A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training

Runpeng Cui, Hu Liu, and Changshui Zhang, *Fellow, IEEE*

***Abstract*—This work develops a continuous sign language (SL) recognition framework with deep neural networks, which directly transcribes videos of SL sentences to sequences of ordered gloss labels. Previous methods dealing with continuous SL recognition usually employ hidden Markov models with limited capacity to capture the temporal information. In contrast, our proposed architecture adopts deep convolutional neural networks with stacked temporal fusion layers as the feature extraction module, and bi-directional recurrent neural networks as the sequence learning module. We propose an iterative optimization process for our architecture to fully exploit the representation capability of deep neural networks with limited data. We first train the end-to-end recognition model for alignment proposal, and then use the alignment proposal as strong supervisory information to directly tune the feature extraction module. This training process can run iteratively to achieve improvements on the recognition performance. We further contribute by exploring the multimodal fusion of RGB images and optical flow in sign language. Our method is evaluated on two challenging SL recognition benchmarks, and outperforms the state-of-the-art by a relative improvement of more than 15% on both databases.**

***Index Terms*—continuous sign language recognition, sequence learning, iterative training, multimodal fusion.**

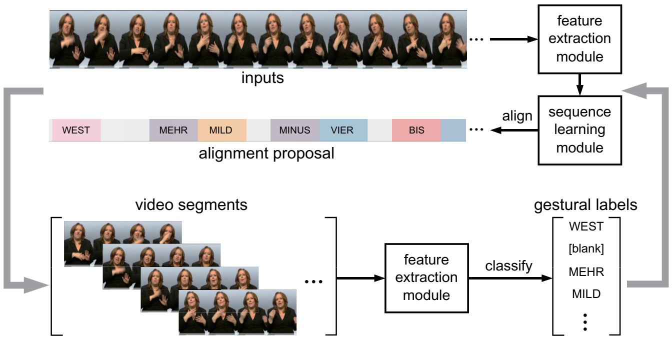


Fig. 1. Iterative training process in our approach. The end-to-end trained full architecture provides alignment proposals, and the feature extractor is further tuned to learn the matching of video segments and gestural labels.

I. Introduction

SIGN language (SL) is commonly known as the primary language of deaf people, and usually collected or broadcast in the form of video. SL is often considered as the most grammatically structured gestural communications [1]. This nature makes SL recognition an ideal research field for developing methods to address problems such as human motion analysis, human-computer interaction (HCI) and user interface design, and makes it receive great attention in multimedia and computer vision [2], [3], [4].

Typical SL learning problems involve isolated gesture classification [3], [5], [6], [7], sign spotting [8], [9], [10], and continuous SL recognition [11], [12], [13]. Generally speaking, gesture classification is to classify isolated gestures to correct categories, while sign spotting is to detect predefined signs from continuous video streams, with precise temporal boundaries of gestures provided for training detectors. Different from these problems, continuous SL recognition is to transcribe videos of SL sentences to ordered sequences of glosses (here we use "gloss" to represent a gesture with its closest meaning in natural languages [1]), and the continuous video streams are provided without prior segmentation. Continuous SL recognition concerns more about learning unsegmented gestures of long-term video streams, and is more suitable for processing continuous gestural videos in real-world systems. Its training also does not require an expensive annotation on temporal boundary for each gesture. Recognizing SL indicates simultaneous analysis and integration of gestural movements and appearance features, as well as disparate body parts [1], and therefore probably using a multimodal approach. In this paper, we focus on the problem of continuous SL recognition on videos, where learning the spatiotemporal representations as well as their temporal matching for the labels is crucial.

Many studies [11], [14], [15], [16] have made their efforts on representing SL with hand-crafted features. For example, hand and joint locations are used in [11], [17], local binary patterns (LBP) is used in [16], histogram of oriented gradients (HOG) is utilized in [15], and its extension HOG-3D is applied in [11]. Recently, deep convolutional neural networks have achieved a tremendous impact on related tasks on videos, *e.g*. human action recognition [18], [19], [20], gesture recognition [6] and sign spotting [9], [10], and recurrent neural networks (RNNs) have shown significant performance on learning the temporal dependencies in sign spotting [4], [21]. Several recent approaches taking advantage of neural networks have also been proposed for continuous SL recognition [12], [13], [22]. In these works, neural networks are restricted to learning frame-wise representations, and hidden Markov models (HMMs) are utilized for sequence learning. However, the frame-wise labelling adopted in [12], [13], [22] is noisy for training the deep neural networks, and HMMs might be

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant 61876095 and Grant 61751308, and Beijing Natural Science Foundation under Grant L172037.

R. Cui, H. Liu and C. Zhang are with Institute for Artificial Intelligence, Tsinghua University (THUAI), State Key Laboratory of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China. (e-mail: {crp16@mails, liuhu15@mails, zcs@mail}.tsinghua.edu.cn)

hard to learn the complex dynamic variations, considering their limited representation capability.

This paper therefore develops a recurrent convolutional neural networks for continuous SL recognition. Our proposed neural model consists of two modules for spatiotemporal feature extraction and sequence learning respectively. Due to the limited scale of the datasets, we find an end-to-end training cannot fully exploit the deep neural network of high complexity. To address this problem, we investigates an iterative optimization process (Fig. 1) to train our recurrent deep neural architecture effectively. We use gloss-level gestural supervision given by forced alignment from end-to-end system to directly guide the training process of the feature extractor. Afterwards, we fine-tune the recurrent neural system with the improved feature extractor, and the system can provide further refined alignment for the feature extraction module. Through this iterative training strategy, our deep neural network can keep learning and benefiting from the refined gestural alignments. The main contributions of our work can be summarized as follows:

1) We develop our architecture with recurrent convolutional neural networks of more learning capacity to achieve state-of-the-art performance on continuous SL recognition, without importing extra supervisory information;
2) We design an iterative optimization process for training our deep neural network architecture, and our approach, with the neural networks better exploited, is proved to take notable effect on the limited training set in contrast to the end-to-end trained system;
3) We propose a multimodal version of our framework with RGB frames and optical flow images, and experiments present that our multimodal fusion scheme provides better representations for the gestures and further improves the performance of the system.

The remainder of this paper is organized as follows. Section II reviews related works on SL recognition. Section III introduces the formulation of our deep neural network for SL recognition, and its iterative optimization scheme. Section IV provides implementation details on our model. Section V presents the experimental results of the proposed method and Section VI concludes the paper.

II. Related Work

SL recognition systems on videos usually consist of a feature extraction module, which extracts sequential representations to characterize gesture sequences, and a temporal model mapping sequential representations to labels.

Many hand-crafted features have been introduced for gesture and SL recognition. These features characterize handshape, appearance and motion cues, by using image pixel intensity [16], gradients [11], [15], [23] and motion trajectories or velocities [8], [11], [17]. In recent years, there has been a growing trend to learn feature representations by deep neural networks. Wu *et al*. [24] employ a deep belief network to extract high-level skeletal joint features for gesture recognition. Convolutional neural networks (CNNs) [25], [26] and 3D convolutional neural networks (3D-CNNs) [9], [10], [4] have also been employed to capture visual cues for hand regions. For instance, Molchanov *et al*. [4] apply 3D-CNNs for spatiotemporal feature extraction from video streams on color, depth and optical flow data. Neverova *et al*. [9] present a multi-scale deep architecture on color, depth data and hand-crafted pose descriptors.

Temporal model is to learn the correspondences between sequential representations and gloss labels. HMMs are the most widely used temporal models in SL recognition [10], [11], [13]. Besides, dynamic time warping (DTW) [16] and SVMs [27] are also used for measuring similarity between gestures. Recently, RNNs have been successfully applied to sequential problems such as speech recognition [28] and machine translation [29], [30], and some progress has also been made for exploring the application of RNNs in SL recognition. Pigou *et al*. [21] propose an end-to-end neural model with temporal convolutions and bidirectional recurrence for sign spotting, which is taken as frame-wise classification in their framework. However, with only weak supervision in sentence level, recurrent neural networks are hard to learn to match the over-length input sequence frame by frame with the ordered labels. Different from their model, we use temporal pooling layers to integrate the temporal dynamics before the bidirectional recurrence. Molchanov *et al*. [4] employ a recurrent 3D-CNN with connectionist temporal classification (CTC) [31] as the cost function for gesture recognition, while in our experiments, we find that our architecture shows a much superior performance compared to 3D-CNN model on the SL recognition benchmarks.

