Boosting Continuous Sign Language Recognition via Cross Modality Augmentation

Junfu Pu[1], Wengang Zhou[1,2], Hezhen Hu[1*], Houqiang Li[1,2]
[1]CAS Key Laboratory of Technology in GIPAS, EEIS Department, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
pjh@mail.ustc.edu.cn,zhwg@ustc.edu.cn,alexhu@mail.ustc.edu.cn,lihq@ustc.edu.cn

ABSTRACT

Continuous sign language recognition (SLR) deals with unaligned video-text pair and uses the word error rate (WER), *i.e.,* edit distance, as the main evaluation metric. Since it is not differentiable, we usually instead optimize the learning model with the connectionist temporal classification (CTC) objective loss, which maximizes the posterior probability over the sequential alignment. Due to the optimization gap, the predicted sentence with the highest decoding probability may not be the best choice under the WER metric. To tackle this issue, we propose a novel architecture with cross modality augmentation. Specifically, we first augment cross-modal data by simulating the calculation procedure of WER, *i.e.,* substitution, deletion and insertion on both text label and its corresponding video. With these real and generated pseudo video-text pairs, we propose multiple loss terms to minimize the cross modality distance between the video and ground truth label, and make the network distinguish the difference between real and pseudo modalities. The proposed framework can be easily extended to other existing CTC based continuous SLR architectures. Extensive experiments on two continuous SLR benchmarks, *i.e.,* RWTH-PHOENIX-Weather and CSL, validate the effectiveness of our proposed method.

CCS CONCEPTS

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

KEYWORDS

Cross Modality Augmentation, Sign Language Recognition

ACM Reference Format:
Junfu Pu, Wengang Zhou, Hezhen Hu, Houqiang Li. 2020. Boosting Continuous Sign Language Recognition via Cross Modality Augmentation. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413931

1 INTRODUCTION

According to the official statistics from World Health Organization (WHO) in 2020, there are around 466 million people with disabling hearing loss, which accounts for over 5% of the world's population. Hearing loss leads to difficulty in hearing conversational speech. As a result, people with hearing loss often use sign language for communication. As a kind of visual language, sign language conveys semantic meanings by gestures and hand movements, together with facial expressions. During the long-term evolution, sign language develops its own characteristic rules and grammar. To facilitate such communication, many research efforts have been devoted to continuous sign language recognition (SLR), which aims to automatically identify the corresponding sign word sequence from a given sign video. It's a transdisciplinary research topic which involves computer vision, natural language processing, and multimedia analysis, *etc.* Due to the expensive labeling cost, the continuous sign videos are generally weakly labeled, which means there is no alignment annotation of text sign words to video frames in the sign video.

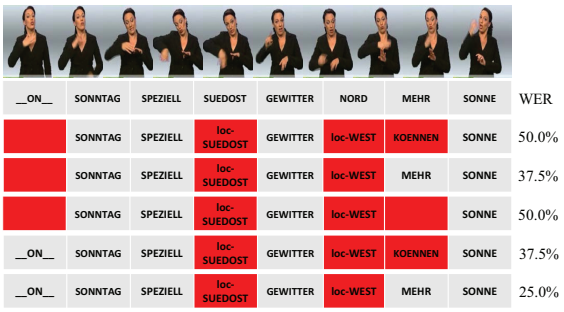


Figure 1: An example of decoding result. It shows five predicted sentences with descent decoding probability from Candidate 1 to 5. The box with red background denotes the false prediction.

Early works [35, 45] on continuous SLR rely on hand-crafted visual features and statistical sequential models, *i.e.,* Hidden Markov Model (HMM). Recently, with the success of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in various computer vision tasks, more and more deep learning based sign language recognition algorithms have been proposed and achieved remarkable performance. Among them, most of the state-of-the-art continuous SLR approaches [2, 10, 29, 47] utilize connectionist temporal classification (CTC), which is a popular technique to deal with sequence-to-sequence transformation without accurate alignment. In CTC based methods, beam search algorithms are used for decoding, which iteratively produces word candidates. Basically, the decoding precision and speed depend on the beam width. The number of candidates for the decoded sequences is also equal to the beam width. In practice, we choose the candidate with maximum decoding probability as the final predicted sentence.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413931

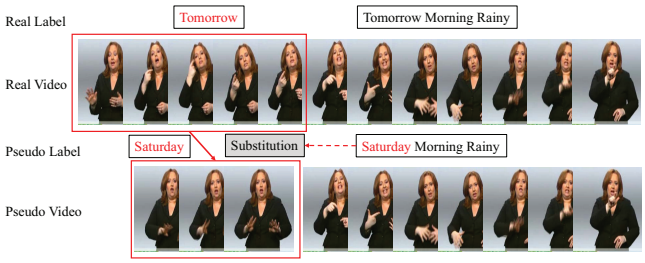
However, due to the inconformity between CTC objective and evaluation metric, the candidate with maximum decoding probability may not be the best one under the evaluation metric, *e.g.*, word error rate (WER). For example, in Figure 1, although Candidate 5 has the minimum decoding probability among all candidates, it is actually the best one with the lowest WER. To quantitatively illustrate this issue, we study the Top-K WER on RWTH-PHOENIX-Weather based on the state-of-the-art method proposed in [10]. Here, Top-K WER is defined as follows: we choose the candidate with the lowest WER out of the K decoded candidates and calculate the average WER over the whole dataset. That is to say, Top-K WER is a lower bound over the decoding results. When the candidate with maximum decoding probability has the lowest WER for all testing samples, WER equals Top-K WER. According to our experiments, the WER on RWTH-PHOENIX-Weather testing set is 23.8%, while Top-5 WER decreases to 19.7%. In order to bridge the performance gap, our target is to make the candidate with the best performance have the maximum decoding probability, which means to minimize the distance between such candidate and sign video.

