Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization

Runpeng Cui* Hu Liu* Changshui Zhang
Department of Automation, Tsinghua University
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
{crp16@mails, liuhu15@mails, zcs@mail}.tsinghua.edu.cn

Abstract

This work presents a weakly supervised framework with deep neural networks for vision-based continuous sign language recognition, where the ordered gloss labels but no exact temporal locations are available with the video of sign sentence, and the amount of labeled sentences for training is limited. Our approach addresses the mapping of video segments to glosses by introducing recurrent convolutional neural network for spatio-temporal feature extraction and sequence learning. We design a three-stage optimization process for our architecture. First, we develop an end-to-end sequence learning scheme and employ connectionist temporal classification (CTC) as the objective function for alignment proposal. Second, we take the alignment proposal as stronger supervision to tune our feature extractor. Finally, we optimize the sequence learning model with the improved feature representations, and design a weakly supervised detection network for regularization. We apply the proposed approach to a real-world continuous sign language recognition benchmark, and our method, with no extra supervision, achieves results comparable to the state-of-the-art.

1. Introduction

Sign language is regarded as the most grammatically structured category of gestural communications. This nature of sign language makes it an ideal test bed for developing methods to solve problems such as motion analysis and human-computer interaction.

Continuous sign language recognition is different from isolated gesture classification [7, 20] or sign spotting [8, 23, 30], which is to detect predefined signs from video stream and the supervision contains exact temporal locations for each sign. In the problem of continuous sign language recognition, each video of sign language sentence is provided with its ordered gloss labels but no time boundaries for each gloss. (We usually use "gloss" to represent sign with its closest meaning in natural languages [24].) Therefore, continuous sign language recognition can be cast as one of the weakly supervised problems, and the main issue is to learn the corresponding relations between the image time series and the sequences of glosses.

Recently, methods using deep convolutional neural networks (CNNs) have achieved breakthroughs in gesture recognition [20] and sign spotting [23, 30], and recurrent neural networks (RNNs) has shown significant results while learning the dynamic temporal dependencies in sign spotting [21, 26].

However, continuous sign language recognition with deep neural networks remains challenging and non-trivial. In this problem, the recognition system is required to achieve representation and sequence learning from the weakly supervised unsegmented video stream. Since video sequences and gloss labels are given in sentence-level, the amount of training data would increase drastically to make the model align the gestures and gloss labels correctly without overfitting. Although RNNs have shown superior performance to hidden Markov models (HMMs) on handling complex dynamic variations in sign recognition [21, 26], with limited amount of training data RNNs are more inclined to end in overfitting.

Furthermore, although deep CNN has been proved to outperform hand-crafted features in almost all computer vision tasks, there is no direct, precise frame-level supervision for CNN's training in this problem. A complex end-to-end model with CNN as visual feature extractor may lead to overfitting as well. This presents the challenge of constructing suitable semantic learning objectives to guide the training process of the feature extractor.

In this paper, we contribute a novel method for real-world sign language recognition from continuous image

*The first two authors contributed equally to this work.

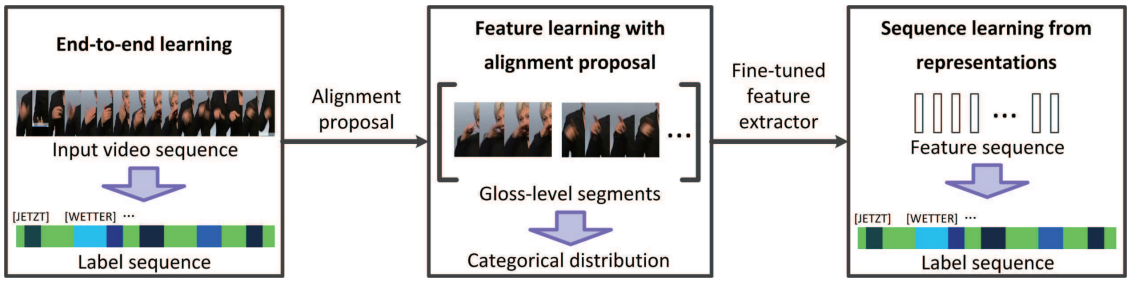

Figure 1. This is the overview of our staged training approach: (1) end-to-end-training the full architecture with feature and sequence learning components to predict the alignment proposal; (2) training the feature extractor with the alignment proposal; (3) training sequence learning component with the improved representation sequence as input, which is given by the fine-tuned feature extractor.

streams. The main contributions of our work can be summarized as follows:

(1) We develop our architecture with recurrent convolutional neural networks to achieve performance comparable to the state-of-the-arts in this weakly supervised problem, without importing extra information;

(2) We fully exploit the representation capability of deep convolutional neural network by segmenting the sentence-level labels to vast amounts of temporal segments with gloss labels, which directly guides the training of deep architecture for feature representation and avoids overfitting efficiently;

(3) We design a three-stage optimization process for training our deep neural network architecture (see Fig. 1), and our approach is proved to take notable effect on the limited training set;

(4) To the best of our knowledge, we are the first to propose a real-world continuous sign language recognition system fully based on deep neural networks in this scope, and we demonstrate its applicability from challenging continuous sign video streams.

