Deep Grammatical Multi-classifier for Continuous Sign Language Recognition

Chengcheng Wei, Wengang Zhou, Junfu Pu, Houqiang Li
CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System
University of Science and Technology of China, Hefei, China
ccwei@mail.ustc.edu.cn, zhwg@ustc.edu.cn, pjh@mail.ustc.edu.cn, lihq@ustc.edu.cn

***Abstract*—In this paper, we propose a novel deep architecture with multiple classifiers for continuous sign language recognition. Representing the sign video with a 3D convolutional residual network and a bidirectional LSTM, we formulate continuous sign language recognition as a grammatical-rule-based classification problem. We first split a text sentence of sign language into isolated words and n-grams, where an n-gram is a sequence of consecutive n words in a sentence. Then, we propose a word-independent classifiers (WIC) module and an n-gram classifier (NGC) module to identify the words and n-grams in a sentence, respectively. A greedy decoding algorithm is employed to integrate words and n-grams into the sentence based on the confidence scores provided by both modules. Our method is evaluated on a Chinese continuous sign language recognition benchmark, and the experimental results demonstrate its effectiveness and superiority.**

***Keywords*-N-gram; Continuous Sign Language Recognition; Multi-classifier**

I. Introduction

Sign language is a non-trivial communication way in the deaf community. However, due to the lack of knowledge about sign language for most hearing people, there is a communication gap between the deaf people and hearing people, which leads to the potential loss of opportunities in education and employment for deaf people. Sign language recognition (SLR) aims to translate sign videos into ordered sign glosses, which helps hearing people understand the content of sign videos. Generally, SLR methods are divided into two categories, *i.e.*, isolated sign language recognition and continuous sign language recognition. Isolated SLR [1], [2], [3] is a kind of fine-grained action recognition task, where each sign video describes a single sign word. In contrast, continuous SLR [4] tackles sign videos describing sign sentences, which is more challenging and practical. To be specific, the accurate semantic boundaries in a sign video are unknown, which makes it difficult to align frames with glosses. Under this background, continuous SLR is essentially a kind of weakly supervised learning task.

To address continuous SLR and its related tasks, several recent works [5], [6], [7], [8] have developed their methods following an encoder-decoder framework [9] with impressive performance. Generally, these methods select decoders based on classical recurrent neural networks (RNN) to generate sign sentences. However, as discussed in [10], [11], there is a problem of *error accumulation* in these sentence generating models. In the training stage, the decoders are typically fed with the ground truth sentence. While in the testing stage, the generation of the next word is dependent on the generated distribution of the previous word. Such kind of dependency leads to severe errors in the generation process.

Our method is free of such error accumulation. Instead of using a classical RNN as the decoder, we design a novel decoding module consisting of multiple grammatical-rule-based classifiers, inspired by [12] and [13] tackling street number recognition and text recognition, respectively. After encoding the original sign video into a feature vector, we propose the word-independent classifiers (WIC) module, which contains a series of classifiers and each recognizes a word from the feature vector. The word classifiers are independent, which gets rid of error accumulation in the testing stage. Besides, there are amounts of common phrases and expressions in sign language, which are expressed in more than one word. They can be utilized as additional supervision and contribute to sentence recognition. To this end, we propose the n-gram classifier (NGC) module, which acts as a grammar-based multi-label classifier to identify the n-grams (*i.e.*, sign phrases) in the sentence. Note that in the n-gram model, unigram, bigram, and trigram represent one word, two adjacent words, and three consecutive words, respectively. In the training stage, we split a sentence into isolated words and n-grams for multiple classification tasks. While in the testing stage, a greedy decoding algorithm is proposed for sentence inference, which integrates words and n-grams into the sentence based on confidence scores provided by the WIC as well as NGC module.

Our main contributions are summarized as follows:

- We propose a novel deep architecture consisting of multiple grammatical-rule-based classifiers for continuous SLR, and we formulate the sentence recognition task as a classification problem of sign words and n-grams by introducing the WIC and NGC modules, providing a new viewpoint to address continuous SLR.
- The proposed WIC module aims to recognize sign words without dependence on the recognition of previous words, which gets rid of error accumulation compared to classical RNN-based sentence generators. In order to make full use of contextual information, we