With the motivation discussed above, in this paper, we present a novel architecture for further boosting the performance of continuous SLR via cross modality augmentation. In continuous SLR, WER is the most important evaluation metric, which is defined as the least operations, *i.e.,* substitution, deletion, and insertion, to transform the target sequence to the reference sequence. To simulate the calculation procedure of WER, we edit the sign video and corresponding text label following the same operations, as illustrated in Figure 2. With such editing, we augment the cross modality data and obtain a pseudo video-text pair. In order to minimize the cross modality distance from the video to ground truth label, while maximizing the distance from video to the pseudo text label, we propose a real-pseudo discriminative loss. Besides, the objective includes alignment-based CTC loss for both real and pseudo video-text pair. A cross modality semantic correspondence loss is also introduced to directly minimize the cross modality distance between real video and real ground truth text label. The proposed framework can be easily extended to other existing CTC based continuous SLR architectures. Extensive experiments on two continuous SLR benchmarks, *i.e.,* RWTH-PHOENIX-Weather and CSL, demonstrate the effectiveness of our proposed method.

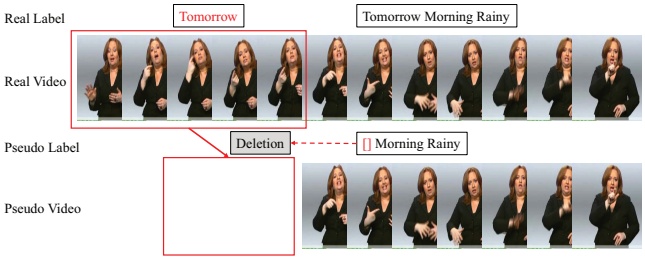
2 RELATED WORK

In this work, we briefly review the key modules and techniques in continuous sign language recognition. Continuous SLR targets at translating the input video into a sequence of glosses in a consistent order. First, the visual encoder transforms the input video into a high-dimensional feature representation. Then the sequential module tries to learn the mapping from this feature representation to the corresponding text sequence. To further refine the recognition result, iterative refinement strategy has been explored with promising results. Besides discussing the above content, we also introduce existing data augmentation techniques in deep learning.

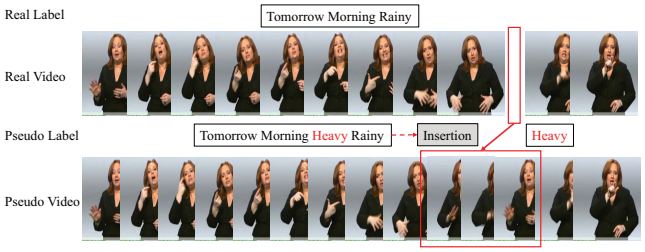
Video representation learning. Discriminative feature representation is crucial for sign language recognition. Early works concentrate on the hand-crafted features, such as HOG or HOG-3D [1, 22], motion trajectories [12, 22, 27] and SIFT [27]. These features are utilized for describing hand shapes, orientations or motion status.



(a) Illustration of the "Substitution" operation.



(b) Illustration of the "Deletion" operation.



(c) Illustration of the "Insertion" operation.

Figure 2: Illustration of different kinds of editing operations.

With the advance of convolutional neural networks (CNNs), many networks are designed for video representation learning. They are based on 2D CNNs [17, 34, 40], 3D-CNNs [3, 30, 31, 38] or a mixture of them [6, 43, 49]. For the task of continuous SLR, various CNNs have been investigated. Oscar *et al.* [25] utilize GoogLeNet [37] as the visual encoder in an end-to-end iterative learning framework. Pu *et al.* [28] and Zhou *et al.* [46] use 3D ResNet [30] and I3D [3] as the feature extractor to jointly model the spatial and temporal information, respectively. There also exist methods [9, 10] using 1D temporal CNNs after 2D CNNs to encode temporal dependency. As one of them, DNF [10] is becomes the most challenging competitor.

Sequential learning. In continuous SLR, there are several popular sequential models, *e.g.* hidden Markov model (HMM), recurrent neural network (RNN) with connectionist temporal classification (CTC) and encoder-decoder network, *etc.*

HMM [21, 25, 26, 42] is one of the most widely used sequential models. Oscar *et al.* [21] embed CNN-LSTM models in each HMM stream following the hybrid approach exploiting the state transitions and the sequential parallelism for sign language recognition. The Recurrent Neural Networks (RNNs), *e.g.* Long Short-Term Memory (LSTM) [18], Gated Recurrent Unit (GRU) [7], have been successfully applied to sequential problems, including speech

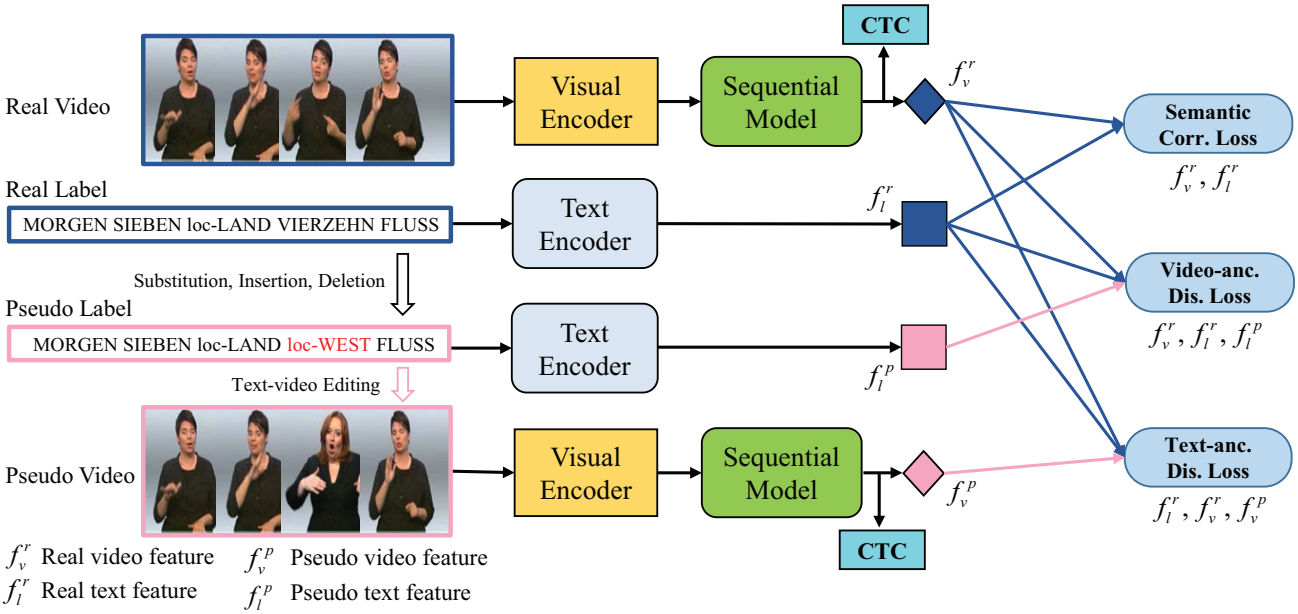


Figure 3: Overview of our proposed framework. The framework consists of a common CNN-TCN visual encoder, sequential model and text encoder. With the cross modality augmentation, we design multiple loss terms to optimize the architecture.

