Iterative Alignment Network for Continuous Sign Language Recognition

Junfu Pu Wengang Zhou Houqiang Li
CAS Key Laboratory of GIPAS, University of Science and Technology of China
pjh@mail.ustc.edu.cn, zhwg@ustc.edu.cn, lihq@ustc.edu.cn

Abstract

In this paper, we propose an alignment network with iterative optimization for weakly supervised continuous sign language recognition. Our framework consists of two modules: a 3D convolutional residual network (3D-ResNet) for feature learning and an encoder-decoder network with connectionist temporal classification (CTC) for sequence modelling. The above two modules are optimized in an alternate way. In the encoder-decoder sequence learning network, two decoders are included, i.e., LSTM decoder and CTC decoder. Both decoders are jointly trained by maximum likelihood criterion with a soft Dynamic Time Warping (soft-DTW) alignment constraint. The warping path, which indicates the possible alignment between input video clips and sign words, is used to fine-tune the 3D-ResNet as training labels with classification loss. After fine-tuning, the improved features are extracted for optimization of encoder-decoder sequence learning network in next iteration. The proposed algorithm is evaluated on two large scale continuous sign language recognition benchmarks, i.e., RWTH-PHOENIX-Weather and CSL. Experimental results demonstrate the effectiveness of our proposed method.

1. Introduction

As one of the most important ways to communicate with the deaf-mute, sign language (SL) is used by millions of people with hearing or spoken damage in their daily life. However, due to the lack of systematic study for sign language, it becomes very difficult for many people to communicate with the deaf-mute. In order to make such communication more convenient, it's necessary to develop an effective algorithm for sign language recognition (SLR). Recently, more and more researchers turn their attention to sign language recognition, not only for the social impacts, but also with academic explorations.

Comparing to isolated SLR [16, 22, 42, 43], *i.e.*, recognition of words or gestures [31], which is similar to action recognition [19, 24], continuous SLR [9, 26, 29] is much more complicated since there is no rigid annotation of text word to video clip for a complete sign video. As a kind of weakly supervised sequence learning task, the key idea for continuous SLR is to learn the mapping between a sign video and its corresponding annotation of text sentence. Continuous SLR task is well defined with a very standard formulation, since the sign translation result is strictly constrained in grammar.

So far, the existing methods on continuous SLR can be grouped into two categories based on the involved feature representation, *i.e.*, hand-crafted feature based and deep learning based methods. Early works [35] mainly use hand-crafted features together with statistical sequence modelling methods such as Hidden Markov Model (HMM) or Hidden Conditional Random Fields (HCRF). Starner *et al.* [35] present two real-time HMM-based systems for recognizing sentence-level continuous American Sign Language (ASL). Later on, Wang *et al.* [40] derive a discriminative sequence model with Hidden Conditional Random Field (HCRF) for gesture recognition, in order to solve the issue of long-range dependencies among observations in HMM.

Recently, benefitting from the development of deep learning, in sign language recognition, there have witnessed some breakthroughs. With the appearance of large scale continuous sign language datasets [9, 23, 25], deep learning based continuous SLR methods gradually become the mainstream. With the powerful video representations by residual network (ResNet) [18] and 3D convolutional neural network (3D-CNN) [33, 37], deep learning methods for continuous SLR achieve state-of-the-art performance. Cui *et al.* [10] propose to use recurrent convolutional neural networks with staged optimization to recognize continuous sign language. Another work [23] with hierarchical attention in latent space also shows the superiority of deep learning to hand-crafted feature based methods.

In this paper, we propose a new deep learning architecture for continuous SLR. Our framework includes a 3D residual network (3D-ResNet) for feature extraction and an encoder-decoder network for sequence modelling. Considering the particularity of continuous SLR crossing computer vision and natural language processing, we explore the technics in video representation and understanding, as well as the sequence modelling with grammar. We unify the visual representation learning and sequence modeling in

our framework and make joint optimization over these two modules. The main contributions of this paper are summarized as follows:

a) A unified deep learning architecture integrating encoder-decoder network and connectionist temporal classification (CTC) for continuous sign language recognition.
b) A soft dynamic time warping (soft-DTW) alignment constraint between the LSTM and CTC decoders, which indicates the temporal segmentation in sign videos.
c) Iterative optimization strategy to train feature extractor and encoder-decoder network alternately with alignment proposals by warping path.

We organize the rest of this paper as follows: after reviewing the related works in Section 2, we elaborate our proposed architecture and iterative optimization algorithm in Section 3 and Section 4, respectively. In Section 5, we conduct a series of experiments with discussions and analysis. At last, we conclude our work in Section 6.

2. Related Works

Video-based continuous SLR systems basically consist of a feature extractor and a sequence modelling module, where the latter is usually achieved via encoder-decoder network or connectionist temporal classification. In this section, we briefly review the works related continuous SLR from the following two aspects.

2.1. Video Representation

Video representation plays a significant role for many computer vision tasks, *e.g.*, action recognition [24, 33, 37] and video captioning [5]. Since Ji *et al.* [24] apply 3D convolutional neural network (3D-CNN) to action recognition task [4, 17], 3D-CNN has become one of the most famous architectures for video representation. Variants of different improved 3D-CNN architectures appear for different vision task. Meanwhile, deep residual network (ResNet) [18] has shown powerful capacity for image representation. Inspired by the recent successes of ResNet in numerous challenging image recognition tasks, Qiu *et al.* develop a new family of building modules named Pseudo-3D (P3D) blocks [33] to replace 2D residual units in ResNet. The potential capacity of combining the residual networks and 3D convolutional networks for video representation is demonstrated in [17].