Due to lack of temporal boundaries for the sign glosses in the image sequences, continuous SL recognition is also a typical weakly supervised learning problem. There have been some attempts focusing on the problem of mining gestures of interest from large amount of SL videos, where signs and annotations are usually coarsely aligned with considerable noise. Different from our problem, they usually take more focus on local temporal dynamics but not long-term dependencies. Buehler *et al*. [15] propose a scoring function based on multiple instance learning (MIL) and search for signs of interest by maximizing the score. Pfister *et al*. [27] use subtitle text, lip and hand motion cues to select candidate temporal windows, and these candidates are further refined using MI-SVM [32]. Chung and Zisserman [33] use a ConvNet learned on image encoding representing human keypoint motion for recognition, and they locate temporal positions of signs via saliency map by back-propagation.

There have been a few works exploring the problem of continuous SL recognition. Gweth *et al*. [34] employ a one-hidden-layer perceptron to estimate posterior from appearance-based features, and use the probabilities as inputs to train an HMM-based recognition system. Koller *et al*. [12], [13], [25] adopt CNNs for feature extraction from cropped hand regions and also use HMMs to model the temporal relationships. As the amount of training data is not sufficient enough, training of deep neural networks is inclined to end in overfitting. To alleviate this problem, Koller *et al*. [12] embed a CNN within a weakly supervised learning framework. Weakly labelled sequence of hand shape annotations are brought in as an initialization, to iteratively train CNN and re-estimate hand shape labels within Expectation Maximization (EM) [35] framework. Similarly, annotations of finger and palm orientations are also imported as weakly supervised information to train CNN [25]. In their later works [13], [22], they use the frame-state alignment, provided by a baseline HMM recognition system, as frame labelling to train the embedded neural networks. In contrast with these works [12], [13], [25], [22], our sequence learning module of recurrent neural networks with end-to-end training shows much more learning capacity and better performance for the dynamic dependencies. Besides, instead of using noisy frame-wise labelling as training targets of neural networks, we adopt the gloss-level alignment proposal to train our feature extraction module, which takes more local temporal dynamics into consideration. Moreover, no extra supervisory information such as hand shape annotations is imported in our approach. Notice that the development of such lexicon requires laborious annotation with expert knowledge, while our method is free from this limitation.

Different from our previous work [36], we propose a distinctive segment-gloss alignment method to learn from the outputs of our sequence learning module, and we provide an explicit illustration for our iterative training scheme, by proving the training of feature extraction module to be maximizing the lower bound of the objective function, instead of using an intuitive approach. We also contribute by investigating more on the multimodal integration of appearance and motion cues in this work.

III. Method

In this work, our proposed architecture adopts a feature extraction module composed of a deep CNN followed by temporal fusion layers, and a sequence learning module using RNNs with bidirectional long short-term memory (Bi-LSTM) architecture.

We propose a novel iterative optimization scheme to effectively train our deep architecture. We use the end-to-end recognition system to generate alignment proposal between video segments and gestural labels. Given the large amount of gestural segments with supervisory labels, we train the feature extraction module and then fine-tune the whole system iteratively. An overview of our approach is presented in Fig. 1. In the remainder of this section, we will first present our model formulation and then introduce its iterative training strategy.

A. Model Formulation

We use a CNN followed by temporal convolutional and pooling layers to learn spatiotemporal representations from input video streams. Letting $\{\boldsymbol{x}_t\}_{t=1}^T$ be the input video stream of length T, the employed CNN f_{CNN} transforms the input sequence into some spatial representation sequence $\{\boldsymbol{r}_t\}_{t=1}^T = f_{\text{CNN}}(\{\boldsymbol{x}_t\}_{t=1}^T)$ with $\boldsymbol{r}_t \in \mathbb{R}^C$, where C is the feature dimensionality. The feature sequence $\{\boldsymbol{r}_t\}_{t=1}^T$ is then processed by stacked temporal convolution and pooling operations $f_{\text{Temp}} : \mathbb{R}^{\ell \times C} \to \mathbb{R}^D$, with temporal stride δ, receptive field ℓ and output dimensionality D, to get:

$$\{\boldsymbol{s}_k\}_{k=1}^K = f_{\text{Temp}}(\{\boldsymbol{r}_t\}_{t=1}^T) = (f_{\text{Temp}} \circ f_{\text{CNN}})(\{\boldsymbol{x}_t\}_{t=1}^T), \quad (1)$$

where $K = T/\delta$ represents the length of extracted spatiotemporal representation sequence $\{\boldsymbol{s}_k\}_{k=1}^K$, and we use $f_{\text{Temp}} \circ f_{\text{CNN}}$ to denote the processing of the proposed feature extraction module. The feature extraction module transforms k-th video segments of length ℓ to representation $\boldsymbol{s}_k$. In general, we set the receptive field approximate to the length of an isolated sign. Therefore, we consider the video segments as approximate "gloss-level".

One shortcoming of unidirectional RNNs is that the hidden states are computed only from previous time steps. However in SL, the gestural performance and meaning is closely related to its previous as well as succeeding contexts. Therefore, Bi-LSTMs [37] are employed to learn the complex dynamics by mapping sequences of spatiotemporal representation to sequences of ordered labels. Bi-LSTM computes the forward and backward hidden sequences by iterating the LSTM computation [38] from $k = 1$ to K and from $k = K$ to 1 respectively:

$$\boldsymbol{h}_k^{\text{f}},\ \boldsymbol{c}_k^{\text{f}} \ = \ f_{\text{LSTM-frw}}(\boldsymbol{s}_k, \boldsymbol{h}_{k-1}^{\text{f}}, \boldsymbol{c}_{k-1}^{\text{f}}), \quad (2)$$

$$\boldsymbol{h}_k^{\text{b}},\ \boldsymbol{c}_k^{\text{b}} \ = \ f_{\text{LSTM-bck}}(\boldsymbol{s}_k, \boldsymbol{h}_{k+1}^{\text{b}}, \boldsymbol{c}_{k+1}^{\text{b}}), \quad (3)$$

where $\boldsymbol{h}_k^{\text{f}},\ \boldsymbol{c}_k^{\text{f}}$ denote the hidden state and memory cell of forward LSTM module $f_{\text{LSTM-frw}}$ at the k-th time step, and $\boldsymbol{h}_k^{\text{b}},\ \boldsymbol{c}_k^{\text{b}}$ denote those of the backward one $f_{\text{LSTM-bck}}$. This scheme helps the recurrent neural networks to exploit future context as well as previous context at the same time. Finally, the output categorical probabilities of M gloss labels at time k are computed through a softmax classifier, which takes the concatenation of hidden states of Bi-LSTM as the input:

$$\boldsymbol{z}_k = \text{softmax}(\boldsymbol{W}[\boldsymbol{h}_k^{\text{f}}; \boldsymbol{h}_k^{\text{b}}] + \boldsymbol{b}), \quad (4)$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and bias vector for the softmax classifier, and we use $[\cdot;\cdot]$ to represent the concatenation operation.

We let $\boldsymbol{\theta} = [\boldsymbol{\theta}^{\text{f}}; \boldsymbol{\theta}^{\text{s}}]$ denote the vector of all parameters employed in the end-to-end recognition system, where $\boldsymbol{\theta}^{\text{f}}$ and $\boldsymbol{\theta}^{\text{s}}$ denote the parameters of the feature extraction and sequence learning module respectively. We will introduce our approach to learning these parameters in the remainder of this section.

B. Training with CTC Objective Function

Since the video streams are unsegmented in continuous SL recognition during training, we introduce connectionist temporal classification (CTC) [31] to solve this transcription problem. CTC is an objective function originally designed for speech recognition, requiring no prior alignments between input and output sequences.

