IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 11, 2020, accepted November 16, 2020, date of publication November 20, 2020,
date of current version December 14, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3039539*

# Multi-Information Spatial–Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation

**QINKUN XIAO, XIN CHANG, XUE ZHANG, AND XING LIU**
Department of Electronics Information and Engineering, Xi'an Technological University, Xi'an 710032, China

Corresponding author: Xing Liu (3194526270@qq.com)

This work was supported in part by the NSFC under Grant 61671362 and Grant 62071366; in part by the Shaanxi Province Natural Science Foundation under Grant 2020JM566; in part by the National Natural Science Foundation of China under Grant 60972095, Grant 61271362, Grant 61671362, and Grant 62071366; and in part by the Natural Science Basic Research Plan of Shaanxi Province in China under Grant 2020JM566.

**ABSTRACT** There are two basic problems in sign language recognition (SLR): (a) isolated word SLR and (b) continuous SLR. Most of the existing continuous SLR methods are extensions of the isolated word SLR methods. These methods use the isolated word SLR results as the basic module and obtain the sentence recognition results through sentence segmentation and word alignment. However, sentence segmentation and word alignment are often not accurate, resulting in a low sentence recognition accuracy. At the same time, continuous SLR usually requires strict sample labels, leading to the difficult task of manual labeling and limited training data availability. To address these challenges, this paper proposes a bidirectional spatial–temporal LSTM fusion attention network (Bi-ST-LSTM-A) for continuous SLR. This approach avoids problems such as sentence segmentation, word alignment, and tedious manual labeling. Our contributions are summarized as follows: (1) we proposed a sign language video feature representation method using a convolutional neural network (CNN) and spatial–temporal LSTM (ST-LSTM) information fusion technology; and (2) we constructed a uniform neural machine translation framework that can be used for complex continuous SLR and gesture recognition of nonspecific people in nonspecific environments. Experiments were carried out on some large continuous sign language datasets. The sign language recognition accuracy reached 81.22% on the 500 CSL dataset, 76.12% on the RWTH-PHOENIX-Weather dataset and 75.32% on the RWTH-PHOENIX-Weather-2014T dataset, thereby illustrating the effectiveness of the proposed framework.

**INDEX TERMS** Continuous SLR, attention, ST-LSTM, neural machine translation, CNN.

## I. INTRODUCTION

The goal of video-based SLR is to convert a video sequence into a sign language text representation [1]–[4]. SLR, and particularly continuous SLR [1], [4], is a relatively new field of human–computer interaction (HCI). Although many researchers have explored this area [8], [9], there are still many challenges and problems.

A key challenge of SLR is the design of visual descriptors to capture SL semantics, such as facial expressions and the shape, direction, and position of hands [1], [3]. As a result, previous studies relied heavily on RGB-D data in the posture/gesture models [6], [7]. Moreover, most of the existing sign language video sequences are recorded using normal cameras that lack depth sensors, thus limiting the practical application of the existing SLR method.

Another challenge of continuous SLR involves time segmentation [8], [9] because sign language actions are diverse and difficult to detect. Accurate sign language video segmentation is a difficult task, and inaccurate segmentation during preprocessing can lead to errors in the subsequent steps. In addition, labeling each isolated sign language word in video is time-consuming, and extending isolated word SLR to continuous SLR is difficult and involves many technical challenges. Most existing SLR studies [6]–[8] are isolated SLR methods; that is, they focus on the identification of a word or phrase. Some methods [8], [9] have explored the extension of the isolated word SLR methods to continuous SLR, which involves the reconstruction of sentence structures.

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

Most existing methods of continuous SLR divide the problem of sentence recognition into three stages: video segmentation, isolated word/phrase recognition, and sentence synthesis. For example, DTW-HMM [8] proposed a coarse segmentation step based on a threshold matrix, followed by a dynamic time wrapping (DTW) algorithm and a bigram model. Dan *et al.* [9] integrated a new HMM-based language model. Recently, transition action modeling [10] has attracted much attention because this approach can be used for video segmentation. However, despite the popularity of this approach, the sign language video segmentation task is still challenging, and transition actions can be subtle and vague. Ultimately, inaccurate segmentation can lead to significant performance loss in the subsequent steps [11].

Video description generation [12]–[15] involves the generation of a short sentence that describes the scene/object/motion of a given video sequence. A popular approach is based on the video-to-text method [13], which connects two layers of LSTM over a CNN. Hierarchical attention networks can be incorporated into LSTM [14], which is characterized by the automatic selection of the most relevant video frames for the task. The LSTM algorithm also has some extensions, such as bidirectional LSTM (Bi-LSTM), layered LSTM [15], and layered attention GRU [16].

In view of some problems in continuous SLR, and inspired by sequence-to-sequence methods [12]–[15], we propose an RGB video continuous SLR method that identifies sign language through data fusion and attention-based machine translation. Our contributions in this paper are two-fold. (1) To achieve the SLR task based on the RGB video data, and inspired by the significant results achieved by deep learning technology in object detection, we proposed a sign language detection and representation framework in RGB video. The method includes a faster Region-CNN (R-CNN) module, a compression tracking module, and a newly proposed CNN-LSTM data fusion model. In previous sign language video feature fusion methods [15], the spatial–temporal information was not fully considered. To fuse the spatial–temporal information of sign language in RGB video sequences, we use spatial–temporal LSTM (ST-LSTM) to fuse hand information and body information in the LSTM units. (2) Inspired by the LSTM video description methods [12], [14] and video-to-text approaches [13], we combine structural information and attention mechanisms to propose the use of recurrent attention networks to bypass time segmentation, which is an extension of LSTM. The scheme involves encoding the entire video and then outputting the complete sentence verbatim. However, the attention network only partially optimizes the probability of generating the next word in the case of a given input video and the previous word, ignoring the relationship between the video and the sentence. Therefore, it may experience robustness issues. To compensate for this consideration, we introduce an attention-based bidirectional LSTM encode–decode model to clearly establish the correlation between the video frame sequences and the text sentences.

The remainder of the paper is organized as follows. Section 2 presents a description of the recognition method, Section 3 describes our experiments, and Section 4 draws some conclusions.

## II. RELATED WORK

Video sign language recognition systems are often composed of a feature extraction module and a sequence signal model. The feature extraction module is used to represent gesture sequences. Moreover, the sequence signal model maps sequence representations to labels. For better gesture recognition, researchers have designed a variety of handmade features, which generally use image gradients [11], [14], [28] and motion trajectories [11], [18], [25] to represent hand shape and skeleton structure.

In recent years, application of the deep neural network to automatic learning feature representation has become widespread. Wu and Shao [29] used a deep belief network to extract high-level skeletal joint features for gesture recognition. Some researchers used the convolutional neural network [30], [31] and three-dimensional convolutional neural network [19], [20] to collect hand visual cues. For example, Molchanov *et al.* [17] applied a 3D CNN to extract the spatiotemporal features of the color, depth, and optical flow data from a video stream. Meanwhile, Neverova *et al.* [19] used color, depth data, and custom-made gesture descriptors to represent sign language features and then established a multiscale deep structure for sign language recognition.

