Connectionist Temporal Fusion for Sign Language Translation

Shuo Wang[1], Dan Guo[1*], Wen-Gang Zhou[2], Zheng-Jun Zha[2], Meng Wang[1]
[1] School of Computer and Information Engineering, Hefei University of Technology
[2] University of Science and Technology of China
shuowang.hfut@gmail.com,guodan@hfut.edu.cn,{zhwg,zhazj}@ustc.edu.cn,eric.mengwang@gmail.com

ABSTRACT

Continuous sign language translation (CSLT) is a weakly supervised problem aiming at translating vision-based videos into natural languages under complicated sign linguistics, where the ordered words in a sentence label have no exact boundary of each sign action in the video. This paper proposes a hybrid deep architecture which consists of a temporal convolution module (TCOV), a bidirectional gated recurrent unit module (BGRU), and a fusion layer module (FL) to address the CSLT problem. TCOV captures short-term temporal transition on adjacent clip features (local pattern), while BGRU keeps the long-term context transition across temporal dimension (global pattern). FL concatenates the feature embedding of TCOV and BGRU to learn their complementary relationship (mutual pattern). Thus we propose a joint connectionist temporal fusion (CTF) mechanism to utilize the merit of each module. The proposed joint CTC loss optimization and deep classification score-based decoding fusion strategy are designed to boost performance. With only once training, our model under the CTC constraints achieves comparable performance to other existing methods with multiple EM iterations. Experiments are tested and verified on a benchmark, *i.e.* the RWTH-PHOENIX-Weather dataset, which demonstrate the effectiveness of our proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**;

KEYWORDS

Temporal COV; BGRU; CTC; Fusion; Sign Language Translation

ACM Reference Format:
Shuo Wang, Dan Guo, Wen-Gang Zhou, Zheng-Jun Zha, and Meng Wang. 2018. Connectionist Temporal Fusion for Sign Language Translation. In 2018 ACM Multimedia Conf. (MM18), October 2226, 2018, Seoul, Republic of Korea. ACM, NY, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240671

*Corresponding author.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240671

1 INTRODUCTION

Sign language is widely used in communication with deaf-mute or some other specified scenarios, such as action-based applications, *e.g.*, virtual reality (VR) and augmented reality (AR). It is always accompanied by complicated variations of hand gesture, skeleton movement, finger orientation and facial expression [6]. Vision-based sign language recognition is a challenging task which still has semantic gaps between visual content and natural language in different modalities. It has attracted many researchers' interest. This task is divided into isolated sign language recognition (ISLR) and continuous sign language translation (CSLT). ISLR is a video classification task which builds the mapping between the visual semantics and vocabulary. Different from ISLR, CSLT is a weakly supervised task which translates the frame stream in a video with no extra alignment information to generate correct ordered words [18, 26]. It means that there is no explicit correspondence between sign language actions in the video and words in the sentence label. Therefore, one challenge of the CSLT problem is to learn each frame classification and arrange generated words in the correct order.

The solution of CSLT usually contains two important steps: visual feature extraction and sequential model learning. As for feature extraction, convolutional neural network (CNN) models have been proved to be powerful in many computer vision tasks. For example, the deep residual network (ResNet) model in [13] has better performance than most other deep architectures and can extract better features than hand-craft features in visual classification tasks. Meanwhile, some 3D-CNN models which capture continuous variation on both spatial and temporal dimensions are widely used in the action recognition tasks [31]. In this paper, we adopt a 3D ResNet model (*i.e.*, the C3D model embedded ResNet, denotes as C3D-ResNet) to obtain clip features of sign language videos.

After feature extraction, there are many sequential learning models. One of the basic architectures is recurrent neural network (RNN) [8] which utilizes hidden units to transmit context information. It has shown a significance in the dynamic sequential data learning. For example, 2D CNN features extracted by VGG [28] and GoogleNet [30] are fed into a bidirectional RNN model with long short-term memory (LSTM) [14] to generate words [5]. In addition to various RNN models, some approaches adopt convolution operations to capture the dynamic sequence variation, such as the temporal convolutional neural network (TCN) in [27]. This idea

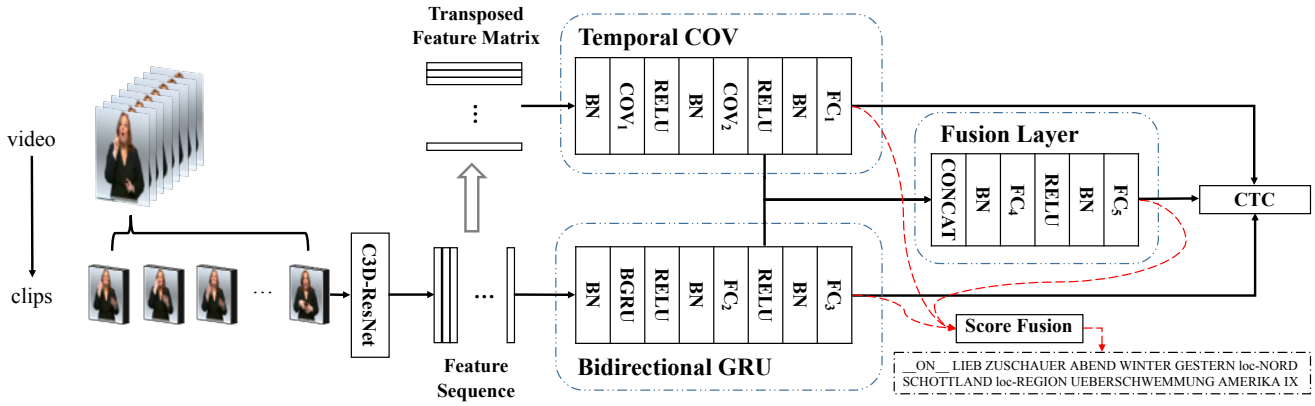

Figure 1: Overview of the proposed CTF (Connectionist Temporal Fusion) framework. Given a video, we firstly extract 3D CNN clip features by the C3D-ResNet model. Then we feed the features into both temporal convolution (TCOV) and bidirectional GRU (BGRU) modules to obtain word classification scores of each clip. After that, a FL module is designed to obtain the complementary relationship between the short-term TCOV and the long-term BGRU temporal learning. Finally, a joint CTC loss optimization and a deep classification score-based fusion strategy are proposed to generate much more correct sentences.

