PARALLEL TEMPORAL ENCODER FOR SIGN LANGUAGE TRANSLATION

Peipei Song *Dan Guo** *Haoran Xin* *Meng Wang*

School of Computer Science and Information Engineering, Hefei University of Technology
beta.songpp@gmail.com, guodan@hfut.edu.cn, jackxin8@foxmail.com, eric.mengwang@gmail.com

ABSTRACT

This paper addresses the sign video interpretation which is a weakly supervised task. Each sign action in videos lacks exact boundaries or labels. We design a Parallel Temporal Encoder (PTEnc) to learn the temporal relation of a sign video from local and global sequential learning views in parallel. PTEnc utilizes the complementarity between the local and global temporal cues. Then, fused encoded feature sequence is fed into a Connectionist Temporal Classification (CTC) based sentence decoder. In addition, in order to enhance the temporal cues in each video, we introduce a reconstruction loss, which performs in an unsupervised way without additional labels. The CTC loss cooperates with the reconstruction loss in an end-to-end training manner. Experimental results on a benchmark dataset demonstrate the effectiveness of the proposed method.

Index Terms— Sign language translation, connectionist temporal classification, parallel temporal encoder, reconstruction loss.

1. INTRODUCTION

Sign language is a kind of language which conveys semantic information by human behaviors. It is challenging to observe the visual variations of gestures, human postures and facial expressions, and translate them into natural language. Thus, automatic Sign Language Translation (SLT) is a cross-modality semantic understanding task [1, 2, 3]. It aims to learn the mapping between visual frame streams and grammatical ordered words. However, there is a huge semantic gap between visual and textual context transformation. Additionally, SLT is a weakly supervised task. Each video is only labeled with an ordered word sequence, but no exact boundaries for each sign action.

Therefore, the challenges of SLT are divided into two aspects: one is to learn discriminative visual features containing temporal relation in the videos, and the other is to address the weakly supervision challenge in this task. In the absence of temporal labels at word level, the accurate alignment between a feature sequence and a sentence is hard to achieve. The former considers good features describing visual content, while the latter focuses on the transformation between visual and textual semantics.

Recently, Convolutional Neural Networks (CNNs) show superior performance in image feature extraction, such as ResNet [4], VGG [5] and GoogleNet [6]. Meanwhile, 3D CNN is widely used in various video analyses which considers both spatial and temporal variations in videos. In this paper, we use a 3D CNN model embedded ResNet (C3D-ResNet [7]) to extract clip features of sign videos. C3D-ResNet explores short-term temporal cues, $i.e.$, the temporal relation in adjacent clips. To learn the global temporal relation in each entire sign video, we propose a Parallel Temporal Encoder (PTEnc), which encodes C3D-ResNet clip features from both local and global sequential learning views. PTEnc utilizes the complementarity between the local and global temporal relation. Then fused encoded feature sequence is fed into a Connectionist Temporal Classification (CTC) based decoder for sentence generation. The CTC loss maximizes the probability of alignments of target sequence.

Furthermore, to address the weakly supervision challenge in SLT, we devote to unsupervised learning strategy [8, 9]. For example, the Neural Machine Translation (NMT) model [9] improves the model performance by a dual translation training process ($i.e.$, English-to-German translation versus German-to-English translation). In this paper, we introduce a reconstruction loss, which measures the distance between original and reconstructed clip features. While minimizing the difference, the model is pushed to learn temporal cues in an unsupervised way. This reconstruction loss is combined with the CTC loss to optimize the proposed model together.

The contributions of the paper are presented as follows:

- We design a Parallel Temporal Encoder (PTEnc) to learn the efficient temporal relation from both local and global sequential learning views among continuous clips in a video.
- We propose a reconstruction loss to enhance the temporal cues in videos, which works in an unsupervised way without additional word-level labels.
- The CTC optimization is used to maximize the alignments probability of the target sequence. Both the CTC and reconstruction losses are jointly drawn into training in an end-to-end manner. The proposed model achieves comparable performance to the state-of-the-arts.