The time series model is a powerful tool for learning the corresponding relationship between a sequence representation and tags. HMM is the most widely used time series model in SL recognition [11], [20]. Dynamic time warping (DTW) [24] and support vector machines [32] are also used to measure the similarity between gestures. In recent years, RNNs have been successfully applied to sequence problems such as speech recognition [33] and machine translation [34], [35]. Pigou *et al.* [26] proposed an end-to-end neural model based on time convolution and bidirectional recursion for sign language recognition. However, due to the weak supervision ability of the recurrent neural network at the sentence level, it is difficult to match the extra-long input sequence with the ordered label frame by frame. Unlike the aforementioned models, we use attention mechanisms to integrate time dynamics before implementing bidirectional recursion.

Compared with existing models [21], [22], [27], [30], the recurrent neural network sequence learning model with end-to-end training presented in this paper has better learning ability and performance with regard to dynamic dependence. First, we do not use noisy frame markers as the training target of the neural network; rather, we use the symbol graphics method of human detection results to train our feature extraction module, which considers more local time dynamics, such as the location of the human face and hands, and other key information. Furthermore, by introducing the spatiotemporal sequence signal model based on the attention mechanism, namely, ST-A-LSTM, we not only integrate hand information

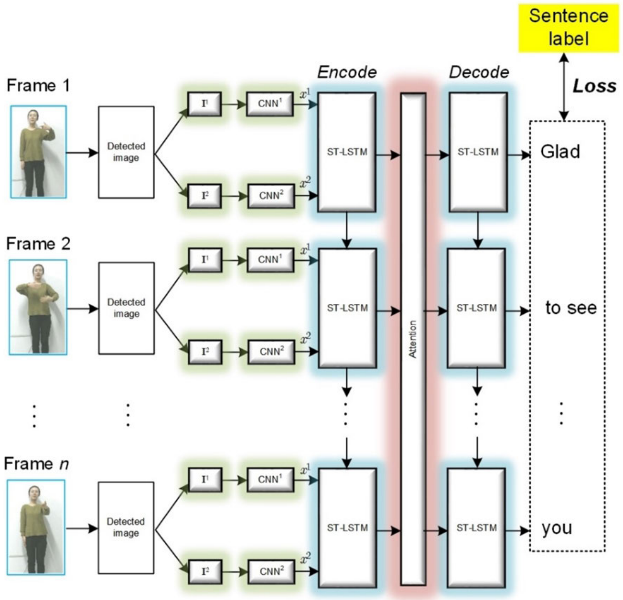

**FIGURE 1. Overview of the proposed continuous SLR method using a Bi-ST-LSTM-A machine translation framework.**

and human body structure information organically through data fusion but also effectively solve the problem that the samples and tags do not show one-to-one correspondence in real time. Moreover, in our method, we do not introduce additional monitoring information, such as hand annotation, which requires expert knowledge and tedious annotation.

We can summarize our contributions in this paper as follows: (1) a sign language video feature representation method is proposed using a CNN and ST-LSTM information fusion technology; and (2) a uniform neural machine translation framework is constructed that can be used for complex continuous SLR and gesture recognition of nonspecific people in nonspecific environments.

## III. RECOGNITION METHODOLOGY

Our method is described in Fig. 1. The method can be broken down into four steps. (1) The RGB video is inputted. For each frame, we detect face and hand patches in the frame image using a faster R-CNN [13]. The frame information is then divided into two parts: local hand information and spatial–temporal information of sign language. (2) These two types of information are then fed into the basic unit of the ST-LSTM and are fused using the proposed fusion method. (3) To combine the attention mechanism with the bidirectional ST-LSTM encode–decode framework, the fused feature sequence is then translated into text sentences. (4) The loss function is defined using the differences between the text sentence labels and the outputted text sentence. The details are discussed below.

### A. FEATURE EXTRACTION USING A CNN

In our research, we used only the RGB image to calculate the sign language feature. In many existing SLR methods [36], [37], both the depth image and RGB image are used. In fact, there are many sign language videos that are only recorded by RGB cameras, and therefore, research on RGB-based SLR is more important.

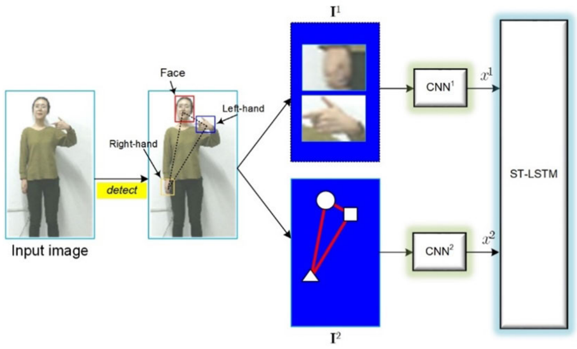

**FIGURE 2. Illustration of feature fusion at the LSTM unit level. The above unit shows calculation for $x^1$, and the bottom unit shows calculation for $x^2$.**

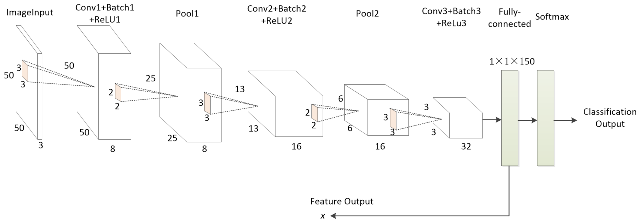

**FIGURE 3. Illustration of image features extraction using a simple CNN.**

Generally, normal-size images are used in SLR to reduce the computational cost of training. Since the hand patch is often small, it is difficult to obtain complete hand information. In contrast to previous research, we used a high-resolution image as the inputted image. Based on this high-resolution image (250 × 250), we used a faster R-CNN [13] and compressive tracking [14] to detect the face and hands, as shown in Fig. 2. The hand patches (reshape size to 50 × 50) are sufficiently large to show subtle details. We use $\mathbf{I}^1$ to denote the hand patch image, and this image is then fed into the CNN to obtain feature $x^1$:

$$x^1 = f_{CNN^1}(\mathbf{I}^1). \tag{1}$$

In this paper, we build a CNN model to extract features of RGB images, as shown in Fig. 3. There are 15 layers in the CNN, namely, an input layer, 3 convolution layers, 3 batch normalization layers, 3 ReLU layers, 2 max pooling layers, 1 fully connected layer, 1 softmax layer and 1 classification output layer. The first convolution layer is computed using 8 different 3 × 3 kernels, and 8 feature maps are obtained; the second convolutional layer is computed with 16 different 3 × 3 kernels, and 16 feature maps are obtained; and the 3rd convolutional layer is computed with 32 different 3 × 3 kernels, and 32 feature maps are obtained. The convolutional layers are used to find the local relationship of the input layer. The corresponding pooling layer is the largest pooling layer in the 2 × 2 neighborhood domain, and the original size of the feature map is obtained. Similarly, the 3rd convolution layer has 32 different 3 × 3 kernels. The ReLU layer is connected

