Video-based Sign Language Recognition without Temporal Segmentation

Jie Huang[1], Wengang Zhou[2], Qilin Zhang[3], Houqiang Li[4], Weiping Li[5]

[1,2,4,5]Department of Electronic Engineering and Information Science, University of Science and Technology of China
[3]HERE Technologies, Chicago, Illinois, USA
[1]hagjie@mail.ustc.edu.cn, {[2]zhwg,[4]lihq,[5]wpli}@ustc.edu.cn, [3]samqzhang@gmail.com

Abstract

Millions of hearing impaired people around the world routinely use some variants of sign languages to communicate, thus the automatic translation of a sign language is meaningful and important. Currently, there are two sub-problems in Sign Language Recognition (SLR), i.e., isolated SLR that recognizes word by word and continuous SLR that translates entire sentences. Existing continuous SLR methods typically utilize isolated SLRs as building blocks, with an extra layer of preprocessing (temporal segmentation) and another layer of post-processing (sentence synthesis). Unfortunately, temporal segmentation itself is non-trivial and inevitably propagates errors into subsequent steps. Worse still, isolated SLR methods typically require strenuous labeling of each word separately in a sentence, severely limiting the amount of attainable training data. To address these challenges, we propose a novel continuous sign recognition framework, the Hierarchical Attention Network with Latent Space (LS-HAN), which eliminates the preprocessing of temporal segmentation. The proposed LS-HAN consists of three components: a two-stream Convolutional Neural Network (CNN) for video feature representation generation, a Latent Space (LS) for semantic gap bridging, and a Hierarchical Attention Network (HAN) for latent space based recognition. Experiments are carried out on two large scale datasets. Experimental results demonstrate the effectiveness of the proposed framework.

Introduction

A key challenge in Sign Language Recognition (SLR) is the design of visual descriptors that reliably captures body motions, gestures, and facial expressions. There are primarily two categories: the hand-crafted features (Sun et al. 2013; Koller, Forster, and Ney 2015) and Convolutional Neural Network (CNN) based features (Tang et al. 2015; Huang et al. 2015; Pu, Zhou, and Li 2016). Inspired by the recent success in CNN (Ran et al. 2017a; 2017b), we design a two-stream 3D-CNN for video feature extraction.

Temporal segmentation is also a difficulty in continuous Sign Language Recognition (SLR). The common scheme to continuous SLR is to decompose it to isolated word recognition problem, which involves temporal segmentation. Temporal segmentation is non-trivial since the transitional movements are diverse and hard to detect, and as a preprocessing

step, inaccurate segmentation could incur errors in the subsequent steps. In addition, it is highly time consuming to label each isolated fragments.

Motivated by video caption with Long-Short Term Memory (LSTM), we circumvent the temporal segmentation with Hierarchical Attention Network (HAN), an extension to LSTM by considering structure information and attention mechanism. The scheme is to feed HAN with the entire video and output complete sentence word-by-word. However, HAN locally optimizes the probability of generating the next word given the input video and previous word, ignoring the relationship between video and sentences (Pan et al. 2015). As a result, it could suffer from robustness issues. To remedy this, we incorporate a Latent Space model to explicitly exploit the relationship between visual video and text sentence.

In summary, the major contributions of the paper are:
- A new two-stream 3D CNN for the generation of global-local video feature representations;
- A new LS-HAN framework for continuous SLR without requiring temporal segmentation;
- Joint optimization of relevance and recognition loss in the proposed LS-HAN framework;
- Compilation of the largest (as of September 2017) open-source Modern Chinese Sign Language (CSL) dataset for continuous SLR with sentence-level annotations.

Related Work

In this section, a brief review of continuous SLR, video subtitle generation, and latent space model is given.

Continuous SLR

Most existing SLR researches (Huang et al. 2015; Guo et al. 2016; Dan Guo and Wang 2017; Liu, Zhou, and Li 2016) fall into the category of isolated SLR, i.e., recognition of words or expressions, similar to action recognition (Cai et al. 2016). A more challenging problem is the continuous SLR, which involves the reconstruction of sentence structures. Most existing continuous SLR methods divide the sentence-to-sentence recognition problem into three stages, temporal segmentation of videos, isolated word/expression recognition (i.e., isolated SLR), and sentence synthesis with a language model. For example, DTW-HMM (Zhang, Zhou,

and Li 2014) proposed a threshold matrix based coarse temporal segmentation step followed by a Dynamic Time Warping (DTW) algorithm and a bi-grammar model. In (Koller, Zargaran, and Ney 2017), a new HMM based language model is incorporated. Recently, transitional movements attract a lot of attention (Dawod, Nordin, and Abdullah 2016; Li, Zhou, and Lee 2016; Yang, Tao, and Ye 2016; Zhang et al. 2016) because they can serve as the basis for temporal segmentation.

Despite its popularity, temporal segmentation is intrinsically difficult: even the transitional movements between hand gestures can be subtle and ambiguous. Inaccurate segmentation can incur significant performance penalty on subsequent steps (Zhang, Zhou, and Li 2014; Koller, Forster, and Ney 2015; Fang, Gao, and Zhao 2007). Worse still, the isolated SLR step typically requires per-video-frame labels, which are highly time consuming.