* Corresponding author.

978-1-5386-6249-6/19/$31.00 ©2019 IEEE

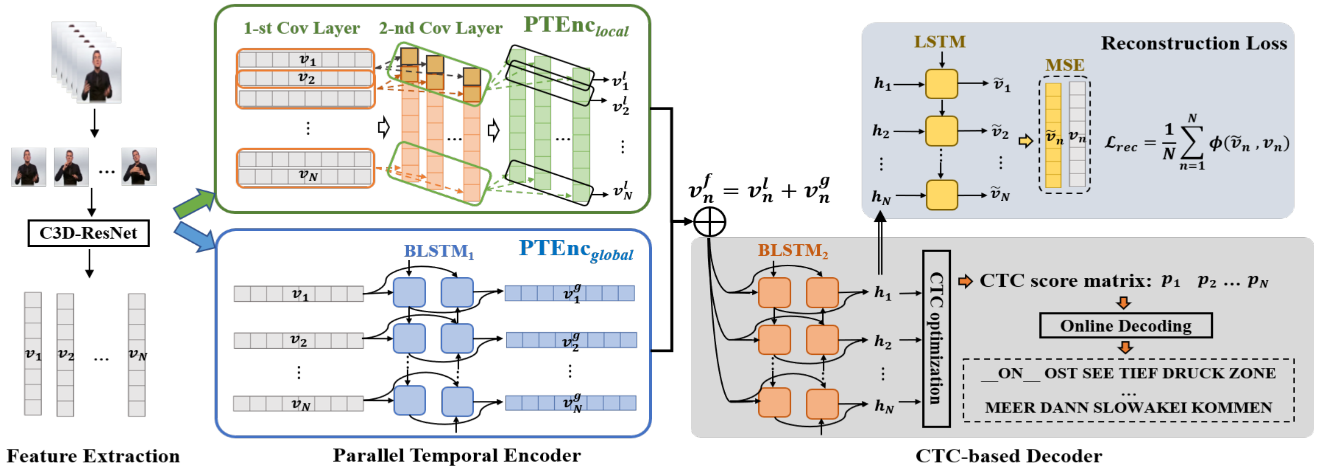

Fig. 1. The overall framework. A pretrained model, C3D-ResNet, is used to extract clip features of each video. The features are fed into sequential encoding modules PTEnc$_{local}$ and PTEnc$_{global}$ in parallel, and the proposed model fuses the local and global temporal correlations. At last, a CTC-based decoder generates a predicted sentence. In addition, we adopt a reconstruction loss in the unsupervised way to further improve the performance of the model.

2. OUR APPROACH

The proposed model is described in Fig.1. The SLT task is to generate a natural language sentence for a given sign video. We firstly divide the video into n clips of equal length, and use a pretrained C3D-ResNet model [7] to extract the clip features. We denote the extracted clip features as $\boldsymbol{\mathcal{V}} = \{v_n\}_{n=1}^{N}$. Then, based on the features, a Parallel Temporal Encoder (PTEnc) are proposed to model both global and local sequential learning in the video. These two branches are fused and fed into a Connectionist Temporal Classification (CTC) module to decode the sign probabilities of clip features. In addition, in order to enhance the temporal cues in the video, we adopt an unsupervised learning idea to measure the distance between original and reconstructed visual features, which is considered as a reconstruction loss. Finally, the proposed model jointly uses the reconstruction loss and a CTC loss to address the SLT problem. Each component of the model is detailed as follows.

2.1. Parallel Temporal Encoder

As shown in Fig.1, PTEnc is comprised of two parallel branches named as PTEnc$_{local}$ and PTEnc$_{global}$. These two branches have the same input and output data sizes.