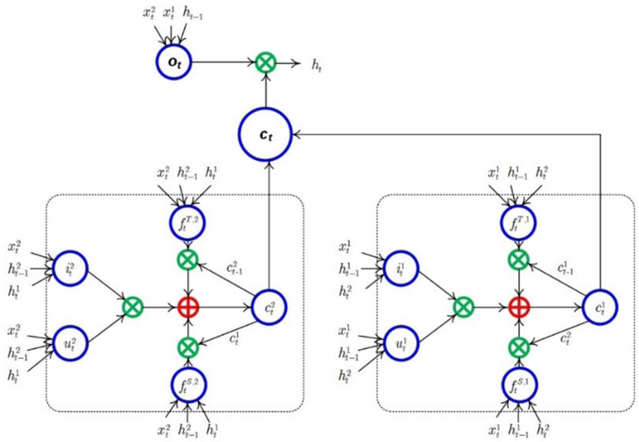
$x_t^2$ $x_t^1$ $h_{t-1}$
$o_t$
$h_t$
$c_t$
$x_t^2$ $h_{t-1}^2$ $h_t^1$
$f_t^{T,2}$
$x_t^2$ $h_{t-1}^2$ $h_t^1$
$i_t^2$
$c_{t-1}^2$
$c_t^2$
$x_t^2$ $h_{t-1}^2$ $h_t^1$
$u_t^2$
$c_t^1$
$f_t^{S,2}$
$x_t^2$ $h_{t-1}^2$ $h_t^1$
$x_t^1$ $h_{t-1}^1$ $h_t^2$
$f_t^{T,1}$
$x_t^1$ $h_{t-1}^1$ $h_t^2$
$i_t^1$
$c_{t-1}^1$
$c_t^1$
$x_t^1$ $h_{t-1}^1$ $h_t^2$
$u_t^1$
$c_t^2$
$f_t^{S,1}$
$x_t^1$ $h_{t-1}^1$ $h_t^2$

**FIGURE 4.** **Illustration of feature fusion in the LSTM unit level. The above unit shows the calculation for $x^1$, and the bottom unit shows the calculation for $x^2$.**

to the fully connected layer. From the fully connected layer of the CNN, the image features $x$ is outputted, which is a 288-dimensional vector, as shown in Fig. 3.

In addition, we used cartoon pictures to simplify the face and hand spatial–temporal information, as shown in Fig. 2. We used three symbols and a thick, solid line with uniform size to draw the cartoon picture. We used "○", "□", and "△" to denote the face location, the left hand's location, and the right hand's location, respectively. The cartoons remove complex environmental factors that can be used to identify tasks for nonspecific people and nonspecific environments. In cartoon pictures, the head and left and right hands are represented by different symbols. The different positions of the three important components and their logical connections determine the spatial configuration of different gestures. Therefore, cartoon pictures can directly reflect the complete spatial information of gestures. We used $\mathbf{I}^2$ to denote the cartoon picture and fed it into the CNN to get feature $x^2$:

$$x^2 = f_{CNN^2}(\mathbf{I}^2). \tag{2}$$

### B. MULTI-INFORMATION FUSION

To effectively combine hand information with spatial–temporal information, and inspired by Huang *et al.* [38], we fused two channels of information together for better SLR performance. However, Jie *et al.* [38] only considered temporal factors of sign language and fused two data streams in a fully connected layer of the CNN. The sign language representation should actually consider spatial factors and temporal factors together.

Inspired by Jun *et al.* [39], the ST-LSTM model was used to represent the spatial–temporal dependencies and relationships among different frames and different body joints. We fused features $x^1$ and $x^2$ in the ST-LSTM [38] unit, as shown in Fig. 4. An ST-LSTM unit at time $t$ is shown in this figure. The unit contains 2 LSTM units: the first is an $x^1$-related LSTM unit (left box), and the second is an $x^2$-related LSTM unit (right box). To describe sign language in a video frame, we use $j$ and $t$ to respectively denote the indices of joints and frames, where $j \in [1, J]$ and $t \in [1, T]$. The input of the ST-LSTM unit is $x_t^j$, and $h_t^{j-1}$ represents the hidden state of the previous joint at time $t$, while $h_t^{j-1}$ is the hidden state of the joint at time $t-1$.

In this paper, we use $E_M$, $\epsilon$ and $\zeta$ to denote the maximum epochs, gradient threshold, and initial learn rate, respectively; the learn rate schedule is 'piecewise', and we use $\eta$, $\xi$, $\delta$, $n_h$ and $b_m$ to denote the learn rate drop period, learn rate drop factor, embedding dimension, number of hidden units, and minibatch size, respectively. To obtain better fusion results, according to the actual samples in the sign language dataset, we initialize the set LSTM parameters to select the adaptive moment estimation (ADAM) method to train the data, and we set $E_M = 250$, $\epsilon = 3$, and $\zeta = 0.001$, $\eta = 125$, $\xi = 0.1$, $\delta = 256$, $n_h = 200$, $b_m = 128$.

The $x^2$-related LSTM unit has two forget gates, $f_t^{T,2}$ and $f_t^{S,2}$, to process the time-related information and spatial-related information, respectively. According to LSTM-based time series processing theory [39], the $x^2$-related LSTM unit can be calculated as follows:

$$\begin{pmatrix} i_t^2 \\ f_t^{s,2} \\ f_t^{T,2} \\ u_t^2 \\ o_t^2 \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \\ \sigma \end{pmatrix} \times \Phi \begin{pmatrix} x_t^2 \\ h_{t-1}^2 \\ h_t^2 \end{pmatrix}, \tag{3}$$

$$c_t^2 = i_t^2 \odot u_t^2 + f_t^{T,2} \odot c_{t-1}^2 + f_t^{S,2} \odot c_t^1, \tag{4}$$

where $x_t \in R^D$ is the input signal of joint $j$ at time $t$, and $i, f, o$ are the input gate, forget gate, and output gate, respectively. $c$ is the $d$-dimensional cell state, $u$ is the modulated input, and $\odot$ denotes the element-wise product $\Phi : R^{D+d} \rightarrow R^{4d}$, which is an affine transformation.

Similar to $x^2$, $x^1$ can be calculated according to the following:

$$\begin{pmatrix} i_t^1 \\ f_t^{S,1} \\ f_t^{T,1} \\ u_t^1 \\ o_t^1 \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \\ \sigma \end{pmatrix} \times \Phi \begin{pmatrix} x_t^1 \\ h_{t-1}^1 \\ h_t^2 \end{pmatrix}, \tag{5}$$

$$c_t^1 = i_t^1 \odot u_t^2 + f_t^{T,1} \odot c_{t-1}^1 + f_t^{S,1} \odot c_t^2. \tag{6}$$

Next, let

$$\mathbf{c}_t = \begin{pmatrix} c_t^1 \\ c_t^2 \end{pmatrix}, \quad o_t = \sigma \left( \Phi \begin{pmatrix} x_t^1 \\ x_t^2 \\ h_{t-1} \end{pmatrix} \right), \tag{7}$$

where $h$ is the fusion hidden state. Finally, the fusion output in ST-LSTM is

$$h_t = o_t \odot \tanh(\mathbf{c}_t). \tag{8}$$

### C. ATTENTION-BASED BIDIRECTIONAL LSTM TRANSLATION SYSTEM

According to the above, we input 2-feature sequence: $x^1 = (x_1^1, \cdots, x_T^1)$ and $x^2 = (x_1^2, \cdots, x_T^2)$, and through ST-LSTM feature fusion, we obtain the hidden state sequence $\mathbf{h} = (h_1, \ldots, h_T)$, which can be considered an encode sequence. In this paper, since we use bidirectional ST-LSTM as the encoder, we have

$$V = q(\mathbf{h}), \quad h_t = [h_t^{\leftarrow}, h_t^{\rightarrow}], \tag{9}$$

where $q(.)$ is an encode function and $V$ is an encoded vector that includes all information of $\mathbf{h}$. $h_t^{\rightarrow}$ is the hidden state computed by the sign language sequence from beginning to end at time $t$, and $h_t^{\leftarrow}$ is hidden state computed by the sign language sequence from end to beginning at time $t$.