is similar to the n-gram language model [1] in natural language processing (NLP) [4]. Thus we propose a temporal convolution module (TCOV) with the 2-stage convolutional operation as shown in Figures 1 and 2. Similar to the original CNN which calculates local spatial relationships among adjacent pixels, our TCOV captures the local temporal relation on adjacent features. Therefore, we employ both TCOV and BGRU modules to learn sign linguistic in different sequential transition views. BGRU is good at remaining global context of the input data, while TCOV concerns much more current local temporal information by calculating adjacent data. Besides, we design a fusion layer module (FL) with multi-layer perceptron (MLP) to integrate the feature embedding learning for further fusion calculation, which aims at modeling the complementation between the BGRU and TCN.

Finally, we propose a connectionist temporal fusion (CTF) mechanism to effectively translate the continuous visual language in a video into a textual language sentence. On one side, in the training process, a joint CTC (connectionist temporal classification) loss optimization is proposed. The influence of the joint loss are reflected as the following three aspects: TCOV helps BGRU to pay more attention to current sub-sequences; BGRU reminds TCOV with the long-term temporal context; and FL measures the mutual accommodation extent of TCOV and BGRU. On the other side, three deep classification score vectors of these modules are fused to promote the total performance. In a nutshell, our CTF insures TCOV, BGRU and FL modules mutually relative (with jointly CTC training) and independent (under each CTC loss calculation). CTF utilizes the merit of each module.

The main contributions of our method are presented as follows. Experimental results on a large real-world continuous sign language translation benchmark, RWTH-PHOENIX-Weather 2014 [18], demonstrate the effectiveness of our method.

- We design an end-to-end trainable network which benefits from both TCOV and BGRU modules. BGRU keeps the long-term temporal context transition pattern (global pattern), while TCOV focuses on short-term temporal pattern (local pattern) on adjacent clip features.
- We propose a fusion layer with MLP which integrates different feature embedding representations to learn the complementary relationship. It measures the mutual accommodation extent of TCOV and BGRU. In addition, a temporal BN detailed in Figure 3 is contributive by conducting 1-dim normalization at each fixed feature position across temporal dimension with sharing parameters. Both them are proved to be very effective in our experiments.
- More importantly, a joint CTC loss optimization and a deep classification score-based decoding fusion strategy are designed to boost performance. With only once training, our model under the CTC constraints achieves comparable performance to other methods with multiple iterations.

The rest of paper is organized as follows. Section 2 reviews related work. Our proposed method is described in Section 3. Section 4 compares our proposed method with other existing methods and gives experimental analysis. In Section 5, we conclude the paper.

2 RELATED WORK

Early work usually used hand-craft features with traditional sequential learning models to address the ISLR problem. For example, the skeletal data of human body (*i.e.*, depth data) was fed into classical sequential learning models, such as

Hidden Markov Model (HMM) [11, 29] and Hidden Conditional Random Fields (HCRF) [35], to recognize action in videos. Besides depth data, Guo *et al.* added the histogram of oriented gradient (HOG) descriptor of hand into an adaptive HMM model for improving the accuracy of sign word classification [9]. Different from ISLR, CSLT is a sequence to sequence task which translates frame stream in video with various actions into a series of meaningful words. Koller *et al.* extracted sequential features of a video by a 3D-HOG algorithm and predicted continuous corresponding words by HMM [18]. In addition to above depth features and various HOG descriptors, hand shape features were used to solve the American sign language translation problem [29].

Recently, more and more deep learning-based methods have been applied to both visual feature extraction and sequential learning and decoding. They are proved to be robust and effective in many computer vision tasks [13, 14]. Some deep learning methods have been used to solve the ISLR and the CSLR problems. In [19], Koller *et al.* used GoogleNet [30] which is a 2D CNN model to extract the feature of each frame. Huang *et al.* use the 3D CNN model (C3D) to capture the spatiotemporal variation of each sign word action in the video [15].

In sequential learning and decoding process, one of the popular deep learning-based methods is recurrent neural network (RNN), including long short-term memory (LSTM) [10, 14], gated recurrent units (GRUs) [3] and various corresponding bidirectional variants [37]. RNN is widely used in many sequence tasks such as visual captioning, visual question answering (VQA) [34, 36, 38] and NLP. The other popular way is temporal convolutional network (TCN), which is proposed for action location and detection [22]. It calculates adjacent data by the shared weight filter and produces the short-term temporal information from sequential data. For example, an end-to-end trainable bidirectional convolutional neural network architecture (*i.e.*, a temporal convolution model) was proposed to translate the sign language video [25].

To solve the matching problem of input and output sequences with different lengths, the connectionist temporal classification function (CTC) is popularly used in unequal sequence alignment tasks, such as speech recognition, text recognition, *ect.* [7]. CTC is also appropriate to deal with the CSLT problem as its difficulty lies in the lack of supervision on accurate temporal segmentation which is the same as speech recognition, but for sign word alignment in the video. Furthermore, the Expectation-Maximization (EM) [23] optimization has been used to obtain better performance [19, 26]. Some work integrated both EM and CTC into a deep learning-based translation model [19, 26]. At M-step, the translated model is trained with CTC optimization and predicts pseudo classification labels at frame level. And at E-step, these pseudo labels are used to optimize another feature extraction model. Therefore, if the translated model obtains better features, the translated performance can be gradually improved with much more reliable pseudo classification labels.

