DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation

Biyi Fang, Jillian Co, Mi Zhang
Michigan State University

ABSTRACT

There is an undeniable communication barrier between deaf people and people with normal hearing ability. Although innovations in sign language translation technology aim to tear down this communication barrier, the majority of existing sign language translation systems are either intrusive or constrained by resolution or ambient lighting conditions. Moreover, these existing systems can only perform single-sign ASL translation rather than sentence-level translation, making them much less useful in daily-life communication scenarios. In this work, we fill this critical gap by presenting *DeepASL*, a transformative deep learning-based sign language translation technology that enables ubiquitous and non-intrusive American Sign Language (ASL) translation at both word and sentence levels. DeepASL uses infrared light as its sensing mechanism to non-intrusively capture the ASL signs. It incorporates a novel hierarchical bidirectional deep recurrent neural network (HB-RNN) and a probabilistic framework based on Connectionist Temporal Classification (CTC) for word-level and sentence-level ASL translation respectively. To evaluate its performance, we have collected 7,306 samples from 11 participants, covering 56 commonly used ASL words and 100 ASL sentences. DeepASL achieves an average 94.5% word-level translation accuracy and an average 8.2% word error rate on translating unseen ASL sentences. Given its promising performance, we believe DeepASL represents a significant step towards breaking the communication barrier between deaf people and hearing majority, and thus has the significant potential to fundamentally change deaf people's lives.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility technologies**;
• **Computing methodologies** → *Neural networks*;

KEYWORDS

Deep Learning; Sign Language Translation; Assistive Technology; Mobile Sensing Systems; Human-Computer Interaction

ACM Reference Format:
Biyi Fang, Jillian Co, Mi Zhang. 2017. *DeepASL*: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. In *Proceedings of 15th ACM Conference on Embedded Networked Sensor Systems, Delft, The Netherlands, November 6–8, 2017 (SenSys'17),* 13 pages.

SenSys'17, November 6–8, 2017, Delft, The Netherlands

ACM ISBN 978-1-4503-5459-2/17/11…$15.00
https://doi.org/10.1145/3131672.3131693

https://doi.org/10.1145/3131672.3131693

1 INTRODUCTION

In the United States, there are over 28 million people considered deaf or hearing disabled [2]. American Sign Language, or ASL in short, is the primary language used by deaf people to communicate with others [3]. Unfortunately, very few people with normal hearing understand sign language. Although there are a few methods for aiding a deaf person to communicate with people who do not understand sign language, such as seeking help from a sign language interpreter, writing on paper, or typing on a mobile phone, each of these methods has its own key limitations in terms of cost, availability, or convenience. As a result, there is an undeniable communication barrier between deaf people and hearing majority.

At the heart of tearing down this communication barrier is the sign language translation technology. Sign language is a language like other languages but based on signs rather than spoken words. A sign language translation system uses sensors to capture signs and computational methods to map the captured signs to English. Over the past few decades, although many efforts have been made, sign language translation technology is still far from being practically useful. Specifically, existing sign language translation systems use motion sensors, Electromyography (EMG) sensors, RGB cameras, Kinect sensors, or their combinations [10, 11, 28, 40, 46] to capture signs. Unfortunately, these systems are either intrusive where sensors have to be attached to fingers and palms of users, lack of resolutions to capture the key characteristics of signs, or significantly constrained by ambient lighting conditions or backgrounds in real-world settings. More importantly, existing sign language translation systems can only translate a single sign at a time, thus requiring users to pause between adjacent signs. These limitations significantly slow down face-to-face conversations, making those sign language translation systems much less useful in daily-life communication scenarios.

In this paper, we present *DeepASL*, a transformative deep learning-based sign language translation technology that enables non-intrusive ASL translation at both word and sentence levels. DeepASL can be embedded inside a wearable device, a mobile phone, a tablet, a laptop, a desktop computer, or a cloud server to enable ubiquitous sign language translation. As such, DeepASL acts as an *always-available virtual sign language interpreter*, which allows deaf people to use their primary language to communicate with the hearing majority in a natural and convenient manner. As an example, Figure 1 illustrates an envisioned scenario where DeepASL is in the form of a wearable device, enabling a deaf person and a hearing individual who does not understand ASL to use their own primary languages to communicate with each other face to face. Specifically, from one side, DeepASL translates signs performed by the deaf person into spoken English; from the other side, DeepASL leverages the

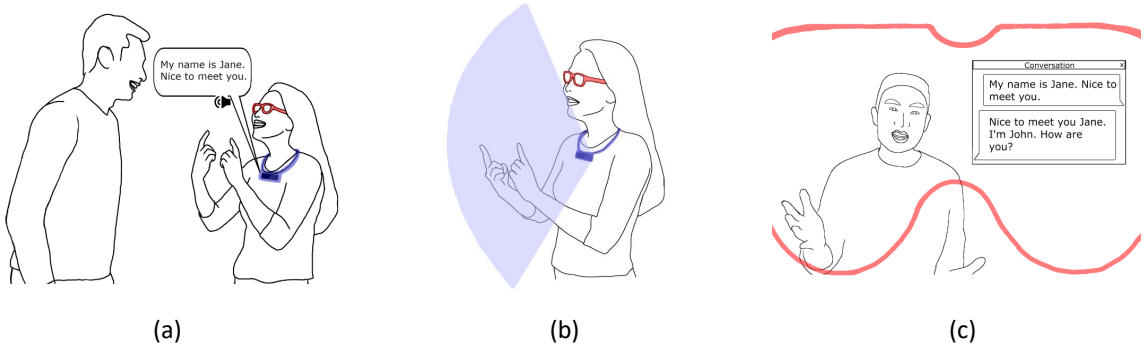

Figure 1: Illustration of an envisioned scenario of real-time two-way communication enabled by DeepASL: (a) DeepASL translates the signs performed by the deaf person into spoken English and broadcasts the translated ASL sentence via a speaker; (b) DeepASL captures the signs in a non-intrusive manner; (c) DeepASL leverages the speech recognition technology to translate spoken English into texts, and projects the texts through a pair of augmented reality (AR) glasses.

speech recognition technology to translate English spoken from the hearing individual into text, and projects the text through a pair of augmented reality (AR) glasses for the deaf person to read.

DeepASL uses Leap Motion [4] – an infrared light-based sensing device that can extract the skeleton joints information of fingers, palms and forearms – to non-intrusively capture the ASL signs performed by a deaf person. By leveraging the extracted skeleton joints information, DeepASL achieves word and sentence-level ASL translation via three innovations. First, DeepASL leverages domain knowledge of ASL to extract the key characteristics of ASL signs buried in the raw skeleton joints data. Second, DeepASL employs a novel hierarchical bidirectional deep recurrent neural network (HB-RNN) to effectively model the spatial structure and temporal dynamics of the extracted ASL characteristics for word-level ASL translation. Third, DeepASL adopts a probabilistic framework based on Connectionist Temporal Classification (CTC) [19] for sentence-level ASL translation. This eliminates the restriction of pre-segmenting the whole sentence into individual words, and thus enables translating the whole sentence end-to-end directly without requiring users to pause between adjacent signs. Moreover, it enables DeepASL to translate ASL sentences that are not included in the training dataset, and hence eliminates the burden of collecting all possible ASL sentences.

Summary of Experimental Results: We have conducted a rich set of experiments to evaluate the performance of DeepASL in three aspects: 1) ASL translation performance at both word level and sentence level; 2) robustness of ASL translation under various real-world settings; and 3) system performance in terms of runtime, memory usage and energy consumption. Specifically, to evaluate the ASL translation performance, we have collected 7,306 samples from 11 participants, covering 56 commonly used ASL words and 100 ASL sentences. To evaluate the robustness, we have collected 1,178 samples under different ambient lighting conditions, body postures when performing ASL, and scenarios with in-the-scene interference and multi-device interference. To evaluate the system performance, we have implemented DeepASL on three platforms with different computing power: 1) a desktop equipped with an Intel i7-4790 CPU and a Nvidia GTX 1080 GPU (desktop CPU and GPU), 2) a Nvidia Jetson TX1 mobile development board equipped with an ARM Cortex-A57 CPU and a Nvidia Tegra X1 GPU (mobile CPU and GPU), and 3) a Microsoft Surface Pro 4 tablet equipped with an Intel i5-6300 CPU (tablet CPU). Our results show that:

- At the word level, DeepASL achieves an average 94.5% translation accuracy. At the sentence level, DeepASL achieves an average 8.2% word error rate on translating unseen ASL sentences and an average 16.1% word error rate on translating ASL sentences performed by unseen users.
- DeepASL achieves more than 91.8% word-level ASL translation accuracy in various ambient lighting conditions, body postures, and interference sources, demonstrating its great robustness in real-world daily communication scenarios.
- DeepASL achieves 282 ms in runtime performance in the worst-case scenario across three platforms for both word-level and sentence translation. It also demonstrates the capability of supporting enough number of inferences for daily usage on both mobile and tablet platforms.

Summary of Contributions: The development of sign language translation technology dates back to the beginning of 90s [44]. However, due to the limitations in both sensing technology and computational methods, limited progress has been made over the decades. The innovative solution provided by DeepASL effectively addresses those limitations, and hence represents a significant contribution to the advancement of sign language translation technology. Moreover, the development of DeepASL enables a wide range of applications. As another contribution of this work, we have designed and developed two prototype applications on top of DeepASL to demonstrate its practical value.