recognition [14], video captioning [33, 44, 48], machine translation [7, 36], *etc.* In continuous SLR, bidirectional LSTM-CTC architecture [2, 9, 10, 29] is employed as a basic model and becomes the most popular one. Besides, there exist some works [16, 20] adopting an attention-aware encoder-decoder network to learn the mapping between visual features and sign glosses. Camgoz *et al.* [8] also utilize the encoder-decoder architecture to extend sign language recognition to sign language translation.

Iterative refinement. Given that the ground truth only provides sentence-level annotations without specific temporal boundaries for each gloss, continuous sign language can be treated as a weakly supervised problem. Current frameworks usually contain a large number of layers and encounter the vanishing gradient problem for low layers, resulting in not fully optimized visual encoder. Recent works demonstrate the importance of the alignments between video clips and sign glosses. The video segment and sign gloss pairs can be treated as the trimmed video classification problem to enhance the visual encoder. In this way, the whole architecture can be optimized in an iterative way for performance boosting. Cui *et al.* [10] and Zhou *et al.* [46] generate the pseudo glosses for video segments by aligning the output probability matrix and output glosses sequence through dynamic programming with notable performance gain. Pu *et al.* [29] utilize utilize the soft Dynamic Time Warping (soft-DTW) as the alignment constraint with the warping path indicating the possible alignments between input video clips and sign words.

Data augmentation in deep learning. Data augmentation is a powerful method to reduce overfitting, which can help the neural network to extract more information from the original dataset. Data augmentation encompasses a series of techniques enhance the quality and size of the training data. It has been successfully applied in various deep learning based approaches. There are many different data augmentation techniques in different tasks. For image-based tasks, *i.e.,* image classification, object detection, *etc*, the image augmentation skills include geometric transformations (*e.g.* rotation, flipping *etc.*), color space transformation (*e.g.* RGB to HSV), kernel filters, random erasing, *etc.* For video-based tasks, *i.e.,* action recognition, tracking, in addition to the image augmentation techniques, video augmentation is performed in temporal dimension by temporal random sampling. In natural language processing tasks, *e.g.* text classification, the operations to augment sentences include synonym replacement, random insertion, random swap, random deletion, *etc.* In existing techniques, the augmented sample share the same label with the original data, and the optimization loss keeps unchanged. In contrast, in our approach, the augmented video or gloss sentence no longer share the same semantic meaning with the original one. We take advantage of the fact and optimize the DNN model with novel triplet losses.

3 OUR APPROACH

In this section, we first give an overview of our framework. After that, we separately discuss the generation of pseudo video-text pairs, network architecture and loss design.

3.1 Overview

Our whole framework is illustrated in Figure 3. During training, given an input video and its corresponding text, we first perform the editing process to create the pseudo text. At the same time, according to the same editing operations, we constitute the pseudo video based on the clip alignment, which is obtained from the refinement stage following [10]. Then we feed the real and pseudo

video into the same recognition framework with shared parameters and calculate the CTC loss, respectively. Further, we explore the relationship between real and pseudo video-text pairs by designing multiple losses to make the network aware of the editing operations and constrain the correspondence of cross-modal data. The final optimization loss is a summation of the CTC loss, real-peseudo discriminative loss and cross modality semantic loss. During the inference stage, the input video is fed into the backbone, *i.e.,* the visual encoder, sequential model and CTC decoding model to output the final prediction text sequence.

3.2 Pseudo Data Generation

Following the definition of word error rate (WER), one of the most widely used evaluation metrics in continuous SLR, we generate pseudo video-text pair by editing raw video and label. WER corresponds to the least operations of substitution, deletion and insertion to transform the target sequence into the reference sequence. In order to simulate the calculation procedure of WER, the videos and labels are edited following these three operations, *i.e.,*, substitution, deletion, and insertion. Given a real video-text pair, we first substitute, insert or delete a word in the real text and repeat this operation a few times. The inserted or substituted new word is randomly picked from the vocabulary in the training set. On the other hand, we perform the same editing operations on the real video according to the alignment extracted in the refinement stage. For a text label and its corresponding sign video, we perform k editing operations, each of which is randomly taken from substitution, deletion, and insertion The editing times k is randomly sample from the range $[1, K]$, where K denotes the maximum editing operations. In this way, we obtain a pseudo video-text pair.

Figure 2 illustrates three different editing operations. Take "Substitution" as an example, as shown in Figure 2a, given a sign video with the label "Tomorrow Morning Rainy", we randomly replace a sign gloss. In this case, sign gloss "Tomorrow" is replaced with "Saturday". Thus, a new pseudo label sequence is generated with such editing operation. After that, all frames corresponding to the sign word "Tomorrow" are also replaced by the frame segment with the meaning of "Saturday" from other videos. Similarly, we can edit the video-text pair with another two operations, *i.e.,*, insertion and deletion. "Insertion" indicates we randomly pick up a text word from the vocabulary and insert it into the original label sequence, while "deletion" means a sign word is randomly deleted from the label sequence.

3.3 Network Architecture

Visual encoder. Visual encoder aims at encoding the input video into semantic feature representation. It consists of a spatial encoder E_{vs} followed by a temporal encoder E_{vt} for spatial-temporal representation. In our implementation, we use the same spatial-temporal backbone proposed in [10] considering its excellent performance. GoogLeNet [37] is selected as our spatial encoder. Temporal encoder contains the architecture of *conv1d-maxpool-conv1d-maxpool*. Specifically, the kernel sizes of 1D temporal convolutional layers and max pooling layers are set as 5 and 2, respectively. The strides for all the layers in V_t are set as 1. Following these settings, the time length is reduced to a quarter of the original video with the receptive field as 16. Given a sign video $\mathbf{V} = \{v_t\}_{t=1}^{T}$ with T frames, the output, *i.e.,* semantic feature representation, is defined as follows,

$$\mathbf{F} = E_{vt}(E_{vs}(\mathbf{V})), \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{C_1 \times T \times H \times W}$ and $\mathbf{F} \in \mathbb{R}^{C_2 \times T/4}$.