2. Related Work

Most systems for sign language recognition consist of a feature extractor to represent the spatial and temporal variations in sign language, and a sequence learning model to learn the correspondence between feature sequences and sequences of glosses. Moreover, continuous sign language recognition [16, 18, 19] is also closely related to weakly supervised learning problem, where precise temporal locations for the glosses are not available. Here we introduce the works related to sign language analysis from these aspects.

Spatio-temporal representations. Many previous works in the area of sign analysis [8, 16, 22, 25] use hand-crafted features for spatio-temporal representations. In recent years, there has been a growing interest in feature extraction with deep neural networks due to the superior representation capability. The neural network methods adopted in gesture analysis include CNNs [17, 18, 26], 3D CNNs [21, 23, 30] and temporal convolutions [26]. However, due to the data insufficiency in the problem of continuous sign language learning, the training of deep neural networks is inclined to end in overfitting. To alleviate the problem, Koller *et al*. [18] integrate CNN into a weakly supervised learning scheme. They use the weakly labelled sequence with hand shape as an initialization to iteratively tune CNN and refine the hand shape labels with Expectation Maximization (EM) algorithm. Koller *et al*. [17] also adopt the finger and palm orientations as the weakly supervision for tuning CNN. Different from [17, 18], we do not require extra annotations in our approach, and we directly use the gloss-level alignment proposals instead of sub-unit labels to help the network training.

Sequence learning. Sign language recognition aims at learning the correspondences between input sequences and sign labels with sequence learning models. HMMs are widely used in sign language analysis for sequence learning from continuous time-series [16, 18, 30]. Recently, RNNs have shown state-of-the-art performance in the task of sign spotting [21, 26]. Pigou *et al*. [26] propose an end-to-end deep architecture with temporal convolutions and bidirectional recurrence. Molchanov *et al*. [21] employ a recurrent 3D-CNN with CTC as the cost function for hand gesture recognition. Their architectures are related to ours, but it is non-trivial to expect that simply applying their methods will work well in continuous sign language recognition, since their aim is not to recognize the whole sign language sentence, but the isolated glosses within the sentences. To the best of our knowledge, we are the first to develop the architecture for continuous sign language recognition fully based on deep neural networks.

Weakly supervised learning. Due to lack of temporal boundaries for the sign glosses in the image sequences, continuous sign language recognition is also a typical weakly supervised learning problem. Cooper and Bowden [6] use method from data mining to extract similar regions from videos, then the scheme of Mean Shift [5] is used to refine

the locations in temporal domain. Buehler *et al*. [3] design a scoring function based on multiple instance learning (MIL) to locate the signs of interest. Pfister *et al*. [25] use cues of subtitles, mouthing and hand motion to search for signs by a discriminative MIL approach. However, most of these methods are concerning the problem of mining isolated signs of interest from large number of sign videos.

A related work to ours is recently proposed by Koller *et al*. [19]. They develop a hybrid CNN-HMM approach, which treats the outputs of CNN as Bayesian posteriors and uses the frame-level hidden states predicted by HMM to tune CNN. There are key differences between our approach and theirs: (1) instead of given frame-level labels with noises as training targets of CNN, we take the temporal variations into account and adopt the gloss-level alignments for training the spatio-temporal feature extractor, and (2) our approach is self-contained and does not require results from other systems for frame-state alignment initialization.

3. Method

Vision-based continuous sign language recognition systems usually take the image sequences of signers' performance as input, and learn to automatically output the gloss labels in right order. In this work, our proposed approach employs CNN with temporal convolution and pooling for spatio-temporal representation learning from video clips, and RNN with long short-term memory (LSTM) module to learn the mapping of feature sequences to sequences of glosses.

To effectively train our deep architecture, we introduce a three-stage optimization process: (1) finding the alignment proposal by end-to-end training the full architecture with feature extractor and sequence learning component; (2) tuning the feature extraction component by using the correspondence between gloss-level segments and categorical probabilities from the alignment proposal; (3) tuning the sequence learning component with improved feature representations as inputs.

To refine the result for sequential prediction, we propose a sign detection net and jointly integrate the detection with the sequence learning outputs as the regularization for sequence learning. An overview of our method is presented in Fig. 1. The remainder of this section discusses our approach in detail.

3.1. Network architecture

Our proposed architecture consists of a CNN with temporal convolution and pooling for spatial and local temporal feature extraction, a bidirectional LSTM [13] for global sequence learning, and a detection network for refining the sequence learning results.