According to World Health Organization (WHO), there are an estimated 360 million people worldwide having disabling hearing loss [6]. While the focus of this paper is on American sign language translation, since our approach is generic at modeling signs expressed by hands, it can be leveraged for developing sign language translation technologies for potentially any of the three hundred sign languages in use around the world [7]. Given its promising performance, we believe DeepASL represents a significant step towards breaking the communication barrier between deaf people and hearing majority, and thus has the significant potential to fundamentally change deaf people's lives.

2 BACKGROUND, STATE-OF-THE-ART, AND DESIGN CHOICE

2.1 Characteristics of ASL

ASL is a complete and complex language that mainly employs signs made by moving the hands [23]. Each individual sign is characterized by three key sources of information: 1) hand shape, 2) hand movement, and 3) relative location of two hands [23, 29]. It is the combination of these three key characteristics that encodes the meaning of each sign. As an example, Figure 2 illustrates how these three characteristics altogether encode the meaning of two ASL signs: "small" and "big". Specifically, to sign "small", one starts with holding both hands in front of her with fingers closed (i.e., hand shape), and then moves two hands towards each other (i.e., hand movement and relative location). In comparison, to sign "big", one starts with extending the thumb and index fingers to form a slightly bent 'L' shape (i.e., hand shape), and then moves two hands away from each other (i.e., hand movement and relative location).

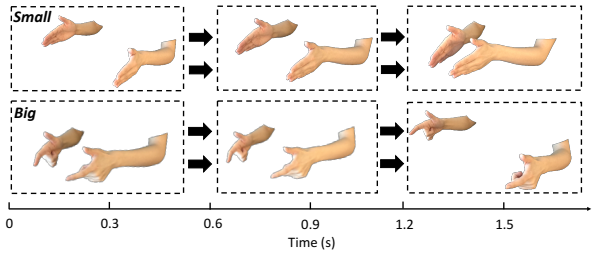

Figure 2: Illustration on how hand shape, hand movement, and relative location of two hands altogether encodes the meaning of two ASL signs: "small" and "big".

It is worthwhile to note that, for illustration purpose, we have selected two of the most distinctive ASL signs to explain the characteristics of ASL. In fact, there are many ASL signs that involve very subtle differences in the three key characteristics mentioned above. Moreover, in real-world scenarios, ASL signs can be expressed under various conditions such as bright vs. poor lighting conditions, walking vs. standing; and indoor vs. outdoor environments. It is the subtle differences and the real-world factors altogether that makes the task of ASL translation challenging.

2.2 State-of-the-Art ASL Translation Systems

Based on the sensing modality the system uses, existing ASL translation systems can be generally grouped into four categories: 1) wearable sensor-based, 2) Radio Frequency (RF)-based, 3) RGB camera-based, and 4) Kinect-based systems. However, each of them has fundamental limitations that prevent it from being practically useful for translating ASL in daily life scenarios. Specifically, wearable sensor-based systems [8, 24–28, 36, 42, 46] use motion sensors (accelerometers, gyroscopes), EMG sensors, or bend sensors to capture the movements of hands, muscle activities, or bending of fingers to infer the performed signs. However, wearable sensor-based systems require attaching sensors to a user's fingers, palms, and forearms. This requirement makes them very intrusive and impractical for daily usage. RF-based systems [32] use wireless signals as a sensing mechanism to capture hand movements. Although this contactless sensing mechanism minimizes the intrusiveness to users, wireless signals have very limited resolutions to "see" the hands. RGB camera-based systems [10, 40, 47], on the other hand, are capable of capturing rich information about hand shape and hand movement without instrumenting users. However, they fail to reliably capture those information in poor lighting conditions or generally uncontrolled backgrounds in real-world scenarios. Moreover, the videos/images captured may be considered invasive to the privacy of the user and surrounding bystanders. Finally, although Kinect-based systems overcome the lighting and privacy issues of the RGB camera-based systems by only capturing the skeleton information of the user body and limbs [11, 12], they do not have enough resolution to capture the hand shape information, which plays a critical role on decoding the sign language.

2.3 Design Choice

In the design of DeepASL, we use Leap Motion as our sensing modality to capture ASL signs [4]. Leap Motion overcomes the fundamental limitations of existing technologies and is able to precisely capture the three key characteristics of ASL signs under real-world scenarios in a non-intrusive manner. Specifically, Leap Motion uses infrared light as its sensing mechanism. This not only enables it to capture the signs in a contactless manner but also makes it "see" the signs in poor lighting conditions. Moreover, Leap Motion is able to extract skeleton joints of the fingers, palms and forearms from the raw infrared images. This preserves the privacy of the user and bystanders, and more importantly, provides enough resolution to precisely capture hand shape as well as hand movements and locations. As an example, Figure 3 illustrates how the ASL signs of two words "small" and "big" are precisely captured by the temporal sequence of skeleton joints of the fingers, palms and forearms.

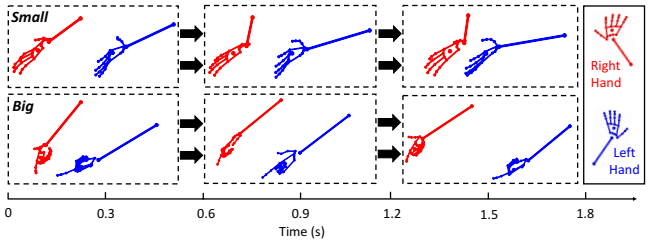

Figure 3: The skeleton joints of two ASL signs: "small" and "big".

To sum up, Table 1 compares Leap Motion with other sensing modalities used in existing sign language translation systems. As listed, Leap Motion has shown its superiority over other sensing modalities on capturing the three key characteristics of ASL signs in a non-intrusive manner without the constraint of ambient lighting condition. We leverage this superiority in the design of DeepASL.

Sensing Modality	Hand Shape	Hand Movement	Hand Location	Intrusive	Lighting Condition
Motion + EMG + Bend	Captured	Captured	No	Yes	Any
RF	No	Captured	No	No	Any
RGB Camera	Captured	Captured	Captured	Yes	Constrained
Kinect	No	Captured	Captured	No	Any
Leap Motion	Captured	Captured	Captured	No	Any

Table 1: Comparison of sensing modalities for ASL translation.

3 CHALLENGES AND OUR SOLUTIONS

Although Leap Motion has shown its superiority over other sensing modalities on capturing key characteristics of ASL signs, there is a *significant gap* between the raw skeleton joints data and the translated ASL. In this section, we describe the challenges on transforming the raw skeleton joints data into translated ASL at both word and sentence levels. We also explain how DeepASL effectively addresses those challenges.

ASL Characteristics Extraction: Leap Motion is *not* designed for ASL translation. Although Leap Motion captures the skeleton joints of the fingers, palms and forearms, the key information that characterizes ASL signs (i.e., hand shape, hand movement, and relative location of two hands) is still buried in the raw skeleton joints data. To address this challenge, we leverage domain knowledge of ASL to extract spatio-temporal trajectories of ASL characteristics from the sequence of skeleton joints during signing, and develop models upon the extracted ASL characteristics for ASL translation.

ASL Characteristics Organization: The extracted ASL characteristics are isolated and unorganized, and thus can not be directly used for ASL translation. This problem is exacerbated when the number of ASL signs to be translated scales up. To address this challenge, we propose a hierarchical model based on deep recurrent neural network (RNN) that effectively integrates the isolated low-level ASL characteristics into an organized high-level representation that can be used for ASL translation.

Similarity between Different Signs: Although each ASL sign is uniquely characterized by its ASL characteristics trajectories, many ASL signs share very similar characteristics at the beginning of their trajectories (see Figure 4 as an example). This similarity confuses traditional RNN which is based on a unidirectional architecture. This is because the unidirectional architecture can only use the past information at each time point in the trajectory to infer the sign being performed. To address this challenge, we propose a bidirectional RNN model which performs inference at each point of the trajectory based on both past and future trajectory information. With the global view of the entire trajectory, our bidirectional RNN model is able to achieve better ASL translation performance.

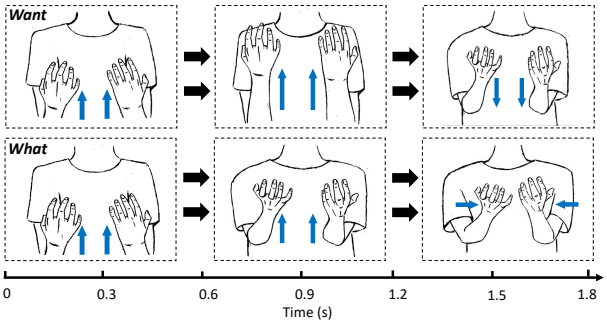

Figure 4: Similarity between two ASL signs: "want" and "what".

ASL Sentence Translation: To translate ASL sentences, existing sign language translation technologies adopt a framework which requires pre-segmenting individual words within the sentence. However, this framework restricts sign language translation technologies to translate one single sign at a time and thus requires users to pause between adjacent signs when signing one sentence. To address this challenge, we propose to adopt a framework based on Connectionist Temporal Classification (CTC) that computes the probability of the whole sentence directly, and therefore, removes the requirement of pre-segmentation.