Video Description Generation

Video description generation (Pan et al. 2015; Venugopalan et al. 2015; Yao et al. 2015) is a relevant research area, which generates a brief sentence describing the scenes/objects/motions of a given video sequences. One popular method is the sequence-to-sequence video-to-text (Venugopalan et al. 2015), a two layer LSTM on top of a CNN. Attention mechanism can be incorporated into a LSTM (Yao et al. 2015), which automatically selects the most likely video frames. There are also some extensions to the LSTM-style algorithms, such as bidirectional LSTM (Bin et al. 2016), hierarchical LSTM (Li, Luong, and Jurafsky 2015) and hierarchical attention GRU (Yang et al. 2016).

Despite many similarities in targets and involved techniques, video description generation and continuous SLR are two distinctive tasks. Video description generation provides a brief summary of the appearance of a video sequence, such as "A female person raises a hand and stretch fingers" while the continuous SLR provides a semantic translation of sign language sentences, such as "A signer says 'I love you' in American Sign Language. "

Latent Space based Learning

Latent space model is a popular tool to bridge the semantic gap between two modalities (Zhang et al. 2015b; 2015a; Zhang and Hua 2015), such as between textual short descriptions and imagery (Pan et al. 2014). For example, the combination of LSTM and latent space model is proposed in (Pan et al. 2015) for video caption generation, in which both an embedding model and an LSTM are jointly learned. However, this embedding model is purely based on the Euclidean distance between video frames and textual descriptions, ignoring all temporal structures. It is suitable for the generation of video appearance summaries, but falls short of semantic translation.

Signing Video Feature Representation

Signing video is characterized primarily by upper body motions, especially hand gestures. The major challenge of hand gesture detection and tracking is the tremendous appearance variations of hand shapes and orientations, in addition to occlusions. Previous researches (Tang et al. 2015; Kurakin, Zhang, and Liu 2012; Sun et al. 2013) on posture/gesture models rely heavily on collected depth maps from Kinect-style RGB-D sensors, which provides a convenient way in 3D modeling. However, a vast majority of existing annotated signing videos are recorded with conventional RGB camcorders. An effective RGB video based feature representation is essential to take advantage of these valuable historical labeled data.

Inspired by the recent success of deep learning based object detection, we propose a two-stream 3-D CNN for the generation of video feature representation, with accurate gesture detection and tracking. This particular two-stream 3-D CNN takes both the entire video frames and cropped/tracked gesture image patches as two separate inputs for each stream, with a late fusion mechanism achieved through shared final fully connected layers. Therefore, the proposed CNN encodes both global and local information.

Gesture Detection and Tracking

Faster R-CNN (Girshick 2015) is a popular object detection method, which is adopted in our proposed method for gesture detection and tracking. We pre-train a faster R-CNN on the VOC2007[1] person-layout dataset[2] with two output units representing hands versus background. Subsequently, we randomly select 400 frames from our proposed CSL dataset and manually annotate the hand locations for fine-tuning. After that, each video is processed frame-by-frame for gesture detection.

The faster R-CNN based detection can fail when the hand-shape varies hugely or is occluded by clothes. To localize gestures in these frames, compressive tracking (Zhang, Zhang, and Yang 2012) is utilized. The compressive tracking model represents target regions in multi-scale compressive feature vectors and scores the proposal regions with a Bayes classifier. The Bayes classifier parameters are updated based on detected targets in each frame, the compressive tracking model is robust to huge appearance variations. Specifically, a compressive tracking model is initialized whenever the faster R-CNN detection fails, with the successfully detections from the immediate prior video frame. Conventional tracking algorithms often suffer from the drift problem, especially with long video sequences. Our proposed method is largely immune to this problem due to limited length of sign videos.

Two-stream 3D CNN

Due to the nature of signing videos, a robust video feature representation requires the incorporation of both global hand locations/motions and local hand gestures. As shown in Fig. 1, we design a two-stream 3-D CNN based on the C3D (Tran et al. 2015), which extracts spatio-temporal features from a video clip (16 frames as suggested in (Tran et

---

[1]http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html

[2]It contains detailed body part labels (*e.g.*, head, hands and feet).
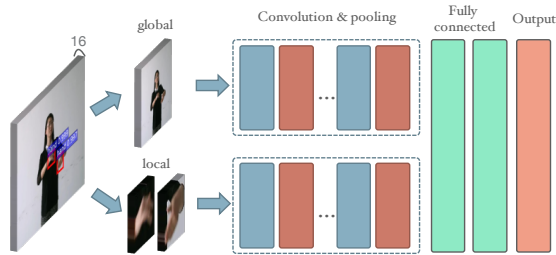
Figure 1: Two-stream 3-D CNN. Input is a video clip containing 16 adjacent frames. Two streams share the same structure as C3D (Tran et al. 2015). Blue and red blocks represent convolutional and pooling layers, respectively. Global-local feature representations are fused in the rightmost two fully connected layers.

al. 2015)). The input to the network is a video clip containing adjacent 16 frames. The upper stream is designed to extract global hand locations/motions, where the input is resized (227×227) complete video frames. The lower stream focuses on the local, detailed hand gestures, where the input is cropped (also 227×227), tracked image patches containing tight bounding boxes of hands. Left and right hand patches are concatenated as multi-channel inputs. Each stream shares the same network structure as the C3D network, including eight convolutional layers and five pooling layers. Two fully connected layers serve as fusion layers to combine global and local information from the upper and lower stream, respectively.