Our recognition system, with $\boldsymbol{\theta} = [\boldsymbol{\theta}^{\text{f}}; \boldsymbol{\theta}^{\text{s}}]$ as the stacked vector of all its filters, takes video stream $\boldsymbol{x} = \{\boldsymbol{x}_t\}_{t=1}^T$ as the input sequence to predict the sequence of gloss labels $\boldsymbol{y}$. We add an extra token "blank" to the gloss vocabulary to model the gestural transitions explicitly. As the categorical probabilities $\boldsymbol{z}_k$ of each gloss (including blank) for segment k has been normalized by the softmax classifier, we let $\Pr(m, k|\boldsymbol{x}; \boldsymbol{\theta}) = z_k^m$, which indicates the emission probability of label m at time step k using the m-th element of $\boldsymbol{z}_k$. In the formulation of CTC, the probability of an alignment $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{K}$ given length K is defined as the product at each time step:

$$\Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{k=1}^{K} \Pr(\pi_k, k|\boldsymbol{x};\boldsymbol{\theta}). \tag{5}$$

The alignments typically include blanks and repeated tokens. Alignments are mapped into the given target sequence $\boldsymbol{y}$ by the many-to-one mapping $\mathcal{B}$, with repeated labels and blanks removed. For example, both the alignment $(-, a, a, -, b)$ and $(-, a, b, -, -)$ correspond to the target sequence (a, b), where a, b are gestural labels, and "$-$" denote the blank label for gestural transition. Letting $\mathcal{V}(\boldsymbol{y}) = \{\boldsymbol{\pi}|\mathcal{B}(\boldsymbol{\pi}) = \boldsymbol{y}\}$ represent all the alignments corresponding to target sequence $\boldsymbol{y}$, the probability of the target transcription can be computed by summing the probabilities of all the alignments corresponding to it:

$$\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{\pi}\in\mathcal{V}(\boldsymbol{y})} \Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}). \tag{6}$$

Using this computation for $\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, the CTC objective function is defined as:

$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta}) = -\log \Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}). \tag{7}$$

The deep neural network for end-to-end SL recognition can then be trained to minimize the CTC objective function.

The calculation of CTC objective function is solved using a dynamic programming algorithm described below [31]. A modified sequence $\boldsymbol{y}'$ is defined by inserting blanks before and after every gestural label, to allow for the blank label in the alignments. Let $U = |\boldsymbol{y}|$ be the number of labels contained in $\boldsymbol{y}$, we have $U' = |\boldsymbol{y}'| = 2U + 1$ as the length of $\boldsymbol{y}'$. We define the forward variable $\alpha(k, u)$ as the summed probability of all paths up to time step k and the prefix of $\boldsymbol{y}'$ with length u. In the formulation of CTC, note that $\alpha(k, u)$ can be calculated recursively as:

$$\alpha(k, u) = \Pr(y'_u, k|\boldsymbol{x};\boldsymbol{\theta}) \sum_{i=g(u)}^{u} \alpha(k-1, i), \tag{8}$$

where

$$g(u) = \begin{cases} u-1 & \text{if } y'_u \text{ is blank or } y'_{u-2} = y'_u \\ u-2 & \text{otherwise} \end{cases} \tag{9}$$

with the boundary conditions given in [31], and we have $\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \alpha(K, U'-1) + \alpha(K, U')$.

C. Learning from Alignments

At this stage, we utilize the alignment proposal given by the end-to-end tuned system to train the feature extraction module. To fully exploit the representation capability of the deep convolutional network $\boldsymbol{\theta}^{\mathrm{f}}$, here we consider using the feature extraction module to maximize the objective function:

$$\mathcal{L}(\boldsymbol{\theta}^{\mathrm{f}}) = \log \Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}). \tag{10}$$

As the feature extractor cannot model the long dynamic dependencies, instead of training it directly with CTC objective, we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}^{\mathrm{f}}) &= \log \Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}) && (11)\\
&= \log \sum_{\boldsymbol{\pi}\in\mathcal{A}} \Pr(\boldsymbol{y}, \boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}) && (12)\\
&\geq \sum_{\boldsymbol{\pi}\in\mathcal{A}} \Pr(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}^{*}) \log \frac{\Pr(\boldsymbol{y}, \boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}})}{\Pr(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}^{*})} && (13)\\
&= \sum_{\boldsymbol{\pi}\in\mathcal{A}} \Pr(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}^{*}) \log \Pr(\boldsymbol{y}, \boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}) + \mathrm{const} && (14)\\
&= Q(\boldsymbol{\theta}^{\mathrm{f}}) + \mathrm{const}, && (15)
\end{aligned}$$

where $\mathcal{A}$ represents all the possible alignments of length K, $\boldsymbol{\theta}^{*}$ denotes the parameters of the trained end-to-end model, and the constant is negative entropy of the distribution $\Pr(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}^{*})$ and therefore independent of $\boldsymbol{\theta}^{\mathrm{f}}$. (13) uses Jensen's inequality for the concave function $\log(\cdot)$.

Note that given a particular alignment $\boldsymbol{\pi}$, the output sequence is uniquely determined as $\mathcal{B}(\boldsymbol{\pi})$. It is easy to find $\Pr(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{x}) = \Pr(\boldsymbol{y}|\boldsymbol{\pi}) = \mathbf{1}(\boldsymbol{\pi} \in \mathcal{V}(\boldsymbol{y}))$ to represent the constraint to possible alignments with given target sequence, where $\mathbf{1}(\cdot)$ represents the indicator function. Then we can write:

$$\Pr(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}^{*}) = \frac{\Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{*})}{\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^{*})} \cdot \mathbf{1}(\boldsymbol{\pi} \in \mathcal{V}(\boldsymbol{y})), \tag{16}$$

and

$$\Pr(\boldsymbol{y}, \boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}) = \Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}) \cdot \mathbf{1}(\boldsymbol{\pi} \in \mathcal{V}(\boldsymbol{y})). \tag{17}$$

Based on the constraint to the alignments, the objective is formulated as:

$$\mathcal{L}(\boldsymbol{\theta}^{\mathrm{f}}) = \log \sum_{\boldsymbol{\pi}\in\mathcal{V}(\boldsymbol{y})} \Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}), \tag{18}$$

and the lower bound is further transformed into:

$$Q(\boldsymbol{\theta}^{\mathrm{f}}) = \sum_{\boldsymbol{\pi}\in\mathcal{V}(\boldsymbol{y})} \frac{\Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{*})}{\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^{*})} \log \Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}). \tag{19}$$

In general, we will not be able to optimize all possible alignments which are corresponding to $\boldsymbol{y}$. As the converged end-to-end system typically outputs single alignment with dominant probability on training examples, we select the alignment in $\mathcal{V}(\boldsymbol{y})$ with the highest probability estimation as the proposal, and use the feature extraction module to learn from it instead. When alignment $\hat{\boldsymbol{\pi}}$ take a dominant probability among all the possible alignments in $\mathcal{V}(\boldsymbol{y})$, we can see that:

$$Q(\boldsymbol{\theta}^{\mathrm{f}}) \approx \frac{\Pr(\hat{\boldsymbol{\pi}}|\boldsymbol{x};\boldsymbol{\theta}^{*})}{\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^{*})} \log \Pr(\hat{\boldsymbol{\pi}}|\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{f}}), \tag{20}$$

where we make an approximation to $Q(\boldsymbol{\theta}^{\mathrm{f}})$ by choosing the alignment with highest probability to represent the integration over possible alignments. The proposed alignment is given by:

$$\hat{\boldsymbol{\pi}} = \operatorname*{arg\,max}_{\boldsymbol{\pi}\in\mathcal{V}(\boldsymbol{y})} \Pr(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta}^{*}). \tag{21}$$

We define $\hat{\alpha}(k,u)$ as the maximum path probability up to time step k and the prefix of $\boldsymbol{y}'$ with length u, and we have initial conditions as:

$$\hat{\alpha}(k,0) = 0,\ 1 \leq k \leq K, \tag{22}$$

$$\hat{\alpha}(1,u) = \begin{cases} \Pr(y'_u, 1|\boldsymbol{x};\boldsymbol{\theta}^*) & u = 1,2 \\ 0 & 2 < u \leq U' \end{cases}. \tag{23}$$

Similar to (8), we can calculate $\hat{\alpha}(k,u)$ recursively as:

$$\hat{\alpha}(k,u) = \Pr(y'_u, k|\boldsymbol{x};\boldsymbol{\theta}^*) \max_{i \in \{i|g(u) \leq i \leq u\}} \hat{\alpha}(k-1,i). \tag{24}$$

With the formulation of CTC and all the notations used before, we develops following algorithm to find the proposed alignment $\hat{\boldsymbol{\pi}} = \{\hat{\pi}_k\}_{k=1}^{K}$.

```
Input: $\hat{\alpha}(k,u),\ 1 \leq k \leq K,\ 1 \leq u \leq U'$
Output: $\hat{\boldsymbol{\pi}}$
$\gamma \leftarrow \arg\max_{i \in \{U'-1, U'\}} \hat{\alpha}(K,i)$.
$\hat{\pi}_K \leftarrow y'_\gamma$.
for $k = K-1$ to $1$ do
    $\gamma \leftarrow \arg\max_{i \in \{i|g(\gamma) \leq i \leq \gamma\}} \hat{\alpha}(k,i)$.
    $\hat{\pi}_k \leftarrow y'_\gamma$.
end for
return $\hat{\boldsymbol{\pi}}$
```

We note that the optimization of (20) contributes to the maximization of original objective $\mathcal{L}(\boldsymbol{\theta}^{\mathrm{f}})$ in (18), by maximizing the weighted likelihood of alignment proposal with the highest estimated probability.