PTEnc$_{local}$. PTEnc$_{local}$ aims to learn the local temporal relation between adjacent clips. We conduct a two-stage convolution module on these adjacent two clip features to learn the local relation.

The two-stage convolution module in PTEnc$_{local}$ have convolution kernels $h_1 \times w_1 \times c_1$ and $h_2 \times w_2 \times c_2$ (abbreviated from $height \times width \times channel$), respectively. PTEnc$_{local}$ is formulated as:

$$\boldsymbol{\mathcal{V}_l} = \left\{v_n^l\right\}_{n=1}^{N} = \left\{Conv_2\left[Conv_1\left(v_n\right)\right]\right\}_{n=1}^{N}. \qquad (1)$$

where $Conv_1$ and $Conv_2$ stand for the operations of two convolution layers, respectively.

We set $h_1 = h_2 = 2, w_1 = 514, w_2 = c_1 = 1024, c_2 = 2048$, and the stride to 1. This means that PTEnc$_{local}$ gradually learns the local temporal relation between adjacent two clip features. To output the consistent temporal dimension N, we set padding as 1 in the convolutional operations. Besides, we also adopt ReLU [10] and one-dimensional batch normalization operation [11], to avoid overfitting and boost the training speed.

PTEnc$_{global}$. PTEnc$_{global}$ aims to learn the global temporal relation of a video. Recurrent Neural Network (RNN) is widely used for capturing sequence temporal relation [12, 13, 14]. Long Short-Term Memory network (LSTM) is a variant of RNN and performs better at long-term dependency problem [15, 16]. Moreover, Bidirectional LSTM (BLSTM) is composed of LSTM units in two directions, which considers the temporal correlation of forward and backward transmissions. Thus, in this paper, we adopt BLSTM as the basic unit of PTEnc$_{global}$.

BLSTM uses two LSTM units to encode input sequence from two directions at each time step t. And it computes the forward and backward hidden states $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. The two vectors are concatenated together as the output $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$[17, 18]. We denote the BLSTM module in PTEnc$_{global}$ as $BLSTM_1$. The PTEnc$_{global}$ can be formulated as follows:

$$\boldsymbol{\mathcal{V}_g} = \{v_n^g\}_{n=1}^{N} = \{BLSTM_1\left(v_n\right)\}_{n=1}^{N}. \qquad (2)$$

Fusion. In order to effectively utilize the complementarity between $\boldsymbol{\mathcal{V}_l}$ and $\boldsymbol{\mathcal{V}_g}$, we integrate them into a fused feature sequence. The fused feature sequence is obtained with sum operation in this paper:

$$\boldsymbol{\mathcal{V}_f} = \left\{v_n^f\right\}_{n=1}^{N} = \left\{v_n^l + v_n^g\right\}_{n=1}^{N}. \qquad (3)$$

2.2. CTC-based Decoder

After encoding aforementioned features, this paper proposes a CTC-based decoder for sentence translation. Since BLSTM excels at modeling forward and backward contexts in the sequential learning compared with basic RNN modules [12, 15], it is more robust and effective in capturing action variations in SLT. In this paper, we choose BLSTM as the unit of decoder:

$$\boldsymbol{\mathcal{H}} = \{h_n\}_{n=1}^{N} = \left\{BLSTM_2\left(v_n^f\right)\right\}_{n=1}^{N}. \tag{4}$$

CTC optimization. Here we use the CTC optimization in [19] as an objective function of the decoder. At first, with the outputs of $BLSTM_2$, we use a fully connected layer FC to embed them into a non-normalized CTC categorical probability sequence with K classes:

$$\boldsymbol{\mathcal{P}} = \{p_n\}_{n=1}^{N} = \{FC(h_n)\}_{n=1}^{N}. \tag{5}$$

where $p_n \in \mathbb{R}^K$ is the CTC categorical probability vector of n-th clip, and K is equal to the vocabulary size plus 1 (the blank symbol '_').