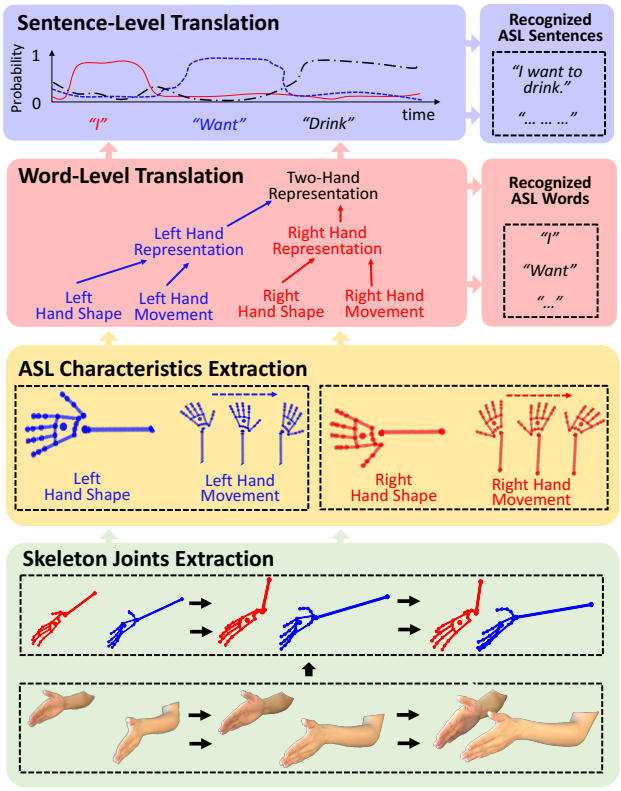

Figure 5: The system architecture of DeepASL.

To the best of our knowledge, DeepASL is the first ASL translation framework that addresses these challenges and achieves accurate ASL translation performance at word and sentence levels.

4 SYSTEM OVERVIEW

Figure 5 provides an overview of the multi-layer system architecture of DeepASL. Specifically, at the first layer, a temporal sequence of 3D coordinates of the skeleton joints of fingers, palms and forearms is captured by the Leap Motion sensor during signing. At the second layer, the key characteristics of ASL signs including hand shape, hand movement and relative location of two hands are extracted from each frame of the sequence, resulting in a number of spatio-temporal trajectories of ASL characteristics. At the third layer, DeepASL employs a hierarchical bidirectional deep recurrent neural network (HB-RNN) that models the spatial structure and temporal dynamics of the spatio-temporal trajectories of ASL characteristics for word-level ASL translation. Finally, at the top layer, DeepASL adopts a CTC-based framework that leverages the captured probabilistic dependencies between words in one complete sentence and translates the whole sentence end-to-end without requiring users to pause between adjacent signs. In the next section, we describe the design of DeepASL in details.

5 SYSTEM DETAILS

5.1 ASL Characteristics Extraction

The skeleton joints data provided by the Leap Motion sensor is noisy in its raw form. As our first step, we apply a simple Savitzky-Golay filter [37] to improve the signal to noise ratio of the raw skeleton joints data. We select the Savitzky-Golay filter because of its effectiveness in smoothing skeleton joints data [16, 48]. Specifically, let $J_{i,j,t} = (x_{i,j,t}, y_{i,j,t}, z_{i,j,t})$, $i = \{left, right\}$, $j = \{1, ..., N\}$, $t = \{1, ..., T\}$ denote the t-th frame of the temporal sequence of the 3D coordinates of the skeleton joints of fingers, palms and forearms of a single ASL sign, where x, y, z denote the 3D coordinates of the skeleton joints, i is the hand index, j is the skeleton joint index (see Figure 6 for the skeleton joints tracked by the Leap Motion sensor), t is the frame index, N denotes the total number of skeleton joints in one hand, and T denotes the total number of frames included in the temporal sequence. The Savitzky-Golay filter is designed as

$$\widetilde{J}_{i,j,t} = (-3J_{i,j,t-2} + 12J_{i,j,t-1} + 17J_{i,j,t} + 12J_{i,j,t+1} - 3J_{i,j,t+2})/35 \quad (1)$$

where $\widetilde{J}_{i,j,t}$ denotes the smoothed 3D coordinates of the skeleton joints in the t-th frame.

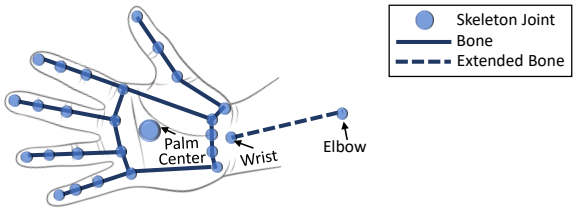

Figure 6: The skeleton joints tracked by the Leap Motion sensor.

Based on the smoothed temporal sequence of the skeleton joints data, we extract the key characteristics of ASL signs including hand shape, hand movement and relative location of two hands from each frame of the sequence. Specifically, since hand shape is independent of the absolute spatial location of the hand and is characterized by the relative distances among skeleton joints of palm and fingers, we extract hand shape information of both left and right hands by zero-centering the palm center of the right hand and then normalizing the 3D coordinates of the skeleton joints to it as

$$S_{i,j,t} = \widetilde{J}_{i,j,t} - \widetilde{J}_{i,j=right\_palm\_center,t}. \quad (2)$$

By doing this, the information of the relative location of the left hand to the right hand is also encoded in $S_{i=left,j,t}$. Lastly, we extract hand movement information of both left and right hands as the spatial displacement of each skeleton joint between two consecutive frames defined as

$$M_{i,j,t} = \begin{cases} (0,0,0), & \text{if } t = 1 \\ \widetilde{J}_{i,j,t} - \widetilde{J}_{i,j,t-1}, & \text{if } t = 2, ..., T. \end{cases} \quad (3)$$

Taken together, the ASL characteristics extracted from each frame of the temporal sequence of 3D coordinates of the skeleton joints result in four spatio-temporal ASL characteristics trajectories that capture information related to: 1) right hand shape, 2) right hand movement, 3) left hand shape (it also encodes the information of the relative location of the left hand to the right hand), and 4) left hand movement, respectively. We denote them as S_{right}, M_{right}, S_{left}, and M_{left} accordingly.

5.2 Word-Level ASL Translation

In this section, we first provide the background knowledge of bidirectional recurrent neural network (B-RNN) and Long Short-Term Memory (LSTM) to make the paper self-contained. We then describe our proposed hierarchical bidirectional deep recurrent neural network (HB-RNN) which is designed upon B-RNN and LSTM for single-sign word-level ASL translation. Finally, we describe the architectures of four comparative models that we use to validate the design choice of our proposed model.

5.2.1 *A Primer on Bidirectional RNN and LSTM.*

RNN is a powerful model for sequential data modeling [18]. It has been widely used and has shown great success in many important tasks such as speech recognition [20], natural language processing [39], language translation [41], and video recognition [15]. Specifically, given an input temporal sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$, where in our case x_t is the t-th frame of the spatio-temporal ASL characteristics trajectories, the hidden states of a recurrent layer $\mathbf{h} = (h_1, h_2, ..., h_T)$ and the output $\mathbf{y} = (y_1, y_2, ..., y_T)$ of a RNN can be obtained as:

$$h_t = \theta_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (4)$$

$$y_t = \theta_y(W_{ho}h_t + b_o) \quad (5)$$

where W_{xh}, W_{hh}, and W_{ho} are connection weight matrices, b_h and b_o are bias values, and θ_h and θ_y are activation functions.

The RNN model described above only contains a single recurrent hidden layer and is unidirectional. One limitation of the unidirectional RNN is that it could only look backward and thus can only access the past information at each time point in the temporal sequence for inference. In the context of sign language translation, this limitation could cause translation errors when different signs share very similar characteristics at the beginning of the signs. To address this limitation, we propose to use bidirectional RNN (B-RNN) [38] as the building block in our design. Figure 7 illustrates the network architecture of a B-RNN. As shown, B-RNN has two separate recurrent hidden layers, with one pointing backward (i.e., backward layer) and the other pointing forward (i.e., forward layer). As such, a B-RNN can look both backward and forward, and can thus utilize both the past and future information at each time point in the temporal sequence to infer the sign being performed.

The recurrent structure of RNN enables it to learn complex temporal dynamics in the temporal sequence. However, it can be difficult to train a RNN to learn long-term dynamics due to the vanishing and exploding gradients problem [21]. To solve this problem, Long

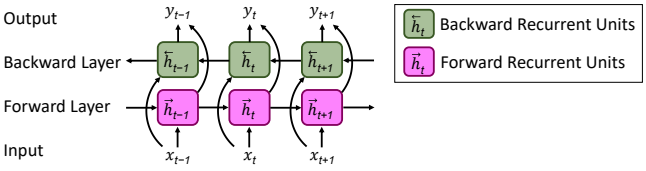

Figure 7: The network architecture of bidirectional RNN (B-RNN).

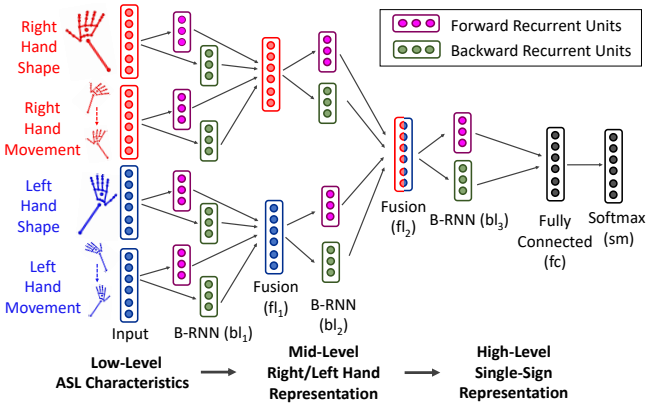

Figure 8: The architecture of the hierarchical bidirectional deep recurrent neural network (HB-RNN) for word-level ASL translation. For ASL signs that are performed using only one hand, only the corresponding half of the HB-RNN model is activated.