As the bidirectional recurrence for sequence learning takes full context into consideration, we assume that the alignment proposal $\hat{\boldsymbol{\pi}}$ typically gives a reliable estimation of temporal localization for most signs, and the spatiotemporal representation from each segment should have strong correspondence to the segment label given by the alignment proposal.

Based on these assumptions, letting $\mathcal{S}$ be the training set as the collection of video stream with its annotation $(\boldsymbol{x},\boldsymbol{y})$, the objective function for feature extraction module training can be transformed from (20) to:

$$\mathcal{L}_{\text{align}}(\boldsymbol{\theta}^{\mathrm{f}}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{S}} \rho(\boldsymbol{x},\boldsymbol{y}) \sum_{k=1}^{K} \log \Pr(\hat{\pi}_k, k|\boldsymbol{x},\boldsymbol{\theta}^{\mathrm{f}}), \tag{25}$$

where $\rho(\boldsymbol{x},\boldsymbol{y}) = \Pr(\hat{\boldsymbol{\pi}}|\boldsymbol{x};\boldsymbol{\theta}^*)/\Pr(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^*)$, and we present the probability of alignment proposal as the product of emission probabilities at each segment. To learn the feature extraction module from alignment proposal, we partition the video sequences to segments with one supervisory gestural label for each, according to the dominant alignment proposal.

The objective function also puts more weights $\rho(\boldsymbol{x},\boldsymbol{y})$ to training examples with estimated alignments of more confidence. In practice, we extend the feature extractor with a softmax layer and tune the parameters by maximizing $\mathcal{L}_{\text{align}}$.

After tuning the feature extractor parameters $\boldsymbol{\theta}^{\mathrm{f}}$ from segment samples provided by alignment, we continue to train the full deep neural architecture with the improved feature extraction module. The objective function of CTC is employed to fine-tune the recognition system, and the fine-tuned system can give further improved alignments. This training procedure can run iteratively until no improvement is observed in the performance of the system.

IV. Model Implementation

In this section, we provide more implementation details of our approach.

A. Model Design

The proposed deep neural architecture consists of a deep CNN followed by temporal operations for representation learning, and Bi-LSTMs for sequence learning.

For experiments with modalities from dominant hands as the inputs, we build the deep convolutional network based on the VGG-S model [39] (from layer `conv1` to `fc6`), which is memory-efficient and shows competitive classification performance on ILSVRC-2012 dataset [40]. The input images, which are the region of dominant hands cropped from original frames, are resized to 101×101 in dimension, and they are then transformed to 1024-dimensional feature vectors through the fully connected layer `fc6`.

The stacked temporal convolution and pooling layers are utilized to generate spatiotemporal representation for each segment. Note that it is hard to learn the extremely long dynamic dependencies with no temporal pooling, while a coarse temporal stride will lead to loss of temporal details. We select the temporal stride δ to ensure sufficient overlapping between neighboring segments, as well as pool the representation sequence to a moderate length. For videos in RWTH-PHOENIX-Weather database, we set $\ell = 16$ frames, $\delta = 4$ frames, and we set $\ell = 25$ frames, $\delta = 9$ frames in experiments on SIGNUM corpus. In the feature extraction module, rectifier and max-pooling are adopted for all the non-linearity and pooling operations.

We use Bi-LSTMs with 2×512 dimensional hidden states and peephole connections to learn the temporal dependencies. The hidden states are then fed into the softmax classifier, with the dimension equal to the vocabulary size.

We also investigate the performance of our training framework with full video frames as the inputs. We use GoogLeNet [41] and also VGG-S net as the deep convolutional network in our feature extractor, and we adopt two stacked Bi-LSTMs to build the sequence learning module. Due to the limitations on GPU memory to fit in the whole system, we fix the parameters of CNN at the end-to-end stage and only tune the sequence learning module. The video frames are resized to 224×224 as the inputs of CNN, transformed to feature vectors after the average pooling layer, and then fed into the temporal fusion layers. The employed GoogLeNet is initialized with the weights pretrained on ILSVRC-2014 dataset [40], and we initialize the feature extractor by fitting it to the alignment proposal generated by the model end-to-end trained on dominant hand images.

B. Multimodal Fusion

To incorporate the appearance and motion information, we also take color image and optical flow for dominant hand regions as the inputs of our deep neural architecture. We adopt sum fusion approach at the `conv5` layer for fusing the two stream networks. It computes element-wise sum of the two

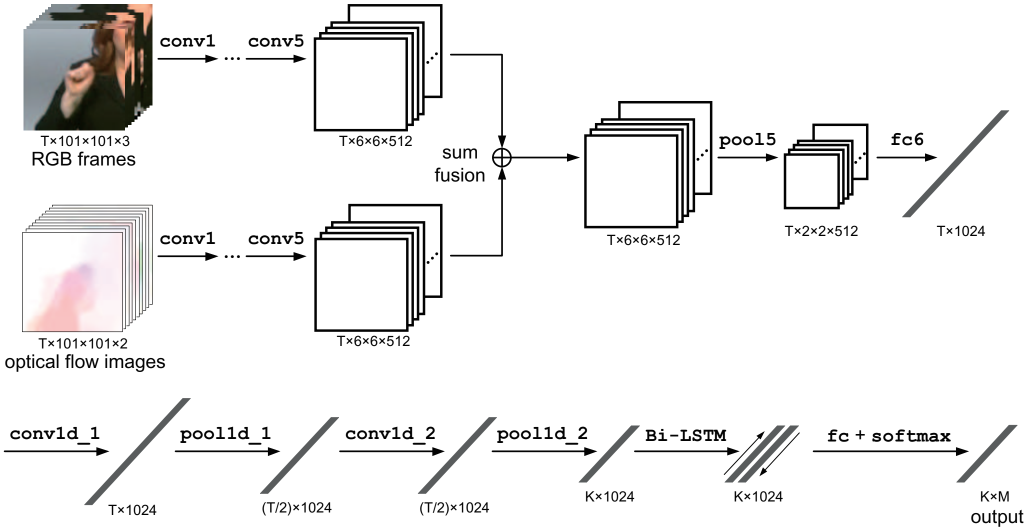


Fig. 2. Deep neural architecture for RGB and optical flow modalities of dominant hands. We take the model developed on RWTH-PHOENIX-Weather 2014 database for example, where T is the length of the input video sequence, $K = T/4$ is the sequence length after temporal pooling, and $M = 1296$ is the vocabulary size (including blank). Each displayed feature map is annotated with its dimensionality respectively. Parameters for different modalities are not shared in this architecture.

feature maps at the same spatial location and channel for the fusion. Our intention here is to put appearance and motion cues at the same spatial position in correspondence, without introducing extra filters in order to join the feature maps together. The sum fusion approach also shows a decent performance on the task of action recognition in video [42] compared to other spatial fusion methods. Our end-to-end architecture for SL recognition from dominant hands is depicted in Fig. 2. Note that parameters for different modalities are not shared before the sum fusion.

In experiments on multiple modalities of full frames, we adopt fusion of color and optical flow at two layers (after `inception_3b` and `inception_4c` in GoogLeNet) similar to [42]. Fig. 3 shows the fusion structures we build for experiments on recognition from multiple modalities of full frames. We also adopt the auxiliary classifiers as in GoogLeNet by adding to temporal fusion layers after `inception_4a` and `inception_4d` during the phase of feature extractor fine-tuning.