At the stage of training, suppose the generated sentence $\boldsymbol{\mathcal{Y}}$ is $\{y_m\}_{m=1}^{M}$. Let $\pi = \{\pi_n\}_{n=1}^{N}$ denote a CTC alignment, which is composed of a sequence of blanks and words. Denote $p_n = \left\{p_n^k\right\}_{k=1}^{K}$, where p_n^k is the probability of label k in vector p_n. The probability of π is given by the product of probabilities:

$$P(\pi) = \prod_{n=1}^{N} P(\pi_n) = \prod_{n=1}^{N} p_n^{\pi_n}. \tag{6}$$

A target sequence can have multiple different alignments. CTC summarizes a many-to-one map as $\mathcal{B}$, which removes all blanks and repeated labels from the alignments. The CTC loss function is defined as:

$$\mathcal{L}_{ctc} = -log \sum_{\pi=\mathcal{B}^{-1}(\boldsymbol{\mathcal{Y}})} P(\pi). \tag{7}$$

where $\mathcal{B}^{-1}(\boldsymbol{\mathcal{Y}}) = \{\pi \mid \mathcal{B}(\pi) = \boldsymbol{\mathcal{Y}}\}$ is the set of all the alignments.

CTC Online Decoding. In the testing stage, we implement online decoding by the $argmax$ function on the CTC score matrix $\{p_n\}_{n=1}^{N}$, and output the word classification labels with the maximum score. A 2-stage greedy strategy on the label sequence is used to merge an output sentence, which removes blank label at 1-th stage and deletes continuous repetitions at 2-th stage.

2.3. Reconstruction Loss

In order to obtain more temporal cues in the video, we introduce a reconstruction loss. The reconstruction loss calculates the distance between original and reconstructed clip features.

As shown in Fig.1, the feature reconstruction process is realized by a normal LSTM unit. The outputs of $BLSTM_2$, $\{h_n\}_{n=1}^{N}$, are taken as the input of LSTM, and the outputs of LSTM are considered as the reconstructed clip features. The clip features reconstruction process is formulated as:

$$\widetilde{\boldsymbol{\mathcal{V}}} = \{\widetilde{v}_n\}_{n=1}^{N} = \{LSTM(h_n)\}_{n=1}^{N}. \tag{8}$$

where $\widetilde{v}_n$ denotes the n-th reconstructed clip feature.

We set the hidden state size of LSTM equal to the dimension of feature v_n, and use the Mean Square Error (MSE) loss to represent the average distance between the original and reconstructed clip features of each video. The reconstruction loss is formulated as:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{n=1}^{N} \phi\left(\widetilde{v}_n, v_n\right). \tag{9}$$

where ϕ denotes the function of the MSE calculation.

2.4. Training and model setting

We use a joint loss defined in Eq.10 to train the model, which contains a CTC loss and a reconstruction loss.

$$\mathcal{L} = \mathcal{L}_{ctc} + \beta\mathcal{L}_{rec} \tag{10}$$

where hyperparameter β is used to balance the proportion of $\mathcal{L}_{rec}$. In this paper, we set $\beta = 0.4$.

In the training process, the clip features are obtained from 18-th layer of C3D-ResNet [7], which is pretrained on a Sign Language Recognition (SLR) dataset [3, 20]. The hidden state size of both $BLSTM_1$ and $BLSTM_2$ is 1024. The hidden state size of $LSTM$ and the dimension of clip features are both 512. The proposed model is train by Adam optimizer [21], with a learning rate 10^{-4}, a batch size 100, a beats ranging from 0.5 to 0.999, and a weight decay 10^{-5}.

3. EXPERIMENT

3.1. Dataset and Evaluation

Dataset. Experiments are conducted on a SLT benchmark dataset RWTH-PHOENIX-Weather 2014 [1], which contains 6841 videos performed by 9 different signers. All videos in the dataset are split into 5672/540/629 for training/validation/test, respectively. Each video corresponds to a sentence in German sign language.