Spatio-temporal feature extractor. Let $\{x_t\}_{t=1}^{T} = (x_1, \cdots, x_T)$ be the input video stream as a sequence of images with time T. We use function $\mathcal{F}$ to represent CNN, which transforms input frames $\{x_t\}_{t=1}^{T}$ to a spatial representation vector sequence $\{f_t\}_{t=1}^{T} = \mathcal{F}(\{x_t\}_{t=1}^{T})$ with $f_t \in \mathbb{R}^c$. We set the stacked temporal convolution with zero-padding and temporal pooling operations as function $\mathcal{P} : \mathbb{R}^{\ell \times c} \rightarrow \mathbb{R}^d$ with receptive field ℓ, temporal stride δ and output dimension d, and each segment with length ℓ is transformed into a spatio-temporal representation:

$$\{s_n\}_{n=1}^{N} = \mathcal{P}(\{f_t\}_{t=1}^{T}) = \mathcal{P}(\mathcal{F}(\{x_t\}_{t=1}^{T})), \tag{1}$$

where $N = T/\delta$ represents the number of segments, and $s_n \in \mathbb{R}^d$ denotes the representation of segment n. The structure of CNN stacked with temporal operations $\mathcal{P} \circ \mathcal{F}$ is the spatio-temporal feature extraction architecture, which transforms video segments into approximate gloss-level representations. In our experiments, we set the receptive field ℓ to 10 frames, which is equal to the median length of isolated glosses provided by [9]. Therefore we dubbed it approximate "gloss-level".

Bidirectional LSTM. The bidirectional LSTM (BLSTM) computes the hidden state sequence by combining the output sequences of LSTM by iterating forwards from $t = 1$ to τ and backwards from $t = \tau$ to 1, which can be simply represented as:

$$\{h_n^{\mathrm{c}}\}_{n=1}^{N} = \mathcal{R}(\{s_n\}_{n=1}^{N}), \tag{2}$$

where $\mathcal{R}$ denotes the temporal modeling function of BLSTM, and h_n^{c} is the BLSTM's output which is going to give the categorical prediction next. Finally, we employ a fully connected layer with softmax to convert the outputs of BLSTM into categorical probabilities of gloss labels with K classes:

$$P_{ij}^{\mathrm{c}} = [\sigma_{\mathrm{cls}}(\phi(h_j^{\mathrm{c}}))]_i = \frac{e^{[\phi(h_j^{\mathrm{c}})]_i}}{\sum_{k=1}^{K} e^{[\phi(h_j^{\mathrm{c}})]_k}}, \tag{3}$$

where $P \in [0,1]^{K \times N}$ and P_{ij}^{c} is the emission probability of label i at time j, σ_{cls} denotes the softmax function performed on the classes, and ϕ represents the linear mapping to $\mathbb{R}^K$ of the fully connected layer. Here we use $[\cdot]_i$ to denote the i-th element of the vector.

Detection net. In the proposed detection net, we employ stacked temporal convolution operations on the spatio-temporal feature vectors $\{s_n\}_{n=1}^{N}$, which is like a sliding-window detection manner along the gloss-level feature sequences. The stacked temporal convolution $\mathcal{C}$ transforms $\{s_n\}_{n=1}^{N}$ into a representation sequence of length N for detection:

$$\{h_n^{\mathrm{d}}\}_{n=1}^{N} = \mathcal{C}(\{s_n\}_{n=1}^{N}). \tag{4}$$

Spatio-temporal feature extractor	
CNN (VGG-S / GoogLeNet)	
conv1D-3-1024 maxpool1D-2 conv1D-3-1024 maxpool1D-2	
Recurrent neural net	**Detection net**
BLSTM-512	conv1D-2-256 conv1D-2-256
fully connected layer softmax	fully connected layer softmax

Table 1. Configuration of our architecture. The parameters for temporal convolution are denoted as "conv1D-[receptive field]-[number of channels]". Temporal pooling layers are annotated with stride, and bidirectional LSTM (denoted by "BLSTM") is with the dimension number of its hidden variables. The output dimensions of the fully connected layers are equal to the size of gloss vocabulary in our architecture.

We pass sequence $\{h_n^{\mathrm{d}}\}_{n=1}^N$ through a softmax layer to get the detection scores as follows:

$$P_{ij}^{\mathrm{d}} = [\sigma_{\mathrm{det}}(\psi(h_j^{\mathrm{d}}))]_i = \frac{e^{[\psi(h_j^{\mathrm{d}})]_i}}{\sum_{k=1}^N e^{[\psi(h_k^{\mathrm{d}})]_i}}, \tag{5}$$

where ψ denotes the linear transformation to $\mathbb{R}^K$, and $P^{\mathrm{d}} \in [0,1]^{K\times N}$. Softmax σ_{det} is different from σ_{cls}. In the detection net, σ_{det} compares the temporal proposals for each class and selects those segments matching the class with higher scores, while σ_{cls} predicts the likely class at each time-step.

The configurations of our proposed architecture is presented in Table 1.

3.2. Alignment proposal by end-to-end learning

At the stage of end-to-end training, our full architecture with feature extractor and sequence learning model takes the image sequences $\boldsymbol{x} = \{x_t\}_{t=1}^T$ as the inputs and learn to output the ordered gloss labels $\boldsymbol{y}$ in an end-to-end manner (see Fig. 2). Since we have no prior knowledge of where the signs occur in the unsegmented image stream, here we employ connectionist temporal classification (CTC) [12] as the objective function of our full architecture.