In the decode stage, let $s$ be the hidden state vector in the decode function and let $y$ be the decode output vector. Based on softmax function calculation, we obtain the optimal decode output:

$$p(y_t|y_{t-1}, \cdots, y_1, V) = \max \frac{\exp(W_y s_t + b_y)}{\sum_{y \in D} \exp(W_y s_t + b_y)}, \tag{10}$$

where $W$ is the weight matrix, $b$ is the bias, and $D$ is the dictionary of sign language words. Generally, the LSTM decode calculation using the attention mechanism can be expressed as:

$$p(y_t|y_{t-1}, \cdots, y_1, V_t) = g(y_t|y_{t-1}, s_t, V_t). \tag{11}$$

We call $V_t$ the context vector, and $V_t$ can be computed by $V_t = \sum_{j=1}^{T} \alpha_{tj} h_j$, where $h_j$ is the encode vector and $\alpha_{tj}$ is the weight coefficient whose value corresponds to the decoding output. A higher value of $\alpha_{tj}$ corresponds to higher correlation with the current decoding output. To obtain the weight of each encoding vector at time $t$, we designed an alignment model: $f(s_{t-1}, h_j)$. Since we aim to calculate the weight distribution of each of the frame encoding features, we used hidden vector $s_{t-1}$ in the decoding function to compare with the encoding vector $h_j$. We used the model $f(s_{t-1}, h_j)$ to align the decode output with the encode input and then normalized the weight of each encoding vector by the softmax function. We used a simple perceptual machine layer to finish the alignment calculations:

$$f(s_{t-1}, h_j) = W_{att}^T \tanh(W_s s_{t-1} + U_{att} h_j + b_{att}), \tag{12}$$

where $W_{att}^T$, $W_s$, $U_{att}$ and $b_{att}$ are parameters related to the attention model. Letting $e_{tj} = f(s_{t-1}, h_j)$, we have $\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} e_{tk}}$. According to the LSTM calculation, we obtain the decode output:

$$\begin{pmatrix} f_t^d \\ i_t^d \\ o_t^d \\ \tilde{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{pmatrix} \left( \Phi_g \begin{pmatrix} y_{t-1} \\ s_{t-1} \\ V_t \end{pmatrix} \right), \tag{13}$$

$$g_t = f_t^d \odot g_{t-1} + i_t^d \odot \tilde{g}_t, \tag{14}$$

where $f_t^d$, $i_t^d$, and $o_t^d$ are the forget gate, input gate, and output gate of the decoding calculation, respectively. $\phi_g$ is an affine transformation.

Finally, we obtain the sign language sentence translation results:

$$s_t = o_t^d \tanh(g_t), \tag{15}$$

$$y_t = soft\max(W_s s_t + b_s). \tag{16}$$

In summary, the input of the Bi-ST-LSTM-A system is the image feature sequence $\mathbf{X}_1, \cdots, \mathbf{X}_T$, where $T$ is the length of the sign language image sequence and $\mathbf{X} = (x^1, x^2)$, $x^1$ and $x^2$ are image features corresponding to images $\mathbf{I}^1$ $\mathbf{I}^2$, respectively, and the system output is the recognition results of a continuous sign language sentence: $(y_1, \cdots, y_n)$, where $n$ is the length of the sentence. In training parts, the loss function can be defined as:

$$Loss = 1 - p(y_1, \cdots, y_n|\mathbf{X}_1, \cdots, \mathbf{X}_T; \theta). \tag{17}$$

We also use the gradient descent method to train the parameters:

$$\theta \leftarrow \theta + \eta \frac{\partial Loss}{\partial \theta}. \tag{18}$$

Finally, the optimization parameters are obtained: $\theta^* = \arg\max_\theta p(y_1, \cdots, y_n|\mathbf{X}_1, \cdots, \mathbf{X}_T; \theta)$.

## IV. RESULTS AND DISCUSSION

### A. DATASET

We tested the performance of our method using 4 sign language datasets: one is a continuous dataset collected by us, and the other 3 datasets are open-source continuous sign language datasets, namely, a Chinese sign language (CSL) dataset [38] and a German sign language dataset, which are both RWTH-PHOENIX-Weather [38] SL datasets, and the RWTH-PHOENIX-Weather-2014T dataset.

In our experiments, only RGB video streams were used. Our CSL dataset was collected in-house and can be used for daily communication. This dataset consisted of 50 continuous common CSL sentences, such as "What's your name?", "Hello, everyone," and so on. Each sentence consisted of 3–5 signs, with a total of 150 different isolated signs in the dataset, including "you," "ID card," and "home". There were 500 instances of each isolated sign, for a total of $150 = 75,000$ instances. We used 70% of instances for training, 15% of instances for validation, and the remaining 15% for testing. Some CSL examples in our dataset are given in Fig. 6.

The 500 CSL dataset [38] contains 25,000 video instances. The total video clip is 100+ hours long and was recorded by 50 actors. Each video instance was labeled semantically by a professional sign language teacher. We used 70% of the instances for training, 15% for validation, and the remaining 15% for testing.

The RWTH-PHOENIX-Weather dataset contains 7,000 weather forecast sentences from nine sign language speakers. All of the videos were 25 frames per second (FPS), with a resolution of 210 × 260. Overall, 80% of the instances were used for training, 10% for validation, and 10% for testing. The RWTH-PHOENIX-Weather-2014T dataset is an expansion of RWTH-PHOENIX-Weather and contains a total of 8257 videos.

The evolution of sentence recognition is different from isolated words recognition. In sentence recognition, the length of the output sentence may not be consistent with the length of

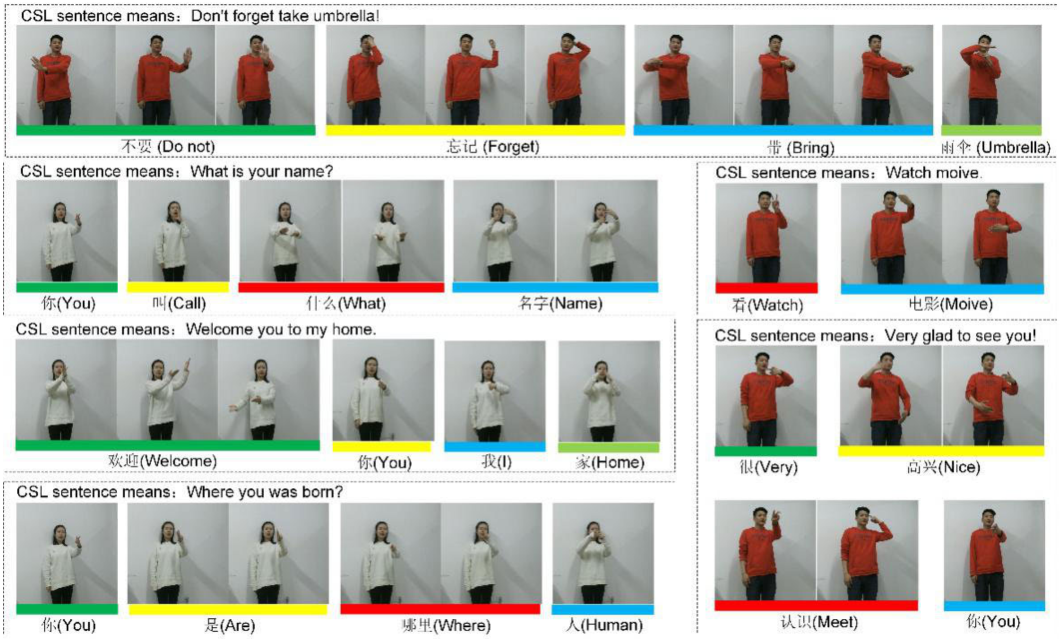
CSL sentence means：Don't forget take umbrella!
不要 (Do not)
忘记 (Forget)
带 (Bring)
雨伞 (Umbrella)
CSL sentence means：What is your name?
你(You)
叫(Call)
什么(What)
名字(Name)
CSL sentence means：Watch moive.
看(Watch)
电影(Moive)
CSL sentence means：Welcome you to my home.
欢迎(Welcome)
你(You)
我(I)
家(Home)
CSL sentence means：Very glad to see you!
很(Very)
高兴(Nice)
CSL sentence means：Where you was born?
你(You)
是(Are)
哪里(Where)
人(Human)
认识(Meet)
你(You)