C. Implementation Details

In our experiments, we use DeepFlow method [43] for optical flow computation. For dominant hand locations which are not provided in the original databases, we adopt the faster R-CNN framework [44] to detect the dominant hand in each video frame. To increase the variability of the training examples, we add random temporal scaling up to ±20% to video streams, intensity noises [45] of standard deviation of 0.2 to the RGB modal, and randomly jitter the height and width of each input image by ±20%. RGB images and optical flows are fed into the neural network with the mean image subtracted. We train the full architecture for 100 epochs using Adam approach [46] with learning rate of 5×10^{-5} and a mini-batch size of 2.

For training the feature extraction module from alignment proposals, we split the pairs of segments and gestural labels into training and validation sets with the ratio of $10 : 1$. We adopt Adam [46] as the stochastic optimization approach with a fixed learning rate of 5×10^{-5} and a mini-batch size of 20. The training process of feature extractor is ceased when $\mathcal{L}_{\text{align}}$ starts to plateau on the validation sets, which is usually no more than 10 epochs for each iteration. To improve the generalization of the deep networks, we also adopt the ℓ_2-penalty for the weights of the network layers with a hyperparameter of 5×10^{-4} to balance the objective and the regularization when tuning the feature extractor and the full system. We stop the iterative training procedure until no performance improvement is observed on validation sets. During our experiments, the training process usually lasts for 3 to 4 iterations.

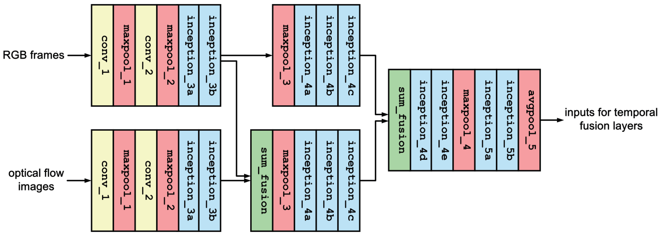


Fig. 3. Fusion structure for RGB and optical flow modalities of full frames. Sum fusion is used after layer `inception_3b` and `inception_4c`. Parameters for different modalities are not shared in this architecture.

Our neural architecture is implemented in Theano [47], and experiments are ran on NVIDIA Titan X GPUs.

V. Experiments

This section reports experiments performed on two benchmarks for continuous SL recognition to validate our approach. In Section V-A, we introduce the databases and the experimental protocol that we follow in the experiments. In the remainder of this section, we present and analyze our experimental results

TABLE I
STATISTICS OF RWTH-PHOENIX-WEATHER 2014 AND SIGNUM SIGNER-DEPENDENT DATABASES

Statistics	Phoenix-2014			SIGNUM	
	Train	Dev	Test	Train	Test
# frames	799,006	75,186	89,472	416,620	114,230
duration [hours]	8.88	0.84	0.99	3.86	1.06
# sentences	5,672	540	629	1,809	531
# gestural instances	65,227	5,540	6,504	11,874	2,979
# signers	9	9	9	1	1
vocabulary size	1,231	460	496	455	385
out-of-vocabulary [%]	-	0.69	0.69	-	0

on public SL recognition benchmarks, and we compare the SL recognition performance of our framework with the state-of-the-arts in Section V-E.

A. Datasets and Experimental Protocol

In this work the experiments are carried out on two publicly available databases, RWTH-PHOENIX-Weather multi-signer 2014 database [48] and SIGNUM signer-dependent set [14].

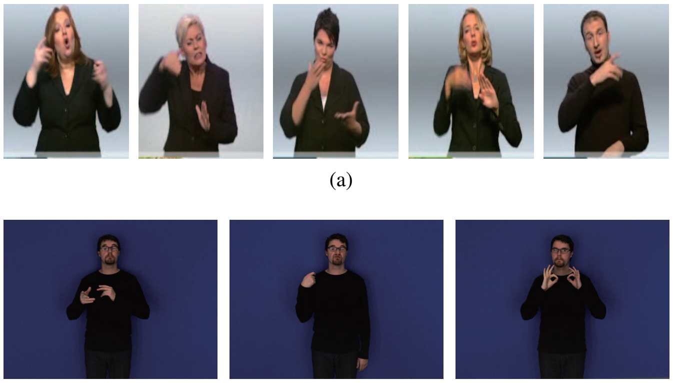
SIGNUM database is created under laboratory conditions, with recording environment (*e.g.* lighting, background, signer's position and clothes) carefully controlled. The signer-dependent subset of SIGNUM corpus contains 603 German SL sentences for training and 177 for testing, each sentence is performed by a native signer three times. The training corpus contains $11,874$ glosses and $416,620$ frames in total, with 455 different gestural categories.

In contrast to SIGNUM corpus, RWTH-PHOENIX-Weather 2014 database is collected from TV broadcasts of weather forecasts. It contains $5,672$ video sequences for training, with $65,227$ gestural instances and $799,006$ frames in total, performed by 9 signers. Each video sequence is for one German SL sentence and performed by a single person. The length of the video sequences ranges from 27 to 300 frames, all at the frame rate of 25 frames per second (fps). The vocabulary of all gestural labels (excluding "blank") is up to $1,295$. RWTH-PHOENIX-Weather 2014 database is a challenging benchmark partially due to its multi-signer settings. Besides, the fast hand motion and blurring in signing also add difficulties to accurate recognition.

We follow the experimental protocol adopted in [11] to split the database into training and testing subsets. The statistics for these two corpora in our experiments are shown in Tabel I. In Fig. 4, we present some example frames from these two databases.

To evaluate the experimental performance quantitatively, in this work we adopt word error rate (WER) as the criterion, which quantifies the dissimilarity of the predicted sequence of labels and the ground truth transcription. More precisely, WER measures the least operations of substitution, deletion and insertion, at the word level, to transform the predicted sequence into the ground truth. WER is defined as:

$$\mathrm{WER} = \frac{\#\mathrm{sub} + \#\mathrm{del} + \#\mathrm{ins}}{\#\mathrm{GT}}, \tag{26}$$


(a)

(b)

Fig. 4. Example frames from the two sign language recognition benchmarks. (a) RWTH-PHOENIX-Weather 2014 database, (b) SIGNUM database.

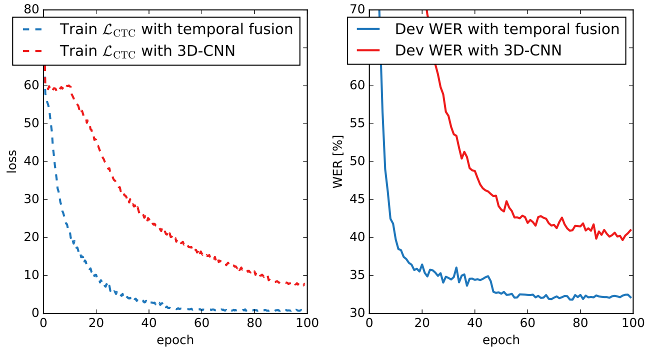


Fig. 5. Training (left) and validation process (right) on RWTH-PHOENIX-Weather 2014 database with our model (temporal fusion) and 3D-CNN model. Red curves show the CTC objective value and validation performance for 3D-CNN model over training epoches, and blue curves are for our approach.

where $\#\mathrm{sub}$, $\#\mathrm{del}$ and $\#\mathrm{ins}$ denote the number of substitutions, deletions and insertions respectively, and $\#\mathrm{GT}$ is the number of labels in the ground truth sequence.

B. Design Choices of Temporal Fusion

Notice that the temporal receptive field ℓ and stride δ are important to the temporal fusion of spatial features across video frames. We compare the performances with different choices of ℓ and δ on RWTH-PHOENIX-Weather 2014 and SIGNUM datasets (see Table II and Table III). We see that changes in visual inputs and modalities have little influence on the superiority of a suitable temporal fusion structure. One possible explanation is that temporal fusion layers generally focus on capturing temporal integration over different time steps, while visual inputs and modalities are more about various spatial properties of gestures, but with closely related dynamic dependencies. Therefore, a suitable temporal fusion structure can also be fit for other visual inputs with close temporal structures. We also observe that overlength temporal sequences with too small δ leads to the failure of the optimization of CTC, and temporal fusion with larger δ also results in the performance deterioration, which can be explained by the loss of temporal details.