CTC is an objective function that integrates all possible alignments between the input and target sequence. We add an extra class "blank" to the gloss vocabulary to explicitly model the transition between two neighboring signs. The CTC alignment π is then a sequence of blank and gloss labels with length N. Let $\boldsymbol{x} = \{x_t\}_{t=1}^T$, the probability $\Pr(\pi|\boldsymbol{x})$ of π is given by the product of probabilities:

$$\Pr(\pi|\boldsymbol{x}) = \prod_{n=1}^N \Pr(\pi_n|\boldsymbol{x}) = \prod_{n=1}^N P_{\pi_n,n}^{\mathrm{c}}, \tag{6}$$

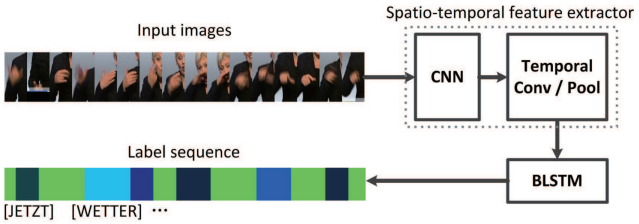

Figure 2. Overview of the end-to-end learning stage of the optimization process.

where π_n is the label of π at time n, and $P_{\pi_n,n}^{\mathrm{c}}$ is the emission probability of π_n at time n.

As presented in [11, 12], the same input and target sequence, which has no blank, can have different alignments due to different ways of blanks separating the gloss labels. We define the many-to-one mapping of alignments onto the target sequence $\boldsymbol{y}$ as $\mathcal{B}$, and the probability of observing $\boldsymbol{y}$ is the sum of probabilities of all alignments corresponding to it:

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\boldsymbol{y})} \Pr(\pi|\boldsymbol{x}), \tag{7}$$

where $\mathcal{B}^{-1}(\boldsymbol{y}) = \{\pi|\mathcal{B}(\pi) = \boldsymbol{y}\}$ is the set of all the alignments. The CTC loss function is defined as:

$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{x},\boldsymbol{y}) = -\log\Pr(\boldsymbol{y}|\boldsymbol{x}). \tag{8}$$

Let $\mathcal{S}$ be the training set, which is the collection of image sequence with its ordered label sequence $(\boldsymbol{x},\boldsymbol{y})$, and $\boldsymbol{w}$ be the stacked vector of all filter parameters employed in the proposed deep architecture, we then define the training objective as:

$$\mathcal{L} = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \mathcal{L}_{\mathrm{CTC}}(\boldsymbol{x},\boldsymbol{y}), \tag{9}$$

where λ is the hyperparameter for regularization. Through the end-to-end training strategy, for each input sequence $\boldsymbol{x}$ our architecture outputs the categorical distribution p_n at each time-step, and we take them as the alignment proposal $P^{\alpha}(\boldsymbol{x}) = \{p_n\}_{n=1}^N$. We use the alignment proposal as stronger supervision to further tune our deep architecture for feature extraction at the later stage.

3.3. Feature learning with alignment proposal

Our end-to-end training stage provides the outputs of BLSTM as approximate alignments between video segments and gloss labels. To fully exploit the representation capability of the deep architecture for feature learning, we take the categorical scores as the gloss-level supervision for segment at each time-step, and we use these segments with

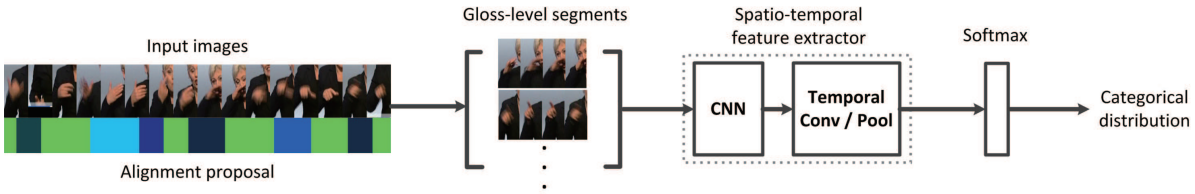

Figure 3. Overview of the feature learning process with alignment proposal.

class probabilities as stronger supervision to directly tune the deep spatio-temporal feature extractor (see Fig. 3).