Short-Term Memory (LSTM) [22] was invented which enables the network to learn when to forget previous hidden states and when to update hidden states given new input. This mechanism makes LSTM very efficient at capturing the long-term dynamics. Given this advantage, we use B-RNN with LSTM architecture in our design to capture the complex temporal dynamics during signing.

5.2.2 Hierarchical Bidirectional RNN for Single-Sign Modeling. Although four spatio-temporal ASL characteristics trajectories have been extracted from the raw skeleton joints data, they are isolated and at low level, and thus can not be directly used for word-level ASL translation. Therefore, we propose a hierarchical model based on bidirectional deep recurrent neural network with the LSTM architecture (HB-RNN) to integrate the isolated low-level ASL characteristics into an organized high-level representation that can be used for word-level ASL translation.

Figure 8 illustrates the architecture of the proposed HB-RNN model. At a high level, our HB-RNN model takes the four spatio-temporal ASL characteristics trajectories as its input, extracts the spatial structure and the temporal dynamics within the trajectories, and combines them in a hierarchical manner to generate an integrated high-level representation of a single ASL sign for word-level ASL translation. As shown, our HB-RNN model consists of seven layers including three B-RNN layers ($bl_{1,2,3}$), two fusion layers ($fl_{1,2}$), one fully connected layer (fc), and one softmax layer (sm). Each of these layers has different structure and thus plays different role in the whole model. Specifically, in the bl_1 layer, the four spatio-temporal ASL characteristics trajectories that capture information related to the right hand shape ($\boldsymbol{S}_{right}$), right hand movement ($\boldsymbol{M}_{right}$), left hand shape ($\boldsymbol{S}_{left}$), and left hand movement ($\boldsymbol{M}_{left}$) are fed into four separate B-RNNs. These B-RNNs capture the spatial structure among skeleton joints and transform the low-level ASL characteristics into new representations of right hand shape, right hand movement, left hand shape, and left hand movement in both forward layer $\overrightarrow{h}$ and backward layer $\overleftarrow{h}$. In the fusion layer fl_1, we concatenate the newly generated representations of right (left) hand shape and right (left) hand movement together as $R^{i,t}_{bl_1} = \{\overrightarrow{h}^t_{bl_1}(\boldsymbol{S}^t_i), \overleftarrow{h}^t_{bl_1}(\boldsymbol{S}^t_i), \overrightarrow{h}^t_{bl_1}(\boldsymbol{M}^t_i), \overleftarrow{h}^t_{bl_1}(\boldsymbol{M}^t_i)\}$, $i = \{right, left\}$, and feed the two concatenations into two B-RNNs in the bl_2 layer separately to obtain an integrated representation of the right (left) hand. Similarly, the two newly generated right and left hand representations are further concatenated together in the fusion layer fl_2 denoted as $R^t_{bl_2}$. This concatenation is then fed into the B-RNN in the bl_3 layer to obtain a high-level representation in both forward layer and backward layer (denoted as $\overrightarrow{h}^t_{bl_3}(R^t_{bl_2})$ and $\overleftarrow{h}^t_{bl_3}(R^t_{bl_2})$) that integrates all the ASL characteristics of a single sign. Finally, we connect $\overrightarrow{h}^t_{bl_3}(R^t_{bl_2})$ and $\overleftarrow{h}^t_{bl_3}(R^t_{bl_2})$ to the fully connected layer fc. The output of fc is summed up across all the frames in the temporal sequence and then normalized by the softmax function in the softmax layer sm to calculate the predicted word class probability given a sequence J:

$$O = \sum_{t=1}^{T} O^t_{fc} \tag{6}$$

$$p(C_k|J) = \frac{e^{O_k}}{\sum_{n=1}^{C} e^{O_n}}, k = 1, ..., C \tag{7}$$

where C denotes the total number of ASL words in the dictionary. By accumulating results and normalizing across all the frames, our model is able to make inference based on the information of the entire sequence. More importantly, it allows our model to handle ASL signs that have different sequence lengths as well as sequence length variation caused by signing speed.

5.2.3 Comparative Models. To validate the design choice of our proposed HB-RNN model, we construct four comparative models as follows:

- **HB-RNN-M:** a hierarchical bidirectional RNN model with hand movement information only. We compare this model with HB-RNN to prove the importance of the hand shape information.
- **HB-RNN-S:** a hierarchical bidirectional RNN model with hand shape information only. We compare this model with HB-RNN to prove the importance of the hand movement information.
- **SB-RNN:** a simple bidirectional RNN model without hierarchical structure. We compare this model with HB-RNN to prove the importance of the hierarchical structure.
- **H-RNN:** a hierarchical unidirectional RNN model with forward recurrent layer only. We compare this model with HB-RNN to prove the significance of the bidirectional connection.

The parameters of the proposed HB-RNN model as well as the four comparative models are listed in Table 2.

5.3 Sentence-Level ASL Translation

In daily-life communication, a deaf person does not sign a single word but a complete sentence at a time. Although the HB-RNN model described in the previous section is capable of transforming the low-level ASL characteristics into a high-level representation for word-level translation, when translating a complete ASL sentence, HB-RNN still requires pre-segmenting the whole sentence into individual words and then connecting every translated word into a sentence in the post-processing. This is not only complicated but

Category	Model	RNN Layer 1	RNN Layer 2	RNN Layer 3
One-Hand ASL Words	HB-RNN-M	$2 \times 1 \times 128$	-	-
	HB-RNN-S	$2 \times 1 \times 128$	-	-
	SB-RNN	$2 \times 1 \times 128$	-	-
	H-RNN	$1 \times 2 \times 64$	$1 \times 1 \times 128$	-
	HB-RNN	$2 \times 2 \times 32$	$2 \times 1 \times 64$	-
Two-Hand ASL Words	HB-RNN-M	$2 \times 2 \times 64$	$2 \times 1 \times 128$	-
	HB-RNN-S	$2 \times 2 \times 64$	$2 \times 1 \times 128$	-
	SB-RNN	$2 \times 1 \times 256$	-	-
	H-RNN	$1 \times 4 \times 64$	$1 \times 2 \times 64$	$1 \times 1 \times 128$
	HB-RNN	$2 \times 4 \times 32$	$2 \times 2 \times 32$	$2 \times 1 \times 64$

Table 2: The parameters of our proposed HB-RNN model and the four comparative models. The parameters follow the format of 1 (unidirectional) or 2 (bidirectional) × #RNNs × #hidden units.

also requires users to pause between adjacent signs when signing one sentence, which is not practical in daily-life communication.

To address this problem, we propose a probabilistic approach based on Connectionist Temporal Classification (CTC) [19] for sentence-level ASL translation. CTC is the key technique that drives the modern automatic speech recognition systems such as Apple Siri and Amazon Alexa [31]. It eliminates the necessity of word pre-segmentation and post-processing, allowing end-to-end translation of a whole sentence. Inspired by its success on sentence-level speech recognition, we propose a CTC-based approach that can be easily built on top of the HB-RNN model described in the previous section for sentence-level ASL translation. Specifically, to realize sentence-level ASL translation based on CTC, we make the following modifications on HB-RNN:

- Let V denote the ASL word vocabulary. We add a blank symbol $\{blank\}$ into the ASL word vocabulary: $V' = V \cup \{blank\}$. Essentially, this blank symbol enables us to model the transition from one word to another within a single sentence.
- We increase the capacity of the RNN layers (i.e., bl_1, bl_2 and bl_3) in HB-RNN to $2 \times 4 \times 32$, $2 \times 2 \times 64$, and $2 \times 1 \times 128$, respectively (see Table 2 for the format definition). This is because ASL sentences are more complex than ASL words and thus require more parameters for modeling.
- Since an ASL sentence consists of multiple signs, we replace the softmax layer in HB-RNN which computes the probability of a single sign with a new softmax layer which computes the probabilities of a sequence of multiple signs.
- Based on the modified softmax layer, the probabilities of all the possible sentences formed by the word included in V can be computed. Given those probabilities, we compute the probability of a target label sequence by marginalizing over all the sequences that are defined as equivalent to this sequence. For example, the label sequence $'SL'$ is defined as equivalent to the label sequences $'SSL'$, $'SLL'$, $'S\_L'$ or $'SL\_'$, where $'\_'$ denotes the blank symbol $\{blank\}$. This process not only eliminates the need for word pre-segmentation and post-processing but also addresses variable-length sequences.
- Finally, we delete adjacent duplicate labels and remove all the blank symbols in the inferred label sequence to derive the translated sentence.

With all the above modifications, the end-to-end sentence-level ASL translation is achieved.

6 EVALUATION

6.1 Experimental Setup

6.1.1 *Dataset Design.*

To evaluate the translation performance of DeepASL at both word and sentence levels as well as its robustness under real-world settings, we have designed and collected three datasets: 1) ASL Words Dataset; 2) ASL Sentences Dataset; and 3) In-the-Field Dataset.