Our most related work is proposed in [5] which uses the bidirectional LSTM (BLSTM) and convolution operation to encode video frames and decode the embedded features to words. It also takes the CNN idea to generate the word classification probabilities in a different view of RNN. Meanwhile, in the training process, the EM iteration is used to fine-tune the feature extraction model with pseudo labels generated by CTC prediction at frame level. By contrast, in our proposed method, at first, we design an end-to-end trainable network which still combines the ideas of CNN and RNN (*i.e.*, TCOV and BGRU modules) in the later explanation in Section 3 but adds a new fusion layer to further learn their complementary relationship. Secondly, we don't use the EM iteration in our framework. We just design a joint CTC operation with once training to constrain the balance of each CTC optimization of these modules (*i.e.*, TCOV, BGRU and FL). Finally, in the testing stage, we adopt a deep classification score-based decoding fusion to utilize the merit of each module. Our results outperform the state-of-the-art with only once end-to-end training by the proposed CTF framework.

3 OUR METHOD

The architecture of our proposed method is described in Figure 1. Given a sign language video $\mathcal{V}$ and the corresponding sentence label with L words $\mathcal{Y} = \{y_1, y_2, ..., y_L\}$, we firstly split the video $\mathcal{V}$ into M clips $\mathcal{C} = \{c_i\}_{i=1}^{M}$ with equal clip lengths and extract each clip feature by the C3D model embedded with Residual Network (C3D-ResNet) [13]. Then we feed the clip features into both a temporal convolutional (TCOV) module and a bidirectional GRU (BGRU) module. In addition, we design a fusion layer (FL) module to learn the complementary relationship of TCOV and BGRU modules. Each above mentioned module is optimized by a CTC loss function. We train our model with a joint CTC loss learning strategy. In the testing stage, a connectionist temporal fusion strategy $O_{fusion} = score\{O_{tcov}, O_{bgru}, O_{fl}\}$ is proposed to improve the performance of the translation, where O_{tcov}, O_{bgru}, and O_{fl} are respective outputs (*i.e.* classification score vectors) of TCOV, BGRU and FL modules.

3.1 Feature extraction by C3D-ResNet

Compared with the 2D-CNN, 3D-CNN considers both spatial and temporal relationships across sequential frames in the video [31]. The ResNet model embedded into 3D CNN models has been proved its effectiveness in many action recognition and detection tasks [12]. It has a strong ability for video representation. In this paper, we adopt the C3D-ResNet model [1] to generate the representation of each clip as follow. Denoting the video with N frames as $\mathcal{V} = \{v_i\}_{i=1}^{N}$, we split $\mathcal{V}$ into clips by dividing operation with sliding window size l and the overlap size o. And the number of clips is calculated by $M = \lfloor \frac{N-o}{l-o} \rfloor$, where function $\lfloor x \rfloor$ returns the max integer that is less than x. Then all clips are represented by the 18th-layer output of C3D-ResNet as

$$\mathcal{F} = \{f_1, f_2, ..., f_M\} = \{\mathbf{\Omega}_\theta(c_i)\}_{i=1}^{M} \quad (1)$$

[1] https://github.com/kenshohara/3D-ResNets-PyTorch

where model parameter θ of C3D-ResNet $\mathbf{\Omega}$ is initialized by the pre-trained model on an ISLR dataset in [26, 40], and $f_i \in \mathbb{R}^d$ is the feature vector of the i-th clip. In this paper, we set $d = 512$, $l = 8$ and $o = 4$, which means half of the frames overlap between the adjacent clips.

3.2 The main framework

Temporal-COV Module: Motivated by the n-gram language model used in the NLP (*i.e.*, Natural Language Processing) task, we design a temporal convolution (TCOV) module to learn the embedding semantics of contiguous features. In the TCOV module, we conduct the convolution operation with n-item contiguous features. Actually, it calculates local convolutional information of adjacent clip features in the view of short-term temporal relation.

Given the clip features $\mathcal{F} \in \mathbb{R}^{d\times M}$, we transform it to $\mathcal{F}' \in \mathbb{R}^{M\times d}$. As depicted in Figure 2, there are 2-layer convolution operations. The TCOV module can be summarized as follow:

$$\begin{aligned} Q' &= \{q'_i\}_{i=1}^{M} = COV_{\Phi_2}[COV_{\Phi_1}(\mathcal{F})] \\ Q &= \{q_i\}_{i=1}^{M} = FC_{\delta_1}(Q') = Q' \cdot W_1 + b_1 \end{aligned} \tag{2}$$

where Φ_1, Φ_2, and δ_1 respectively denotes the model parameters of COV_1, COV_2 and a full connected (the 1st FC in Figure 1, denoted as FC_1) layers.

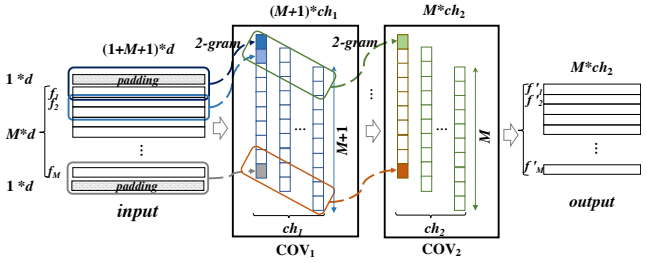

Figure 2: Illustration of temporal convolution operations in the TCOV module. With the filter number ch_1 and ch_2, we learn the feature embedding transformation with twice 2-gram (2-item) temporal convolution operations.