Sequential model. The sequential model captures temporal dependency among the semantic feature representations generated by visual encoder, and learn the mapping between visual features and sign glosses. We select Bidirectional Long Short-Term Memory (BLSTM) S_{bi}, which captures the temporal dependency in both forward and backward time steps. It takes the feature sequence as input and generates hidden states $\mathbf{F}_v$ as follows,

$$\mathbf{F}_v = S_{bi}(\mathbf{F}), \tag{2}$$

where $\mathbf{F}_v \in \mathbb{R}^{C_3 \times T/4}$ and C_3 indicates the units of the hidden states. After that, the hidden states $\mathbf{H}$ is utilized as the input of a fully-connected layer f_c and a softmax layer f to generate the probability matrix for all the time steps as follows,

$$\mathbf{P} = f(f_c(\mathbf{F}_v)), \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{N \times T/4}$ and N indicates the number of glosses.

Text Encoder. For the semantic correspondence between the visual feature and gloss sequence, we utilize the text label encoder E_T to map the gloss sequence $\boldsymbol{s}$ into the same latent space as the visual features as follows,

$$\mathbf{F}_l = E_t(\boldsymbol{s}), \tag{4}$$

where $\mathbf{F}_l \in \mathbb{R}^{C_3 \times T/4}$ and a two-layer BLSTM is also utilized as the text encoder.

3.4 Objective Function

To optimize the network, we use three different kinds of loss functions, *i.e.,*, alignment loss, real-pseudo discriminative loss, and cross modality semantic correspondence loss. For each stream, in order to learn the alignment between video and text sequence, connectionist temporal classification (CTC) is introduced. CTC is proposed to deal with two unsegmented sequences without accurate alignment. CTC introduces a blank label out of the vocabulary to account for transitions and silence without precise meaning. There may exist several alignment paths π between the input sequence and target sequence. The probability of each path π is written as follows,

$$p(\pi|\mathbf{V}) = \prod_{t=1}^{T} p(\pi_t|\mathbf{V}), \tag{5}$$

where π_t is the label at time step t, T is the number of frames in video. A many-to-one mapping $\mathcal{B}$ is defined to remove reduplicated words and blank labels. The conditional probability of the target sequence $\boldsymbol{s}$ is calculated as the summation of the probabilities of all alignment paths, which is formulated as follows,

$$p(\boldsymbol{s}|\boldsymbol{V}) = \sum_{\pi \in \mathcal{B}^{-1}(s)} p(\pi|\boldsymbol{V}), \tag{6}$$

where $\mathcal{B}^{-1}$ is the inverse mapping of $\mathcal{B}$. The final CTC objective function is defined as the negative log probability of $p(\boldsymbol{s}|\boldsymbol{V})$, written as follows,

$$\mathcal{L}_{\text{CTC}} = -\ln p(\boldsymbol{s}|\boldsymbol{V}). \tag{7}$$

Denote the real video and pseudo video as $\mathbf{V}_r$ and $\mathbf{V}_p$, respectively. For the two basic streams with real video and pseudo video, we define two CTC loss as alignment loss $\mathcal{L}_A$, written as follows

$$\mathcal{L}_A = \mathcal{L}^r_{\text{CTC}} + \mathcal{L}^p_{\text{CTC}}, \tag{8}$$

where $\mathcal{L}^r_{\text{CTC}}$ and $\mathcal{L}^p_{\text{CTC}}$ are the CTC loss for real video and pseudo video, respectively. The alignment loss targets at maximizing the total probabilities of all alignment paths between the sign video and the label sequence.

In our method, the video data and text label are mapped into the same latent space, which makes it possible for distance measurement. For the input data modalities, *i.e.*,, real video, real label, pseudo video, pseudo label, the corresponding feature representations are denoted as $f^r_v, f^r_l, f^p_v, f^p_l$ from Equation (2) and Equation (4), respectively. We divide the features into two groups, which are (f^r_v, f^r_l, f^p_l) and (f^r_l, f^r_v, f^p_v), respectively. For (f^r_v, f^r_l, f^p_l), the distance of the feature representations between real video and real label is supposed to be closer than that between real video and pseudo label. That is to say, in such triplet, the feature representation of real video is regarded as the anchor, with the real and pseudo label as a positive and negative sample, respectively. We use triplet loss as the objective function to minimize the distance from the anchor to the positive sample, and maximize the distance from the anchor to the negative sample. The real video anchor based real-pseudo discriminative loss is defined as follows,

$$\mathcal{L}_{D_v} = \mathcal{L}_{trip}(f^r_v, f^r_l, f^p_l) = \max\left(\mathcal{D}(f^r_v, f^r_l) - \mathcal{D}(f^r_v, f^p_l) + \alpha, 0\right), \tag{9}$$

where $\mathcal{L}_{trip}(\cdot)$ means triplet loss [19], $\mathcal{D}$ is the distance function, α is a margin. For another group, with the same purpose, the real text anchor based real-pseudo discriminative loss is defined as follows,

$$\mathcal{L}_{D_l} = \mathcal{L}_{trip}(f^r_l, f^r_v, f^p_v) = \max\left(\mathcal{D}(f^r_l, f^r_v) - \mathcal{D}(f^r_l, f^p_v) + \alpha, 0\right). \tag{10}$$