Here we extend the gloss-level encoding architecture $\mathcal{P} \circ \mathcal{F}$ with a softmax layer φ, which transforms the input $\boldsymbol{x}$ into:

$$\varphi(\mathcal{P} \circ \mathcal{F}(\boldsymbol{x})) = \{\varphi(s_n)\}_{n=1}^{N}, \tag{10}$$

where $\varphi(s_n) \in [0,1]^K$ is the predicted categorical distribution for time-step n. Given the alignment proposal as $P^{\alpha}(\boldsymbol{x}) = \{p_n\}_{n=1}^{N}$, which provides the categorical distribution of target at each time-step, we define the objective for gloss-level alignment as:

$$\mathcal{L}_{\text{align}}(\boldsymbol{x}, P^{\alpha}(\boldsymbol{x})) = \frac{1}{N}\sum_{n=1}^{N} d_{\text{KL}}(p_n \| \varphi(s_n)), \tag{11}$$

where we use Kullback-Leibler divergence d_{KL} to measure the distribution difference between $\varphi(s_n)$ and p_n. Therefore, the training objective for this stage can be presented as:

$$\mathcal{L} = \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \mathcal{L}_{\text{align}}(\boldsymbol{x}, P^{\alpha}(\boldsymbol{x})). \tag{12}$$

We take the alignment proposal as the assignment of learning targets to the temporal segments, which provides numerous gloss-level training samples with stronger supervision, and we use this scheme to tune the feature extractor for much better spatio-temporal representations.

3.4. Sequence learning from representations

At this stage, we adopt the tuned feature extractor to provide representation sequence $\{s_n\}_{n=1}^{N}$ for video stream $\boldsymbol{x}$, and we further train the sequence learning model to learn the ordered labels with CTC loss, taking the representations as inputs.

To further improve the generalization capability of our recurrent neural network and avoid overfitting, inspired by the weakly supervised object detection scheme developed in [2], here we propose the detection net for sign glosses to implicitly locate them in the temporal sequences. By optimizing the detection scheme jointly with the CTC objective function, the deep network architecture not only learns to predict the gloss sequence, but also aligns label with the according video segment more precisely, thus improving the generalization capability of the model.

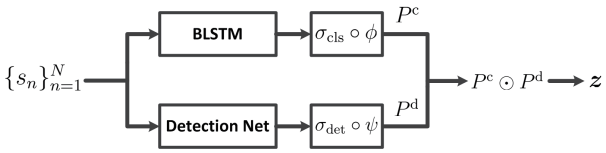

Figure 4. This illustrates the approach to integrating the classification and detection scores.

Unlike visual object detection from images [2], there is no proposal method to segment and generate candidate temporal intervals of interest from time-series. Therefore, we construct the detection net, as introduced in section 3.1, performing like a sliding-window detection approach along the gloss-level feature sequences. We use element-wise product of P^{c} and P^{d} and sum up the scores over the temporal proposals:

$$z_k = \sum_{n=1}^{N} P_{kn}^{\text{c}} \cdot P_{kn}^{\text{d}}, \tag{13}$$

where we take $z_k \in (0,1)$ as the score that gloss k occurs in this sign language videos. Fig. 4 illustrates the approach to integrating the scores of detection and classification. We let $\mathcal{Y}$ be the set of glosses contained in target sequence $\boldsymbol{y}$, and $\mathcal{A}$ be the gloss dictionary without "blank". Then the objective function of training the detection net is given by:

$$\mathcal{L}_{\text{det}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k\in\mathcal{A}\setminus\mathcal{Y}} \log(1 - z_k) + \sum_{k\in\mathcal{Y}} \log z_k. \tag{14}$$

We take $\mathcal{L}_{\text{det}}$ as the regularization on the prediction of temporal locations, and the objective function for training the sequence learning model at this stage is presented as:

$$\mathcal{L} = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} (\mathcal{L}_{\text{CTC}} + \mu\mathcal{L}_{\text{det}})(\boldsymbol{x}, \boldsymbol{y}), \tag{15}$$

where λ and μ are hyperparameters for regularization, and $\boldsymbol{w}$ is the stacked vector of all filter parameters in our proposed sequence learning architecture.

By integrating the scores of sequential prediction and gloss detection together, the model is trained not only with sequential cues, but also with the consideration of the context in detection, and it is expected to predict more precise alignments in accord with the detection outputs. Moreover, the detection net can be seen as a component of multi-task learning with shared weights for representation, which encourages the sequence learning to be further improved.

4. Experiments

In this section we analyze the performance of our approach on continuous sign language recognition.

4.1. Implementation details

Dataset and evaluation. We evaluate our method on RWTH-PHOENIX-Weather multi-signer 2014 [10], which is a publicly available benchmark dataset for continuous sign language recognition. This dataset contains 5,672 sentences in German sign language for training with 65,227 sign glosses and 799,006 frames in total. These videos are performed by 9 signers, and each video contains a single gloss sentence.

To evaluate the performance quantitatively, we employ word error rate (WER) as the criterion, which is widely-used in the scope of continuous sign language recognition. WER measures the least operations of substitution, deletion and insertion to transform the reference sequence into the hypothesis:

$$\mathrm{WER} = \frac{\#\mathrm{sub} + \#\mathrm{del} + \#\mathrm{ins}}{\#\text{words in reference}}, \tag{16}$$

where $\#\mathrm{sub}$, $\#\mathrm{del}$ and $\#\mathrm{ins}$ stand for the number of required substitutions, deletions and insertions, respectively.