If we set the parameter format of a convolution layer as (number of channels, height, width, stride, padding), the first COV layer was set as (ch_1, n, d, 1, 1) and the second COV layer has parameters (ch_2, n, ch_1, 1, 0), where n is the number in n-gram setting, we set n=2 in the paper; and ch_1 and ch_2 are respective filter numbers of COV_1 and COV_2, we set $ch_1 = 1024$ and $ch_2 = 2048$. To maintain the consistence to the temporal dimension M, we set the padding parameter in COV_1 as 1 and strides $s_1 = s_2 = 1$. As described in Figure 2, $\mathcal{F}'' = padding(\mathcal{F}')$ is in the size of $[1 + M + 1, 512]$, and by the stride operation, it finally returns to M on the height dimension again. Thus we have COV_1 with parameters [1024, 2, 512, 1, 1] and COV_2 with [2048, 2, 1024, 1, 0]. In fact, TCOV gradually magnifies the embedding representation of visual features with contiguous n-items. We try to learn more detailed semantics on visual features on temporal dimension by convolution operations. Until now, the output of temporal convolutional network is in size of $[M, 2048]$. Besides, we also use the normal function ReLU [21] and the Batch normalization (BN) operation [17] after each convolution layer to avoid training over-fitting and boost the training speed in our experiments.

After 2-stage convolution operations, a fully connected layer is used to calculate the classification score vector of each clip. The dimension of weight matrix W_1 is set as $[2048, k]$ and bias b_1 is a $[k]$-dim vector, where the k is the size of vocabulary Voc. Finally, the output of the TCOV module is in size of $[M, k]$.

Here we specially introduce the usage of BN in our paper. Traditional BN calculates normalization elements of a whole image feature map and updates the BN parameters γ and β during the training process. The output of BN is always calculated by $output = \frac{input - mean[input]}{\sqrt{Var[input]}} * \gamma + \beta$, where $input$ is an image feature map. But in this part, we adopt a one-dimensional BN strategy on the temporal dimension. Taking the clip feature matrix $\mathcal{F}' \in \mathbb{R}^{M\times d}$ as an example, as shown in Figure 3, the 1-dim BN is conducted on a fixed feature dimension to measure the normalization relation across temporal dimension, where $input$ for the proposed BN is a M-dim vector. As the feature dimension is d, we implement d times BN operations on $\mathcal{F}'$ with the same sharing BN parameters. Here the temporal BN is not used for feature normalization, but for clip normalization on each fixed feature dimension. It still belongs to considering the temporal relation but with the convolutional calculation way.

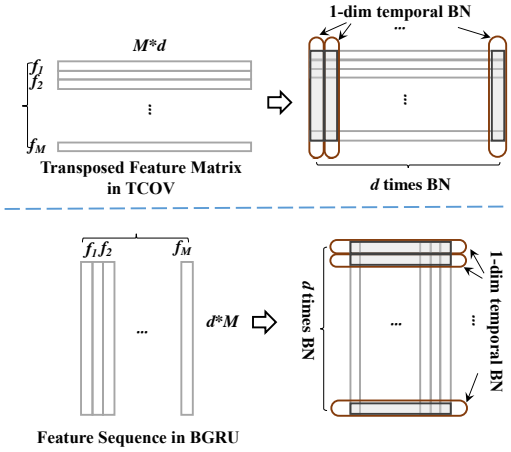

Figure 3: Illustration of 1-dim temporal BN.

Bidirectional-GRU Module: In addition to the above mentioned TCOV in the view of short-term temporal convolution operations with n-items, we propose a bidirectional-GRU (BGRU) module to handle the global sequence learning on strict long short-term temporal dimension. Compared with classical unidirectional RNN models [3], the basic BGRU

model alleviates gradient disappearance and has the ability to model much more longer temporal dependency. Here we use a basic BGRU unit to calculate both the forward pass of sequential clip features from $i = 1$ to M and the backward pass from $i = M$ to 1. Defining the $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ as the forward and the backward outputs of a BGRU unit, the output of i-th clip by the BGRU layer is concatenated as:

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{3}$$

where we set the hidden unit sizes of both $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ to 2048 in the paper, thus h_i is a 4096-dim vector.

After the BGRU layer, we also take ReLU and BN operations as in the TCOV module to speed up the training process and promote the performance of training. As shown in Figure 3, the same temporal BN is conducted on the clip dimension (*i.e.* temporal dimension) in BGRU module. Besides, we set two full connected layers (FC_2 and FC_3) followed by the same function (4):

$$P = \{p_i\}_{i=1}^{M} = FC_\delta(H) = H \cdot W + b \tag{4}$$

where H is the input variable feeded into the FC_2 or the FC_3 layer, and the model parameters $\delta_2|[W_2, b_2]$ and $\delta_3|[W_3, b_3]$ respectively correspond to FC_2 and FC_3.

There are different settings of TCOV and BGRU modules, which result in the dimension of feature embedding repressions transformed as 512-1024-2048 in TCOV, while 512-4096-8192 in BGRU. The reason is that to obtain a better performance, we should better gradually increase the feature mapping dimension to meet the spatial expanding characteristic of convolutional operation; but as for the sequential learning RNN mode, 4096 is the hidden unit size of BGRU and in normal situation, RNN models usually have better performance with larger hidden unit size. Thus we set the maximum value under the up-limit of calculation capacity of GPU. Under the same condition, we set $W_2 \in [8192, 4096], b_2 \in [4096]$. Finally, the dimension of weight matrix W_3 is set as $[4096, k]$ and bias b_1 is a $[k]$-dim vector. The output of the BGRU module is in size of $[M, k]$ too.

Fusion Layer (FL) Module: To discover more deeply latent complementary relationship between the short-term TCOV and the long-term BGRU modules, a fusion layer module based on MLP is designed. Note that the last FC layer in either TCOV or BGRU module is severed for outputting the word classification score vector for later CTC calculation. If we try to get the appropriate feature embedding representations in both TCOV and BGRU for fusion, we have to go back to former steps in our model as in Figure 1. From the prior setting, the outputs of the "RELU" layer in TCOV and BGRU are respectively in size of $[M, 8192]$ and $[M, 2048]$, therefore the concatenated input for FL is in size of $[M, 10240]$.