Let `Ck` denote a temporal convolutional layer with 1024-dimensional filters of size k. `Pk` denotes a temporal max-pooling layer with stride k and kernel size k. Due to the consistent superior performances, we set the temporal fusion layers as `C5-P2-C5-P2` for RWTH-PHOENIX-Weather 2014 database, and `C5-P3-C5-P3` for SIGNUM database in our experiments. For these two benchmarks, the selected strides make the feature sequence before Bi-LSTMs about 4 times as long as label sequence on average. As for receptive field ℓ, we suggest that it should be better to cover the approximate length of single gestures in corpus, so that feature extractor can learn the representation for gestures with complete "gloss-level" information.

We also implement a recurrent 3D-CNN architecture [4] for the continuous SL recognition task. The recurrent 3D-CNN model reaches WER of 39.64% on development set and WER of 39.50% on test set. In contrast, our end-to-end training architecture achieves WER of 32.21% on development set and 32.70% on test set. Fig. 5 shows a typical example of the training process. Our approach with temporal fusion converges more quickly over training epochs, and also achieves a much better performance on validation set than 3D-CNN architecture. This result suggests that our proposed model is a more suitable structure than 3D-CNN models on this corpus.

C. Recognition with Multimodal Cues

In this section, we present our experimental results with multimodal setups on both databases. On RWTH-PHOENIX-Weather 2014 database (see Table IV), the multimodal fusion scheme brings complementary information to the appearance inputs, and shows a consistently superior performance in contrast to learning from single modality for all network configurations. Moreover, the performances of our neural networks, for all different inputs of modality and network configurations, are improved by iterative training scheme consistently, although the last training iteration sometimes sees a slight decrease in performance. Among all the experimental configurations on RWTH-PHOENIX-Weather 2014 database, Our system, with multimodal fusion scheme and GoogLeNet architecture in feature extraction module, presents the best performance, achieving a WER of 23.10% on development set and 22.86% on test set, which benefits from both multimodal fusion and iterative training, with relative improvements of 6.87% and 6.08% on test set respectively.

In Table V we can observe similar system performances on SIGNUM database. Our deep neural networks consistently benefit from the proposed iterative training scheme as well as multimodal fusion. The system with architecture of VGG-S net and modal fusion shows the best performance, with WER of 2.80%, where the multimodal integration approach and the iterative training brings improvements of 1.13% and 0.37% in WER respectively.

To demonstrate the effectiveness of our proposed method qualitatively, some recognition results with different settings are shown in Fig. 6. We can see that both our multimodal fusion design and iterative training strategy improve the recognition performance and help to provide more precise predictions to the input videos.

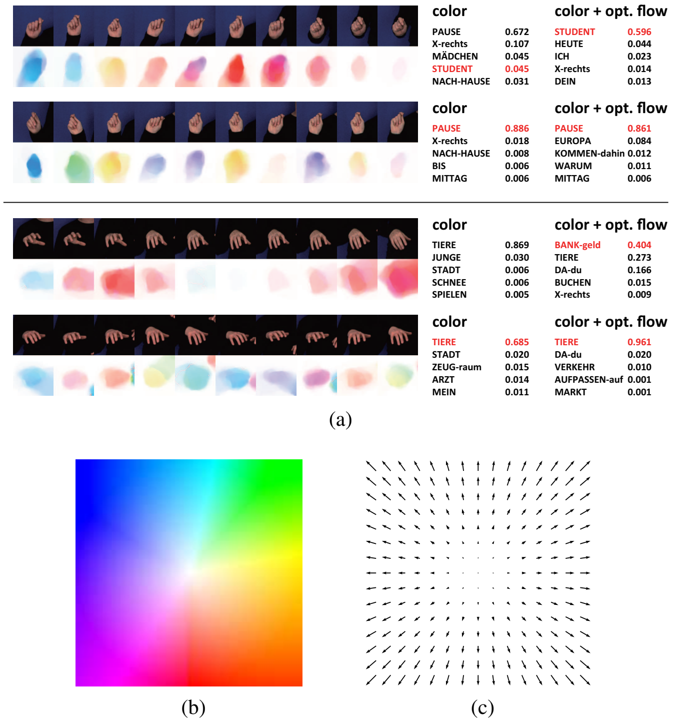


Fig. 7. Comparison of gesture classifiers trained from alignments with different modalities. (a) Classification results on some unseen gestural sequences. We list top-5 predictions of classifiers trained with color modality and multimodal fusion respectively, and annotate the ground truth label for each sequence in red color. We see that both classifiers can give correct labels on "PAUSE" and "TIERE". However, the classifier learning from color modality only, fails to discriminate gestures with similar hand shape but different types of motion, *e.g.* "BANK-geld" and "TIERE". (b)(c) The visualization of flow fields. Orientation and magnitude of flow vectors are represented by hue and saturation respectively.

D. Gesture Classification with Feature Extractor

At the stage of learning from alignments, we use the feature extraction module with a softmax classifier to learn the aligned gestural labels from video segments. To prove the interpretability of this training process, in this section we directly evaluate our trained gesture classifiers on the task of isolated gesture recognition without finetuning.

We evaluate our classifier on the signer-dependent subset of isolated gestures in SIGNUM database, which contains $1,350$ utterances of 450 classes. The isolated gestures are all performed by the same signer as the SL sentences. We use the classifier with temporal convolution and pooling layers to process the image stream, and simply take the mean pooling of the predictions for testing.

It is interesting to find that our classifier with color image modal shows a 63.70% top-1 accuracy and a 86.37% top-5 accuracy, and the classifier with fusion of color and optical flow achieves 75.70% for top-1 accuracy and 91.93% for top-5 accuracy on the 450 gestural classes. This result illustrates that our sequence learning module with bidirectional recurrence can provide reliable alignments between frames and gestural labels, thus we can get decent gesture classifier by training with feature extraction module on this supervision. Notice also that our classifier with multimodal integration presents a higher classification accuracy. In Fig. 7, we can find that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2889563, IEEE Transactions on Multimedia

TABLE II
COMPARISON OF DEV WERS FOR DIFFERENT TEMPORAL CONVOLUTION AND POOLING PARAMETERS ON RWTH-PHOENIX-WEATHER 2014 DATABASE

Temporal Layers	ℓ	δ	Right hand		Full frame	
			color	opt. flow	color	opt. flow
`C5-P2`	6	2	-[1]	-	-	-
`C3-P2-C3-P2`	10	4	35.02±0.04[2]	41.64±0.24	33.93±0.13	45.62±0.36
`C5-P2-C5-P2`	16	4	**32.21±0.22**	**39.88±0.24**	**33.27±0.20**	**44.51±0.26**
`C3-P2-C3-P2-C3-P2`	22	8	37.17±0.23	43.64±0.05	37.15±0.25	46.57±0.25

[1] The network fails to optimize the CTC objective function in this case.
[2] WERs are expressed as percentages, the lower the better.

TABLE III
COMPARISON OF WERS FOR DIFFERENT TEMPORAL CONVOLUTION AND POOLING PARAMETERS ON SIGNUM DATABASE

Temporal Layers	ℓ	δ	Right hand		Full frame	
			color	opt. flow	color	opt. flow
`C5-P2-C5-P2`	10	4	-	-	-	-
`C3-P3-C3-P3`	17	9	7.99±0.17	24.09±0.05	4.66±0.04	7.24±0.25
`C5-P3-C5-P3`	25	9	**6.95±0.12**	**19.79±0.19**	**4.21±0.06**	**6.68±0.12**
`C5-P4-C5-P4`	36	16	10.57±0.05	21.82±0.07	5.33±0.02	8.13±0.11