Image preprocessing. In order to provide comparable results, all the input images are cropped right (dominant) hand patches provided by RWTH-PHOENIX-Weather multi-signer 2014 dataset. The size of cropped patches is 92×132 pixels. In our experiment, these crops are resized to the size of 101×101 (for VGG-S [4]) or 224×224 (for GoogLeNet [28]), from which the mean image is subtracted.

CNN pretraining. At this stage, we focus on pretraining the CNN for spatial representations before the stacked 1-dimensional temporal convolution and pooling. We first initiate our CNN with VGG-S model pretrained on ILSVRC-2012 dataset [27]. We choose this "relatively shallow" network mainly in consideration of GPU memory constraints when jointly optimizing CNN and RNN. Then, we apply the nonlinearity of $\tanh$ to the last fully connected layer and fine-tune the network on the training set with triplets of patches including positive and negative pairs as in the work of PN-Net [1]. We select two neighbouring frames from one sentence within 3-frame intervals as a positive patch pair and one frame from another sentence with no shared gloss as the negative patch. We train the network using stochastic gradient descent (SGD) with a fixed learning rate of 5×10^{-5} and a momentum of 0.9. We set the batch size to 48 and stop pretraining after 16,000 iterations.

Alignment proposal by end-to-end learning. We remove the last fully connected layer from the pretrained VGG-S model and add to it the stacked temporal convolution and pooling as the spatio-temporal feature extractor. We propose a BLSTM as the sequence learning model, and we use the objective function given in Eq. 9 with $\lambda = 5 \times 10^{-4}$ to train the full architecture. We employ ADAM [15] as the stochastic optimization approach with a fixed learning rate of 5×10^{-5}. We apply temporal scaling up to $\pm 20\%$ as the approach for data augmentation to increase the variability of the video sequences.

Feature learning with alignment proposal. We start training the feature extractor with the alignment proposal given by the end-to-end training stage. This alignment proposal is employed to generate numerous video segments with the according categorical scores as supervision. We split training and validation set of pairs with segment and categorical score at a ratio of 10:1 and guarantee that the video segments extracted from the same sentence fall within the same set. At the stage of tuning the representation learning architecture, we employ GoogLeNet [28] pretrained on ILSVRC-2014 [27] as the CNN model, which shows better performance on the problem of large-scale image classification. To extend its temporal receptive field to be compatible with the video segment, we apply a modification to the ConvNet by inserting a stacked temporal convolution with max-pooling layer before each classifier of GoogLeNet. We employ ADAM [15] as the stochastic optimization approach. We set a fixed learning rate to 5×10^{-5}, batch size to 20 and stop the finetuning after 16,000 iterations.

Sequence learning from representations. At the stage of training the sequence learning model, we take the feature map from layer "pool5/7x7_s1" of fine-tuned GoogLeNet as the representation for each patch of right hand in the video stream. We use the feature sequence as input to tune the sequence learning architecture with objective function defined in Eq. 15, where we set $\lambda = 5 \times 10^{-4}$, $\mu = 0.5$ and a fixed learning rate to 5×10^{-5}.

4.2. Results

Design choices. We analyze the performance and effect of each individual component in our proposed approach. In the phase of end-to-end training for alignment proposal, we look into the ingredients of our feature and sequence learning architecture. We substitute 3D-CNN [21, 29] for our proposed CNN with stacked temporal convolution and

pooling, and we also assess the utility of pre-training the C-NN with loss employed in PN-Net [1] from video frames. Besides, we also try to find out the effect of different recurrent sequence learning models. Continuous sign language recognition results of these experiments are listed in Table 2. We use "ConvTC+BLSTM" to denote our proposed architecture with VGG-S net pretraining on ISLVRC 2012, and "+pretrain" for further applying pre-training with loss in PN-Net [1] to our proposed model.

We observe from Table 2 that our proposed approach for spatio-temporal representations in a later fusion manner [14] outperforms the recurrent 3D-CNN in this problem by a large margin. We think the reason for CNN with temporal convolutions performing better is that there are less parameters compared to 3D-CNN with the same number of layers, thus it is less prone to overfitting. We also notice that pretraining our model on the right hand patches from training videos further improves the performance, since the network learns from the similarity and continuity of the hand shape streams.

We analyze the effect of each component in our sequence learning model at the stage of sequence learning, where we get the spatio-temporal representations from feature extractor and tune the recurrent component. To understand the utility of our proposed detection net, we remove it from our model in this experiment. We also substitute different models of recurrence for the employed BLSTM. In Table 3 we observe that BLSTM gives the best performance among the recurrent models, since it can fully take advantage of the information from the context. Notice that the employment of detection net achieves consistently superior performance compared to learning the sequential mapping with BLSTM alone. The results demonstrate the effectiveness of our proposed temporal detection net. Moreover, we observe that by fine-tuning the feature extractor with the alignment proposal, all the results given by different sequence learning models outperform the best one at the stage of end-to-end training. This demonstrates that the representation learning process is crucial and contributes greatly to the performance improvement.

Alignment evaluation. We further analyze the effect of our proposed approach from the perspective of alignment performance.