The final real-pseudo discriminative loss $\mathcal{L}_D$ is the summation of these two parts, written as follows,

$$\mathcal{L}_D = \mathcal{L}_{D_v} + \mathcal{L}_{D_l}. \tag{11}$$

The real-pseudo discriminative loss focuses on the relative distance between the video and text label data. For real video-text pair, the distance between the features of such pair in latent space is expected to get as close as possible. Hence, we directly minimize the distance of the real video-text pair, called cross modality semantic correspondence loss. The loss function is defined as the distance metric,

$$\mathcal{L}_S = \mathcal{D}(f^r_v, f^r_l), \tag{12}$$

where $\mathcal{D}$ is the distance metric function. Considering the lengths of video and text are variable, to calculate the distance between two variable length sequences, the distance metric function between f^r_v and f^r_l is defined as the dynamic time warping (DTW) discrepancy. Denoting the cost between f^r_v and f^r_l at different time steps t_1 and t_2 as $d(t_1, t_2)$, DTW typically uses dynamic programming to efficiently find the best alignment that minimizes the overall cost. Define the DTW distance for subsequences $f^r_v(1:i) = (f^r_v(1), f^r_v(2), \cdots, f^r_v(i))$ and $f^r_l(1:j) = (f^r_l(1), f^r_l(2), \cdots, f^r_l(j))$ as $D_{i,j}$, which can be written as

$$D_{i,j} = d(i,j) + \min(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}). \tag{13}$$

In our experiments, d is calculated as the cosine distance, written as follows,

$$d(i,j) = 1 - \frac{f^r_v(i) \cdot f^r_l(j)}{||f^r_v(i)|| \cdot ||f^r_l(j)||}. \tag{14}$$

To make DTW distance differentiable, a continuous relaxation of the minimum operator [4, 11] is introduced with a smoothing parameter $\gamma \geq 0$

$$\min{}^{\gamma}(a_1, \cdots, a_n) := \begin{cases} \min\limits_i a_i & \gamma = 0. \\ -\gamma \log \sum\limits_i e^{-a_i/\gamma} & \gamma \geq 0. \end{cases} \tag{15}$$

With the formulation of DTW, $\mathcal{D}(f^r_v, f^r_l)$ is calculated as follows,

$$\mathcal{D}(f^r_v, f^r_l) = D_{T,N}, \tag{16}$$

where T is the length of f^r_v and N is the length of f^r_l. The distance in triplet loss used in Equation (9) and Equation (10) is also calculated by DTW since the lengths of the items in triplet are variable.

The final objective loss function is defined as follows,

$$\mathcal{L} = \lambda \mathcal{L}_A + (1-\lambda)(\mathcal{L}_D + \mathcal{L}_S), \tag{17}$$

where λ is a hyper-parameter, which indicates the weighted summation of these loss terms. Since $\mathcal{L}_D$ and $\mathcal{L}_S$ have the same distance metrics, we combine them and perform weighted sum with $\mathcal{L}_A$.

4 EXPERIMENTS

In this section, we conduct comprehensive experiments to validate the effectiveness of our method. We first review our benchmark datasets and evaluation metrics. Then we perform ablation studies on each part of our proposed framework. Finally, we compare our method with state-of-the-art approaches on two benchmark datasets.

4.1 Dataset and Evaluation

We perform our experiments on two benchmark datasets, *i.e.*, RWTH-PHOENIX-Weather multi-signer [22], RWTH-PHOENIX-Weather signer-independent [25] and CSL [20] dataset. RWTH-PHO ENIX-WEATHER multi-signer dataset, focusing on the German sign language, is one of the most popular benchmark datasets in continuous SLR. This dataset is recorded from a public television broadcast from a monocular RGB camera at 25 frames per second (fps), with a resolution of 210×260. It contains a total of 6,841 sentences with a total vocabulary size of 1,295 sign words performed by 9 different signers. Besides the training set, two independent sets are proposed for evaluation, *i.e.*, dev and test set. Each set accounts for about 10% of the size of the training set. All 9 signers appear in these 3 sets. Additionally, there is another signer-independent setting, where it chooses 8 signers for training and leaves out signer #5 for evaluation. CSL [20] is a continuous Chinese sign language dataset containing 5,000 videos performed by 50 signers. It contains a total of 100 different sentences with a vocabulary size of 100 sign words in daily life. This dataset is divided into the training and test set, containing 4,700 and 300 videos, respectively. The detailed statistics are listed in Table 1.

We utilize multiple evaluation metrics for continuous SLR. Word error rate (WER) is one of the commonly used metrics. It is actually an edit distance, indicating the minimum number of operations,

Table 1: Statistical data on RWTH-PHOENIX-Weather multi-signer, signer-independent and CSL datasets.

Statistics	RWTH-PHOENIX-Weather Multi-Signer			RWTH-PHOENIX-Weather Signer-Independent			CSL	
	Train	Dev	Test	Train	Dev	Test	Train	Test
#signers	9	9	9	8	1	1	50	50
#frames	799,006	75,186	89,472	612,027	16,460	26,891	963,228	66,529
#duration (h)	8.88	0.84	0.99	6.80	0.18	0.30	10.70	0.74
#vocabulary	1,231	460	496	1,081	239	294	178	20
#videos	5,672	540	629	4,376	111	180	4,700	300

i.e., substitution, deletion and insertion, required to convert the predicted sentence to the reference one:

$$WER = \frac{n_i + n_d + n_s}{L}, \tag{18}$$

where n_i, n_d, and n_s are the number of operations for insertion, deletion, and substitution, respectively. We also calculate the ratio of correct words to the reference words, denoted as Acc-w. Besides, we adopt some semantic metrics in Neural Language Processing (NLP) and Neural Machine Translation (NMT), including BLEU, METEOR, CIDEr and ROUGE-L.

4.2 Implementation Details

In our experiment, we optimize our framework in a staged strategy following the previous methods [10, 29]. First, the backbone GoogLeNet adopts the parameters pre-trained on ILSVRC-2014 [32] dataset. For the end-to-end training stage, the whole framework is supervised by the loss in Equation (17). In the sequential model and text encoder, BLSTMs both have two layers with the hidden states set to 1024. Adam optimizer is utilized with the learning rate as 5e-3 and batch size as 3. The network is trained for 50 epochs till convergence. In continuous SLR, it is crucial for the visual encoder to produce robust feature representation. The loss terms utilized for the first stage have limited contribution to the low layers of the visual encoder due to the vanishing gradient, which makes the visual encoder not fully optimized. Therefore, we use the CTC decoding method to generate pseudo labels for video clips.

After that, in the second stage, we utilize these clip-label pairs for classification. Specifically, we add a fully-connected layer on top of the visual encoder and use the cross-entropy loss to supervise its learning. It is optimized by stochastic gradient descent (SGD) with totally 40 epochs. The initial learning rate is set as 5e-3 and with 10x reduction when loss saturates. The input clip length is 16. We set the batch size as 32 and weight decay as 1e-4, respectively. Embedded with the optimized visual encoder at this stage, we then train the whole framework using the loss in Equation (17) again. Through such optimization strategy, our visual encoder cooperates with the sequential model for better performance. Our whole framework is implemented on PyTorch and experiments are performed on NVIDIA Tesla V100.