The full connected layers (FC_4 and FC_5) in this module follow formula (4) too, where $\delta_4|[W_4, b_4]$ and $\delta_5|[W_5, b_5]$ are the model parameters of this proposed MLP structure. We set W_4 and W_5 to the sizes of $[10240, 1024]$ and $[1024, k]$, respectively. In this part, we also adopt ReLU and the proposed 1-dim temporal BN to this FL module.

3.3 CTC optimization and score fusion

CTC optimization: In this paper, connectionist temporal classification (CTC) is an objective function to find a decoded sentence $\mathcal{Y}$ with the maximum sum of probabilities of various alignments $\{\pi\}$ between input and target sequences [7]. It focuses on the correct word order without strict feature alignment boundaries corresponding to sequential words in a sentence. In other words, CTC is a measurement metric for weakly-supervised learning.

Given a input sequence as $\mathcal{X}$, the probability of a CTC alignment path π is defined as follow:

$$p(\pi|\mathcal{X}) = \prod_{j=1}^{|\mathcal{X}|} p(\pi_j|\mathcal{X}), \forall \pi_j \in Voc' \tag{5}$$

where π has the same sequence length as $\mathcal{X}$, π_j is the j-th element of π and $|\mathcal{X}|$ is the sequence length of $\mathcal{X}$. Here Voc'=$Voc \cup \{\text{'\_'}\}$. CTC introduces a new "blank" label ('_') into vocabulary Voc.

To transform π into a variable sentence $\mathcal{Y}$, CTC introduces a many-to-one mapping operation $\mathcal{B}$ which removes "blank" and repeated words in π, *e.g.*, $\mathcal{B}$(_ *a a* _ _ *book*)= $\{a\ book\}$. Therefore, the probability of a labeling $\mathcal{Y} = (y_1, y_2, ..., y_L)$ with L words is calculated as the sum of the probabilities of all word alignments corresponding to it:

$$p(\mathcal{Y}|\mathcal{X}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathcal{Y})} p(\pi|\mathcal{X}) \tag{6}$$

where $\mathcal{B}^{-1}(\mathcal{Y}) = \{\pi|\mathcal{B}(\pi) = \mathcal{Y}\}$.

Finally, taking clips $\mathcal{C} = \{c_i\}_{i=1}^{M}$ as the input $\mathcal{X}$, the CTC loss is defined as follow:

$$\mathcal{L}_{CTC} = -\log p(\mathcal{Y}|\mathcal{C}) \tag{7}$$

We propose a joint CTC end-to-end training optimization for the sequence to sequence learning with unequal video lengths. It combines three CTC losses from TCOV, BGRU and FL modules to jointly learn the short-term, the long-term and the mutual complementary relation as follow:

$$\mathcal{L} = \rho_1 \mathcal{L}_{CTC}(tcov) + \rho_2 \mathcal{L}_{CTC}(bgru) + \rho_3 \mathcal{L}_{CTC}(fl) \tag{8}$$

where ρ_1, ρ_2 and ρ_3 are hyper-parameters controlling the weight of each module. ρ_1, ρ_2, and ρ_3 are discussed in our experiments. We train the model by ADAM optimization with an initialized learning rate 10^{-4}, weight decay 0.00001, and beats ranging from 0.5 to 0.999. After training 30 epochs, we reduce the learning rate from 10^{-4} to 10^{-5}.

Online deep score decoding: In the testing process, we use a fusion strategy based on deep classification scores to decode a generated sentence. With the outputs of the i-th clip in TCOV, BGRU and FL modules $\{O_{tcov,i}, O_{bgru,i}, O_{fl,i}\} \in \mathbb{R}^{[M,k]}$, we use the *softmax* operation to normalize each CTC score vector, and sum different normalized score vectors as follow:

$$O^j_{fusion,i} = \frac{1}{|Mo|} \sum_{mo \in Mo = \{tcov, bgru, fl\}} \frac{e^{O^j_{mo,i}}}{\sum_{j'=1}^{k} e^{O^{j'}_{mo,i}}} \tag{9}$$

where j denotes the j-th position in a score vector, Mo is the module set {TCOV, BGRU and FL}, and here $|Mo|$=3.

As for the decoding phase, we used the function *argmax* on $O_{fusion,i}$ and output the i-th word classification label with the maximum score value. Thus we obtain M word labels and adopt a 2-stage greedy strategy to remove redundant labels (*i.e.*, "blank"'_' and continuous repeated words). For example, delete the "blank " label at the 1-st stage, "*I _I _ have _a a _ book*" $\rightarrow$ "*I I have a a book*". Since that the adjacent clips have 50% overlapping frames ($l = 8$ and $o = 4$) which is easy to generate redundant labels, we delete continuous repetitions in the 2-nd stage, e.g., "*I I have a a book*" $\rightarrow$ "*I have a book*".

4 EXPERIMENTS

This section introduces a SLT dataset benchmark and evaluation metrics in our experiments. We compare our method with other existing methods and give experimental analyses and discussion.

4.1 Dataset and evaluation

We mainly experiment on a German continuous sign language dataset, namely RWTH-PHOENIX-Weather 2014 [18], which is a common benchmark for the SLT task. This dataset contains 6841 videos performed by 9 signers. The details of the dataset are available in Table 1. Note that the vocabulary of the TRAIN set does not contain all the words in VAL and TEST sets. We split the videos in TRAIN/VAL/TEST sets into 190536/17908/21349 clips, respectively.

Table 1: RWTH-PHOENIX-Weather Dataset.