ASL Words Dataset: Since it is impossible to collect all the words in the ASL vocabulary, we target ASL words that are representative of each category of the ASL vocabulary. In particular, we have selected 56 ASL words from five word categories: pronoun, noun, verb, adjective and adverb. Table 3 lists the selected 56 ASL words. These words are among the top 200 most commonly used words in ASL vocabulary. Among these 56 words, 29 are performed by two hands and the rest 27 are performed by one hand (right hand).

Category	Words
pronoun	who, I, you, what, we, my, your, other
noun	time, food, drink, mother, clothes, box, car, bicycle, book, shoes, year, boy, church, family
verb	want, dontwant, like, help, finish, need, thankyou, meet, live, can, come
adjective	big, small, hot, cold, blue, red, gray, black, green, white, old, with, without, nice, bad, sad, many, sorry, few
adverb	where, more, please, but

Table 3: The ASL Words Dataset (two-hand words are underlined).

ASL Sentences Dataset: We have followed the dataset design methodology used in Google's LipNet (i.e., sentence-level lipreading) [9] to design our ASL Sentences Dataset. Specifically, we design the ASL sentences by following a simple sentence template: $subject^{(4)} + predicate^{(4)} + attributive^{(4)} + object^{(4)}$, where the superscript denotes the number of word choices for each of the four word categories, which are designed to be $\{I, you, mother, who\}$, $\{dontwant, like, want, need\}$, $\{big, small, cold, more\}$ and $\{time, food, drink, clothes\}$, respectively. Based on this sentence template, a total of 256 possible sentences can be generated. Out of these 256 sentences, we hand picked 100 meaningful sentences that people would use in daily communication. Example meaningful sentences are "I need more food" and "Who want cold drink".

In-the-Field Dataset: In daily life, deaf people may need to use ASL to communicate with others under various real-world settings. We consider three common real-world factors that can potentially affect the ASL translation performance: 1) lighting conditions, 2) body postures; and 3) interference sources. For lighting conditions, we collected data from both indoor poor lighting scenario and outdoor bright sunlight scenario. For body postures, we collected data when signs are performed while the signer stands or walks. For interference sources, we considered two interference sources: people and device. In terms of people interference, data was collected while another person stands in front of Leap Motion with both of her hands appearing in the viewing angle of the Leap Motion sensor. This setup simulates the communication scenario between a deaf person and a normal hearing person. In terms of device interference, data was collected while another person is wearing Leap Motion

standing near the user. This setup simulates the communication scenario where there are more than one person using DeepASL.

6.1.2 *Participants.*

Our study is approved by IRB. Due to the IRB constraint, we could only recruit people with normal hearing ability to participate in the study. We recruited 11 participants and hosted an 3-hour tutorial session to teach them how to perform the target ASL signs using online ASL tutorial videos. The 11 participants (four female) are between 20 to 33 (μ = 24.2) years old, weighted between 49 kg to 86 kg (μ = 74 kg) and are between 155 cm to 185 cm tall (μ = 173 cm).

6.1.3 *Summary of Datasets.*

Table 4 summarizes the amount of data collected in the three datasets. Specifically, for ASL Words Dataset, we collected 56 ASL words with 10 (±3) samples of each word from each of the 11 participants. In total, 3, 068 and 3, 372 samples of one-hand and two-hand ASL words were collected, respectively. For ASL Sentences Dataset, we randomly collected 80 (±3) out of the 100 meaningful sentences from each of the 11 participants. In total, 866 sentences were collected. For In-the-Field Dataset, for each of the six scenarios, we randomly selected 25 out of the 56 ASL words and collected 3 (±1) samples of each word from three out of the 11 participants. To the best of our knowledge, our datasets are the largest and the most comprehensive datasets in the sign language translation literature [10, 14, 26, 32, 43, 47].

Category	ASL Words		ASL Sentences	In-the-Field	Total
Subcategory	One-hand	Two-hand			
Duration (s)	7541.7	8616.3	5094.3	2498.4	23750.7
Frames	821846	949310	507001	259431	2537588
Samples	3068	3372	866	1178	8484

Table 4: Summary of datasets

6.1.4 *Evaluation Metrics and Protocol.*

Evaluation Metrics: We use different metrics to evaluate the translation performance of DeepASL at the word and sentence levels. Specifically, at the word level, we use word translation accuracy, confusion matrix, and Top-K accuracy as evaluation metrics. At the sentence level, we use word error rate (WER) as the evaluation metric, which is defined as the minimum number of word insertions, substitutions, and deletions required to transform the prediction into the ground truth, divided by the number of words in the ground truth. WER is also the standard metric for evaluating sentence-level translation performance of speech recognition systems.

Evaluation Protocol: At both word and sentence levels, we use leave-one-subject-out cross-validation as the protocol to examine the generalization capability of DeepASL across different subjects. In addition, to evaluate the performance of DeepASL in translating unseen sentences, we randomly divide ASL sentences into ten folds, making sure that each fold contains unique sentences to the rest nine folds. We then use ten-fold cross-validation as the protocol.

6.2 Word-Level Translation Performance

Figure 11 shows the average word-level ASL translation accuracy across 11 participants. Overall, DeepASL achieves an average accuracy of 94.5% on translating 56 ASL words across 11 participants. This is a very impressive result considering it is achieved based on leave-one-subject-out cross validation protocol. Further, we observe that the margin between the highest (participant#1, 98.4%) and the lowest (participant#11, 90.6%) accuracies is small. This indicates that our HB-RNN model is capable of capturing the key characteristics of ASL words. Furthermore, the standard deviation of these accuracies is as low as 2.4%, which also demonstrates the generalization capability of our model across different users.

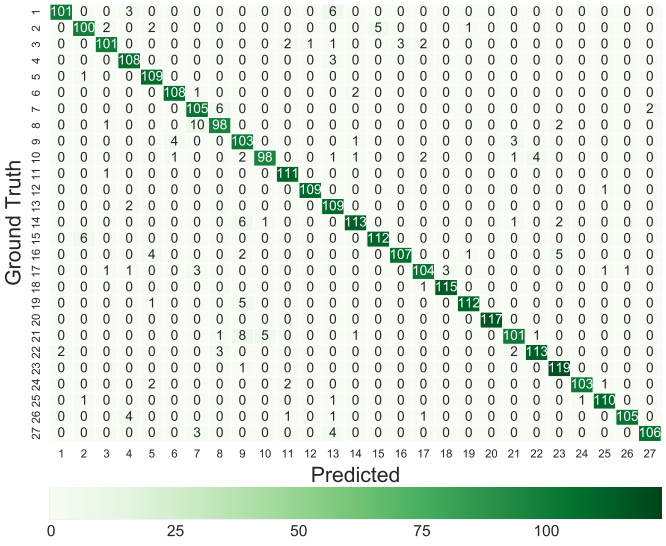

Figure 9: Confusion matrix of 27 one-hand ASL words.

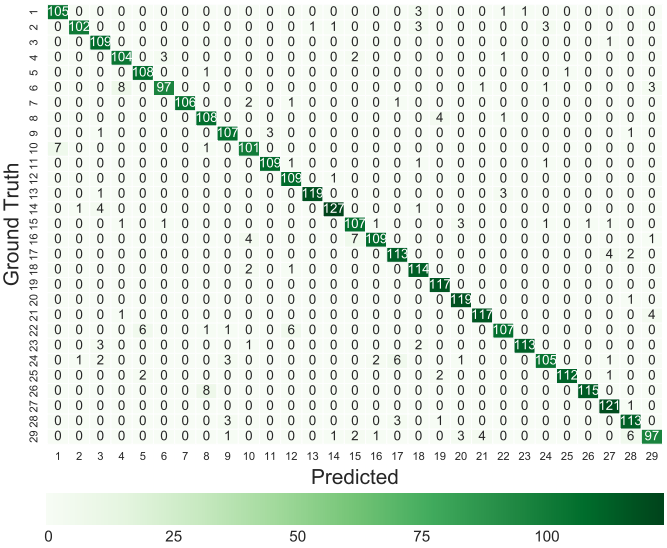

Figure 10: Confusion matrix of 29 two-hand ASL words.

To provide a more detailed view of the result, Figure 9 and 10 show the confusion matrices of translating 27 one-hand ASL words and 29 two-hand ASL words, respectively. As shown in Figure 9, among all the 27 one-hand ASL words, only *please* (word#20) achieves 100% in both precision and recall. This is because *please* has very distinctive hand shape and hand movement characteristics. In contrast, *hot* (word#9) achieves the lowest precision of 81.1% and *bad* (word#21) achieves the lowest recall of 86.3%. Similarly, as shown

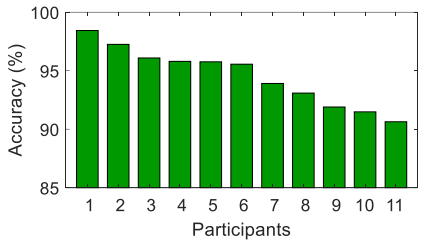

Figure 11: Word-level ASL translation accuracy across 11 participants.

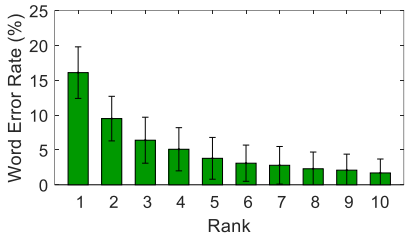

Figure 12: Top-10 WER in translating ASL sentences of unseen participants.