2.2. Sequence Modelling

The end-to-end sequence learning methods are typically grouped in two types: attention-based encoder-decoder [7, 8, 38] network and connectionist temporal classification (CTC)-based network [12, 21]. Encoder-decoder network is first proposed for machine translation in [7]. The encoder-decoder architecture consists of two recurrent neural networks (RNN) that act as a pair of encoder and decoder pair. The encoder maps a variable-length source sequence to a fixed-length vector, while the decoder maps the vector representation back to a variable-length target sequence. Although the encoder-decoder network has been widely used for speech recognition [8] and video captioning [2], there still remains some limitation when modelling long-term dependency. To overcome this issue, Bahdanau *et al.* [1] introduce attention mechanism into encoder-decoder network to learn the correspondence between source sequence and target sequence. Following that, more and more different attention methods [2, 30, 41, 36] are proposed to improve the encoder-decoder networks for specific tasks.

Connectionist temporal classification (CTC) [12] is another end-to-end sequence learning model for speech and hand writing recognition [13, 21]. CTC is able to deal with unsegmented input data, and learn the correspondence between the input sequence and output sequence. It is appropriate for continuous SLR, since continuous SLR is somehow a kind of weakly supervised sequence learning problem. With the superiority of CTC, Cui *et al.* [10] achieve the state-of-the-art performance for continuous SLR.

3. Alignment Network Architecture

In this section, we present a novel deep learning framework for continuous SLR. Our method integrates the encoder-decoder network and connectionist temporal classification into a unified deep architecture. To explore the correspondence between the input sequence and target translation, we use soft dynamic time warping (soft-DTW) to align the CTC-decoder and LSTM-decoder.

3.1. Framework and Formulation

Continuous SLR deals with a sequence mapping from a video with T frames $\mathbf{V} = \{x_t \in \mathbb{R}^{h\times w\times c}\} = \{x_t\}_{t=1}^{T}$ to a L-word sequence $\boldsymbol{s} = \{s_i \in \mathcal{V} | i = 1, \cdots, L\}$, where $h \times w$ is the size of image x_t, c is 3 for an RGB video. The mathematic formulation of continuous SLR is based on Bayes decision theory, and the translated sentence $\widehat{\boldsymbol{s}}$ is estimated with the most probable word sequence among all possible sequences $\boldsymbol{s}^*$ as follows,

$$\widehat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}\in\boldsymbol{s}^*} p(\boldsymbol{s}|\boldsymbol{V}). \tag{1}$$

Figure 1 illustrates the framework of our continuous SLR system. The input to the framework is sign video with paired sentence-level annotation. Our continuous SLR system consists of the following four tiers of neural network.

1) **Feature Extractor** With the input of video clip sequence, 3D-ResNet coverts it into a fixed-length feature, which summarizes the spatial and temporal information.
2) **Sequence Encoder** The sequential video descriptors extracted by 3D-ResNet are modelled by a 2-layer Bidirectional Long Short-Term Memory (Bi-LSTM) encoder.

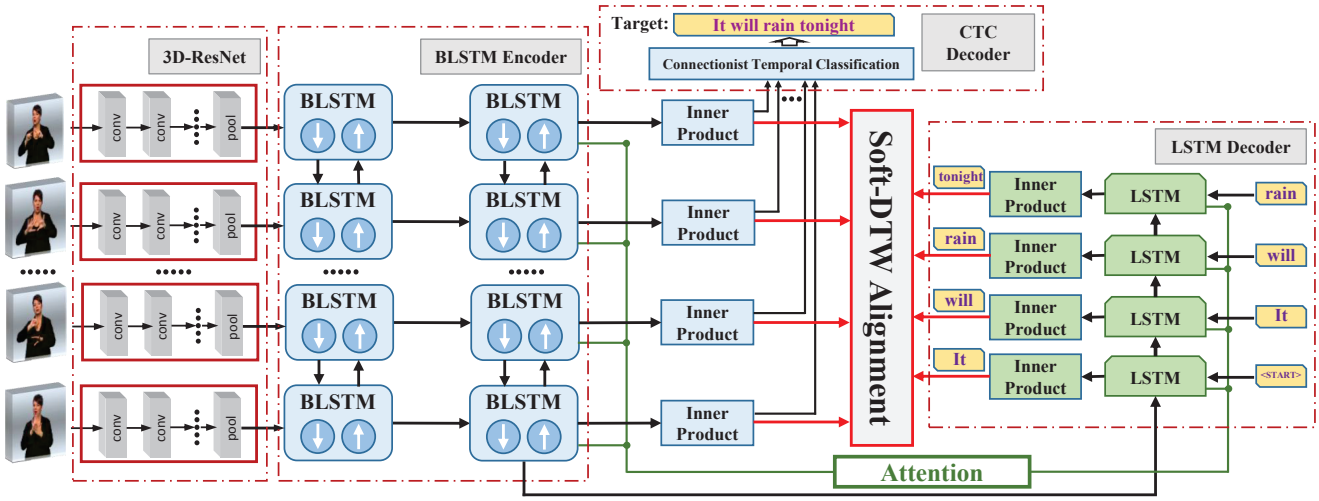

Figure 1: Overview of our SLR framework. The system consists of a 3D-ResNet and an encoder-decoder network with connectionist temporal classification. The CTC decoder and LSTM decoder are aligned with soft dynamic time warping constraint. The inner-product layer projects the BLSTM and LSTM outputs into categorical probabilities for word recognition.