We observe from Fig. 5 that, the BLSTM with only CTC loss shows inferior prediction, both in error rate and in location accuracy, to the full model. We address that CTC only optimizes the sequence to sequence correspondence but takes no consideration of alignment. Our sliding-window-based detection net implicitly aligns the segment-level detection score with the sequential prediction, which results in a better alignment. Thus the employment of our detection net is more inclined to possess better generalization performance for the entire model.

Model setup	Validation del / ins / WER	Test del / ins / WER
C3d+BLSTM	45.6 / 2.5 / 76.8	46.9 / 2.8 / 77.6
ConvTC+RNN	19.5 / 6.8 / 53.8	18.9 / 7.6 / 53.7
ConvTC+LSTM	21.4 / 6.3 / 50.9	20.7 / 6.8 / 51.3
ConvTC+BLSTM	16.8 / 6.8 / 47.8	15.8 / 7.9 / 47.3
+pretrain	16.3 / 6.7 / **46.2**	15.1 / 7.4 / **46.9**

Table 2. Recognition results for end-to-end training stage on RWTH-PHOENIX-Weather 2014 multi-signer dataset in [%]. "C3d" stands for the 3D-CNN structure employed in [21, 29], "ConvTC" for our proposed feature extraction architecture with VGG-S net pretrained on ISLVRC 2012, and "+pretrain" for our model further pretrained with PN-Net [1] loss on the right hand patches from training set.

Model setup	Validation del / ins / WER	Test del / ins / WER
Our-end2end	16.3 / 6.7 / 46.2	15.1 / 7.4 / 46.9
RNN	19.6 / 5.4 / 45.0	18.1 / 6.2 / 44.8
LSTM	18.1 / 5.7 / 43.3	17.1 / 6.6 / 43.6
BLSTM	14.9 / 6.7 / 41.4	15.1 / 7.1 / 41.9
BLSTM+det net	13.7 / 7.3 / **39.4**	12.2 / 7.5 / **38.7**

Table 3. Recognition results for sequence learning stage on RWTH-PHOENIX-Weather 2014 multi-signer dataset in [%]. We assess the performance of different recurrent models and our proposed detection net. "BLSTM+det net" stands for the employed model with bidirectional LSTM and detection net, and "Our-end2end" for the full model with best performance in the stage of end-to-end training.

From Fig. 5 we also observe significant gain of alignment performance after finetuning the feature extractor. This observation is consistent with the results of WER in Table 3.

Results on multiple signers. In our experiments, no schemes are specifically taken or designed for the inter-signer variations. The amounts of training samples for the 9 signers are unbalanced in this dataset, with the three most sampled signers account for 26.0%, 22.8%, 14.7% and three least 0.5%, 0.8%, 2.9%, while the WERs (in %) for these signers on validation set are 36.0, 38.6, 43.8 and 45.8, 43.3, 38.7 respectively. This indicates that our system can learn the shared representations among different signers and to some extent handle the inter-signer variations.

Comparisons. In Table 4, We evaluate our proposed method together with the state-of-the-arts on RWTH-PHOENIX-Weather multi-signer 2014 dataset. We observe that our approach achieves comparable performance to the state-of-the-arts without using extra supervision, which contains a sign language lexicon mapping signs to hand shape sequences. Moreover, our approach using only infor-

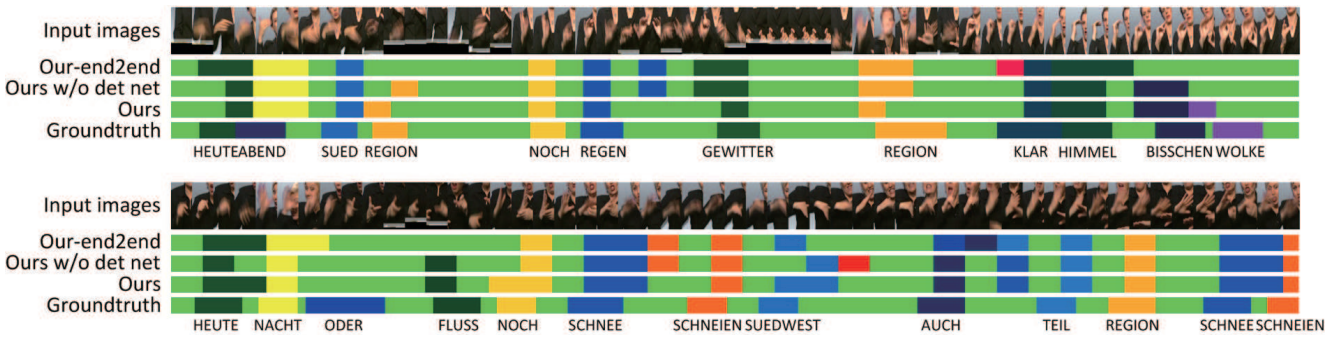

Figure 5. Two examples for qualitative alignment results on gloss sentence videos from test set. Colors are used to represent different glosses and horizontal axis to represent time. "Our-end2end" stands for the full model at end-to-end stage, "Ours" stands for the top-performing model with BLSTM and detection net at sequence learning stage, and "Ours w/o det net" stands for the BLSTM sequence learning model without the detection net. We manually annotate the groundtruth by comparing with the gloss samples provided by RWTH-PHOENIX-Weather 2012 dataset [9].