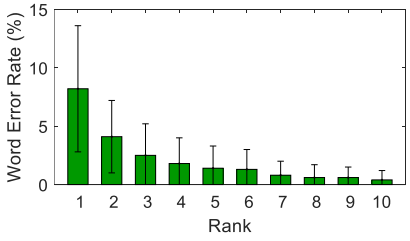

Figure 13: Top-10 WER in translating unseen sentences.

in Figure 10, among all the 29 two-hand ASL words, *big* (word#7) achieves 100% in precision and *bicycle* (word#19) achieves 100% in recall, whereas *with* (word#15) has the lowest precision of 90.7% and *but* (word#29) has the lowest recall of 84.3%.

6.3 The Necessity of Model Components

To validate the design choice of our proposed HB-RNN model, we compare the translation performance between HB-RNN and four comparative models introduced in section 5.2.3. To make a fair comparison, we set the number of hidden units of each model such that the total number of parameters of each model is roughly the same. We use the ASL Words Dataset to evaluate these models. Table 5 lists the evaluation results in terms of average Top-K ($K = 1, 2, 3$) accuracies and standard deviations across 11 participants. As listed, our proposed HB-RNN model outperforms all four comparative models across all three Top-K metrics. Specifically, our HB-RNN model achieves an 5.1%, 5.0%, 3.4% and 0.8% increase in average Top-1 accuracy over the four comparative models, respectively. This result demonstrates the superiority of our HB-RNN model over the four comparative models. It also indicates that the hand shape information, hand movement information, the hierarchical structure, as well as the bidirectional connection capture important and complimentary information about the the ASL signs. By combining these important and complimentary information, the best word-level translation performance is achieved.

Model	Top-1 (%)	Top-2 (%)	Top-3 (%)	Note
HB-RNN-M	89.4 ± 3.1	95.4 ± 1.8	97.2 ± 1.2	No hand shape
HB-RNN-S	89.5 ± 2.4	94.9 ± 1.7	97.0 ± 1.3	No hand movement
SB-RNN	91.1 ± 3.4	96.5 ± 1.7	98.2 ± 1.2	No hierarchical structure
H-RNN	93.7 ± 1.7	97.1 ± 0.9	98.1 ± 0.6	No bidirectional connection
HB-RNN	**94.5 ± 2.4**	**97.8± 1.3**	**98.7 ± 0.9**	**Our model**

Table 5: Comparison of word-level ASL translation performance between HB-RNN and four comparative models.

6.4 Sentence-Level Translation Performance

6.4.1 Performance on Unseen Participants.
We first evaluate the performance of DeepASL on translating ASL sentences using leave-one-subject-out cross-validation protocol. Figure 12 shows the results in Top-10 WER. Specifically, the Top-1 WER is 16.1 ± 3.7%. It indicates that for a 4-word sentence, there is only an average 0.64 words that needs either substitution, deletion or insertion. This is a very promising results considering: 1) there are 16 candidate classes (16 ASL words that construct these sentences) in each frame of the sequence; 2) we do not restrict the length or the word order of the sentence during inference and thus there is an enormous amount of possible label sequences; and 3) no language model is leveraged to help improve the performance.

6.4.2 Performance on Unseen Sentences.
We further conduct an experiment to evaluate the performance of DeepASL on translating unseen sentences (i.e., sentences not included in the training set). The results are illustrated in Figure 13. Specifically, the Top-1 WER is 8.2 ± 5.4%. This indicates that there is only an average 0.33 out of 4 words that needs substitution, deletion and insertion. This is a very promising result considering that the translated sentences are not included in the training set. As a result, it eliminates the burden of collecting all possible ASL sentences.

6.5 Robustness of ASL Translation in the Field

Table 6 lists the word-level ASL translation performance achieved on the In-the-Field Dataset.

Impact of Lighting Conditions: Under poor lighting condition, DeepASL achieves 96.8 ± 3.1% accuracy. It indicates that the poor lighting condition has very limited impact on the performance of DeepASL. Under outdoor sunlight condition, DeepASL achieves 91.8 ± 4.1% accuracy. This result indicates that the significant portion of infrared light in the sunlight also has very limited impact on the performance of DeepASL.

Impact of Body Postures: DeepASL achieves 92.2 ± 3.0% and 94.9 ± 4.5% on walking and standing postures, respectively. The accuracy only drops slightly comparing to previous ASL word recognition result, indicating that DeepASL could also capture information with either standing or sitting body posture. Moreover, this result also demonstrates the advantage of Leap Motion over inertial sensors which are very susceptible to human body motion artifacts.

Impact of Interference Sources: DeepASL achieves 94.7 ± 3.0% and 94.1 ± 1.3% on people in-the-scene interference and multi-device interference, respectively. In the first scenario, the accuracy is comparable to previous ASL word recognition result, meaning that DeepASL is robust to this two interference scenarios. We observe that spaced with social distance, Leap Motion is rarely confounded by the hands of an interferer. This is because the cameras of Leap Motion both have fish-eye angle view, making the far objects too small to be detected. As a matter of fact, effective range of Leap Motion is designed to be no more than approximately 80 cm [5], much less than the social distance. On the other hand, our system

is not affected by multiple Leap Motion present in the ambient environment, indicating that DeepASL is robust when multiple devices are being used at the same time. This is because Leap Motion only uses infrared to illuminate the space where ASL is performed and hence the illumination does not have impact on the infrared images captured by the sensor.

Category	Lighting Condition		Body Posture		Interference Source	
Subcategory	Poor	Bright	Walk	Stand	People	Device
Accuracy (%)	96.8 ± 3.1	91.8 ± 4.3	92.2 ± 3.0	94.9 ± 4.5	94.7 ± 3.4	94.1 ± 1.3

Table 6: In-the-field ASL translation performance.

6.6 System Performance

To examine the system performance, we have implemented DeepASL on three platforms with five computing units: 1) desktop CPU and GPU, 2) mobile CPU and GPU, and 3) tablet CPU. Our goal is to profile the system performance of DeepASL across platforms with different computing powers. Specifically, we use a desktop installed with an Intel i7-4790 CPU and a Nvidia GTX 1080 GPU to simulate a cloud server; we use a mobile development board that contains an ARM Cortex-A57 CPU and a Nvidia Tegra X1 GPU to simulate augmented reality devices with built-in mobile CPU/GPU[1]; and we use Microsoft Surface Pro 4 tablet and run DeepASL on its Intel i5-6300 CPU. The specs of the computing units are listed in Table7.

To provide a comprehensive evaluation, we evaluate the system performance of three models: 1) one-hand ASL word translation model; 2) two-hand ASL word translation model; and 3) ASL sentence translation model. In the following, we report their system performance in terms of runtime performance, runtime memory performance, and energy consumption.

Platform	CPU		RAM	GPU		
	Cores	Speed		Cores	GFLOPS	Speed
Desktop	8	3.6GHz	16GB	2560	8228	1.67GHz
Mobile	4	1.9GHz	4GB	256	512	1GHz
Tablet	2	2.4GHz	4GB	-	-	-

Table 7: The specs of the three hardware platforms.

6.6.1 *Runtime Performance.*

An *inference* contains two parts: data fetching/preprocessing and forward feeding of deep network. Because the time consumed by data fetching/preprocessing is negligible comparing to forward feeding, we report only total *inference* time. We run 200 ASL word/sentence recognition and report the average runtime performance. The results of runtime performance of three models on five computing units of three platforms are shown in Figure 14. To give a straightforward view, we order the time cost in an ascending order. At a high level, *inference* takes much longer when running on the CPUs than on the GPUs across three models. In detail, PC-CPU is 8× to 9× faster than PC-GPU; TX1-CPU is 14× to 28× faster than TX1-GPU. There are two reasons: (1) during inference, only one sample is passing through the network, which substantially limits the component eligible for parallel computation; and (2) our models are built on RNN, meaning that its time-dependency nature intrinsically eliminate the possibility of parallelism. Therefore, we argue that during inference CPUs are better choice than GPUs. As such, DeepASL does not need high-end GPU to do inference. It is also worth pointing out that the runtime of ASL sentence recognition is longer than word recognition. This is because HB-RNN-CTC for sentence recognition has about twice as many parameters as HB-RNN for word recognition. Equally important, we observe that the time cost on all three CPUs are less than 282 ms (ASL sentence recognition on TX1-CPU) which means DeepASL achieves *real-time* recognition performance.

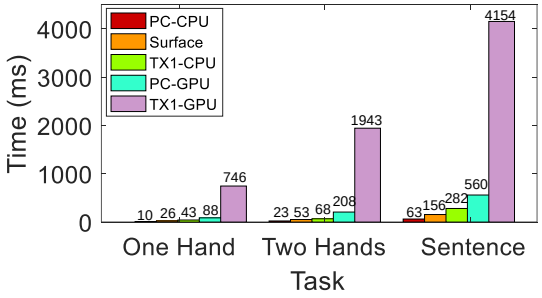

Figure 14: Runtime performance of the three models.