3) **Target Decoders** To predict the target sequence, two decoders are embedded into the network, which are connectionist temporal classification (CTC) decoder and LSTM decoder, respectively.

4) **Alignment Constraint** The soft-DTW constraint is used to align the CTC-decoder and LSTM-decoder, which both describe the probability distribution of the target sequence.

The following parts of this section will elaborate each module of our framework in detail.

3.2. Video Representation: 3D-ResNet

3D-CNN has been widely applied for video representation in action recognition [24, 37] and sign language recognition [23, 32], and achieves state-of-the-art performance. Considering the successes of residual network in different computer vision tasks, we use 3D residual network (3D-ResNet) to represent video clips, which inherits the superiority of both two models.

Given a sign video $\mathbf{V} = (x_1, \cdots, x_T) = \{x_t\}_{t=1}^{T}$ with T frames, where x_i is the i^{th} frame in the video, a sliding window is moved along the image sequence to generate a set of ordered video clips. In this way, the sign video is denoted as $\mathbf{V} = (v_1, \cdots, v_N) = \{v_t\}_{t=1}^{N}$ with N clips. We use $\boldsymbol{\mathcal{F}_\theta}$ to represent 3D-ResNet feature extractor, where $\boldsymbol{\theta}$ is the network weight. For each video clip v_t, we get the representation $f_t = \boldsymbol{\mathcal{F}_\theta}(v_t) \in \mathbb{R}^d$ by passing it through the 3D-ResNet, where d is the dimension of the video feature. Thus, the sign video is represented as a sequence of 3D-CNN features as follows,

$$\mathbf{F}^N = (f_1, \cdots, f_N) = \{\boldsymbol{\mathcal{F}}_\Theta(v_t)\}_{t=1}^{N}. \tag{2}$$

Considering the GPU memory and computational cost for low latency, we use an 18-layer 3D-ResNet, which is light and powerful enough for sign video representation.

3.3. Temporal Encoder: Bidirectional LSTMs

Recurrent neural network has made huge success for various of sequence processing tasks, *e.g.*, speech recognition [14, 21], neural machine translation [6], and video captioning [2]. One of the most popular RNN architectures is Long Short-Term Memory (LSTM) [20], which preserves the long term dependencies to avoid vanishing gradients compared to traditional RNN. LSTM units use purpose-built memory cells to store and pass information, which is better to explore the long term dependencies. The current status of the LSTM unit is described with cell state C_t and hidden state h_t. The most fancy idea of LSTM is the use of gate structures that optionally let information through.

One shortcoming of LSTM is that it only models the correlations between the current input and the previous time steps. The inputs after current time step t make no contribution when generating the LSTM output. In continuous SLR, the sign video represents a semantic sentence with grammatical rules, which means both forward and backward frames should be taken into consideration. To this end, we use a bidirectional LSTM (BLSTM) to encoder the input sign video. The basic idea of BLSTM is to present the training sequence forwards and backwards to two separate LSTMs, and concatenate the two outputs before feeding to the deeper layer. This means that for current time step, the output of BLSTM has the complete sequential information over all time steps before and after it. We use $\mathcal{R}$ to represent BLSTM, then the output of encoder is denoted as follows,

$$\mathbf{E} = \{e_t\}_{t=1}^{N} = \mathcal{R}(\{f_t\}_{t=1}^{N}). \tag{3}$$

The outputs are embedded into non-normalized categorical probabilities of word-level labels in the size of vocabulary by a fully-connected layer as follows,

$$y_t = W_{fc1} \cdot e_t + b_{fc1}. \tag{4}$$

For a sign video with N clips, the probability distribution characterized by BLSTM can be written as follows,

$$\boldsymbol{Y} = (Y_{t,l}) = [y_1, y_2, \cdots, y_N]^T, \tag{5}$$

where $Y_{t,l}$ is the probability of t^{th} clip belonging to word l.

3.4. Target Decoders: LSTM and CTC

To decode the target sentence from the sign video, we use two kinds of decoders, *i.e.*, LSTM decoder with attention mechanism and CTC decoder.

3.4.1 Attention-aware LSTM Decoder

Following the BLSTM encoder, the LSTM decoder generates corresponding sentence from the encoder output. After all input clips going through the BLSTM, the LSTM decoder is fed with the beginning-of-sentence (<BOS>) tag, which prompts the network to start decoding the current hidden states into a sequence of words. In training stage, the model maximizes the log-likelihood of the target sentence given the hidden states and the previous words. While in inference, we choose the word with maximum probability until it emits the ending-of-sentence (<EOS>) token.

We apply attention mechanism [1] for LSTM decoder. The decoder output for the k^{th} word is written as follows,

$$d_k = \mathbf{Decoder}_{lstm}(c_k, s_k, h^d_{k-1}), \tag{6}$$

where c_k is context vector, s_k and h^d_{k-1} are embedded word and hidden state of the decoder, respectively. The LSTM is connected to an inner-product layer to project the LSTM output into categorical probability with M classes, where $M = |\mathcal{V}|$ is the vocabulary size. The final activation of the inner-product layer is defined as follows,

$$z_k = W_{fc2} \cdot d_k + b_{fc2}. \tag{7}$$

Similar to Section 3.3, the probability distribution of the translated sentences is formulated as follows,

$$\boldsymbol{Z} = (Z_{k,l}) = [z_1, z_2, \cdots, z_L]^T, \tag{8}$$

where L is the length of sentence, and $Z_{k,l}$ is the probability of s_k belonging to word label l given s_{k-1}.