**FIGURE 5.** Illustration of CSL words in our CSL dataset.

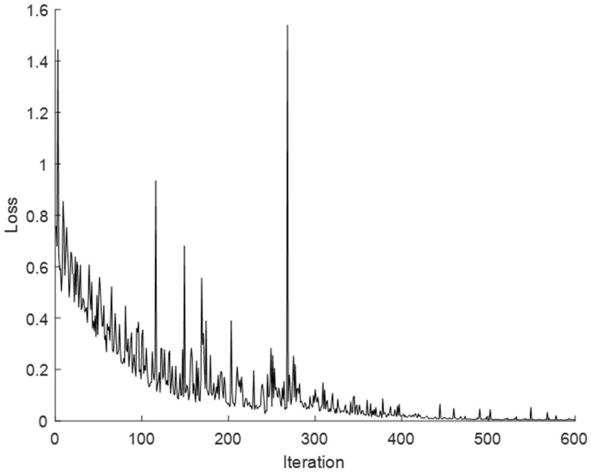

**FIGURE 6.** Training process of the proposed Bi-ST-LSTM-A sign language recognition model.

the annotation sentence, which means that there may be an increase in output, deletion, and replacement errors. To consider various errors and describe the accuracy of sentence recognition, we use the following metrics [43]:

$$\text{Accuracy} = 1 - \frac{S+I+D}{N} \times 100\%, \tag{19}$$

where $S$, $I$, and $D$ represent the minimum number of replace, insert, and delete operations, respectively, required to convert the hypothetical sentences to true annotations.

## B. PERFORMANCE EVALUATION ON OUR DATASET

### 1) ACCURACY

In our dataset, 50 sentences were established to test the continuous CSL recognition framework proposed in this paper. These sentences were composed of 150 isolated sign words and included common phrases such as "Do not forget to bring an umbrella," "This is my business card," and "Is there a room?"

**TABLE 1.** Results of continuous CSL recognition for various framework settings. (In the experiment, the adaptive moment estimation (ADAM) method is selected for data training with $E_M = 250$, $\epsilon = 3$, $\zeta = 0.001$, $\eta = 125$, $\xi = 0.1$, $\delta = 256$, $n_h = 200$, and $b_m = 128$).

| Method (Number of sentences) | Accuracy |
|---|---|
| LSTM (10) (Liu *et al.* [7]) | $91.13 \pm 1.63\%$ |
| LSTM (30) (Liu *et al.* [7]) | $80.15 \pm 2.81\%$ |
| LSTM (50) (Liu *et al.* [7]) | $72.22 \pm 1.41\%$ |
| LSTM + HMM (10) (Guo *et al.* [56]) | $93.23 \pm 2.52\%$ |
| LSTM + HMM (30) (Guo *et al.* [56]) | $82.32 \pm 2.61\%$ |
| LSTM + HMM (50) (Guo *et al.* [56]) | $77.23 \pm 3.32\%$ |
| LSTM + Attention (10) (Ma *et al.* [57]) | $92.35 \pm 2.91\%$ |
| LSTM + Attention (30) (Ma *et al.* [57]) | $85.98 \pm 1.75\%$ |
| LSTM + Attention (50) (Ma *et al.* [57]) | $75.78 \pm 1.78\%$ |
| Bi-ST-LSTM-A (10) | $\mathbf{98.56 \pm 0.62\%}$ |
| Bi-ST-LSTM-A (30) | $86.84 \pm 1.33\%$ |
| Bi-ST-LSTM-A (50) | $78.87 \pm 2.42\%$ |

The visual receptive field and time window length are highly important for spatial feature fusion with a large time span. The fusion effects of the proposed method under different combinations are compared on our database. As shown in Tab. 1 and Tab. 2, we obtain some recognition results based on different parameter settings, and from the results, we observe that our proposed method obtains the best performance regardless of the parameter settings.

We also find that the combination of the time series model and sequential neural network (LSTM + HMM model) can effectively improve the accuracy of sign language recognition. The combination of the attention model and sequential neural network (LSTM + attention) can obviously improve the recognition accuracy of continuous sign language. Hence, it is best to consider the temporal and spatial model sequence neural network. One possible explanation for these findings

**TABLE 2.** **Results of continuous CSL recognition for various framework settings. (In the experiment, the adaptive moment estimation (ADAM) method is selected for data training with $E_M = 250$, $\epsilon = 5$, $\zeta = 0.005$, $\eta = 125$, $\xi = 0.05$, $\delta = 256$, $n_h = 200$, and $b_m = 128$).**

| Method (Number of sentences) | Accuracy |
|---|---|
| LSTM (10) (Liu *et al.* [7]) | 93.13 ± 1.53% |
| LSTM (30) (Liu *et al.* [7]) | 85.15 ± 1.31% |
| LSTM (50) (Liu *et al.* [7]) | 74.21 ± 2.41% |
| LSTM + HMM (10) (Guo *et al.* [56]) | 93.23 ± 2.12% |
| LSTM + HMM (30) (Guo *et al.* [56]) | 87.52 ± 3.21% |
| LSTM + HMM (50) (Guo *et al.* [56]) | 75.53 ± 2.12% |
| LSTM + Attention (10) (Ma *et al.* [57]) | 93.35 ± 1.21% |
| LSTM + Attention (30) (Ma *et al.* [57]) | 85.98 ± 2.75% |
| LSTM + Attention (50) (Ma *et al.* [57]) | 74.78 ± 2.18% |
| Bi-ST-LSTM-A (10) | **97.56 ± 1.21%** |
| Bi-ST-LSTM-A (30) | 89.84 ± 2.33% |
| Bi-ST-LSTM-A (50) | 79.87 ± 1.42% |

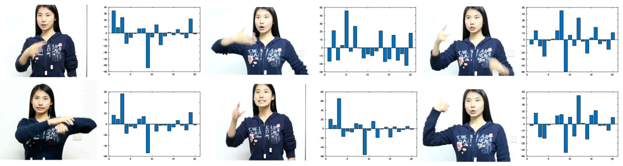

**FIGURE 7.** **Some examples of sign language feature representation in the video frame image. Different gestures (left) correspond to different features (right).**

is that the time fusion layer usually focuses on capturing information integration at different times, while visual input involves more spatial attributes of a gesture but is closely related to the dynamic correlation of hand shape. Therefore, a suitable spatiotemporal fusion structure can be applied to the visual input with a close temporal structure.