Evaluation. Word Error Rate (WER) reflects the similarity between the predicted sequences and reference sequence at word level. As defined in the Eq. 11, a lower WER means a better performance.

$$WER = \frac{\#sub + \#del + \#ins}{\#words\_num} \tag{11}$$

$$del = \frac{\#del}{\#words\_num} \tag{12}$$

$$ins = \frac{\#ins}{\#words\_num} \tag{13}$$

where $\#sub$, $\#del$ and $\#ins$ measure the least operations of substitution, deletion, and insertion referenced to the ground truth, respectively. $\#words\_num$ is the number of words in the reference sequence.

Table 1. Comparison on the RWTH-PHOENIX-Weather 2014 dataset. 'Extra supervision' means additional knowledge or cues are imported, such as [22] introduces an already trained sign language dictionary. 'r-hand', 'traj' and 'face' denote additional features of right-hand poses, trajectories and facial expressions, respectively. 'Iterations' is the times of offline iterative optimizations, based on the Expectation Maximization (EM) algorithm.

Method	Extra	Modality			Iterations	VAL		TEST	
	supervision	r-hand	traj	face		des / ins	WER	des / ins	WER
HOG-3D [1]		√			1	25.8 / 4.2	60.9	23.2 / 4.1	58.1
CMLLR [1]		√	√	√	1	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1M-Hands [22]	√	√			3	19.1 / 4.1	51.6	17.5 / 4.5	50.2
1M-Hands [1, 22]	√	√	√	√	3	16.3 / 4.6	47.1	15.2 / 4.6	45.1
SubUNets [23]			√		1	14.6 / 4.0	40.8	14.3 / 4.0	40.7
Staged-Opt [2]		√			3	13.7 / 7.3	39.4	12.2 / 7.5	38.7
CNN-Hybrid [24]	√	√			3	12.6 / 5.1	38.3	11.1 / 5.7	38.8
Dilated CNN [3]					5	8.3 / 4.8	**38.0**	7.6 / 4.8	**37.3**
Ours					1	12.7 / 5.5	38.1	11.9 / 5.6	38.3

3.2. Comparison and Analysis

Ablation studies. The results of the ablation studies are illustrated in Table 2. $Ours(L)_{ctc}$ is a variant that only uses a single-branch encoder PTEnc$_{local}$ with the CTC loss $\mathcal{L}_{ctc}$, $Ours(G)_{ctc}$ covers only PTEnc$_{global}$, and $Ours(P)_{ctc}$ combines PTEnc$_{local}$ and PTEnc$_{global}$. These models are trained with a single CTC loss. In contrast, $Ours(L)$, $Ours(G)$ and $Ours(Full)$ are trained with both $\mathcal{L}_{ctc}$ and $\mathcal{L}_{rec}$.

Compared to $Ours(L)_{ctc}$ and $Ours(G)_{ctc}$, $Ours(P)_{ctc}$ performs better. The WER is reduced by 1.4/1.4 and 1.0/1.1 on val/test, respectively. This indicates the effectiveness of the proposed PTEnc, which utilizes the complementarity between local and global temporal relation.

By introducing the reconstruction loss, $Ours(L)$, $Ours(G)$ and $Ours(Full)$ are superior to their originals, $i.e.$, $Ours(L)_{ctc}$, $Ours(G)_{ctc}$ and $Ours(P)_{ctc}$, respectively. The reconstruction loss brings 3%∼4% reduction of WER. This verifies that the joint optimization of CTC and reconstruction losses is resultful for SLT. Meanwhile, among these variants of our method, Ours(Full) performs best. This demonstrates the effectiveness of the global and local temporal cues combination and the joint loss optimization again.