3.4.2 CTC Decoder

Connectionist temporal classification (CTC) [12] is a popular sequence learning algorithm, which models the mapping between input sequence and target sequence. The output of inner-product layer following with BLSTM encoder is corresponded to the probability distribution of word labels. CTC approach decodes the target sentence from the probability matrix $\boldsymbol{Y}$ explained in Section 3.3 by introducing a blank label ($*$) as an assistant token. Define a path $\pi = (\pi_1, \cdots, \pi_T), \pi_t \in \mathcal{V} \bigcup \{*\}$ on input sequence, where $\mathcal{V}$ is the sign vocabulary. The probability for path π given sign video $\boldsymbol{V} = \{v_t\}_{t=1}^N$ is calculated as follows,

$$p(\pi|\boldsymbol{V}) = \prod_{t=1}^{N} p(\pi_t|v_t) = \prod_{t=1}^{N} Y_{t,\pi_t}. \tag{9}$$

To get the final decoded sequence without blanks, CTC defines a many-to-one mapping $\mathcal{M}$, which removes the repeated labels and blanks, *e.g.*, $\mathcal{M}(r * aa * i * n) = \mathcal{M}(r * a * i * n) = rain$. The probability of the sentence $\boldsymbol{s} = (s_1, \cdots, s_L)$ decoded by CTC is the summation of the probabilities for all possible paths as follows,

$$p_{ctc}(\boldsymbol{s}|\boldsymbol{V}) = \sum_{\pi \in \mathcal{M}^{-1}(\boldsymbol{s})} p(\pi|\boldsymbol{V}), \tag{10}$$

where $\mathcal{M}^{-1}$ is the inverse mapping of $\mathcal{M}$, *i.e.*, $\mathcal{M}^{-1}(\boldsymbol{s}) = \{\pi|\mathcal{M}(\pi) = \boldsymbol{s}\}$.

3.5. Sequence Alignment: Soft DTW

We apply two kinds of decoders to our network introduced in Section 3.4. Essentially, there are somehow potential correlations between these two probability distributions $\boldsymbol{Y}$ and $\boldsymbol{Z}$ for CTC decoder and LSTM decoder, since they both describe the same target sentence. Hence, we aim to maximize the similarity between $\boldsymbol{Y}$ and $\boldsymbol{Z}$. However, the length of sentences generated from different decoders may not equal each other. To evaluate the similarity between various length sequences, we use soft dynamic time warping (soft-DTW) [11] to get the distance between $\boldsymbol{Y}$ and $\boldsymbol{Z}$, as well as the warping path.

Soft-DTW is a differentiable learning distance between time series, building upon the original dynamic time warping (DTW) [34] discrepancy. The DTW algorithm is used to find the minimal accumulating distance of two sequences and the temporal warping path. Given two sequences $\boldsymbol{u} = (u_1, \cdots, u_m)$ and $\boldsymbol{v} = (v_1, \cdots, v_n)$, the DTW distance for subsequence $\boldsymbol{u}^i = (u_1, \cdots, u_i)$ and $\boldsymbol{v}^j = (v_1, \cdots, v_j)$ is denoted as $D_{i,j}$ and defined as follows,

$$D_{i,j} = d_{i,j} + \min(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}), \tag{11}$$

where

$$d_{i,j} = ||u_i - v_j||_2. \tag{12}$$

In order to make the DTW discrepancy differentiable, soft-DTW algorithm is taken by introducing the generalized min operator, with a smoothing parameter $\gamma \geq 0$ [11]:

$$\min_i{}^{\gamma} \{a_i\} = \begin{cases} \min_i\{a_i\}, & \gamma = 0. \\ -\gamma \log \sum_i e^{-a_i/\gamma}, & \gamma > 0. \end{cases} \tag{13}$$

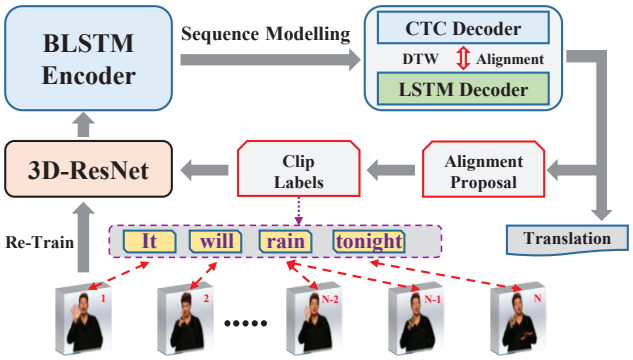

Figure 2: Illustration of our iterative training algorithm. After encoding the sequential features extracted by 3D-ResNet, the CTC decoder and LSTM decoder decode them into sign glosses. The decoders also generate the alignment proposal with the warping path by soft-DTW to fine-tune the 3D-ResNet in next iteration.

With the basic formulation of soft-DTW, the distance between the probability distributions $\boldsymbol{Y}$ and $\boldsymbol{Z}$ is defined as

$$\mathcal{D}_p = D_{N,L}(\boldsymbol{Y}, \boldsymbol{Z}), \tag{14}$$

where N and L are the sequence length for $\boldsymbol{Y}$ and $\boldsymbol{Z}$, respectively.

We can recover the warping path by backtracking. The warping path indicates the possible alignment between sign clips and words, which is a fine-grained understanding for sign videos. Denote the warping path as $\Pi = \{(p, q)|p \leq N, q \leq L\}$, the label ℓ_p of the p^{th} clip is obtained by

$$\ell_p = s_q. \tag{15}$$

4. Optimization and Decoding

In this section, we will introduce the objective function and iterative training algorithm to optimize the network. Besides, a joint decoding approach combining the CTC decoder and LSTM decoder is proposed for better recognition.