Besides, data augmentation is crucial for relieving over-fitting. During training, the video is randomly cropped at the same spatial location along the time dimension, with the resolution of 224×224. Then it is randomly flipped horizontally. Temporally, we randomly discard 20% frames individually. During testing, the video is center cropped at the same spatial location with the resolution of 224×224. All the frames in the video are fed into the framework.

4.3 Ablation Study

We perform ablation studies on the effectiveness of different parts in our framework.

Hyper parameter λ. We study the impact of λ in Equation (17) and the result is shown in Figure 4. The experiments are conducted on RWTH-PHOENIX-Weather multi-signer independent dataset and we utilize the WER result on the dev set as our choosing criterion. It can be seen that the WER result gets improved with a gradual and flexuous process and achieves the best when $\lambda = 0.9$. Notice that $\lambda = 1$ corresponds to baseline which optimized with only CTC loss. When $\lambda = 0$, the network does not converge, so there is no result shown in Figure 4. Unless stated, we utilize it as our default hyper parameters in the following experiments.

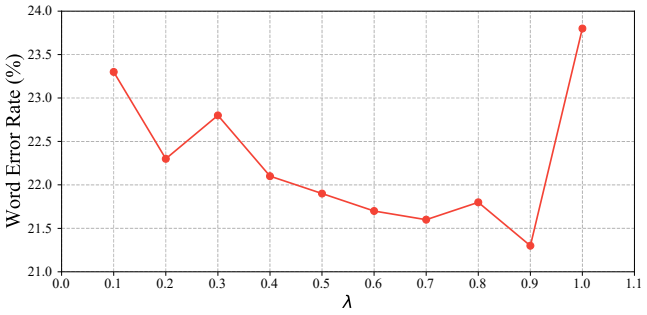


Figure 4: Effects of different hyper parameter λ in Equation (17) on the RWTH-PHOENIX-Weather multi-signer dataset.

Effectiveness on the pseudo video-text pairs. We compare the effectiveness of our pseudo video and text respectively on the RWTH-PHOENIX-Weather multi-signer dataset in Table 2. "Video only" denotes that we only generate pseudo video after editing and supervise the whole framework by the loss terms except the pseudo text related ones. It can be observed that the WER result is improved by 1.8% and 1.9% over the baseline on the dev and test set, respectively. "Text only" denotes that we only generate pseudo text after editing. The improvement on the WER result is similar to the former, with 1.9% and 2.0% on the dev and test set, respectively. When the pseudo video-text pair is generated after editing, the performance is further improved to 21.3% and 21.9% on the dev and

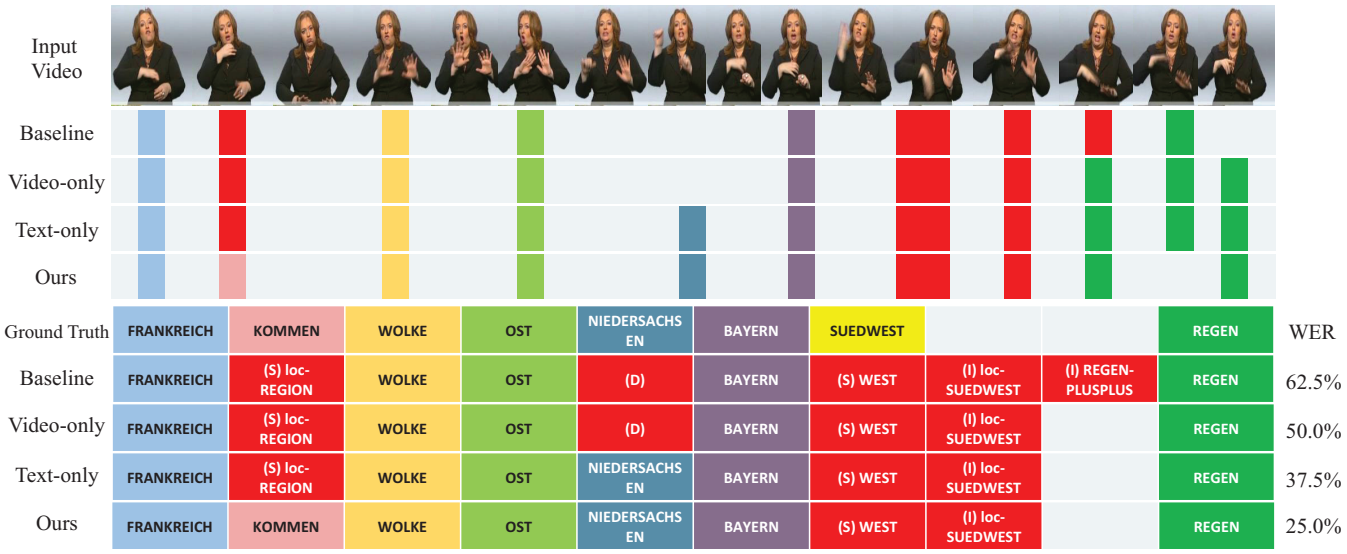


Figure 5: An example on the the dev set of RWTH-PHOENIX-Weather multi-signer dataset. In the figure, the first row represents the input frame sequences. The medium part indicates the sign word with maximum probability at each time step. The bottom part shows the final predicted sentence. Red symbols denotes the wrongly predicted words. "D", "S", and "I" stand for deletion, substitution, and insertion, respectively.

Table 2: An ablation study on the effectiveness of the pseudo video-text pair on the RWTH-PHOENIX-Weather multi-signer dataset (the lower the better).

Methods	Dev		Test	
	del / ins	WER	del / ins	WER
Baseline	7.8 / 3.5	23.8	7.8 / 3.4	24.4
Video only	7.7 / 3.0	22.0	7.0 / 2.8	22.5
Text only	8.1 / 2.8	21.9	7.8 / 2.4	22.4
Ours	7.3 / 2.7	**21.3**	7.3 / 2.4	**21.9**

test set, respectively. It can be concluded that the generated video and text are both beneficial for the performance boost.