The proposed technique was tested on a PC with an Intel Core *i*5 CPU with 8 GB of RAM and the Microsoft Windows 10 operating system. We used two Titan graphics cards for training acceleration. The training process with our dataset is shown in Fig. 6 for 75,000 video samples: 70% of the samples were selected as training samples, there was a total of 600 epochs, and the training time was 13 hours 54 min. The value of the loss function decreased over time.

We compared continuous CSL recognition performance for different data sizes and various frameworks. As shown in Tab. 1 and Tab. 2, the recognition accuracy gradually decreased with increasing quantity of data, making large-scale sign language recognition difficult. For example, as shown in Tab. 1, the LSTM recognition accuracy for 10 samples was 91.13%, but it decreased to 72.22% with 50 samples; this result may have been caused by a decline in the isolated word recognition or increased sentence complexity. Regardless, the LSTM + attention and LSTM + HMM approach outperformed the LSTM method, with Bi-ST-LSTM-A producing the highest recognition accuracy (78.87%) across all 50 sentences. These results indicate that multi-information fusion is beneficial for automated segmentation of sign language sentences, independent of the data sampling environment.

Additional details are shown in Figs. 7 and 8. In Fig. 7, some examples for sign language feature presentation are shown. We can see that our proposed sign language feature representation is effective and that different gestures correspond to different feature representations.

An attention matrix is also given in Fig. 8. We can see that the sign language sentence can be translated well by the attention model. Four attention score matrixes are presented in Fig. 8, and the attention scores highlight the relationship between the source and translated sequences. Each attention score can be calculated by:

$$\mathbf{score}_t^{att} = softmax(\sum_{j=1}^{T} \alpha_{tj} h_j Z_j^{enc}), \qquad (20)$$

where the $softmax(.)$ is an activation function, and the attention score at each time step is the dot product of the hidden state $h_j$ and the learnable attention weights $\alpha_{tj}$ multiplied by the encoder output $Z_j^{enc}$. The encoder output is obtained by $Z = \text{embedding}(\mathbf{X}, W_{input})$, where the embedding function maps numeric indices to the corresponding vector given by the input weights $W_{input}$.

### 2) TIME COST

We also test the time cost of proposed method, with the time cost comparison results given in Tab. 3. From the results, we can see that the LSTM method uses the shortest time for training, and the testing time is 0.22 s, while at the same time, our method uses the longest time for training. In total, the testing time of all LSTM-based methods is almost same. The disadvantage of our method is the longer training time, while it has the advantage of the highest recognition accuracy, and the testing time is almost the same as those of the other methods.

## C. PERFORMANCE EVALUATION ON THE OPEN-SOURCE DATASET

We also compared our method with existing approaches on 3 open-source databases. For fair comparison, we set several LSTM-based methods as baselines.

### 1) BASELINES AND CRITERIA

To fully evaluate our model, we compare the proposed method with several baseline methods. The first baseline is long-short term memory (LSTM). The LSTM-based method [5] directly translates the video into natural language. In [19], the "fc7" layer feature was extracted from each frame to feed the feature sequences into the LSTM network at each moment, and the LSTM outputs a corresponding word at each moment. The second baseline is video-to-text (S2VT) [13], which is a stacked two-layer LSTM where the first layer encodes video frames. S2VT uses the CNN output as an input feature, and once all the frames are read, the model generates a sentence verbatim. The third baseline is LSTM combined with an attention mechanism (LSTM-A) [16]. This model takes advantage of the global time structure and focuses on the most relevant time frames. The fourth baseline is LSTM-E [41], a model based on visual semantic embedding. Here, the CNN is used to extract the visual features of the selected video frames for a given video. Video representations are generated by pooling the values of these visual features.

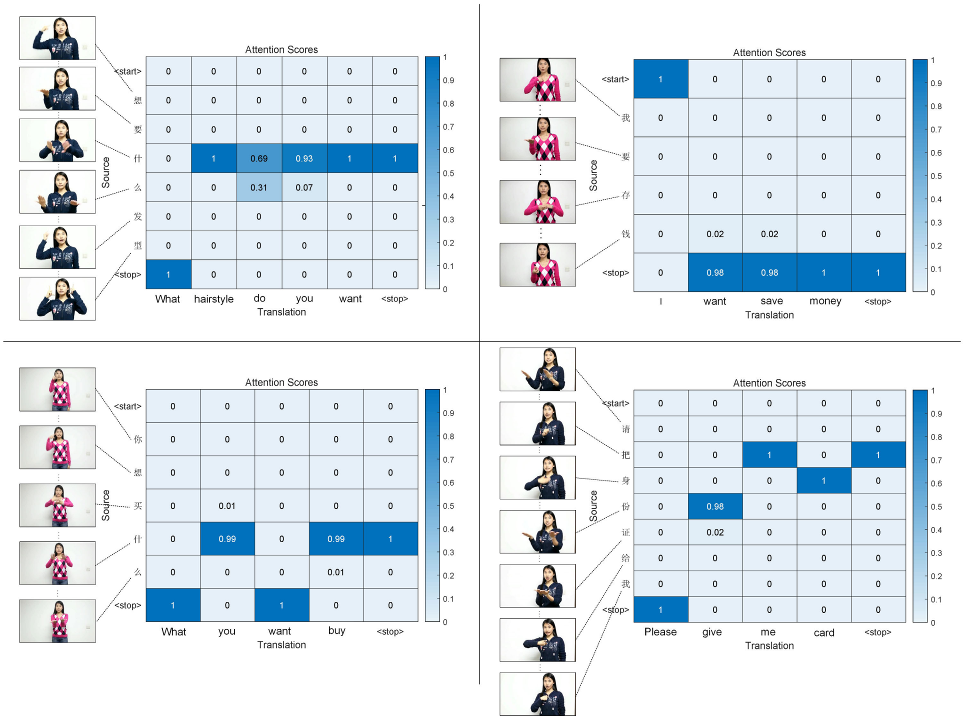

**FIGURE 8.** Example attention score matrix. The proposed end-to-end sign language translation system can translate key words well by using the attention mechanism.

**TABLE 3.** Time cost comparison of the proposed method and other methods.

| Method (Number of sentences) | Train time | Test time |
|---|---|---|
| LSTM (10) (Liu *et al.* [7]) | 0 h 51 m 25 s | 0.22 s |
| LSTM + HMM (10) (Guo *et al.* [56]) | 1 h 12 m 23 s | 0.23 s |
| LSTM + Attention (10) (Ma *et al.* [57]) | 1 h 28 m 43 s | 0.25 s |
| Bi-ST-LSTM-A (10) | 2 h 45 m 13 s | 0.23 s |

Then, at the same time, the visual semantic embedding model of LSTM is used to generate the video sentences and measure the distance between the video and the sentences.