4.1. Objective Function

In Section 3.4, we describe two kinds of decoders. Both LSTM decoder and CTC decoder are trained with maximum-likelihood criterion. Given a sign video $\boldsymbol{V}$ and its corresponding annotation $\boldsymbol{s} = (s_1, \cdots, s_L)$, the loss function for CTC decoder is formulated as

$$\mathcal{L}_{ctc} = -\ln p_{ctc}(\boldsymbol{s}|\boldsymbol{V}), \tag{16}$$

where $p_{ctc}(\boldsymbol{s}|\boldsymbol{V})$ is the posterior probability of $\boldsymbol{s}$ given $\boldsymbol{V}$ which is defined in Equation 10.

For LSTM decoder, the probability of $\boldsymbol{s}$ given $\boldsymbol{V}$ is

$$p_{lstm}(\boldsymbol{s}|\boldsymbol{V}) = \prod_{i=1}^{L} p(s_i|s_{i-1}) = \prod_{i=1}^{L} \boldsymbol{Z}_{i,s_i}. \tag{17}$$

Similar to $\mathcal{L}_{ctc}$, the LSTM loss function is defined as

$$\mathcal{L}_{lstm} = -\ln p_{lstm}(\boldsymbol{s}|\boldsymbol{V}). \tag{18}$$

Besides, there is an alignment term for CTC decoder and LSTM decoder, which is constrained by soft-DTW distance. In order to make the two probability distributions get closer to each other, we define an alignment loss as

$$\mathcal{L}_{align} = \mathcal{D}_p(\boldsymbol{Y}, \boldsymbol{Z}), \tag{19}$$

where $\mathcal{D}_p$ is described in Equation 14.

We jointly train the network and the objective function for optimization is presented as

$$\mathcal{L} = \lambda\mathcal{L}_{ctc} + (1-\lambda)\mathcal{L}_{lstm} + \mathcal{L}_{align} + \mu||\omega||^2, \tag{20}$$

where λ is a tunable hyper-parameter which balances the potential significance of the two decoders, and $\mu||\omega||^2$ is a regularization term to avoid overfitting.

4.2. Optimization Strategy

While recognizing continuous sign videos, 3D-ResNet plays a significant role as feature representation learning module. Representative features contribute to good performance. When training the network in an end-to-end way, the objective loss has limited contribution to the learning of parameters for low layers of 3D-ResNet due to the chain rules of back-propagation. To alleviate this issue, an alternative is to learn explicit 3D-CNN features by optimizing feature extractor directly with clip level labels. However, in our continuous SLR task, such labels are unavailable.

To address the above problem, we propose to use soft-DTW alignment proposals as pseudo-labels to learn representative 3D-CNN features, and optimize feature extractor and sequence learning module in an EM-like iterations, as shown in Figure 2. In our method, we firstly use 3D-ResNet to extract features from a sign video. After that, we train the encoder-decoder network by minimizing the total loss $\mathcal{L}$. After convergence, the network provides the warping path between the input clips and words by soft-DTW. For a better feature representation of 3D-ResNet, we use the alignment proposal described in Equation 15 as the supervision for video clips to fine-tune the feature extractor (3D-ResNet) with cross entropy classification loss. With the optimized 3D-ResNet, we extract features with stronger representative capacity to train the encoder-decoder network in next iteration. These two parts of the network are alternately optimized until both of them converge to optimum.

4.3. Decoding

This section introduces the decoding method, which potentially utilizes both benefits of CTC decoder and attention-aware LSTM decoder. Our network allows CTC

decoder and LSTM decoders to decode the sign video independently. To combine the superiority of both decoders, we use a two-pass re-ranking approach to fuse the results. In inference stage, CTC decoder obtains a set of complete hypotheses sentence as candidates using beam search. We re-rank the candidates using both CTC and LSTM decoders. Suppose we have K candidates $C = \{\boldsymbol{s}^i | i = 1, \cdots, K\}$, the score for hypotheses sentence $\boldsymbol{s}^i$ is represented as

$$r(\boldsymbol{s}^i) = \alpha \ln p_{ctc}(\boldsymbol{s}^i|\boldsymbol{V}) + (1-\alpha) \ln p_{lstm}(\boldsymbol{s}^i|\boldsymbol{V}) + \beta \ln L_i, \tag{21}$$

where α is a tunable parameter, L_i is the length of $\boldsymbol{s}^i$, and $\beta \ln L_i$ is an additional length term to balance the score for long sequence. Given K-best hypotheses produced by CTC decoder via beam search, we determine the final result $\widehat{\boldsymbol{s}}$ by

$$\widehat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} r(\boldsymbol{s}). \tag{22}$$

5. Experiments

We provide extensive experiments to evaluate the effectiveness of our method. The datasets and evaluation metric are introduced firstly. After that, we give the details about experimental setup and analyze the experimental results.

5.1. Dataset and Evaluation

We conduct our experiments on two public datasets, which are RWTH-PHOENIX-Weather multi-signer [25] for German SLR and CSL [23] for Chinese SLR, respectively. RWTH-PHOENIX-Weather dataset contains around 7K sign videos within a total of 77K words. RGB videos and their corresponding annotations are provided. The annotations are about weather forecast in German Sign Language. All videos are of 25 frames per second (FPS) with the resolution of 210×260. The dataset is divided into three parts: $5,672$ instances for training, 540 for validation, and 629 for testing. The CSL dataset has 178 Chinese words which are mostly used in daily communication. The corpus contains 100 sentences. Each sentence is performed by 50 signers. Therefore, there are 5,000 videos in this dataset. In average, 5 words (phases) are included in each sentence.