The two-stream CNN is first pre-trained with an isolated SLR dataset (Pu, Zhou, and Li 2016), after which all weights are fixed and the last SoftMax and fully-connected layer are discarded. During the test phase, this CNN is truncated at the first fully connected layer (as a feature extractor) and each test video is divided into 16-frame clips with a temporal sliding window and subsequently fed into the two-steam 3D CNN. The 4096 dimensional output of the truncated 3D CNN is the desired global-local feature representation for this particular 16-frame video clip. Consequently, each video is denoted as a sequence of such 4096 dimensional feature vectors.

Proposed LS-HAN Model

In this section, we present the Hierarchical Attention Network (HAN) for continuous SLR in a latent space. HAN is an extension to LSTM, which incorporates the attention mechanism based on the structure of input. In the proposed joint learning model, the optimization function takes into account both the video-sentence relevance error E_r in a latent space, and a recognition error E_c by HAN,

$$\min_{\theta_r, \theta_c} \frac{1}{N} \sum_{i=1}^{N} \lambda_1 E_r(V^{(i)}, S^{(i)}; \theta_r) \\ + (1-\lambda_1) E_c(V^{(i)}, S^{(i)}; \theta_r, \theta_c) + \lambda_2 R \, , \quad (1)$$
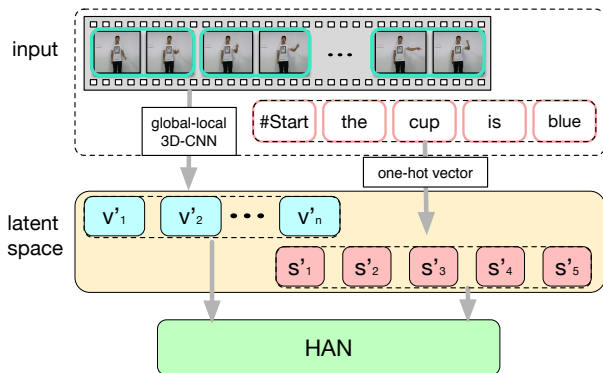

Figure 2: The proposed LS-HAN framework. Input are video paired with annotation sentence. Video is represented with global-local features and each word is encoded with one-hot vector. They are mapped into the same latent space to model video-sentence relevance. Based on the mapping results, we utilize HAN for automatic sentence generation.

where N denotes the number of instances in the training set, and i_{th} instance being a video with annotated sentence $(V^{(i)}, S^{(i)})$. θ_r and θ_c denote parameters in the latent space and HAN, respectively. R is a regularization term. Eq. (1) represents the minimization of the mean loss over training data with some regularizations. The balance between the loss term and the regularization term is achieved by weights λ_1 and λ_2.

In the following, the term E_r is first formulated in the video-sentence latent space, followed by details on sentence recognition with HAN and the formula of term E_c. Finally, the overall training and testing processes of LS-HAN are presented.

Video-sentence Latent Space

As shown in Fig. 2, the input to our framework is videos with paired annotated sentences. Video is represented with the proposed global-local features, while each word of the annotated sentence is encoded with a "one-hot" vector[3].

Let the videos be $V = (v_1, v_2, ..., v_n)$, and the sentence be $S = (s_1, s_2, ..., s_m)$, where $v_i \in \mathbb{R}^{D_c}$ is the D_c-dimensional global-local feature of the i_{th} video clip. n denotes the total number of clips in this video, $s_j \in \mathbb{R}^{D_w}$ is the "one-hot" vector of j_{th} word of sentence, D_w is vocabulary size and m represents the total number of words in the current sentence.

The goal of the latent space is to construct a space to bridge semantic gaps, we map $V \in \mathbb{R}^{D_c \times n}$ and $S \in \mathbb{R}^{D_w \times m}$ to the same latent space as $f_v(V) = (v'_1, v'_2, ..., v'_n)$ and $f_s(S) = (s'_1, s'_2, ..., s'_m)$, where f_v and f_s are mapping function for video features and sentence features, respectively,

$$f_v(x) = T_v x \quad \text{and} \quad f_s(x) = T_s x \, , \quad (2)$$

---

[3] A one-hot vector is a binary index used to distinguish words with a given vocabulary. All bits are '0', except the ith bit being '1' for the ith word in this vocabulary.
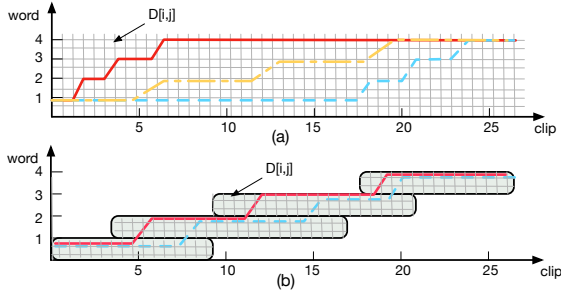
Figure 3: Associated warping paths generated by DTW. X-axis represents the frame index, y-axis indicates the word sequence index. Grids represent the matrix elements $D[i,j]$. (a) shows three possible alignment paths of raw DTW. (b) shows the alignment paths of Window-DTW which are bounded by windows.