### 2) EVALUATION ON THE CSL DATASET

Tab. 1 summarizes the identification results compared with those of the baseline methods on the continuous CSL dataset. Because our framework is based on LSTM, it is compared to LSTM [5], S2VT [13], LSTM-A [16], LSTM-E [41], and HAN [38].

We tested the sentence recognition accuracy by compensating for the missing alignment during the identification process. The proposed method achieved the highest accuracy. While this alignment is not a true alignment result, we still believe that our proposed scheme is feasible. It is important to note that although all of the LSTM-based models are similar to our model, one of the key differences is that many LSTM-based methods ignore or simplify temporal or spatial information in order to simplify the calculation during the embedding process. We chose to not only retain temporal information but also retain spatial information and optimize the alignment of video sentences. In addition, we compared our model with the traditional continuous SLR algorithms, such as CRF [45] and DTW-HMM [8]. These models require presegmentation of the video when they are identified, possibly leading to segmentation errors. From the comparison results, we can observe that our method presents improved SLR accuracy due to the avoidance of time segmentation. In Tab. 2, we compare recognition performances based on different parameter settings, and it is observed that regardless of the parameter values, our method always obtains the best test results.

**TABLE 4.** Results of continuous CSL recognition for various framework settings in the open-source CSL dataset [38]. The ADAM method is selected for data training, and for LSTM-based method, we set $E_M = 300$, $\epsilon = 3$, $\zeta = 0.001$, $\eta = 100$, $\xi = 0.01$, $\delta = 200$, $n_h = 100$, and $b_m = 50$. The other methods select optimal parameters for training.

| Method | Accuracy |
|---|---|
| LSTM (Liu *et al.* [7]) | $72.33 \pm 1.35\%$ |
| S2VT (Venugopalan *et al.*, [13]) | $73.91 \pm 1.56\%$ |
| LSTM-A (Yang *et al.* [16]) | $75.15 \pm 1.82\%$ |
| LSTM-E (Venugopalan *et al.* [41]) | $75.62 \pm 1.11\%$ |
| HAN (Huang *et al.* [38]) | $78.95 \pm 2.02\%$ |
| Bi-ST-LSTM-A | $\mathbf{80.21 \pm 2.13\%}$ |
| CRF (Lafferty *et al.* [45]) | $70.26 \pm 1.13\%$ |
| DTW-HMM (Zhang *et al.* [5]) | $66.52 \pm 1.57\%$ |

### 3) EVALUATION ON THE RWTH-PHOENIX-WEATHER-2014 DATASET

We evaluate the performance of the proposed method by comparison with some state-of-art methods on the RWTH-Phoenix-Weather-2014 dataset, including HMM [11], Deep Hand [21], recurrent CNN [44], CNN-LSTM-HMMs [49], CNN-TEMP-RNN [46], CNN-Hybrid [22] and SubUNet [53].

Tab. 6 shows the comparison results of continuous SLR on the RWTH-PHOENIX-Weather dataset. Some CNN-based

**TABLE 5.** Results of continuous CSL recognition for various framework settings in the open-source CSL dataset [38]. The ADAM method is selected for data training, and we set $E_M = 250$, $\epsilon = 5$, $\zeta = 0.005$, $\eta = 125$, $\xi = 0.05$, $\delta = 256$, $n_h = 200$, and $b_m = 128$. The other methods select optimal parameters for training.

| Method | Accuracy |
|---|---|
| LSTM (Liu *et al.* [7]) | $71.43 \pm 1.95\%$ |
| S2VT (Venugopalan *et al.* [13]) | $72.81 \pm 1.03\%$ |
| LSTM-A (Yang *et al.* [16]) | $76.25 \pm 2.72\%$ |
| LSTM-E (Venugopalan *et al.* [41]) | $77.42 \pm 2.91\%$ |
| HAN (Huang *et al.* [38]) | $79.25 \pm 2.91\%$ |
| Bi-ST-LSTM-A | $\mathbf{81.22 \pm 2.61}\%$ |
| CRF (Lafferty *et al.* [45]) | $70.26 \pm 1.13\%$ |
| DTW-HMM (Zhang *et al.* [5]) | $66.52 \pm 1.57\%$ |

**TABLE 6.** Performance comparison of different methods on the RWTH-PHOENIX-Weather dataset. (For the LSTM-based method, we select the ADAM method to train the data and set the LSTM parameters as $E_M = 300$, $\epsilon = 3$, $\zeta = 0.001$, $\eta = 100$, $\xi = 0.01$, $\delta = 200$, $n_h = 100$, and $b_m = 50$. The other methods select optimal parameters for training).

| Method | Accuracy |
|---|---|
| HMM (Koller *et al.* [42]) | $45.11 \pm 2.02\%$ |
| Deep Hand (Koller *et al.* [22]) | $53.91 \pm 3.68\%$ |
| Recurrent CNN (Cui *et al.* [50]) | $62.62 \pm 2.15\%$ |
| CNN-LSTM-HMMs (Koller *et al.* [55]) | $73.62 \pm 1.59\%$ |
| CNN-TEMP-RNN (Cui *et al.* [52]) | $75.33 \pm 2.23\%$ |
| CNN-Hybrid (Koller *et al.* [23]) | $62.13 \pm 3.22\%$ |
| SubUNets (Camgoz *et al.* [59]) | $60.21 \pm 2.06\%$ |
| Bi-ST-LSTM-A | $\mathbf{76.12 \pm 1.02}\%$ |

methods always obtain good recognition results, such as deep hand and recurrent CNN. Both deep hand [21] and the recurrent CNN [44] are extensions of the CNN: the former combines the CNN with EM algorithms, while the latter combines the RNN and the CNN. These methods not only exploit the feature learning ability of the CNN but also utilize the time-series modeling ability of the iterative EM and RNN. Our approach uses a similar idea but additionally employs an attention model to strengthen the key content of translation. The approach also uses high-resolution image detection to obtain subtle hand local information and improve the recognition accuracy and robustness of identification. Then, our approach uses the ST-LSTM attention network to generate sign language sentences. The comparison results show that the proposed Bi-ST-LSTM-A network is superior to the other state-of-the-art methods.

### 4) EVALUATION ON THE RWTH-PHOENIX-WEATHER-2014T DATASET

We also use the RWTH-Phoenix-Weather-2014T [4] continuous sign language dataset, which is an extended database of RWTH-Phoenix-Weather-2014, to evaluate the performance of the proposed method. This dataset provides spoken language translations and gloss-level annotations for German sign language videos of weather broadcasts. The dataset is built by 9 different signers and contains a total of 8257 videos. The dataset includes 2887 different isolated sign language words. The videos' size is 210 × 260. We also divide the dataset into three sets for training, validation and testing, and there is no overlap with the previous version of the dataset in any subset. As shown in Tab. 7, the Bi-ST-LSTM-A achieves 75.32% accuracy on the test set. Comparison to some state-of-the-art methods, such as the CNN-LSTM-HMM method [49], Re-Sign [27] and CNN-Temp-CNN [46], shows that our method has the highest recognition accuracy.