	#TRAIN	#VAL	#TEST
Num of videos	5672	540	629
Voc size	1231	461	497

Word Error Rate (WER) is a metric to measure the similarity between two sentences. Given the pair of a generated sentence and the ground truth, it counts the least operations of substitution, deletion, and insertion referenced to the ground truth as formula (10). Lower WER means the fewer word errors. We denote the number of words in the ground truth sentence as $\#num\_words$. There are two auxiliary evaluations "del" and "ins", which represent the proportions of deletion and insertion operations calculated as follows:

$$WER = \frac{\#insertions + \#deletions + \#substitutions}{\#num\_words} * 100\% \quad (10)$$

$$del = \frac{\#deletions}{\#num\_words} * 100\% \quad (11)$$

$$ins = \frac{\#insertions}{\#num\_words} * 100\% \quad (12)$$

4.2 Model validation

This paper focuses on the combined CTC optimization and fusion. With experimental experiences, we find that our CTF model with the hyper-parameter settings $\rho_1 = 1, \rho_2 = 1$ and $\rho_3 = 0.5$ has the best performance. In this subsection, we test some main contributions of our CTF model and give analysis and discussion.

1-dim temporal BN. Under above mentioned hyper-parameters setting, we test each decoding performance of TCOV, BGRU and FL modules with the proposed BN. The proposed BN is contributive to improve the performance of each module as shown in Table 2. For example, "BGRU w/o BN" means removing the BN step (without BN) in the BGRU module, "TCOV w/o BN" and "FL w/o BN" have the same meanings as "BGRU w/o BN". With the BN, the WER performance of the BGRU module reduces from 43.2% to 39.7% and 42.5% to 39.9% on VAL and TEST sets, respectively. There are also approximate 3% improvement on the FL module with BN. In addition, the output of the FL module has the best decoding performance.

Table 2: Comparison on 1-dim temporal BN.

Dataset	BGRU w/o BN	BGRU	TCOV w/o BN	TCOV	FL w/o BN	FL
VAL	43.2	**39.7**	43.8	**42.6**	42.5	**39.1**
TEST	42.5	**39.9**	41.8	**41.6**	42.1	**39.4**

Fusion Layer. This part verifies the effectiveness of the FL module. "w/o FL" means removing the FC module in the proposed CTF framework. Thus it is equal to remove $L_{CTC}(fl)$ in formula 8 and $O_{fl,i}$ in formula 9. The influence of the FL module is shown in the Table 3. Both TCOV and BGRU combined with FL are optimized slightly by 1% improvement on VAL and TEST sets. However, FL performs better than TCOV and BGRU. Especially the CTF model (Fusion $\mathcal{L}$)using formula 9 with three modules together has the best performance, which is much better than others.

Table 3: Comparison on Fusion Layer.

	Method	VAL			TEST		
		del	ins	WER	del	ins	WER
w/o FL	TCOV	13.8	6.3	43.7	13.9	6.5	42.7
	BGRU	11.5	7.9	40.7	10.5	8.0	40.3
	Fusion $\mathcal{L}$	14.7	5.0	39.5	13.7	4.8	38.2
FL	TCOV	14.9	6.3	**42.6**	14.4	6.5	**41.6**
	BGRU	10.8	7.3	**39.7**	9.8	8.1	**39.9**
	FL	11.5	5.8	**39.1**	11.1	6.4	**39.4**
	Fusion $\mathcal{L}$	12.8	5.2	**37.9**	11.9	5.6	**37.8**

Discussion on hyper-parameter ρ_3. Here we further discuss the influence of FL in the whole CTF model. To keep the consistency, we set hyper-parameters $\rho_1 = 1$ and $\rho_2 = 1$, and just change ρ_3 from 0.5 to 5.5. Figure 4 illustrates the performance curves of different fusion configurations with various module combinations. Results show that "Fusion" (*i.e.*

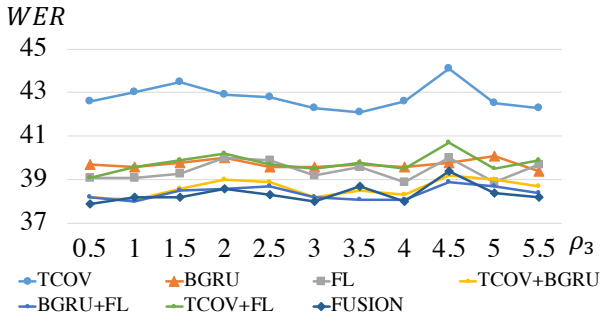

Figure 4: The Performances on hyper-pamamter ρ_3.

Table 4: Comparisons with Different CTC Optimizations.

Method	VAL			TEST		
	del	ins	WER	del	ins	WER
$\mathcal{L}_{CTC}$ & single FL	14.3	4.9	40.1	13.5	5.1	39.8
$\mathcal{L}$ & single FL	11.5	5.8	39.1	11.1	6.4	39.4
$\mathcal{L}$ & total CTF	12.8	5.2	**37.9**	11.9	5.6	**37.8**

the whole CTF model) stably performs better than others. $\rho_3 = 0.5$ is still the best setting for our CTF model. The reason is that in the CTF model, the most contributive module is BGRU, and TCOV is a supplementary module. TCOV and BGRU come to a compromise using FL. But FL prefers to BGRU. Therefore, if we set a larger ρ_3, it is equal to magnify the proportion of BGRU and weaken the contribution of TCOV. Thus $\rho_3 = 0.5$ is an appropriate weighting value to maintain the activity of TCOV and keep the diversity and balance of TCOV, BRGU and FL.

Joint CTC Loss optimization. Here we experiment the end-to-end training of CTF with $\mathcal{L}_{CTC}(fl)$ and $\mathcal{L}$, respectively. As shown in Table 4, we just take the measurement on CTC score outputs of both single module FL and total CTF. We know that the outputs of single FL with $\mathcal{L}$ has slight WER improvement than $\mathcal{L}_{CTC}(fl)$. There is more than 2% improvement on deletion operations, while -1% negative effect on insertion operations. It means under $\mathcal{L}$, FL can effectively eliminate redundant words with jointly short-term and long-term context considerations, but still needs the help of TCOV which has a strong capability on recognizing some special new words with local semantics (mostly corresponding to insertion operations). To fix this insufficiency, we validate score-based decoding fusion below. Anyhow total CTF with $\mathcal{L}$ still has an obvious improvement on the performance WER.