$T_v \in \mathbb{R}^{D_s \times D_c}$ and $T_s \in \mathbb{R}^{D_s \times D_w}$ are transformation matrices that project video content and semantic sentence into common space. D_s is the dimension of the latent space.

To measure the relevance between $f_v(V)$ and $f_s(S)$, the Dynamic Time Warping (DTW) algorithm is used to find the minimal accumulating distance of two sequences and the temporal warping path,

$$D[i,j] = min(D[i-1,j], D[i-1,j-1]) + d(i,j) , \quad (3)$$

$$d(i,j) = ||T_v v_i - T_s s_j||_2 , \quad (4)$$

where $D[i,j]$ denotes the distance between $(v'_1, ..., v'_i)$ and $(s'_1, ..., s'_j)$, and $d(i,j)$ denotes the distance between v'_i and s'_j. Thus we define the loss of video-sentence with single instance as,

$$E_r(V, S; \theta_r) = D(n, m) . \quad (5)$$

This DTW algorithm assumes that words appearing earlier in a sentence should also appear early in video clips, i.e., a monotonically increasing alignment. Per our evaluation dataset, most signers vastly prefer simple sentences to compound ones, thus simple sentences with a single clause makes up the majority of the dataset. Therefore, approximate monotonically increasing alignment can be assumed.

The associated warping path is recovered using backtracking. This warping path is normally interpreted as the alignment between frames and words. For better alignment, the Windowing-DTW (Biba and Xhafa 2011) is used, as shown in Fig. 3, and $D[i,j]$ is only computed within the windows. In the following experiments, the window length is $\frac{n}{2}$, except the first and last one at boundary 0 and n. All adjacent windows have $\frac{n}{4}$ overlap.

Recognition with HAN

Inspired by recent sequence-to-sequence models (Venugopalan et al. 2015), the recognition problem is formulated as the estimation of log-conditional probability of sentences
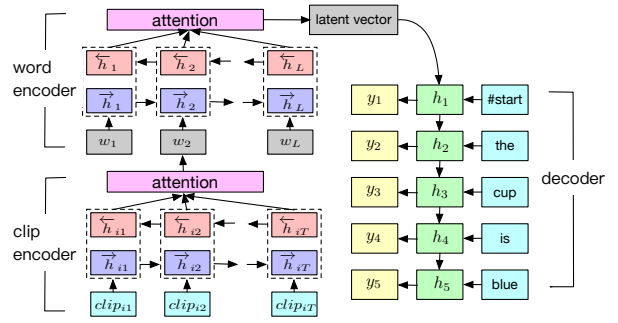

Figure 4: HAN encodes videos hierarchically and weights the input sequence via a attention layer. It decodes the hidden vector representation to a sentence word-by-word.

given videos. By minimizing the loss, contextual relationship among the words in sentences can be kept.

$$E_c(V, S; \theta_r, \theta_c) = \log p(s'_1, s'_2, ..., s'_m | v'_1, v'_2, ..., v'_n; \theta_c)$$
$$= \sum_{t=1}^{m} \log p(s'_t | v'_1, ..., v'_n, s'_1, ..., s'_{t-1}; \theta_c) . \quad (6)$$

First, input frame sequences are encoded as a latent vector representation on a per-frame basis, followed by decoding from each representation to a sentence, one word at a time. We extend the encoder in HAN to reflect the hierarchical structures in its inputs (clips form word and words form sentence) and incorporate the attention mechanism. The modified model is an unnamed variant of HAN.

As shown in Fig. 4, the inputs (in blue) are clip sequences and word sequences represented in latent space. This model contains two encoders and a decoder. Each encoder is a bidirectional LSTM with a attention layer; while the decoder is a single LSTM. The clip encoder encodes the video clips aligning to a word. As shown in Fig. 5, we empirically try three alignment strategies:

1. Split clips to two subsequences;
2. Split to subsequences every two clips;
3. Split evenly to 7 subsequences, since each sentence contains 7 words on average in the training set.

The outputs of bidirectional LSTM pass through the attention layer to form a word-level vector, where the attention layer acts as information selecting weights to the inputs. Subsequently, the sequences of word-level vectors are projected into a latent vector representation, where decoding is carried out with annotation sentence. During the decoding, $\#Start$ is used as the start symbol to mark the beginning of a sentence and $\#End$ as the end symbol that indicates the end of a sentence. At each time stamp, words encoded with "one-hot vector" are mapped into the latent space and fed to the LSTM (denoted in green) together with the hidden state from the previous timestamp. The LSTM cell output h_t is used to emitted word y_t. A softmax function is applied to obtain the probability distribution over the words $\hat{y}$ in the
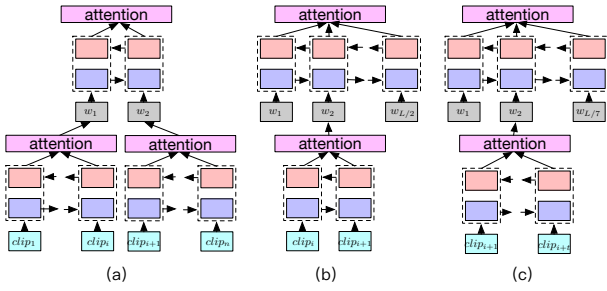
Figure 5: Alignment reconstruction in the testing phase. (a) Split clips to two subsequences and encode as HAN; (b) Split to subsequences every two adjacent clips; (c) Split evenly to 7 subsequences, where 7 is average sentence length in the training set.