To qualitatively show its effectiveness, we further visualize an example as shown in Figure 5. It can be seen the hypothesis sentence of the baseline method shows deletion, substitution and insertion compared with the ground truth sentence. With only pseudo video or text inserted into our framework, the WER result gets improved to some extent, *e.g.* "Video-only" method corrects the failure insertion of word "REGEN-PLUSPLUS". When using both pseudo video and text, the WER result is further improved by a large margin on this sentence, which also shows the complementary effect of the pseudo video-text pair.

Effects of the maximum editing operations. We perform experiments on the effects of the maximum editing operations. As shown in Table 3, "K" indicates the maximum number of editing operations. The number of editing operations on the original video-text pair is randomly selected in the range from 1 to K. It can be seen that it reaches the lowest WER when the max number of operations is 3. It can be explained that the lower number of operations makes the framework more concentrate on distinguishing the fine-grained differences between the real and pseudo video-text pairs. Unless stated, we set the default maximum operations as $K = 3$ in the following experiments.

Table 3: Effects of the maximum editing operations on the RWTH-PHOENIX-Weather signer-independent dataset (the lower the better).

K	1	2	3	4	5	6
del	7.7	7.1	7.3	8.0	8.0	7.9
ins	2.9	3.1	2.7	2.6	2.6	2.7
WER	21.9	21.7	21.3	21.8	21.8	21.7

4.4 Comparison with the State-of-the-art Methods

We perform extensive experiments and compare with other state-of-the-art methods on two benchmark datasets, including RWTH-PHOENIX-Weather multi-signer, signer-independent and CSL dataset.

Evaluation on the RWTH-PHOENIX-Weather multi-signer dataset. The results on RWTH-PHOENIX-Weather multi-signer dataset are shown in Table 5. CMLLR [22] and 1-Million-Hand [23] are classical methods using hand-crafted features with traditional HMM models. CMLLR designs specific features to describe signs from different aspects of SLR, *e.g.* HOG-3D features, trajectories with position, high-level face features and temporal derivatives. With the advance of deep learning, researchers utilize CNNs to adaptively extract feature representations with significant performance gain. SubUNets [2] solves simultaneous alignment and recognition problems in an end-to-end framework by incorporating the CNN-BLSTM architecture supervised by the CTC loss. Re-sign [25]

Table 4: Evaluation on CSL dataset. (↑ indicates the higher the better, while ↓ indicates the lower the better.)

Methods	Acc-w ↑	BLEU-1 ↑	CIDEr ↑	ROUGE-L ↑	METEOR ↑	WER ↓
ELM [5]	0.175	0.376	0.028	0.120	0.388	0.987
LSTM & CTC [13, 18]	0.332	0.343	0.241	0.362	0.111	0.757
S2VT (3-layer) [39]	0.461	0.475	0.477	0.465	0.186	0.652
HLSTM-attn [16]	0.506	0.508	0.605	0.503	0.205	0.641
HRF-Fusion [15]	0.445	0.450	0.398	0.449	0.171	0.672
IAN [29]	0.670	0.724	**3.946**	0.716	0.383	0.327
Ours	**0.747**	**0.784**	3.006	**0.782**	**0.390**	**0.245**

Table 5: Evaluation on RWTH-PHOENIX-Weather multi-signer dataset (the lower the better).

Methods	Dev		Test	
	del / ins	WER	del / ins	WER
CMLLR [22]	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1-Million-Hand [23]	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid [24]	12.6 / 5.1	38.3	11.1 / 5.7	38.8
SubUNets [2]	14.6 / 4.0	40.8	14.3 / 4.0	40.7
RCNN [9]	13.7 / 7.3	39.4	12.2 / 7.5	38.7
Re-sign [25]	-	27.1	-	26.8
Hybrid CNN-HMM [26]	-	31.6	-	32.5
CNN-LSTM-HMM [21]	-	26.0	-	26.0
CTF [41]	12.8 / 5.2	37.9	11.9 / 5.6	37.8
Dilated [28]	8.3 / 4.8	38.0	7.6 / 4.8	37.3
IAN [29]	12.9 / 2.6	37.1	13.0 / 2.5	36.7
DNF (RGB) [10]	7.8 / 3.5	23.8	7.8 / 3.4	24.4
Ours	7.3 / 2.7	**21.3**	7.3 / 2.4	**21.9**

Table 6: Evaluation on RWTH-PHOENIX-Weather signer-independent dataset (the lower the better).

Methods	Dev		Test	
	del / ins	WER	del / ins	WER
Re-sign [25]	-	45.1	-	44.1
Baseline [10]	9.2 / 4.3	36.0	9.5 / 4.6	35.7
Ours	11.1 / 2.4	**34.8**	11.4 / 3.3	**34.3**

presents an iterative re-alignment approach with further embedding a HMM to correct the frame labels and continuously improves its performance. DNF [10] explores the suitable CNN-BLSTM framework and the function of multiple input modalities, becoming the most competitive method. Even compared with these challenging methods, our method still achieves a new state-of-the-art result in this dataset, *i.e.,* 21.3% and 21.9% WER on the dev and test set, respectively. It surpasses the best competitor with 2.6% and 2.5% on the dev and test set, respectively.

Evaluation on the RWTH-PHOENIX-Weather signer- independent dataset. The signer-independent subset is created based on the RWTH-PHOENIX-Weather, where we test on a single individual who has not been seen during training. In Table 6, we compare existing the methods on this dataset. It can be observed that our method still outperforms all the methods in this dataset, achieving 34.8% and 34.3% WER on the dev and test set, respectively. Compared with the best WER results on its multi-signer counterpart, the independent setting is a more challenging one, with over 10% WER reduction.