6.6.2 *Runtime Memory Usage.*

Next we report the memory cost on various platforms in CPU and GPU mode. Again, we briefly report the average memory usage on three models. Because memory allocation highly depends on operating system and it is difficult to unravel the memory usage for each part, we only report the total memory usage of each model. For all three platforms, we report physical RAM usage and GPU memory usage. We report these usages because they reflect the memory cost of each model and might indicate the potential improvements. To clearly reflect the influence of three models on CPU, we report the RAM usage that is subtracted by the RAM usage before doing *inference*. The total RAM usage without loading our model is 2891 MB on desktop, 931 MB on TX1 and 1248 MB on Surface. Figure 15 shows the memory cost on five computing units of three platforms. We observe that memory cost of ASL sentence *inference* is larger than two hand ASL word *inference*, which is larger than one hand ASL word. The reason is that in the ASL sentence model, there are more hidden units, thus demanding more allocated memory.

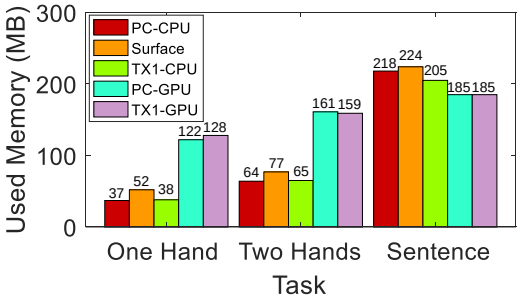

Figure 15: Runtime memory usage of the three models.

[1]Since Mircrosoft Hololens currently does not support hosting USB clients, we could not implement DeepASL in Mircrosoft Hololens to test its system performance.

6.6.3 Energy Consumption.
To evaluate the power consumption of DeepASL, we use PWRcheck DC power analyzer [1] to measure the power consumption of both TX1 and Surface tablet. We run 200 ASL word/sentence recognition and report the average power consumption. Table 8 lists the average power consumption of TX1 and Surface respectively. We report the power consumption of TX1 to simulate augmented reality devices because the TX1 is designed for mobile device real-time artificial intelligence performance evaluation and thus reflects the portion of power consumed by *inference* in augmented reality devices. We observe that for TX1 the power consumption of performing inference on GPU is much larger than CPU. This is because: (1) due to the RNN structure of our model, limited amount of computation can be parallelled, making GPU is less efficient in inference than CPU; and (2) performing inference on GPU also involves processing on CPU (for data loading etc.) and thus costs almost twice as much power as CPU alone.

Platform	Task	Power (W)	Time (ms)	Energy (mJ)
TX1	Idle	3.54	-	-
	One-hand ASL word (CPU)	5.92	42.9	254.0
	Two-hand ASL word (CPU)	6.13	66.1	417.5
	ASL sentence (CPU)	6.02	281.7	1695.8
	One-hand ASL word (GPU)	12.31	746.2	9185.7
	Two-hand ASL word (GPU)	12.16	1943.4	23631.7
	ASL sentence (GPU)	11.75	4153.6	48804.8
Surface	Sleep	1.63	-	-
	Screen-on	8.32	-	-
	ASL Dictionary App-on	15.75	-	-
	One-hand ASL word	23.67	26.1	591.7
	Two-hand ASL word	24.21	52.7	1117.8
	ASL sentence	22.13	156.2	3456.7

Table 8: Energy consumption on TX1 and Surface.

Finally, in Table 9, we report the estimated number of ASL word/sentence recognition that can be completed by TX1 and Surface, using fully-charged battery of Hololens (16.5 Wh) and Surface (38.2 Wh), respectively. For TX1, the number of inferences of CPU is 36×, 57× and 29× larger than those of its GPU for three model respetively. It means that in terms of performing *inference*, CPU is more suitable. Meanwhile, despite the power consumption from other sources, a Hololens/Surface equal volume battery could support enough number of inferences within one day.

Platform	Task	CPU	GPU
TX1	One-hand ASL word	233888	6467
	Two-hand ASL word	142291	2514
	ASL sentence	35028	1217
Surface	One-hand ASL word	232420	-
	Two-hand ASL word	123031	-
	ASL sentence	39784	-

Table 9: Estimated number of inferences on TX1 and Surface with a 16.5 Wh (Hololens) and 38.2 Wh (Surface) battery, respectively.

7 APPLICATIONS

The design of DeepASL enables a wide range of applications. To demonstrate the practical value of DeepASL, we have developed two prototype applications based on DeepASL. In this section, we briefly describe these two prototype applications.

7.1 Application#1: Personal Interpreter

Use Scenario: For the first application, DeepASL is used as a *Personal Interpreter*. With the help of an AR headset, *Personal Interpreter* enables real-time two-way communications between a deaf person and peole who do not understand ASL. Specifically, on one hand, *Personal Interpreter* uses speech recognition technology to translate spoken languages into digital texts, and then projects the digital texts to the AR screen for the deaf person to see; on the other hand, *Personal Interpreter* uses ASL recognition technology to translate ASL performed by the deaf person into spoken languages for peole who do not understand ASL.

Implementation: We implemented *Personal Interpreter* as a Microsoft Hololens application. Since Mircrosoft Hololens currently does not support hosting USB clients, we could not implement DeepASL in Mircrosoft Hololens. Instead, we transmitted the ASL recognition results to Hololens via TCP/IP. Figure 16 illustrates the usage scenario and a screenshot from AR perspective. As shown, the recognized ASL sentence is displayed in the green dialogue box in the Hololens application. The tablet-AR set is burdensome to the deaf people, but we envision that in the future, the AR headset will be miniaturized and hence is much less burdensome for people to wear on a daily basis. Meanwhile, the future AR headset will be able to host a USB device, enabling direct data transmission from Leap Motion to itself.

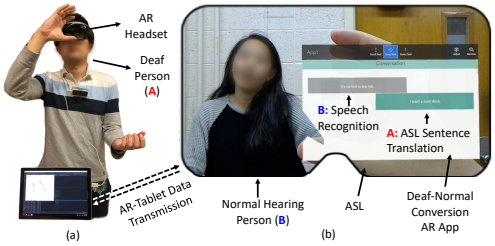

Figure 16: The *Personal Interpreter* application: (a) a deaf person performing ASL while wearing a Microsoft Hololens AR headset; (b) a screenshot from AR perspective.

7.2 Application#2: ASL Dictionary

Use Scenario: For the second application, DeepASL is used as an *ASL Dictionary* to help a deaf person look up unknown ASL words. Spoken languages (e.g., English) allow one to easily look up an unknown word via indexing. Unfortunately, this is not the case for ASL. Imagine a deaf child who wants to look up an ASL word that she remembers how to perform but forgets the meaning of it. Without the help of a person who understands ASL, there is not an easy way for her to look up the ASL word. This is because unlike spoken languages, ASL does not have a natural form to properly index each gesture. *ASL Dictionary* solves this problem by taking the sign of the ASL word as input and displays the meaning of this ASL word in real time.

Implementation: We implemented *ASL Dictionary* as a Microsoft Surface tablet application. Figure 17 illustrates the usage scenario and a screenshot of the tablet application.

Figure 17: The *ASL Dictionary* application: (a) a deaf person looking up an ASL word "help"; (b) a screenshot of the tablet application.

8 DISCUSSION

Impact on ASL Translation Technology: DeepASL represents the first ASL translation technology that enables ubiquitous and non-intrusive ASL translation at both word and sentence levels. It demonstrates the superiority of using infrared light and skeleton joints information over other sensing modalities for capturing key characteristics of ASL signs. It also demonstrates the capability of hierarchical bidirectional deep recurrent neural network for single-sign modeling as well as CTC for translating the whole sentence end-to-end without requiring users to pause between adjacent signs. Given the innovation solution it provides and its promising performance, we believe DeepASL has made a significant contribution to the advancement of ASL translation technology.

Initiative on ASL Sign Data Crowdsourcing: Despite months of efforts spent on data collection, our dataset still covers a small portion of the ASL vocabulary. To make DeepASL being able to translate as many words as in the ASL vocabulary and many more sentences that deaf people use in their daily-life communications, we have taken an initiative on ASL sign data crowdsourcing. We have made our data collection toolkit publicly available. We hope our initiative could serve as a seed to draw attentions from people who share the same vision as we have, and want to turn this vision into reality. We deeply believe that, with the crowdsourced efforts, ASL translation technology can be significantly advanced. With that, we could ultimately turn the vision of tearing down the communication barrier between the deaf people and the hearing majority into reality.

9 RELATED WORK

Our work is related to two research areas: 1) sign language translation; and more broadly 2) mobile sensing systems.

Sign Language Translation Systems: Over the past few decades, a variety of sign language translation systems based on various sensing modalities have been developed. Among them, systems based on wearable sensors have been extensively studied [8, 24–28, 36, 42, 46]. These systems use motion sensors, EMG sensors, bend sensors, or their combinations to capture hand movements, muscle activities, or bending of fingers to infer the performed signs. For example, Wu *et al.* developed a wrist-worn device with onboard motion and EMG sensors which is able to recognize 40 signs [46]. Another widely used sensing modality in sign language translation systems is RGB camera [10, 40, 47]. For example, Starner *et al.* are able to recognize 40 ASL words using Hidden Markov Model with a hat-mounted RGB camera [40]. There are also some efforts on designing sign language translation systems based on Microsoft Kinect [11, 12]. As an example, by capturing the skeleton joints of the user body and limbs using Microsoft Kinect, Chai *et al.* are able to recognize Chinese Sign Language by matching the collected skeleton trajectory with gallery trajectories [12]. Most recently, researchers have started exploring using Leap Motion to build sign language translation systems [13, 30]. However, these systems are very limited in their capabilities in the sense that they can only recognize static ASL signs by capturing hand shape information. In contrast, DeepASL captures both hand shape and movement information so that it is able to recognize dynamic signs that involve movements. Most importantly, compared to all the existing sign language translation systems, DeepASL is the first framework that enables end-to-end ASL sentence translation.