vocabulary voc.

$$p(y_t|h_t) = \frac{\exp(W_{y_t}h_t)}{\sum_{\hat{y} \in voc} \exp(W_{\hat{y}}h_t)}, \qquad (7)$$

where $W_{y_t}, W_{\hat{y}}$ are parameter vectors in the softmax layer. All next words are obtained based on the probability in Eq. (7) until the sentence end symbol is emitted.

Inspired by (Yang et al. 2016), the coherence loss in Eq. (6) can be further simplified as,

$$
\begin{aligned}
E_c(V, S; \theta_r, \theta_c) &= \sum_{t=1}^{m} \log p(s'_t|h_{t-1}, s'_{t-1}) \\
&= \sum_{t=1}^{m} \log \frac{\exp(W_{s_t}h_t)}{\sum_{\hat{s}' \in voc} \exp(W_{\hat{s}'}h_t)} .
\end{aligned}
\qquad (8)
$$

Learning and Recognition of LS-HAN Model

With previous formulations of the latent space and HAN, Eq. (1) is equivalent to,

$$
\begin{aligned}
\min_{T_v, T_s, \theta} \frac{1}{N} \sum_{i=1}^{N} &\lambda_1 E_r(V^{(i)}, S^{(i)}; T_v, T_s) \\
&+ (1 - \lambda_1) E_c(V^{(i)}, S^{(i)}; T_v, T_s, \theta) \\
&+ \lambda_2 (\|T_v\|^2 + \|T_s\|^2 + \|\theta\|^2) ,
\end{aligned}
\qquad (9)
$$

where T_v and T_s are latent space parameters, θ is a HAN parameter. The optimization of LS-HAN requires the partial derivatives of Eq. (9) with respect to parameters T_v, T_s and θ and the solutions of such equations. Thanks to the linearity, the two loss terms and the regularization item could be optimized separately. For simplicity, we only elaborate the computation of partial derivatives of E_r and E_c in the simplest case (with only one sample), ignoring unrelated coefficients and regularization terms.

Consider the partial derivative of E_r with respect to matrices T_v and T_s. According to Eq. (5),

$$\frac{\partial E_r(V, S; T_v, T_s)}{\partial T_v} = \frac{\partial D(n, m)}{\partial T_v} , \qquad (10)$$

Substitution of Eq. (3) in Eq. (10) leads to

$$
\begin{aligned}
\frac{\partial D(n, m)}{\partial T_v} = &\min(\frac{\partial D(n-1, m)}{\partial T_v}, \frac{\partial D(n-1, m-1))}{\partial T_v}) \\
&+ \frac{\partial d(n, m)}{\partial T_v}.
\end{aligned}
\qquad (11)
$$

Eq. (11) is recursive and allows for gradient back-propagation through time. As a result, $\frac{\partial D(n,m)}{\partial T_v}$ can be represented by $\frac{\partial d(i,j)}{\partial T_v}$ for $i < n, j < m$,

$$\frac{\partial D(n, m)}{\partial T_v} = f\left(\left\{\frac{\partial d(i, j)}{\partial T_v}| i < n, j < m\right\}\right) , \qquad (12)$$

where $f(\cdot)$ denotes a function that can be represented. Since the term $\frac{\partial d(i,j)}{\partial T_v}$ can be obtained according to Eq. (4), $\frac{\partial D(n,m)}{\partial T_v}$ can be computed by the chain rule. Likewise, $\frac{\partial E_r(V,S;T_v,T_s)}{\partial T_s}$ could be similarly computed.

Consider the partial derivative of HAN error E_c with respect to T_v, T_s and the network parameter θ. Regular stochastic gradient descent is utilized, with gradients calculated by the Back-propagation Through Time (BPTT) (Werbos 1990), which recursively back-propagates gradients from current to previous timestamps. After gradients are propagated back to the inputs, $\frac{\partial E_c(V,S;T_v,T_s,\theta)}{\partial V'}$ and $\frac{\partial E_c(V,S;T_v,T_s,\theta)}{\partial S'}$ are obtained. With Eq. (2), further simplification could be obtained,

$$\frac{\partial E_c(V, S; T_v, T_s, \theta)}{\partial T_v} = \frac{\partial E_c(V, S; T_v, T_s, \theta)}{\partial V'} \cdot V , \qquad (13)$$

$$\frac{\partial E_c(V, S; T_v, T_s, \theta)}{\partial T_s} = \frac{\partial E_c(V, S; T_v, T_s, \theta)}{\partial S'} \cdot S . \qquad (14)$$

Therefore, all gradients with respect to all parameters could be obtained. The objective function in Eq. (9) can be minimized accordingly.