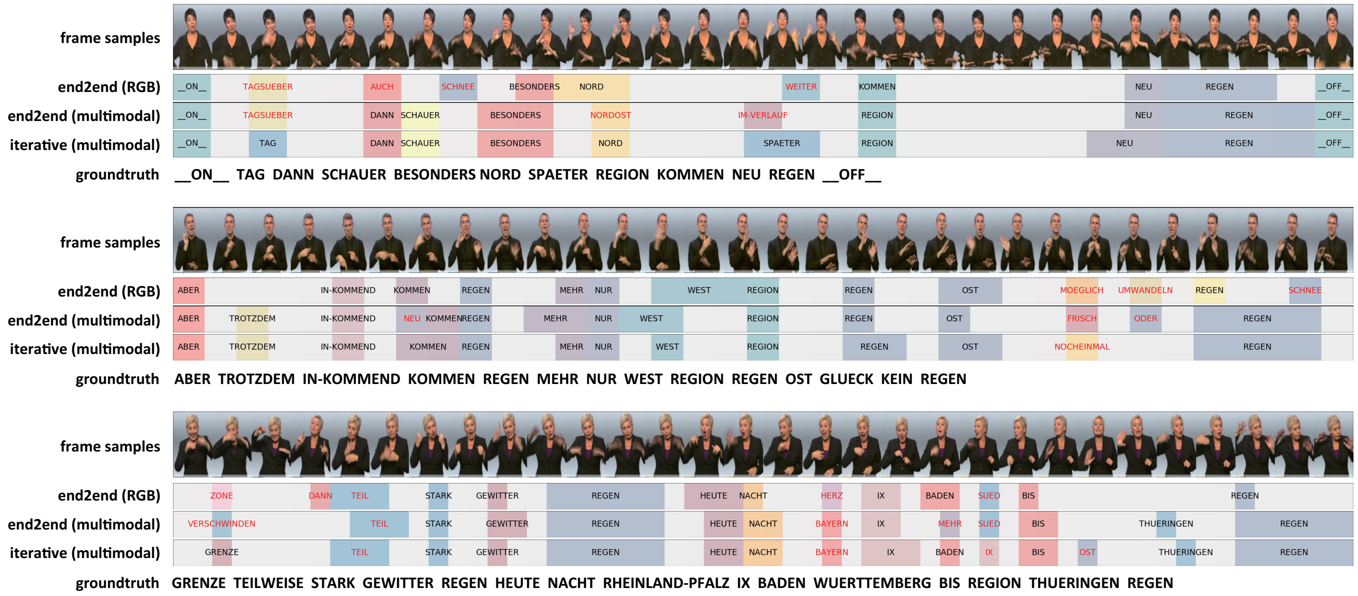


Fig. 6. Recognition results with multiple modalities of full frames on test set. Our architecture with modal fusion and iterative training gives a better prediction of the gestures. Recognition errors of substitutions and insertions are annotated in red.

TABLE VI
RECOGNITION DEV/TEST WERS ON SIGNER INDEPENDENT EXPERIMENT

Iteration	Dev		Test	
	del / ins	WER	del / ins	WER
0	16.48 / 4.17	53.44±0.54	14.40 / 5.87	53.44±0.37
1	13.77 / 3.66	40.76±0.33	13.01 / 4.07	41.28±0.58
2	13.31 / 4.37	40.50±0.22	12.48 / 3.95	39.94±0.40
3	12.57 / 4.51	**39.82±0.15**	11.70 / 4.21	39.63±0.47
4	13.22 / 4.48	39.90±0.26	12.70 / 4.47	40.17±0.25
5	12.25 / 4.94	40.04±0.08	12.12 / 4.86	39.52±0.41
6	11.54 / 4.68	40.07±0.22	11.99 / 5.17	39.49±0.47

classifier learning from complementary modalities, other than color images only, can make more accurate inference. Fig. 8 further shows an example, where our model using multimodal fusion scheme, which has better inference performance on gestural labels, provides a more accurate alignment proposal.

E. Signer Independent Recognition

To evaluate the performance of our approach dealing with inter-signer variations, we present a signer independent experiment for continuous SL recognition in this section. Using the same experimental configurations as SI5 corpus in [22], we remove the video sequences of signer 5 from the training set, which takes 22.85% off from the whole set, and we evaluate our trained recognition system only on sequences of signer 5 in development and test set. We use the alignments from our end-to-end architecture trained on color images of

TABLE IV
RECOGNITION DEV/TEST WERS WITH MULTIPLE MODALITIES ON RWTH-PHOENIX-WEATHER 2014 DATABASE

Input	Modality	Iteration			
		1	2	3	4
Right hand (VGG-S)	color	32.21±0.22 / 32.70±0.11	**31.72±0.08** / 31.51±0.17	31.96±0.02 / 32.15±0.18	-
	opt. flow	39.88±0.24 / 39.94±0.05	**39.62±0.12** / 38.99±0.37	39.62±0.07 / 39.29±0.24	-
	color + opt. flow	31.64±0.06 / 31.40±0.43	**31.07±0.09** / 30.78±0.13	31.36±0.17 / 31.18±0.16	-
Full frame (VGG-S)	color	33.27±0.20 / 33.69±0.22	**32.34±0.18** / 32.30±0.24	32.65±0.23 / 32.71±0.15	-
	opt. flow	44.51±0.26 / 43.31±0.53	**43.76±0.18** / 42.71±0.11	44.22±0.09 / 42.84±0.23	-
	color + opt. flow	32.24±0.09 / 32.53±0.16	**31.56±0.09** / 31.54±0.15	32.12±0.09 / 32.00±0.28	-
Full frame (GoogLeNet)	color	25.96±0.16 / 26.53±0.36	24.40±0.15 / 24.97±0.07	**23.81±0.13** / 24.43±0.16	23.85±0.26 / 24.62±0.32
	opt. flow	39.03±0.21 / 37.96±0.21	**37.88±0.27** / 37.61±0.23	38.15±0.33 / 37.68±0.26	-
	color + opt. flow	24.41±0.35 / 24.25±0.24	**23.10±0.14** / 22.86±0.18	23.38±0.16 / 22.87±0.22	-

TABLE V
RECOGNITION WERS WITH MULTIPLE MODALITIES ON SIGNUM DATABASE

Input	Modality	Iteration					
		1	2	3	4	5	6
Right hand (VGG-S)	color	6.95±0.12	4.74±0.06	**4.66±0.02**	4.67±0.03	-	-
	opt. flow	19.79±0.19	11.00±0.09	9.74±0.02	9.52±0.16	**8.31±0.11**	9.09±0.05
	color + opt. flow	4.66±0.09	4.06±0.03	**3.86±0.02**	4.01±0.03	-	-
Full frame (VGG-S)	color	4.21±0.06	**3.93±0.03**	4.08±0.07	-	-	-
	opt. flow	6.68±0.12	6.51±0.05	**6.20±0.16**	6.50±0.04	-	-
	color + opt. flow	3.17±0.09	2.96±0.07	**2.80±0.08**	2.95±0.12	-	-
Full frame (GoogLeNet)	color	3.96±0.08	**3.58±0.13**	3.75±0.03	-	-	-
	opt. flow	5.17±0.15	**4.72±0.06**	4.74±0.28	-	-	-
	color + opt. flow	3.21±0.04	**2.98±0.12**	3.08±0.19	-	-	-

TABLE VII
PERFORMANCE COMPARISON OF CONTINUOUS SL RECOGNITION APPROACHES ON RWTH-PHOENIX-WEATHER 2014 AND SIGNUM DATABASES

Model	Modality	RWTH-PHOENIX-Weather 2014				SIGNUM	
		Dev		Test		Test	
		del / ins	WER	del / ins	WER	del / ins	WER
v. Agris *et al.* [14]	hands + face	-	-	-	-	-	12.7
Gweth *et al.* [34]	full frame + right hand	-	-	-	-	2.1 / 1.5	11.9
HMM [11]	right hand + face + trajectory	21.8 / 3.9	55.0	20.3 / 4.5	53.0	1.7 / 1.7	10.0
1-Mio-Hands [12]	right hand + face + trajectory	16.3 / 4.6	47.1	15.2 / 4.6	45.1	0.9 / 1.6	7.6
CNN-Hybrid [13]	right hand	12.6 / 5.1	38.3	11.1 / 5.7	38.8	1.4 / 1.4	7.4
Re-Sign [22]	full frame	-	27.1	-	26.8	-	4.8
Ours (VGG-S)	right hand	9.35 / 4.03	31.72±0.08	8.62 / 3.95	31.51±0.17	1.62 / 0.63	4.66±0.02
Ours (VGG-S)	right hand + optical flow	8.77 / 3.72	31.07±0.09	8.47 / 3.25	30.78±0.13	1.09 / 0.60	3.86±0.02
Ours (VGG-S)	full frame	11.06 / 4.23	32.34±0.18	10.32 / 4.00	32.30±0.24	1.75 / 0.32	3.93±0.03
Ours (VGG-S)	full frame + optical flow	9.51 / 4.37	31.56±0.09	9.09 / 4.27	31.54±0.15	1.07 / 0.23	**2.80±0.08**
Ours (GoogLeNet)	full frame	7.83 / 3.48	23.81±0.13	7.79 / 3.37	24.43±0.16	1.52 / 0.43	3.58±0.13
Ours (GoogLeNet)	full frame + optical flow	7.33 / 3.27	**23.10±0.14**	6.73 / 3.29	**22.86±0.18**	1.10 / 0.32	2.98±0.12

right hands in SI5 corpus as the initial alignment proposal. The recognition results after each iteration of optimization are shown in Table VI.

