, Tsinghua University, Beijing, China, in 2013 and 2018, respectively.

Previously, she was a Postdoctoral Research Fellow with Tsinghua University. She is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include RFID, Internet of Things, wireless sensing, and mobile computing.



Yu Wang (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in computer science from Tsinghua University, Beijing, China, in 1998 and 2000, respectively, and the Ph.D. degree in computer science from Illinois Institute of Technology, Chicago, IL, USA, in 2004.

He is currently a Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. He has published over 200 papers in peer-reviewed journals and conferences, with four best paper awards. His research interest includes wireless networks, smart sensing, and mobile computing.

Prof. Wang is a recipient of the Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities in 2006 and the Outstanding Faculty Research Award from the College of Computing and Informatics, University of North Carolina at Charlotte in 2008. He has served as a general chair, a program chair, and a program committee member. for many international conferences (such as IEEE IPCCC, ACM MobiHoc, IEEE INFOCOM, IEEE GLOBECOM, and IEEE ICC). He has served as the Editorial Board Member of several international journals, including IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He is the ACM Distinguished Member in 2020.