During the testing phase, the proposed LS-HAN is used to translate signing videos sentence-by-sentence. Each video is divided into clips with a sliding window algorithm. The alignment information need to be reconstructed. Empirical experiments verify that the strategy 3 outperforms others, therefore it is employed in the testing phase. After encoding, the start symbol "$\#Start$" is fed to HAN indicating the beginning of sentence prediction. During each decoding timestamp, the word with the highest probability after the softmax is chosen as the predicted word, with its representation in the latent space fed to HAN for the next timestamp, until the emission of the end symbol "$\#End$".

Experiments

In this section, datasets are introduced followed by evaluation comparison. Additionally, a sensitivity analysis on a trade-off parameter is included.

Datasets

Two open source continuous SLR datasets are used in the following experiments, one for CSL and the other is the

Table 1: Statistics on Proposed CSL Video Dataset

RGB resolution	1920×1080	# of signers	50
Depth resolution	512×424	Vocab. size	178
Video duration (sec.)	10~14	Body joints	21
Average words/instance	7	FPS	25
Total instances	25,000	Total hours	100+

German sign language dataset RWTH-PHOENIX-Weather (Koller, Forster, and Ney 2015). The CSL dataset in Tab. 1 is collected by us and released on our project web page[4]. A Microsoft Kinect camera is used for all recording, providing RGB, depth and body joints modalities in all videos. The additional modalities should provide helpful additional information as proven in hyper-spectral imaging efforts (Ran et al. 2017a; Zhang et al. 2011; 2012; Abeida et al. 2013), which is potentially helpful in future works. In this paper, only the RGB modality is used. The CSL dataset contains 25K labeled video instances, with $100+$ hours of total video footage by 50 signers. Every video instance is annotated with a complete sentence by a professional CSL teacher. 17K instances are selected for training, 2K for validation, and the rest 6K for testing. The RWTH-PHOENIX-Weather dataset contains 7K weather forecasts sentences from 9 signers. All videos are of 25 frames per second (FPS) and at resolution of 210×260. Following (Koller, Forster, and Ney 2015), 5,672 instances are used for training, 540 for validation, and 629 for testing.

Experimental Setting

Per (Tran et al. 2015), videos are divided into 16-frame clips with 50% overlap, with frames cropped and resized at 227×227. The outputs of the 4096-dimensinoal $fc6$ layer from 2-stream 3D CNN are clip representations. The following parameters are set based on our validation set. The dimension of latent space and the size of hidden layer in HAN are both 1024. The trade-off parameter λ_1 in Eq. (9) of relevance loss and coherence loss is set to 0.6. The regularization parameter is empirically set to 0.0002.

Evaluation Metrics

Predicted sentence can suffer from errors including word substitution, insertion and deletion errors, following (Fang, Gao, and Zhao 2007; Starner, Weaver, and Pentland 1998; Zhang, Zhou, and Li 2014),

$$Accuracy = 1 - \frac{S + I + D}{N} \times 100\% , \qquad (15)$$

where S, I, and D denote the minimum number of substitution, insertion and deletion operations needed to transform a hypothesized sentence to the ground truth. N is the number of words in ground truth. Note that, since all errors are accounted against the accuracy rate, it is possible to get negative accuracies.

Table 2: Continuous SLR Results. Methods in bold text are the original and modified versions of the proposed method.

Methods	Accuracy
LSTM (Venugopalan et al. 2014)	0.736
S2VT (Venugopalan et al. 2015)	0.745
LSTM-A (Yao et al. 2015)	0.757
LSTM-E (Pan et al. 2015)	0.768
HAN (Yang et al. 2016)	0.793
CRF (Lafferty, McCallum, and Pereira)	0.686
LDCRF (Morency, Quattoni, and Darrell 2007)	0.704
DTW-HMM (Zhang, Zhou, and Li 2014)	0.716
LS-HAN (a)	0.792
LS-HAN (b)	0.805
LS-HAN (c)	**0.827**

Results and Analyses

The continuous SLR results over our CSL dataset is summarized in Tab. 2. Since our framework is based on LSTM, we compare with LSTM, S2VT, LSTM-A, LSTM-E and HAN, which are also related to LSTM. Note that LSTM-E that jointly learns LSTM and embedding layer is much similar to our method. One of the key differences is that LSTM-E ignores temporal information for brevity during its embedding process, while we choose to retain temporal structural information while optimizing video-sentence correspondence. Given the result that our method achieve 0.059 higher accuracy than LSTM-E, our solution to model video-sentence relevance is a better choice.

Besides, we also make a comparison with continuous SLR algorithms: CRF, LDCRF and DTW-HMM. These models require segmentation when do recognition, which may incur and propagate the inaccuracy. Our method obtains 0.141, 0.123 and 0.111 higher accuracy, respectively, showing the advantage of circumventing the temporal segmentation.

As presented at the bottom of Tab. 2, we test the schemes that makes up the missing of alignment during recognition, as mention in previous subsection *Learning and Recognition of LS-HAN Model*. We see that LS-HAN with scheme (c) achieves the highest accuracy. Although this alignment is not the true alignment result, we still suppose this scheme is feasible for our model considering the significant results.