**TABLE 7.** Performance comparison of different methods on the RWTH-PHOENIX-Weather dataset.(For the LSTM-based method, we select the ADAM method to train the data and set the LSTM parameters as $E_M = 300$, $\epsilon = 3$, $\zeta = 0.001$, $\eta = 100$, $\xi = 0.01$, $\delta = 200$, $n_h = 100$, and $b_m = 50$, while other methods select optimal parameters for training).

| Method | Accuracy |
|---|---|
| CNN-LSTM-HMMs (Koller *et al.* [49]) | $72.44 \pm 2.13\%$ |
| Re-Sign (Koller *et al.* [27]) | $72.52 \pm 1.83\%$ |
| CNN-TEMP-RNN (Cui *et al.* [46]) | $74.22 \pm 1.45\%$ |
| Bi-ST-LSTM-A | $\mathbf{75.32 \pm 2.11}\%$ |

## V. CONCLUSION

In this paper, a continuous SLR framework based on an ST-LSTM fusion attention network is proposed. We call it Bi-ST-LSTM-A, and it bypasses the sequence segmentation steps. The SL video features are produced by a dual-stream CNN model: one stream analyzes global motion information, while the other focuses on local gesture representation. The ST-LSTM is used for spatial–temporal information fusion, and then,an attention-based Bi-LSTM framework is introduced to measure the correlation between the video and the sentence. Finally, the transformation between the video and the sentence is established by the Bi-ST-LSTM-A network, and the sentence recognition is realized through encoding and decoding operations.

## ACKNOWLEDGMENT

*Qinkun Xiao and Xin Chang contributed equally to this work.*

## REFERENCES

[1] A. Shamama, S. S. Kumar, V. Snehanshu, and A. Vishal, "Hand gesture recognition: A survey," in *Nanoelectronics, Circuits and Communication Systems—Proceeding of NCCS* (Lecture Notes in Electrical Engineering), vol. 511. Ranchi, India: Springer-Verlag, 2019, pp. 365–371.

[2] D. A. Kumar, A. S. C. S. Sastry, P. V. V. Kishore, E. K. Kumar, and M. T. K. Kumar, "S3DRGF: Spatial 3-D relational geometric features for 3-D sign language representation and recognition," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 169–173, Jan. 2019.

[3] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 234–245, Jan. 2019.

[4] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7784–7793.

[5] J. Zhang, W. Zhou, and H. Li, "A threshold-based HMM-DTW approach for continuous sign language recognition," in *Proc. Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*, 2014, pp. 237–240.

[6] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3416–3424.

[7] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2871–2875.

[8] D. Guo, W. Zhou, H. Li, and M. Wang, "Online early-late fusion based on adaptive HMM for sign language recognition," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, pp. 1–18, Jan. 2018.

[9] D. Guo, W. Zhou, M. Wang, and H. Li, "Sign language recognition based on adaptive HMMS with data augmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2876–2880.

[10] W. Yang, J. Tao, and Z. Ye, "Continuous sign language recognition using level building based on fast hidden Markov model," *Pattern Recognit. Lett.*, vol. 78, pp. 28–35, Jul. 2016.

[11] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Comput. Vis. Image Understand.*, vol. 141, pp. 108–125, Dec. 2015.

[12] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4594–4602.

[13] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4534–4542.

[14] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4507–4515.

[15] J. Li, T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1–10.

[16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[17] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network," in *Proc. ICCV*, 2016, pp. 4207–4215.

[18] G. D. Evangelidis, G. Singh, and R. Horaud, "Continuous gesture recognition from articulated poses," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 595–607.

[19] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 474–490.

[20] D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, and J.-M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.

[21] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3793–3802.

[22] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid CNN-HMM for continuous sign language recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[23] P. Buehler, A. Zisserman, and M. Everingham, "Learning sign language by watching TV (using weakly aligned subtitles)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2961–2968.

[24] L.-C. Wang, R. Wang, D.-H. Kong, and B.-C. Yin, "Similarity assessment model for Chinese sign language videos," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 751–761, Apr. 2014.

[25] C. Monnier, S. German, and A. Ost, "A multi-scale boosted detector for efficient and robust gesture recognition," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 491–502.

[26] L. Pigou, A. V. D. Oord, S. Dieleman, M. M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *Int. J. Comput. Vis.*, vol. 126, pp. 430–439, Oct. 2016.

[27] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4297–4305.

[28] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 814–829.

[29] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 724–731.

[30] O. Koller, H. Ney, and R. Bowden, "Automatic alignment of HamNoSys subunits for continuous sign language recognition," in *Proc. Int. Conf. Lang. Resour. Eval. Workshops*, 2016, pp. 121–128.

[31] L. Pigou, S. Dieleman, P. J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 572–578.

[32] T. Pfister, J. Charles, and A. Zisserman, "Large-scale learning of sign language by watching TV (using co-occurrences)," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–11.

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

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[36] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[37] Z. Kaihua, Z. Lei, and Y. Ming-Hsuan, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 864–877.

[38] H. Jie, Z. Wengang, Z. Qilin, L. Houqiang, and L. Weiping, "Video-based sign language recognition without temporal segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2257–2264.

[39] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[40] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-based 3D-CNNs for large-vocabulary sign language recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2822–2832, Sep. 2019.

[41] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 1494–1504.

[42] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[43] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1311–1325, Dec. 2018.

[44] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1610–1618.

[45] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[46] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1880–1891, Jul. 2019.

[47] H. Wang, X. Chai, and X. Chen, "A novel sign language recognition framework using hierarchical Grassmann covariance matrix," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2806–2814, Nov. 2019.

[48] C. D. D. Monteiro, F. M. Shipman, S. Duggina, and R. Gutierrez-Osuna, "Tradeoffs in the efficient detection of sign language content in video sharing sites," *ACM Trans. Accessible Comput.*, vol. 12, no. 2, pp. 1–16, Jul. 2019.

[49] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2306–2320, Sep. 2020.

[50] G. Dan, Z. Wengang, L. Houqiang, and W. Meng, "Hierarchical LSTM for sign language translation," in *Proc. AAAI*, 2018, pp. 6845–6852.

[51] Q. Ma, S. Tian, J. Wei, J. Wang, and W. W. Y. Ng, "Attention-based spatio-temporal dependence learning network," *Inf. Sci.*, vol. 503, pp. 92–108, Nov. 2019.

[52] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, and P. Daras, "Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space," *IEEE Access*, vol. 8, pp. 91170–91180, 2020.

[53] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "SubUNets: End-to-end hand shape and continuous sign language recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3075–3084.

**QINKUN XIAO** was born in 1974. He received the Ph.D. degree from Northwestern Polytechnic University in 2007. From 2006 to 2007, he was engaged in Visiting Research with the University of Victoria, Canada. From 2007 to 2009, he was Post-Doctoral Researcher with Tsinghua University. He is currently a Professor with Xi'an Technological University. His research interests include object recognition and information retrieval, dynamic Bayesian network and image processing.

**XIN CHANG** was born in 1996. She is currently pursuing the degree with Xi'an Technological University. Her research interests include motion recognition and video information processing.

**XUE ZHANG** was born in 1996. She is currently pursuing the degree with Xi'an Technological University. Her research interests include motion recognition and video information processing.

**XING LIU** was born in 1975. He received the master's degree from Northwestern Polytechnical University in 2008 and the Ph.D. degree in aeronautical and astronautical manufacturing engineering from Northwestern Polytechnical University in 2014. In 2015, he joined the School of Electronic Information Engineering, Xi'an Technological University, where he is currently a Professor. His research interests include ammunition design, guidance, and control.