We observe that the iterative training notably improves the performance of the recognition system, with more than 25% relative decrease in WER on both development and test sets. Our architecture achieves WER of 39.82% on development set and 39.63% on test, with no notable decrease in performance on further training iterations. Compared to the results of 45.1% on development and 44.1% on test set reported in [22], our results reduce WER of the state-of-the-art by a margin around 5%, which is a relative improvement of more than 10%.

Note that in experiments of multi-signer configurations, our approach achieves an overall WER of 23.10% on development and 22.86% on test set, and on those sequences of signer 5, the WERs on development and test set are 21.28% and 20.48% respectively. We can observe that the signer independent recognition is much more challenging than that on multi-signer settings. Besides, the reduction in training examples can also partially explain the decreasing performance of the deep neural architecture.

F. Performance Comparison

Table VII compares the performances of our approach and the state-of-the-arts. We can observe that our approach achieves the best performance on both benchmarks. For systems only using RGB images as inputs, our approach outperforms the state-of-the-art method on RWTH-PHOENIX-Weather 2014 database by 2.4% in WER (24.43% vs 26.8%), which is a relative improvement of 9.0%. On SIGNUM bench-

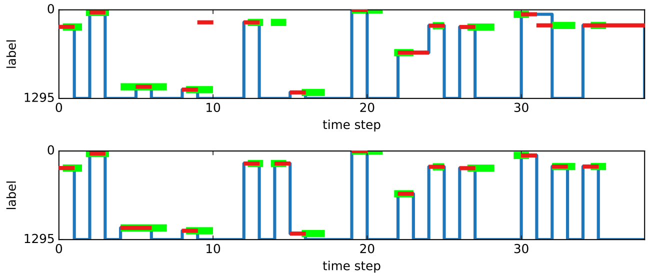


Fig. 8. Alignments for training example #1582 from RWTH-PHOENIX-Weather 2014 database, using color modality (top) and multimodal fusion scheme (bottom). The x-axis represents time steps of the feature sequence after temporal pooling with a stride of 4 frames. The y-axis represents gloss labels, with token #1295 representing "blank". The blue lines are the most probable labels that our networks predict at each time step, the red lines denote the alignment proposal given by our approach, and the green lines show the ground truth alignment manually annotated. Our modal fusion scheme exploits complementary visual cues, and provides a more precise alignment. In contrast, model with color modality fails to make correct inference around time step 15 and 33, resulting in a worse alignment proposal.

mark, our system learning from color modality, with WER of 3.58%, also sees an improvement of 1.2% in contrast to the state-of-the-art with WER of 4.8%. Performance improvements are further gained by introducing multimodal fusion to our framework. The best result on RWTH-PHOENIX-Weather 2014 dataset sees an improvement of WER from 26.8% to 22.86% on test set, and on SIGNUM corpus, our system reduces WER of the state-of-the-art from 4.8% to 2.80%.

VI. Conclusion

In this paper, we develop a continuous SL recognition system with recurrent convolutional neural networks on multimodal data of RGB frames and optical flow images. In contrast to previous state-of-the-art methods, our framework employs recurrent neural networks as the sequence learning module, which shows a superior capability of learning temporal dependencies compared to HMMs. The scale of training data is the bottleneck in fully training a deep neural network of high complexity on this task. To alleviate this problem, we propose a novel training scheme to make our feature extraction module fully exploited to learn the relevant gestural labels on video segments, and keep on benefitting from the iteratively refined alignment proposals. We develop a multimodal fusion approach to integrate appearance and motion cues from SL videos, which presents better spatiotemporal representations for gestures. We evaluate our model on two publicly available SL recognition benchmarks, and experimental results present the effectiveness of our method, where both the iterative training strategy and the multimodal fusion contribute to a better representation and the performance improvements.

There are several directions for future work. First, since gestures consist of simultaneous related channels of information, the integration approach of multiple modalities needs more exploration. It would also be interesting to exploit prior knowledge on sign language, such as subunits, to help the model learn from the limited data. Another promising research path is to develop other sequence learning approach, such as attention-based methods, to make better use of the temporal dependencies.

References

[1] S. C. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 873–891, 2005.

[2] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[3] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with Kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, 2015.

[4] 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. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4207–4215.

[5] H. Cooper, E. J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," *J. Mach. Learning Research*, vol. 13, pp. 2205–2231, 2012.

[6] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2015, pp. 1–7.

[7] G. Zen, L. Porzi, E. Sangineto, E. Ricci, and N. Sebe, "Learning personalized models for facial expression analysis and gesture recognition," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 775–788, 2016.

[8] G. D. Evangelidis, G. Singh, and R. Horaud, "Continuous gesture recognition from articulated poses," in *Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 595–607.

[9] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 474–490.

[10] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, 2016.

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

[12] 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. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3793–3802.

[13] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: hybrid CNN-HMM for continuous sign language recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016.

[14] U. Von Agris, M. Knorr, and K.-F. Kraiss, "The significance of facial features for automatic sign language recognition," in *8th IEEE Int. Conf. Autom. Face Gesture Recog.*, 2008, pp. 1–6.

[15] P. Buehler, A. Zisserman, and M. Everingham, "Learning sign language by watching TV (using weakly aligned subtitles)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2961–2968.

[16] L.-C. Wang, R. Wang, D.-H. Kong, and B.-C. Yin, "Similarity assessment model for Chinese sign language videos," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 751–761, 2014.

[17] C. Monnier, S. German, and A. Ost, "A multi-scale boosted detector for efficient and robust gesture recognition," in *Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 491–502.

[18] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2625–2634.

[19] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[20] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream CNN," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, 2017.

[21] 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," *Int. J. Comput. Vis.*, pp. 1–10, 2015.

[22] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[23] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 814–829.

[24] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 724–731.

[25] O. Koller, H. Ney, and R. Bowden, "Automatic alignment of HamNoSys subunits for continuous sign language recognition," in *Int. Conf. Language Resources and Evaluation Workshops*, 2016, pp. 121–128.

[26] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 572–578.

[27] T. Pfister, J. Charles, and A. Zisserman, "Large-scale learning of sign language by watching TV (using co-occurrences)," in *Proc. Brit. Mach. Vis. Conf.*, 2013.

[28] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learning*, 2014, pp. 1764–1772.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learning Representations*, 2014.

[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[31] A. Graves, S. Fernàndez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learning*, 2006, pp. 369–376.

[32] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 577–584.

[33] J. S. Chung and A. Zisserman, "Signs in time: Encoding human motion as a temporal image," in *Eur. Conf. Comput. Vis. Workshop on Brave New Ideas for Motion Representations*, 2016.

[34] Y. L. Gweth, C. Plahl, and H. Ney, "Enhanced continuous sign language recognition using pca and neural network features," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2012, pp. 55–60.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[36] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[37] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

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

[39] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014.

[40] 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," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[41] 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. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[42] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1933–1941.

[43] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1385–1392.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[46] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2015.

[47] Theano Development Team, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[48] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-Weather," in *Int. Conf. Language Resources and Evaluation*, 2014, pp. 1911–1916.

Runpeng Cui received his B.S. degree from Tsinghua University, Beijing, China, in 2013, He is currently a Ph.D. student at the State Key Laboratory of Intelligent Technology and Systems, Department of Automation, Tsinghua University, Beijing, China. His research interests include deep learning and computer vision.

Hu Liu received his B.S. degree from Tsinghua University, Beijing, China, in 2015, He is currently a Master's student at the State Key Laboratory of Intelligent Technology and Systems, Department of Automation, Tsinghua University, Beijing, China. His research interests include deep learning and computer vision.

Changshui Zhang received his B.S. degree from Peking University, Beijing, China, in 1986, and Ph.D. degree from Tsinghua University, Beijing, China, in 1992. He is currently a professor of Department of Automation, Tsinghua University. His research interests include pattern recognition and machine learning. He became a member of IEEE in 2002, IEEE Senior Member in 2015, and IEEE Fellow in 2018.