Tab. 3 shows a comparison of our result with the recently published results. For a fair comparison, we only utilize hands sequences, as with all competing methods. Both Deep Hand and Recurrent CNN are extensions to CNN. The former combines CNN with an EM algorithm, while the latter proposes the RNN-CNN, which takes advantage of both the feature learning capability of CNN and the temporal sequence modeling capability of the iterative EM and RNN. Our approach shares a similar idea but goes a step further, which bridges semantic gap with a latent space, then applies HAN to hypothesize semantic sentence. The presented LS-HAN outperforms both Deep Hand and Recurrent CNN.

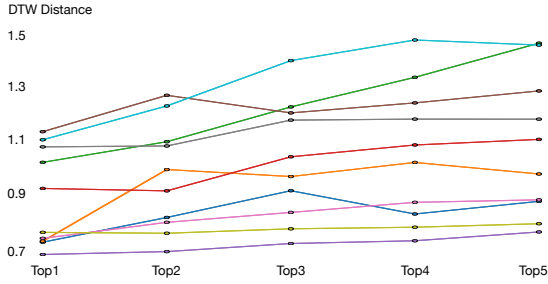[4]http://mccipc.ustc.edu.cn/mediawiki/index.php/SLR_Dataset

Figure 6: LSTM-based and Latent Space similarity measure comparison.

Table 3: Continuous SLR on RWTH-PHOENIX-Weather.

Methods	Accuracy
(Koller, Forster, and Ney 2015)	0.444
Deep Hand (Koller, Ney, and Bowden 2016)	0.549
Recurrent CNN (Cui, Liu, and Zhang 2017)	0.613
LS-HAN (only hand sequence)	0.617

Relationship Between HAN and LS

In the proposed LS-HAN, video-sentence correlations are indicated by both the most likely sentence outputs from HAN and the distance metrics in the Latent Space. Theoretically, these two measures should be identical, therefore, an empirical test is carried out and the results are presented in Fig. 6. 10 video sequences are randomly selected and fed into the HAN, with each video generating 5 most likely sentences. In Fig. 6, these 5 mostly likely sentences are denoted as vertices on a polyline and sorted with descending probabilities along the X-axis. In addition, the DTW distances in the Latent Space between the video (indicated by the polyline) and each sentences (denoted by the vertices) are visualized along the Y-axis. Theoretically, if these two similarity measures are identical, all polylines in Fig. 6 should be monotonically increasing. Practically there are some noises but the increasing trends are within our expectations.

Sensitivity Analysis on Parameter Selections

The robustness of the proposed LS-HAN with respect to different selections of parameters are summarized in this section. A sample sensitivity analysis on the trade parameter λ_1 in Eq. (9) is presented here and illustrated in Fig. 7. LS-HAN recognition accuracy is evaluated with 2.5k instances in the validation set with varying λ_1 values. Expectedly, extreme λ_1 values lead to high error rate; while the optimal choice is approximately 0.6. Such reasonable sensitivity behaviors also verify the validity of establishing an HAN model in the video-sentence latent space.

Conclusion

In this paper, the LS-HAN framework is proposed for continuous SLR is proposed, which eliminated both the error-prone temporal segmentation and the sentences synthesis in post-processing steps. For video representation, a two-stream 3D CNN generates highly informative global-local
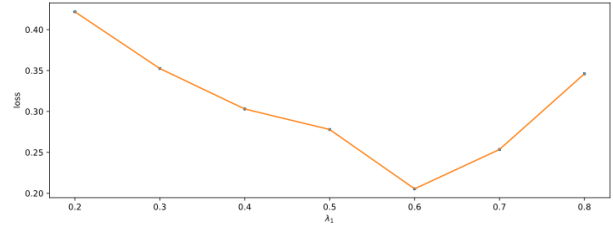

Figure 7: Validation error rate with respect to varying trade-off parameter λ_1 in Eq. (9).

features, with one stream focused on global motion information and the other on local gesture representations. A Latent Space is subsequently introduced via an optimization of labeled video-sentence distance metrics. This latent space captures the temporal structures between signing videos and annotated sentences by aligning frames to words. Our future work could involve the extension of the LS-HAN to longer compound sentences and real-time translation tasks.

Acknowledgments

References

Abeida, H.; Zhang, Q.; Li, J.; and Merabtine, N. 2013. Iterative sparse asymptotic minimum variance based approaches for array processing. *Signal Processing, IEEE Transactions on* 61(4):933–944.

Biba, M., and Xhafa, F. 2011. *Learning Structure and Schemas from Documents*, volume 375. Springer.

Bin, Y.; Yang, Y.; Shen, F.; Xu, X.; and Shen, H. T. 2016. Bidirectional long-short term memory for video description. In *Proceedings of the ACM on Multimedia Conference*, 436–440.

Cai, X.; Zhou, W.; Wu, L.; Luo, J.; and Li, H. 2016. Effective active skeleton representation for low latency human action recognition. *IEEE Transactions on Multimedia* 18(2):141–154.

Cui, R.; Liu, H.; and Zhang, C. 2017. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7361–7369.

Dan Guo, Wengang Zhou, H. L., and Wang, M. 2017. Online early-late fusion based on adaptive hmm for sign language recognition. In *ACM Transactions on Multimedia Computing Communications and Applications*.