In continuous SLR, word error rate (WER) is the most widely-used metric to evaluate the performance. WER is essentially an edit distance. In other words, WER indicates the least operations of substitution, insertion, and deletion to transform the predict sentence into the reference sequence:

$$\text{WER} = \frac{\#\text{substitution} + \#\text{insertion} + \#\text{deletion}}{\text{length of reference}}. \tag{23}$$

Besides, following this work [15], we use some other evaluation metrics on CSL dataset, including ***precision*** and ***Acc-w***, which are the ratio of strictly correct sentences and the ratio of correct words in reference sentence, respectively. We also adopt semantic evaluation metrics which widely used in image caption and neural machine translation (NMT), *i.e.*, CIDEr, BLEU, ROUGE-L, and METEOR.

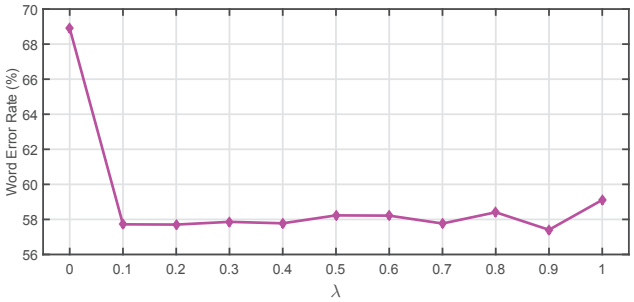

Figure 3: The effect of weight parameter λ in Equation 20 at iteration-0.

5.2. Experimental Setting

Our model consists of two modules: 3D-ResNet for feature learning and encoder-decoder network with soft-DTW alignment for sequence learning. We use iterative optimization strategy described in 4.2 to train these two parts alternately. In this section, the experiments for parameter selection are conducted on RWTH-PHOENIX-Weather dataset.

The input of 3D-ResNet is required to be fixed-length video clips. Hence, we conduct a sliding window on raw videos to generate clips. The window size is set to be 8 with a stride of 4, which means there is 50% overlap between adjacent clips. The activations of 512-dimensional $pool5$ layer from 3D-ResNet are extracted as the representation of video clips. While training the feature extractor, we use stochastic gradient descent (SGD) optimizer to train our network. The initial learning rate and weight decay are set to be 1×10^{-3} and 5×10^{-5}, respectively. At the initial step, to extract features for encoder-decoder network, the 3D-ResNet is pre-trained on an isolated sign language recognition dataset released in [43]. The hidden states of the 2-layer BLSTM encoder is set to be 1024.

In order to set an optimal weight λ in Equation 20, we conduct experiments with different λ using the features extracted in initial step, as shown in Figure 3. For $0 < \lambda < 1$, we use jointly re-ranking decoding algorithm introduced in Section 4.3. The hyper-parameters α and β in Equation 21 are set to 0.85 and 0.7, respectively. Note that when $\lambda = 0$ or 1, it means we only use one of the decoders for training and inference without soft-DTW alignment. From the results, we find that $\lambda = 0.9$ is the best option. Hence, all following experiments use the setup of $\lambda = 0.9$.

5.3. Results on RWTH-PHOENIX-Weather

In this section, we show the performance comparisons on RWTH-PHOENIX-Weather. We analyze the performance for different optimization iterations and give an example illustrating the alignment between video clips and annotation.

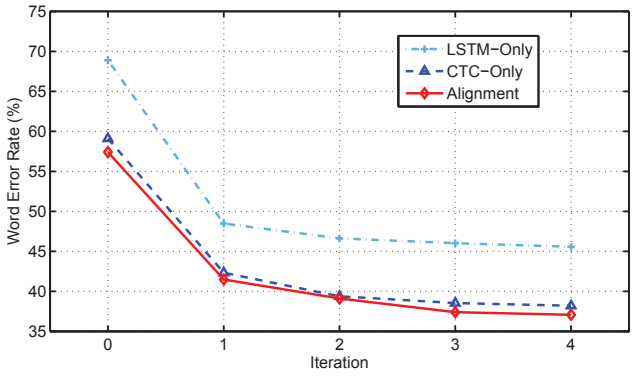

Figure 4: Performance comparison for alignment mechanism.

Iterations	Dev (%)			Test (%)		
	del /	ins	WER	del /	ins	WER
Iter-0	19.46 /	2.74	57.72	20.26 /	2.49	57.90
Iter-1	15.01 /	2.69	41.48	14.12 /	2.22	40.38
Iter-2	13.16 /	2.83	39.11	13.40 /	2.74	39.17
Iter-3	12.68 /	2.93	37.39	12.94 /	2.58	37.56
Iter-4	12.86 /	2.64	37.07	12.97 /	2.47	36.71

Table 1: Word error rate (WER) for different iterations on RWTH-PHOENIX-Weather-2014 (the lower the better).

5.3.1 Iterative Optimization Results

Our network is optimized by iterative training. Table 1 shows the performance on dev set and test set in different iterations. In this table, "del" and "ins" stand for deletion error and insertion error, respectively. It can be observed that the word error rate (WER) declines with the training iterations progress, which demonstrates the effectiveness of the iterative optimization strategy. After 4 iterations, we stop iterative training progress, since the WER does not decline anymore and the network converges to optimum.