Score fusion. With the best hyper-parameter setting $\rho_1 = 1$, $\rho_2 = 1$ and $\rho_2 = 0.5$, we discuss the fusion performances with different module configurations by formula (9). As shown in the Table 5, the combination of two modules ({TCOV, BGRU}, {TCOV, FL} or {BGRU, FL}) performs better than the single module (TCOV, BGRU or FL), and the combination of three modules together for fusion is the best

Table 5: Comparison on Score Fusion.

Fusion Module Set	VAL			TEST		
	del	ins	WER	del	ins	WER
{TCOV}	14.9	6.3	42.6	14.4	6.5	41.6
{BGRU}	10.8	7.3	39.7	9.8	8.1	39.9
{FL}	11.5	5.8	39.1	11.1	6.4	39.4
{TCOV, BGRU}	13.3	5.5	38.2	12.0	5.9	38.1
{(TCOV, FL}	13.5	5.4	39.1	12.9	5.4	38.9
{BGRU, FL}	11.6	5.8	38.2	10.7	6.5	38.5
{TCOV, BGRU, FL}	12.8	5.2	**37.9**	11.9	5.6	**37.8**

configuration. This result validates the complementary relationship of each module. The proposed score fusion utilizes the advantage of each module.

An example is shown in Figure 5. Given a video with 27 clips, we decode multiple sentences according to the outputs of modules TCOV, BGRU, FL and the total CTF framework. Referenced to "ground-truth", the values of the evaluation metric WER of BGRU, TCN, FL and our CTF are 16%, 50%, 16% and 0% respectively. BGRU is good at segmenting continuous clip regions for word alignment. TCOV excels at discovering special new words with local semantics among adjacent features, such as recognizing words "NOCH", "DONNERSTAG" and "IN-KOMMEND" at clips 16,19 and 21. FL is a compromise between TCOV and BGRU. In this example, FL almost votes to BGRU. Finally, with score fusion, our CTF trusts TCOV with word "IN-KOMMEND" at clip 19, while standing at the side of BGRU and FL at clips 16 and 21. It demonstrates the effectiveness of our CTF model.

4.3 Comparison with other existing methods

As shown in Table 6, "r-hand', "traj' and "face" respectively denote extracting feature descriptors from "right hand" images, "trajectory motion" with skeletal information and "face" images. "Extra supervision" means utilizing additional offline optimizations, such as using multiple EM iterations on a hybrid CNN-HMM framework for weak supervision [18–20]. Here we analyze the differences among these models. Both HOG-3D [18] and CMLLR [18] belong to traditional HMM-based learning with different hand-craft features. Then turning to deep features, Cui et. al. proposed a three-step training optimization [5]. In [2, 19], both hand features and global image features were used to solve the CSLT problem. [16] introduced a hierarchical attention mechanism. And the model in [26] is trained five times by the EM optimization procedure. Our proposed method has just one-step training using only global image features without any other optimization except for the joint CTC-based fusion. And our result outperforms the state-of-the-art with only once end-to-end training. Our method is effective and competitive.

Besides, we also verify the generalization of our method on a Chinese sign language dataset [16]. We split the dataset into training and testing sets as in [16]. Experimental results

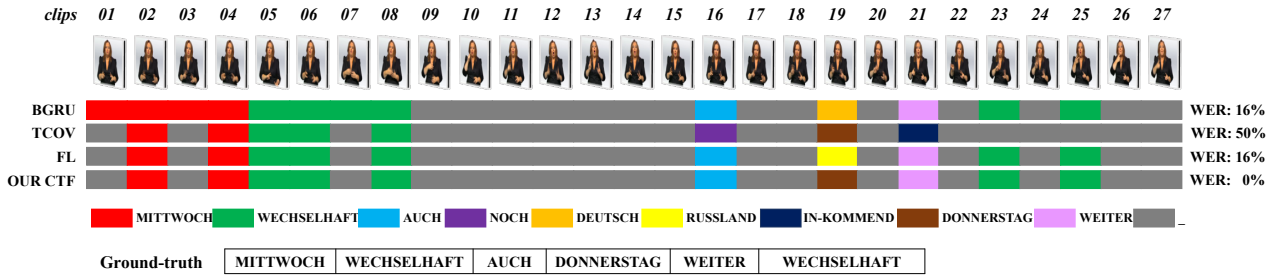

Figure 5: An example of generated sentences by different modules in our CTF model.

Table 6: Compared with other existing methods on RWTH-PHOENIX-Weather Dataset.

Method	Extra supervision	Modality			VAL		TEST	
		r-hand	traj	face	del / ins	WER	del / ins	WER
HOG-3D [18]		√			25.8 / 4.2	60.9	23.2 / 4.1	58.1
CMLLR [18]		√	√	√	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1-Mio-Hands [19]	√	√			19.1 / 4.1	51.6	17.5 / 4.5	50.2
1-Mio-Hands [18, 19]	√	√	√	√	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid [20]	√	√			12.6 / 5.1	38.3	11.1 / 5.7	38.8
Staged Optimization [5]		√			13.7 / 7.3	39.4	12.2 / 7.5	38.7
SubuNets [2]		√			14.6 / 4.0	40.8	14.3 / 4.0	40.7
Dilated CNN [26]					8.3 / 4.8	38.0	7.6 / 4.8	**37.3**
LS-HAN [16]					-	-	-	38.3
OUR CTF					12.8 / 5.2	**37.9**	11.9 / 5.6	37.8

Table 7: Evaluation on a Chinese sign language dataset

Method	LSTM [33]	S2VT [32]	LSTM-A [38]	LSTM-E [24]	HAN [37]	DTW-HMM [39]	LS-HAN [16]	OUR CTF
WER	26.4	25.5	24.3	23.2	20.7	28.4	17.3	**11.2**

are shown in the Table 7. DTW-HMM belongs to traditional HMM-based sequential learning, and other LSTM-based methods all adopted the encoding-decoding framework widely used in NLP domain. Among these methods, HAN and LS-HAN introduced the attention mechanism to measure the influences of all input sources to current decoding position. By contrast, we propose a novel view to address the CSLT problem. Our model directly decodes the classification label along temporal dimension of the input sequence. We focus on word alignment with the joint CTC-based loss optimization and score fusion by integrating temporal-COV, BGRU and FL modules. Experimental results demonstrate our CTF has already improved by 6.1% of WER.