Dawod, A. Y.; Nordin, M. J.; and Abdullah, J. 2016. Gesture segmentation: automatic continuous sign language technique based on adaptive contrast stretching approach. *Middle-East Journal of Scientific Research* 24(2):347–352.

Fang, G.; Gao, W.; and Zhao, D. 2007. Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Transactions on Systems, Man and Cybernetics* 37(1):1–9.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of ICCV*, 1440–1448.

Guo, D.; Zhou, W.; Wang, M.; and Li, H. 2016. Sign language recognition based on adaptive hmms with data augmentation. In *IEEE International Conference on Image Processing*, 2876–2880.

Huang, J.; Zhou, W.; Li, H.; and Li, W. 2015. Sign language recognition using 3d convolutional neural networks. In *IEEE International Conference on Multimedia and Expo*, 1–6.

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

Koller, O.; Ney, H.; and Bowden, R. 2016. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3793–3802.

Koller, O.; Zargaran, S.; and Ney, H. 2017. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kurakin, A.; Zhang, Z.; and Liu, Z. 2012. A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference*, 1975–1979. IEEE.

Lafferty, J.; McCallum, A.; and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, volume 1, 282–289.

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Li, K.; Zhou, Z.; and Lee, C.-H. 2016. Sign transition modeling and a scalable solution to continuous sign language recognition for real-world applications. *ACM Transactions on Accessible Computing* 8(2):7.

Liu, T.; Zhou, W.; and Li, H. 2016. Sign language recognition with long short-term memory. In *IEEE International Conference on Image Processing*, 2871–2875.

Morency, L.-P.; Quattoni, A.; and Darrell, T. 2007. Latent-dynamic discriminative models for continuous gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Pan, Y.; Yao, T.; Tian, X.; Li, H.; and Ngo, C.-W. 2014. Click-through-based subspace learning for image search. In *Proceedings of ACM International Conference on Multimedia*, 233–236.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2015. Jointly modeling embedding and translation to bridge video and language. *arXiv preprint arXiv:1505.01861*.

Pu, J.; Zhou, W.; and Li, H. 2016. Sign language recognition with multi-modal features. In *Pacific Rim Conference on Multimedia*, 252–261. Springer.

Ran, L.; Zhang, Y.; Wei, W.; and Zhang, Q. 2017a. A hyperspectral image classification framework with spatial pixel pair features. *Sensors* 17(10).

Ran, L.; Zhang, Y.; Zhang, Q.; and Yang, T. 2017b. Convolutional neural network-based robot navigation using uncalibrated spherical images. *Sensors* 17(6).

Starner, T.; Weaver, J.; and Pentland, A. 1998. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(12):1371–1375.

Sun, C.; Zhang, T.; Bao, B.-K.; Xu, C.; and Mei, T. 2013. Discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics* 43(5):1418–1428.

Tang, A.; Lu, K.; Wang, Y.; Huang, J.; and Li, H. 2015. A real-time hand posture recognition system using deep neural networks. *ACM Transactions on Intelligent Systems and Technology* 6(2):21.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, 4489–4497.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; and Saenko, K. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, 4534–4542.

Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10):1550–1560.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. J.; and Hovy, E. H. 2016. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 1480–1489.

Yang, W.; Tao, J.; and Ye, Z. 2016. Continuous sign language recognition using level building based on fast hidden markov model. *Pattern Recognition Letters* 78:28–35.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C.; Larochelle, H.; and Courville, A. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, 4507–4515.

Zhang, Q., and Hua, G. 2015. Multi-view visual recognition of imperfect testing data. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, 561–570. ACM.

Zhang, Q.; Abeida, H.; Xue, M.; Rowe, W.; and Li, J. 2011. Fast implementation of sparse iterative covariance-based estimation for array processing. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, 2031–2035. IEEE.

Zhang, Q.; Abeida, H.; Xue, M.; Rowe, W.; and Li, J. 2012. Fast implementation of sparse iterative covariance-based estimation for source localization. *The Journal of the Acoustical Society of America* 131(2):1249–1259.

Zhang, Q.; Hua, G.; Liu, W.; Liu, Z.; and Zhang, Z. 2015a. Auxiliary training information assisted visual recognition. *IPSJ Transactions on Computer Vision and Applications* 7:138–150.

Zhang, Q.; Hua, G.; Liu, W.; Liu, Z.; and Zhang, Z. 2015b. Can visual recognition benefit from auxiliary information in training? In *Computer Vision – ACCV 2014*, volume 9003 of *Lecture Notes in Computer Science*, 65–80. Springer International Publishing.

Zhang, J.; Zhou, W.; Xie, C.; Pu, J.; and Li, H. 2016. Chinese sign language recognition with adaptive hmm. In *IEEE International Conference on Multimedia and Expo*, 1–6.

Zhang, K.; Zhang, L.; and Yang, M.-H. 2012. Real-time compressive tracking. In *European Conference on Computer Vision*, 864–877. Springer.

Zhang, J.; Zhou, W.; and Li, H. 2014. A threshold-based hmm-dtw approach for continuous sign language recognition. In *Proceedings of ACM International Conference on Internet Multimedia Computing and Service*, 237.