Evaluation on the CSL dataset. We perform experiments on the challenging split of the CSL dataset in Table 4. This split is difficult due to the unseen combination and occurrence of words, and different semantic context in the test set. We compare our approach with other challenging methods, such as HRF-Fusion [15], HLSTM-atten [16] and IAN [29]. HLSTM-atten treats this task as sign language translation and proposes a hierarchical-LSTM (HLSTM) encoder-decoder model with visual content and word embedding, and utilize temporal attention for performance boost. IAN proposes to use the visual encoder and encoder-decoder sequence learning network with iterative refinement. Compared with these methods, our method also achieves the state-of-the-art performance on most of the evaluation metrics on this dataset. It should be noted that our method mimics the editing process and surpasses the best competitor with 8.2% on the WER result, which is consistent with our optimization target. Besides, it also achieves the state-of-the-art performance on most semantic evaluation metrics, such as BLEU-1, ROUGE-L, METEOR, *etc.*

5 CONCLUSION

In this paper, we attempt to tackle the issue of inconformity between the CTC objective and the evaluation metric via cross modality augmentation. Following the operations in the definition of WER, *i.e.,* substitution, deletion and insertion, we edit the real video-text pair to generate its corresponding pseudo counterpart. Besides constraining the semantic correspondence between the video and text, we design a discriminative loss to make the network aware of the differences between the real and pseudo video-text pair. Our proposed framework can be easily extended to other existing CTC based continuous SLR networks. We conduct experiments on two continuous SLR benchmarks, *i.e.,* RWTH-PHOENIX-Weather and CSL dataset. Experimental results validate the effectiveness of our proposed method with notable performance gain over previous methods, especially on the WER metric.

ACKNOWLEDGMENTS

This work was supported in part to Dr. Houqiang Li by NSFC under contract No. 61836011, and in part to Dr. Wengang Zhou by NSFC under contract No. 61822208 & 61632019 and Youth Innovation Promotion Association CAS (No. 2018497).

REFERENCES

[1] Patrick Buehler, Andrew Zisserman, and Mark Everingham. 2009. Learning sign language by watching TV (using weakly aligned subtitles). In *CVPR*. 2961–2968.

[2] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. 2017. SubUNets: End-to-end hand shape and continuous sign language recognition. In *ICCV*. 3075–3084.

[3] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*. 6299–6308.

[4] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. 2019. D^3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*. 3546–3555.

[5] Xi Chen and Markus Koskela. 2014. Using appearance-based hand features for dynamic RGB-D gesture recognition. In *ICPR*. 411–416.

[6] Changmao Cheng, Chi Zhang, Yichen Wei, and Yu-Gang Jiang. 2019. Sparse temporal causal convolution for efficient action modeling. In *ACM MM*. 592–600.

[7] 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).

[8] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *CVPR*. 7784–7793.

[9] Runpeng Cui, Hu Liu, and Changshui Zhang. 2017. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *CVPR*. 7361–7369.

[10] Runpeng Cui, Hu Liu, and Changshui Zhang. 2019. A deep neural framework for continuous sign language recognition by iterative training. *TMM* 21, 7 (2019), 1880–1891.

[11] Marco Cuturi and Mathieu Blondel. 2017. Soft-DTW: A differentiable loss function for time-series. In *ICML*. 894–903.

[12] Georgios D Evangelidis, Gurkirt Singh, and Radu Horaud. 2014. Continuous gesture recognition from articulated poses. In *ECCV*. 595–607.

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*. 369–376.

[14] Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*. 1764–1772.

[15] Dan Guo, Wengang Zhou, Anyang Li, Houqiang Li, and Meng Wang. 2019. Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation. *TIP* 29 (2019), 1575–1590.

[16] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2018. Hierarchical LSTM for sign language translation. In *AAAI*.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

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

[19] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*. 84–92.

[20] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based sign language recognition without temporal segmentation. In *AAAI*.

[21] Oscar Koller, Cihan Camgoz, Hermann Ney, and Richard Bowden. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *TPAMI* (2019).

[22] Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *CVIU* 141 (2015), 108–125.

[23] 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 *CVPR*. 3793–3802.

[24] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. 2016. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *BMVC*.

[25] Oscar Koller, Sepehr Zargaran, and Hermann Ney. 2017. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *CVPR*. 4297–4305.

[26] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. 2018. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *IJCV* 126, 12 (2018), 1311–1325.

[27] Tomas Pfister, James Charles, and Andrew Zisserman. 2013. Large-scale learning of sign language by watching TV (Using Co-occurrences).. In *BMVC*.

[28] Junfu Pu, Wengang Zhou, and Houqiang Li. 2018. Dilated convolutional network with iterative optimization for continuous sign language recognition. In *IJCAI*.

[29] Junfu Pu, Wengang Zhou, and Houqiang Li. 2019. Iterative alignment network for continuous sign language recognition. In *CVPR*. 4165–4174.

[30] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3D residual networks. In *ICCV*. 5533–5541.

[31] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. 2019. Learning spatio-temporal representation with local and global diffusion. In *CVPR*. 12056–12065.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *IJCV* (2015), 211–252.

[33] Xiangxi Shi, Jianfei Cai, Shafiq Joty, and Jiuxiang Gu. 2019. Watch it twice: Video captioning with a refocused video encoder. In *ACM MM*. 818–826.

[34] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*. 568–576.

[35] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time American sign language recognition using desk and wearable computer based video. *TPAMI* 20, 12 (1998), 1371–1375.

[36] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*. 3104–3112.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*. 1–9.

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatio-temporal features with 3D convolutional networks. In *ICCV*. 4489–4497.

[39] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *ICCV*.

[40] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*. 20–36.

[41] Shuo Wang, Dan Guo, Wen-gang Zhou, Zheng-Jun Zha, and Meng Wang. 2018. Connectionist temporal fusion for sign language translation. In *ACM MM*. 1483–1491.

[42] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. 2016. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *TPAMI* 38, 8 (2016), 1583–1597.

[43] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatio-temporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*. 305–321.

[44] Shijie Yang, Liang Li, Shuhui Wang, Dechao Meng, Qingming Huang, and Qi Tian. 2019. Structured stochastic recurrent network for linguistic video prediction. In *ACM MM*. 21–29.

[45] Jihai Zhang, Wengang Zhou, Chao Xie, Junfu Pu, and Houqiang Li. 2016. Chinese sign language recognition with adaptive HMM. In *ICME*. 1–6.

[46] Hao Zhou, Wengang Zhou, and Houqiang Li. 2019. Dynamic pseudo label decoding for continuous sign language recognition. In *ICME*. 1282–1287.

[47] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2020. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*.

[48] Yongqing Zhu and Shuqiang Jiang. 2019. Attention-based densely connected LSTM for video captioning. In *ACM MM*. 802–810.

[49] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. 2018. ECO: Efficient convolutional network for online video understanding. In *ECCV*. 695–712.