Besides, Figure 4 gives the comparison for whether there is alignment mechanism in the network. As the Figure shows, *CTC-Only* and *LSTM-Only* correspond to training with only CTC loss $\mathcal{L}_{ctc}$ or LSTM cross entropy loss $\mathcal{L}_{lstm}$, respectively. *Align* means the network is trained with alignment constraint and jointly decodes the sentence with both decoders. From Figure 4, we notice that the network with alignment constraint outperforms another two networks with different objective functions at every iteration. The experimental results show that alignment mechanism works well in our proposed network.

5.3.2 Alignment and Comparisons

In this section, we give an example qualitatively describing the alignment between the input video and its corresponding annotation. Additionally, we discuss the performance of our method together with the state-of-the-arts on RWTH-PHOENIX-Weather multi-signer dataset.

Methods	Dev (%)		Test (%)	
	del / ins	WER	del / ins	WER
1-Mio-Hands [25, 27]	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid [28]	12.6 / 5.1	38.3	11.1 / 5.7	38.8
SubUNet [3]	14.6 / 4.0	40.8	14.3 / 4.0	40.7
Staged-Opt [10]	13.7 / 7.3	39.4	12.2 / 7.5	38.7
CTF [39]	12.8 / 5.2	37.9	11.9 / 5.6	37.8
Dilated-SLR [32]	8,3 / 4.8	38.0	7.6 / 4.8	37.3
LS-HAN [23]	-	-	-	38.3
Ours (LSTM)	13.8 / 3.3	45.6	13.6 / 3.3	46.1
Ours (CTC)	11.4 / 3.8	38.2	11.9 / 3.5	37.9
Ours (Align-end2end)	12.6 / 2.2	69.1	22.0 / 2.6	69.3
Ours (Align-iOpt)	12.9 / 2.6	**37.1**	13.0 / 2.5	**36.7**

Table 2: Word error rate (WER) on RWTH-PHOENIX-Weather-2014 (the lower the better).

Figure 5 shows an example[1] of alignment results from Dev set. All clips are from the same sign video by order. Each clip is aligned to its corresponding word. The period of appearance for different sign word may be different in the sign video. Our network has the capacity of exploring the sequential alignment.

We evaluate the performance of our approach on the large-scale continuous SLR benchmark RWTH-PHOENIX-Weather, and the comparison results [2] to different methods are shown in Table 2. 1-Mio-Hands [25, 27] achieves an WERs of 47.1% and 45.1% on dev set and test set, respectively, by embedding a CNN within an iterative EM algorithm. CNN-Hybrid [28] introduces an end-to-end embedding of a CNN into a HMM, while interpreting the outputs of CNN in a truly Bayesian fashion. The basic architectures in SubUNet [3] and Staged-Opt [10] are both CNN+BLSTM+CTC. The main difference is that Staged-Opt proposes a staged optimization algorithm with detection net, and it achieves a better performance than SubUNet. Another two works CTF [39] and Dilated-SLR [32] are both CTC-based approach. In addition, LS-HAN [23] is an encoder-decoder framework with hierarchical attention mechanism for better recognition.

Comparing to the results which use only one of the decoders, *i.e.*, *LSTM* or *CTC*, for training and inference, the network using soft-DTW alignment for both decoders with iterative optimization strategy achieves the best performances. We also train our network in an end-to-end way, denoted as *Align-end2end*. However, the results are not good enough. These comparative experiments illustrate both the alignment mechanism and iterative optimization work well in our approach.

[1] Video ID: 03February_2010_Wednesday_tagesschau_default-0.

[2] Since WER is the summation of insertion error, deletion error, and substitution error, we only list 3 of them without substitution error.

Figure 5: An example for alignment results between the video clips and sentence annotation in German from Dev set.

Method	Split I					Split II					
	Precision	BLEU-1	CIDEr	ROUGE-L	METEOR	*Acc-w*	BLEU-1	CIDEr	ROUGE-L	METEOR	WER
LSTM&CTC [12, 20]	0.858	0.936	8.632	0.940	0.646	0.332	0.343	0.241	0.362	0.111	0.757
S2VT [38]	0.897	0.902	8.512	0.904	0.642	0.457	0.466	0.479	0.461	0.189	0.670
S2VT (3-layer) [38]	0.903	0.911	8.592	0.911	0.648	0.461	0.475	0.477	0.465	0.186	0.652
HLSTM (SYS sampling) [15]	0.910	0.935	8.907	0.938	0.683	0.459	0.463	0.476	0.462	0.173	0.630
HLSTM [15]	0.924	0.942	9.019	0.944	0.699	0.482	0.487	0.561	0.481	0.193	0.662
HLSTM-attn [15]	0.929	0.948	9.084	0.951	0.703	0.506	0.508	0.605	0.503	0.205	0.641
Ours	**0.939**	**0.980**	**9.342**	**0.981**	**0.713**	**0.670**	**0.724**	**3.946**	**0.716**	**0.383**	**0.327**

Table 3: Evaluation on CSL Dataset Split I for seen sentence recognition and Split II for unseen sentence recognition (the lower the better for WER, the higher the better for other metrics).

5.4. Results on CSL

The CSL dataset contains a smaller vocabulary comparing with RWTH-PHOENIX-Weather. We use the same hyper-parameters on both datasets. Following this work [15], the training set and testing set are generated with two different strategies. (a) **Split I - signer independent test**: We use the videos performed by 40 signers for training, and the remaining videos of 10 signers for testing. The sentences of training and testing sets are the same, while the signers are different. (b) **Split II - unseen sentence test**: We choose 94 sentences ($94 \times 50 = 3700$ videos) for training, and the remaining 6 sentences ($6 \times 50 = 300$ videos) for testing. The sentences in testing set are different from which in training set, while the vocabulary in testing set is a subset of vocabulary in training set.