Mobile Sensing Systems: Our work is also broadly related to research in mobile sensing systems. Prior mobile sensing systems have explored a variety of sensing modalities that have enabled a wide range of innovative applications. Among them, accelerometer, microphone and physiological sensors are some of the mostly explored sensing modalities. For example, Mokaya *et al.* developed an accelerometer-based system to sense skeletal muscle vibrations for quantifying skeletal muscle fatigue in an exercise setting [33]. Nirjon *et al.* developed MusicalHeart [35] which integrated a microphone into an earphone to extract heartbeat information from audio signals. Nguyen *et al.* designed an in-ear sensing system in the form of earplugs that is able to capture EEG, EOG, and EMG signals for sleep monitoring [34]. Recently, researchers have started exploring using wireless radio signal as a contactless sensing mechanism. For example, Wang *et al.* developed WiFall [45] that used wireless radio signal to detect accidental falls. Fang *et al.* used radio as a single sensing modality for integrated activities of daily living and vital sign monitoring [17]. In this work, we explore infrared light as a new sensing modality in the context of ASL translation. It complements existing mobile sensing systems by providing a non-intrusive and high-resolution sensing scheme. We regard this work as an excellent example to demonstrate the usefulness of infrared sensing for mobile systems. With the incoming era of virtual/augmented reality, we envision infrared sensing will be integrated into many future mobile systems such as smartphones and smart glasses.

10 CONCLUSION

In this paper, we present the design, implementation and evaluation of DeepASL, a transformative deep learning-based sign language translation technology that enables ubiquitous and non-intrusive ASL translation at both word and sentence levels. At the word level, DeepASL achieves an average 94.5% translation accuracy over 56 commonly used ASL words. At the sentence level, DeepASL achieves an average 8.2% word error rate on translating unseen ASL sentences and an average 16.1% word error rate on translating ASL sentences performed by unseen users over 100 commonly used ASL sentences. Given the innovation solution it provides and its promising performance, we believe DeepASL has made a significant contribution to the advancement of ASL translation technology.

ACKNOWLEDGMENTS

We would like to thank Dr. John Stankovic for being the shepherd of this paper. We are also grateful to the anonymous SenSys reviewers for their valuable reviews and insightful comments. This research was partially funded by NSF awards #1565604 and #1617627.

REFERENCES

[1] 2016. PWRcheck DC power analyzer. http://www.westmountainradio.com/product_info.php?products_id=pwrcheck. (2016).

[2] 2017. American Deaf And Hard of Hearing Statistics. http://www.ncra.org/Government/content.cfm?ItemNumber=9450. (2017).

[3] 2017. American Sign Language | NIDCD. https://www.nidcd.nih.gov/health/american-sign-language. (2017).

[4] 2017. Leap Motion. https://www.leapmotion.com/. (2017).

[5] 2017. Leap Motion API. (2017). https://developer.leapmotion.com/documentation/python/api/Leap_Classes.html.

[6] 2017. WHO | Deafness and hearing loss. http://www.who.int/mediacentre/factsheets/fs300/en/. (2017).

[7] 2017. Wikipedia | List of sign languages. https://en.wikipedia.org/wiki/List_of_sign_languages. (2017).

[8] Kalpattu S Abhishek, Lee Chun Fai Qubeley, and Derek Ho. 2016. Glove-based hand gesture recognition sign language translator using capacitive touch sensor. In *Electron Devices and Solid-State Circuits (EDSSC), 2016 IEEE International Conference on*. IEEE, 334–337.

[9] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. LipNet: Sentence-level Lipreading. *arXiv preprint arXiv:1611.01599* (2016).

[10] Helene Brashear, Thad Starner, Paul Lukowicz, and Holger Junker. 2003. Using Multiple Sensors for Mobile Sign Language Recognition. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC '03)*. IEEE Computer Society, Washington, DC, USA, 45–. http://dl.acm.org/citation.cfm?id=946249.946868

[11] Xiujuan Chai, Guang Li, Xilin Chen, Ming Zhou, Guobin Wu, and Hanjing Li. 2013. Visualcomm: A tool to support communication between deaf and hearing persons with the kinect. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 76.

[12] Xiujuan Chai, Guang Li, Yushun Lin, Zhihao Xu, Yili Tang, Xilin Chen, and Ming Zhou. 2013. Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR*.

[13] Ching-Hua Chuan, Eric Regina, and Caroline Guardino. 2014. American Sign Language recognition using leap motion sensor. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*. IEEE, 541–544.

[14] Fabio Dominio, Mauro Donadeo, and Pietro Zanuttigh. 2014. Combining multiple depth-based descriptors for hand gesture recognition. *Pattern Recognition Letters* 50 (2014), 101–111.

[15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

[16] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1110–1118.

[17] Biyi Fang, Nicholas D. Lane, Mi Zhang, Aidan Boran, and Fahim Kawsar. 2016. BodyScan: Enabling Radio-based Sensing on Wearable Devices for Contactless Activity and Vital Sign Monitoring. In *The 14th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 97–110.

[18] Alex Graves et al. 2012. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer.

[19] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 369–376.

[20] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.

[21] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. (2001).

[22] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[23] Jack Hoza. 2007. *It's not what you sign, it's how you sign it: politeness in American Sign Language*. Gallaudet University Press.

[24] Kehkashan Kanwal, Saad Abdullah, Yusra Binte Ahmed, Yusra Saher, and Ali Raza Jafri. 2014. Assistive Glove for Pakistani Sign Language Translation. In *Multi-Topic Conference (INMIC), 2014 IEEE 17th International*. IEEE, 173–176.

[25] Jonghwa Kim, Johannes Wagner, Matthias Rehm, and Elisabeth André. 2008. Bi-channel sensor fusion for automatic sign language recognition. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 1–6.

[26] Vasiliki E Kosmidou and Leontios J Hadjileontiadis. 2009. Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data. *IEEE transactions on biomedical engineering* 56, 12 (2009), 2879–2890.

[27] Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, and Jihai Yang. 2010. Automatic recognition of sign language subwords based on portable accelerometer and EMG sensors. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. ACM, 17.

[28] Yun Li, Xiang Chen, Xu Zhang, Kongqiao Wang, and Z Jane Wang. 2012. A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data. *IEEE transactions on biomedical engineering* 59, 10 (2012), 2695–2704.

[29] Scott K Liddell. 2003. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press.

[30] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. 2014. Hand gesture recognition with leap motion and kinect devices. In *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 1565–1569.

[31] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Haşim Sak, Alexander Gruenstein, Françoise Beaufays, and Carolina Parada. 2016. Personalized speech recognition on mobile devices. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 5955–5959.

[32] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. 2014. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 541–551.

[33] Frank Mokaya, Roland Lucas, Hae Young Noh, and Pei Zhang. 2016. Burnout: a wearable system for unobtrusive skeletal muscle fatigue estimation. In *Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on*. IEEE, 1–12.

[34] Anh Nguyen, Raghda Alqurashi, Zohreh Raghebi, Farnoush Banaei-kashani, Ann C Halbower, and Tam Vu. 2016. A Lightweight And Inexpensive In-ear Sensing System For Automatic Whole-night Sleep Stage Monitoring. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 230–244.

[35] Shahriar Nirjon, Robert F Dickerson, Qiang Li, Philip Asare, John A Stankovic, Dezhi Hong, Ben Zhang, Xiaofan Jiang, Guobin Shen, and Feng Zhao. 2012. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 43–56.

[36] Nikhita Praveen, Naveen Karanth, and MS Megha. 2014. Sign language interpreter using a smart glove. In *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on*. IEEE, 1–5.

[37] Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

[38] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[39] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 129–136.

[40] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 12 (1998), 1371–1375.

[41] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[42] Noor Tubaiz, Tamer Shanableh, and Khaled Assaleh. 2015. Glove-based continuous Arabic sign language recognition in user-dependent mode. *IEEE Transactions on Human-Machine Systems* 45, 4 (2015), 526–533.

[43] Dominique Uebersax, Juergen Gall, Michael Van den Bergh, and Luc Van Gool. 2011. Real-time sign language letter and word recognition from depth data. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 383–390.

[44] Ulrich Von Agris, Jörg Zieren, Ulrich Canzler, Britta Bauer, and Karl-Friedrich Kraiss. 2008. Recent developments in visual sign language recognition. *Universal Access in the Information Society* 6, 4 (2008), 323–362.

[45] Yuxi Wang, Kaishun Wu, and Lionel M Ni. 2017. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing* 16, 2 (2017), 581–594.

[46] Jian Wu, Zhongjun Tian, Lu Sun, Leonardo Estevez, and Roozbeh Jafari. 2015. Real-time American sign language recognition using wrist-worn motion and surface EMG sensors. In *Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on*. IEEE, 1–6.

[47] Zahoor Zafrulla, Helene Brashear, Harley Hamilton, and Thad Starner. 2010. A novel approach to american sign language (asl) phrase verification using reversed signing. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 48–55.

[48] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. 2016. Co-occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 3697–3703. http://dl.acm.org/citation.cfm?id=3016387.3016423