Model setup	Extra supervision	Modality			Validation		Test	
		r-hand	traj	face	del / ins	WER	del / ins	WER
HOG-3D [16]		✓			25.8 / 4.2	60.9	23.2 / 4.1	58.1
[16] CMLLR		✓	✓	✓	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1-Mio-Hands [18]	✓	✓			19.1 / 4.1	51.6	17.5 / 4.5	50.2
1-Mio-Hands [18]+[16]	✓	✓	✓	✓	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid [19]	✓	✓			12.6 / 5.1	**38.3**	11.1 / 5.7	38.8
Our-end2end		✓			16.3 / 6.7	46.2	15.1 / 7.4	46.9
Ours		✓			13.7 / 7.3	39.4	12.2 / 7.5	**38.7**

Table 4. Performance comparison of different continuous sign language recognition approaches on RWTH-PHOENIX-Weather 2014 multi-signer dataset in [%]. "r-hand" stands for right hand and "traj" stands for trajectory motion. "Extra supervision" imported in [18] contains a sign language lexicon mapping signs to hand shape sequences, and the best result of [19] uses [18]+[16] as the initial alignment.

mation from dominant hand even outperforms those multi-modal methods by a large margin. These results can quantitatively demonstrate the effectiveness of our approach.

Notice that our approach performs comparably to the CNN-Hybrid system [19]. It should be clarified that our system is self-contained and does not need the initial alignment imported from other systems. While in the CNN-Hybrid system, the best performance is achieved with the help of initial alignment provided by the approach of [18]+[16]. Therefore, the multi-modal information and extra supervision is implicitly imported to aid the optimization of their system. However, it seems that suitable exploitation of extra supervision is crucial to the notable improvement of performance, and we may investigate it in the future work. Besides, it is also necessary to extend the algorithm to a multi-modal version to integrate complementary cues for further improvements.

5. Conclusion

In this paper, we have proposed a deep architecture with recurrent convolutional neural network for continuous sign language recognition. We have designed a staged optimization process for training our deep neural network architecture. We fully exploit the representation capability of CNN with tuning on vast amounts of gloss-level segments and effectively avoid overfitting with the deep architecture. We have also proposed a novel detection net for regularization on the consistency between sequential predictions and detection results. The effectiveness of our approach is demonstrated on a challenging benchmark, where we have achieved the performance comparable to the state-of-the-art.

Acknowledgement

This work is supported by 973 Program (2013CB329503), Natural Science Foundation of China (Grant No. 61473167 and No. 61621136008), and the German Research Foundation (DFG) in Project Crossmodal Learning DFC TRR-169.

References

[1] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. PN-Net: Conjoined triple deep network for learning local image descriptors. *arXiv*, 2016.

[2] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *Proc. CVPR*, 2016.

[3] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching TV (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC*, 2014.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[6] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. CVPR*, 2009.

[7] H. Cooper, E. J. Ong, N. Pugeault, and R. Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205–2231, 2012.

[8] G. D. Evangelidis, G. Singh, and R. Horaud. Continuous gesture recognition from articulated poses. In *ECCV Workshops*, 2014.

[9] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. Ney. RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus. In *Language Resources and Evaluation Conference*, 2012.

[10] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney. Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-Weather. In *Language Resources and Evaluation Conference*, 2014.

[11] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

[12] A. Graves, S. Fernàndez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, 2006.

[13] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*, 2014.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

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

[17] O. Koller, H. Ney, and R. Bowden. Automatic alignment of HamNoSys subunits for continuous sign language. In *Language Resources and Evaluation Conference Workshops*, 2016.

[18] O. Koller, H. Ney, and R. Bowden. Deep hand: how to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *Proc. CVPR*, 2016.

[19] O. Koller, S. Zargaran, H. Ney, and R. Bowden. Deep sign: hybrid CNN-HMM for continuous sign language recognition. In *Proc. BMVC*, 2016.

[20] P. Molchanov, S. Gupta, K. Kim, and J. Kautz. Hand gesture recognition with 3D convolutional neural networks. In *CVPR Workshops*, 2015.

[21] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *Proc. CVPR*, 2016.

[22] C. Monnier, S. German, and A. Ost. A multi-scale boosted detector for efficient and robust gesture recognition. In *ECCV Workshops*, 2014.

[23] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. In *ECCV Workshops*, 2014.

[24] S. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):873–891, 2005.

[25] T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. BMVC*, 2013.

[26] L. Pigou, A. v. d. Oord, S. Dieleman, M. M. Van Herreweghe, and J. Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *arXiv*, 2015.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*, 2015.

[30] D. Wu, L. Pigou, P.-J. Kindermans, N. Le, L. Shao, J. Dambre, and J.-M. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597, 2016.