We pre-train 3D-ResNet on the isolated SLR dataset [43]. Since the vocabularies in CSL dataset are all from isolated SLR dataset, we get good enough performances without iterations. The performances of our method comparing with existing methods over the CSL dataset are summarized in Table 3. We compare our method with LSTM&CTC, S2VT [38], and HLSTM [15] over both splits. Experimental results show that our method outperforms the state-of-the-art methods over **Split I** with signer-independence test. In continuous SLR, it's quite difficult to recognize the sentences which are not appeared in training set. To evaluate the capability of our method for such case, we conduct experiments on CSL **Split II**, and the performances comparing with other methods are shown in Table 3 (Split II). Our method outperforms the state-of-the-art methods by a large margin over all evaluation metrics, including *Acc-w*, CIDEr, BLEU, ROUGE-L, METEOR, and WER. Experimental results on **Split II** indicate that our method has a strong capability to deal with the unseen sentence recognition problem.

6. Conclusions

In this paper, we propose a new deep architecture based on 3D-ResNet and encoder-decoder network with connectionist temporal classification by iterative optimization for continuous SLR. We jointly train encoder-decoder network by minimizing CTC loss and cross-entropy loss, additionally with a soft-DTW alignment constraint. The clip labels generated by the warping path, which aligns each clip to its corresponding sign word, are regarded as the supervision to fine-tune the feature extractor. The 3D-ResNet feature extractor and encoder-decoder sequence modelling network are alternately optimized step by step. Our method achieves better performance on two public continuous SLR datasets than the existing methods. Experimental results demonstrate the effectiveness and superiority of our approach.

Acknowledgement

This work was supported in part to Dr. Houqiang Li by 973 Program (No. 2015CB351803) and NSFC (No. 61836011), and in part to Dr. Wengang Zhou by NSFC (No. 61822208 and 61632019), Young Elite Scientists Sponsorship Program By CAST (2016QNRC001), and the Fundamental Research Funds for the Central Universities.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2, 4

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*, 2017. 2, 3

[3] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. SubUNets: End-to-end hand shape and continuous sign language recognition. In *ICCV*, 2017. 7

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[5] Shizhe Chen, Jia Chen, Qin Jin, and Alexander Hauptmann. Video captioning with guidance of multimodal latent topics. In *ACM MM*, 2017. 2

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 3

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014. 2

[8] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NIPS*, 2015. 2

[9] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *CVPR*, 2018. 1

[10] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *CVPR*, 2017. 1, 2, 7

[11] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *ICML*, 2017. 4

[12] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. 2, 4, 8

[13] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *TPAMI*, 31(5):855–868, 2009. 2

[14] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 2005. 3

[15] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical LSTM for sign language translation. In *AAAI*, 2018. 6, 8

[16] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Online early-late fusion based on adaptive HMM for sign language recognition. *TOMM*, 14(1):8, 2018. 1

[17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet? *arXiv preprint arXiv:1711.09577*, 2017. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2

[19] Alejandro Hernandez Ruiz, Lorenzo Porzi, Samuel Rota Bulò, and Francesc Moreno-Noguer. 3D CNNs on distance matrices for human action recognition. In *ACM MM*, 2017. 1

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 3, 8

[21] Takaaki Hori, Shinji Watanabe, and John Hershey. Joint CTC/attention decoding for end-to-end speech recognition. In *ACL*, 2017. 2, 3

[22] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Attention based 3D-CNNs for large-vocabulary sign language recognition. *TCSVT*, 2018. 1

[23] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, 2018. 1, 3, 6, 7

[24] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013. 1, 2, 3

[25] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *CVIU*, 141:108–125, 2015. 1, 6, 7

[26] Oscar Koller, Hermann Ney, and Richard Bowden. Read my lips: Continuous signer independent weakly supervised viseme recognition. In *ECCV*, 2014. 1

[27] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *CVPR*, 2016. 7

[28] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: hybrid CNN-HMM for continuous sign language recognition. In *BMVC*, 2016. 7

[29] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *CVPR*, 2017. 1

[30] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015. 2

[31] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks. In *CVPR*, 2016. 1

[32] Junfu Pu, Wengang Zhou, and Houqiang Li. Dilated convolutional network with iterative optimization for continuous sign language recognition. In *IJCAI*, 2018. 3, 7

[33] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *ICCV*, 2017. 1, 2

[34] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Transactions on Acoustics, Speech, and Signal Processing*, 1978. 4

[35] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *TPAMI*, 20(12):1371–1375, 1998. 1

[36] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NIPS*, 2015. 2

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 1, 2, 3

[38] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, 2015. 2, 8

[39] Shuo Wang, Dan Guo, Wengang Zhou, Zheng-Jun Zha, and Meng Wang. Connectionist temporal fusion for sign language translation. In *ACM MM*, 2018. 7

[40] Sy Bor Wang, Ariadna Quattoni, L-P Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *CVPR*, 2006. 1

[41] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[42] Fang Yin, Xiujuan Chai, and Xilin Chen. Iterative reference driven metric learning for signer independent isolated sign language recognition. In *ECCV*, 2016. 1

[43] Jihai Zhang, Wengang Zhou, Chao Xie, Junfu Pu, and Houqiang Li. Chinese sign language recognition with adaptive HMM. In *ICME*, 2016. 1, 6, 8