Main Comparisons. We compare $Ours(Full)$ with the state-of-the-arts on RWTH-PHOENIX-Weather 2014 [1] dataset. As shown in Table 1, our model achieves comparable performance to the state-of-the-arts without extra supervision and multi-modality information. And our approach even outperforms some methods using offline iterative optimizations by a large margin. Specifically, 1M-Hands [1, 22] imports a trained aforehand sign language dictionary as extra supervision. CNN-Hybrid [24] introduces initial alignments provided by 1M-Hands [1, 22]. Compared with the end-to-end method SubUNets [23], the WER of our method is reduced by 2.7/2.4 on val/test. Besides, Dilated CNN [3] has the closest performance to ours. However, it achieved the best performance with the help of five times offline iterative optimizations. Without offline iterative optimization, the WER of it is 60.3/59.7 on val/test. In contrast, our model is also trained in end-to-end manner and the WER is 38.1/38.3 on val/test. This means, on an end-to-end training, our model improves performance by almost 37%/36% on val/test compared to Dilated CNN [3]. These results quantitatively demonstrate the effectiveness of our approach.

Table 2. Performance comparison of different variants of our method.

Variant	VAL		TEST	
	des / ins	WER	des / ins	WER
$Ours(L)_{ctc}$	14.6 / 5.2	41.0	14.2 / 5.1	41.2
$Ours(G)_{ctc}$	12.6 / 5.1	40.6	11.8 / 5.7	40.9
$Ours(P)_{ctc}$	13.3 / 5.2	39.6	12.4 / 5.6	39.8
$Ours(L)$	13.0 / 5.1	39.3	12.5 / 5.2	39.2
$Ours(G)$	11.4 / 5.2	38.8	10.6 / 5.9	39.3
$\boldsymbol{Ours(Full)}$	12.7 / 5.5	**38.1**	11.9 / 5.6	**38.3**

4. CONCLUSION

This paper proposes a Parallel Temporal Encoder (PTEnc) for sign language translation, which learns both the local and global temporal relation of video. And a CTC-based decoder is explored to translate sentences. To address the weakly supervised challenge in SLT, we also introduce a reconstruction loss, which measures the distances between the original and reconstructed visual features. Finally, both the CTC and reconstruction losses are used to realize the end-to-end optimization process. Experimental results on RWTH-PHOENIX-Weather 2014 [1] exhibit the performance of the proposed model.

5. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (NSFC) under grant 61876058, and the National Key Research and Development Program of China under grant 2018YFC0830103.

6. REFERENCES

[1] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015.

[2] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *CVPR*, 2017, pp. 7361–7369.

[3] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterative optimization for continuous sign language recognition.," in *IJCAI*, 2018, pp. 885–891.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[7] D. Tran, J. Ray, Z. Shou, S. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *arXiv preprint arXiv:1708.05038*, 2017.

[8] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W. Ma, "Dual learning for machine translation," in *NIPS*, 2016, pp. 820–828.

[9] Z. Tu, Y. Liu, L. Shang, X. Liu, and H. Li, "Neural machine translation with reconstruction," in *AAAI*, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[12] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*. IEEE, 2013, pp. 6645–6649.

[13] C. Lea, M. D Flynn, R. Vidal, A. Reiter, and G. D Hager, "Temporal convolutional networks for action segmentation and detection," in *CVPR*, 2017, pp. 156–165.

[14] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 430–439, 2018.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, 2015, pp. 4507–4515.

[17] M. Schuster and K. K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*. ACM, 2006, pp. 369–376.

[20] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, "Chinese sign language recognition with adaptive hmm," in *ICME*. IEEE, 2016, pp. 1–6.

[21] D. P Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] 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 *CVPR*, 2016, pp. 3793–3802.

[23] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *ICCV*. IEEE, 2017, pp. 3075–3084.

[24] O. Koller, O Zargaran, H. Ney, and R. Bowden, "Deep sign: hybrid cnn-hmm for continuous sign language recognition," in *BMVC*, 2016.