5 CONCLUSION

This paper proposes a connectionist temporal fusion optimization for sign language translation. Considering different properties of TCOV, BGRU and FL modules, it simultaneously captures the short-term, long-term and complementary relation among sequential clip features in a given sign language video. In the process of training, we conduct the joint CTC loss optimization; and as for the testing process, we use an online fusion strategy on the deep classification scores to decode the video into languages. Experiments on a large continuous sign language benchmark RWTH-PHOENIX-Weather-2014 dataset verify the effectiveness of our proposed method compared with the state-of-art work. For the future work, we will focus on integrating other optimizations into our framework, such as attention-based models or EM iterative optimization.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61472392, 61432019, 61622211, 61620106009, 61632007, 61732008, and 61725203.

REFERENCES

[1] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based N-gram Models of Natural Language. *Computational Linguistics* 18, 4 (1992), 467–479.
[2] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. 2017. Subunets: End-to-end Hand Shape and Continuous Sign Language Recognition. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. 3075–3084.
[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.

2014. Learning Phrase Representations using RNN Encoder-decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078* (2014).
[4] Gobinda G Chowdhury. 2003. Natural Language Processing. *Annual Review of Information Science and Technology (ARIST)* 37, 1 (2003), 51–89.
[5] Runpeng Cui, Hu Liu, and Changshui Zhang. 2017. Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 1610–1618.
[6] PR Futane, RV Dharaskar, and VM Thakare. 2012. A Comparative Study for Approaches for Hand Sign Language. In *IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET)*. 36–39.
[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*. 369–376.
[8] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech Recognition with Deep Recurrent Neural Networks. In *Acoustics, Speech, and Signal Processing (ICASSP)*. 6645–6649.
[9] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2017. Online Early-late Fusion based on Adaptive HMM for Sign Language Recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 1 (2017), 1–18.
[10] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2018. Hierarchical LSTM for Sign Language Translation.. In *Association for the Advancement of Artificial Intelligence (AAAI)*.
[11] Dan Guo, Wengang Zhou, Meng Wang, and Houqiang Li. 2016. Sign language recognition based on adaptive hmms with data augmentation. In *Image Processing (ICIP), 2016 IEEE International Conference on*. 2876–2880.
[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Learning Spatio-temporal Features with 3D Residual Networks for Action Recognition. In *ICCV Workshop on Action, Gesture, and Emotion Recognition*, Vol. 2. 4.
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 770–778.
[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9, 8 (1997), 1735–1780.
[15] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. 2015. Sign Language Recognition using 3D Convolutional Neural Networks. In *International Conference on Multimedia and Expo (ICME)*. 1–6.
[16] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based Sign Language Recognition without Temporal Segmentation. *arXiv preprint arXiv:1801.10111* (2018).
[17] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167* (2015).
[18] Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers. *Computer Vision and Image Understanding* 141 (2015), 108–125.
[19] Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Deep Hand: How to Train a CNN on 1 Million Hand Images when Your Data is Continuous and Weakly Labelled. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 3793–3802.
[20] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. 2016. Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In *British Machine Vision Conference (BMVC)*. 12.
[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*. 1097–1105.
[22] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. 2016. Temporal Convolutional Networks: A Unified Approach to Action Segmentation. In *European Conference on Computer Vision (ECCV)*. 47–54.
[23] Geoffrey McLachlan and Thriyambakam Krishnan. 2007. *The EM Algorithm and Extensions*. Vol. 382.
[24] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly Modeling Embedding and Translation to Bridge Video and Language. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 4594–4602.
[25] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. 2018. Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video. *International Journal of Computer Vision (IJCV)* 126, 2-4 (2018), 430–439.
[26] Junfu Pu, Wengang Zhou, and Houqiang Li. 2018. Dilated Convolutional Network with Iterative Optimization for Coutinuous Sign Language Recognition. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 885–891.
[27] Colin Lea Michael D Flynn René and Vidal Austin Reiter Gregory D Hager. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. (2017), 1003–1012.
[28] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).
[29] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time American Sign Language Recognition using Desk and Wearable Computer based Video. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 20, 12 (1998), 1371–1375.
[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. 2015. Going Deeper with Convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.
[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. 4489–4497.
[32] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to Sequence-video to Text. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. 4534–4542.
[33] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating Videos to Natural Language using Deep Recurrent Neural Networks. *arXiv preprint arXiv:1412.4729* (2014).
[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 3156–3164.
[35] Sy Bor Wang, Ariadna Quattoni, L-P Morency, David Demirdjian, and Trevor Darrell. 2006. Hidden Conditional Random Fields for Gesture Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Vol. 2. 1521–1527.
[36] Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. Ask Me Anything: Free-form Visual Question Answering based on Knowledge from External Sources. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 4622–4630.
[37] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*. 1480–1489.
[38] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing Videos by Exploiting Temporal Structure. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. 4507–4515.
[39] Jihai Zhang, Wengang Zhou, and Houqiang Li. 2014. A Threshold-based HMM-DTW Approach for Continuous Sign Language Recognition. In *International Conference on Internet Multimedia Computing and Service (ICIMCS)*. 237.
[40] Jihai Zhang, Wengang Zhou, Chao Xie, Junfu Pu, and Houqiang Li. 2016. Chinese Sign Language Recognition with Adaptive HMM. In *International Conference on Multimedia and Expo (ICME)*. 1–6.