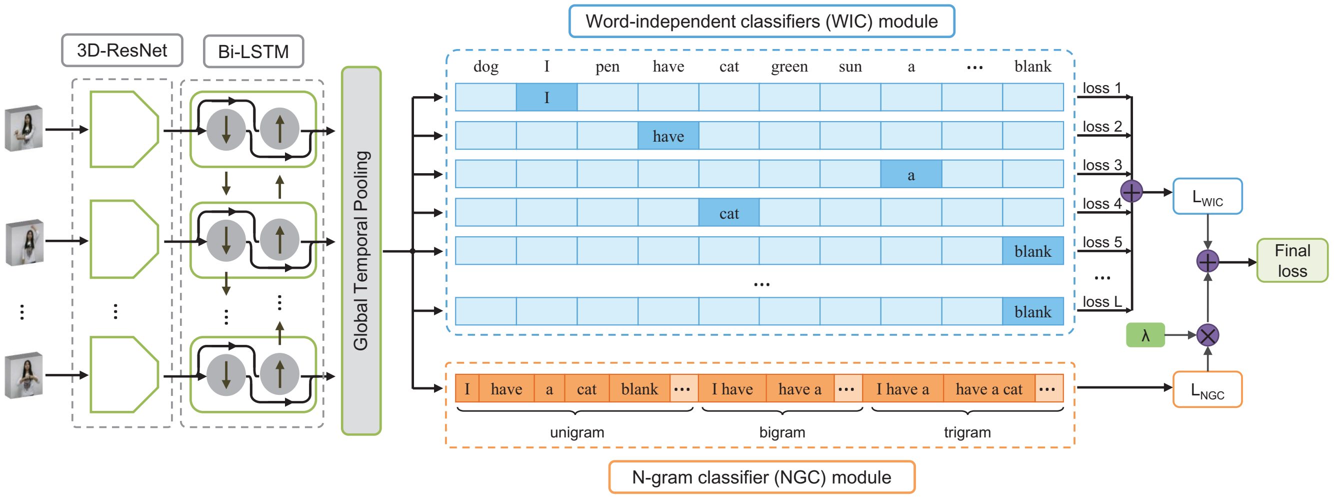

Figure 1. Overview of our proposed architecture. For a sign video, the 3D residual network (3D-ResNet) and bidirectional long short-term memory (Bi-LSTM) are employed for feature representation and contextual learning, respectively. The following global temporal layer summarizes the video into a fixed-length vector. The proposed word-independent classifiers (WIC) module, which consists of a series of word classifiers, recognizes sign words in order. While the n-gram classifier (NGC) module containing only a multi-label classifier identifies sign phrases to help the recognition. We train the network by joint cross-entropy loss, with the coefficient λ balancing two kinds of losses from WIC and NGC module.

represent phrases as n-grams and introduce the NGC module to disentangle the multi-label classification task, which is the first attempt in continuous SLR to the best of our knowledge.

- We evaluate our method on a large-scale continuous SLR dataset with extensive experiments, and the results demonstrate the effectiveness and superiority of our proposed approach.

The rest of this paper is organized as follows: after introducing the related works in Section II, we elaborate our method in Section III. In Section IV we conduct extensive experiments and give an analysis of the results. At last, we summarize this paper in Section V.

II. Related Work

In this section, we first introduce feature extraction and sequence learning, which are two subtasks for video-based continuous SLR. Then two most common categories of continuous SLR approaches are discussed, *i.e.*, methods based on the encoder-decoder framework and connectionist temporal classification (CTC) [14].

Generally, continuous SLR is a cross-modal task, involving vision and language. Early related works [15], [16] typically feed hand-crafted features to statistical sequence models such as hidden Markov models (HMM) and conditional random fields (CRF). In recent years, convolutional neural networks (CNN) and recurrent neural networks (RNN) have achieved impressive success in the fields of computer vision and natural language processing, respectively. More and more SLR approaches employ CNNs for visual feature extraction [17], [18], [19] and RNNs for temporal sequence modelling [20], [21], [22], which improve the performance significantly. Compared to 2D CNNs, 3D CNNs are specially designed for video feature extraction [23] and soon widely used in action recognition [24], [25]. Inspired by this, several works [6], [26], [27] employ 3D CNNs as spatio-temporal feature extractors for continuous SLR. In addition, some other works [28], [29], [30] use 2D CNNs as well as temporal convolutions for spatial and temporal feature extraction. RNNs have a great capacity for sequence processing tasks, such as speech recognition [31], neural machine translation [32], [33] and continuous SLR, thanks to its recurrent topology. The long short-term memory (LSTM) [34] and gated recurrent unit (GRU) [9] are both commonly used RNN cells in continuous SLR.

The encoder-decoder framework is originally proposed for neural machine translation [9], [35]. Later on, this kind of architecture is successfully introduced into continuous SLR. Huang *et al.* [5] propose a hierarchical network with the attention mechanism [32], based on the encoder-decoder framework. It encodes videos hierarchically and decodes the latent vectors to sign sentences. On the other hand, CTC is first designed for speech recognition [14] as a sequence alignment model. Afterwards, it is used in handwriting recognition [36], [37], lip reading [38], [39], and sign language recognition [18], [21], [27], achieving impressive performance. Current state-of-the-art continuous SLR approaches are mainly developed based on the encoder-decoder framework as well as the CTC model. Different from existing approaches, our method with the WIC and NGC module formulates continuous SLR as a grammatical-rule-based classification problem and provides a new alternative network to the classic CTC or encoder-decoder based networks.

III. Our Method

Fig. 1 shows the architecture of our proposed method. First, we split a sign video into a sequence of video clips and use the 3D residual network (3D-ResNet) for spatio-temporal video clip representation. Like most of the sequence learning approaches, the bidirectional long short-term memory (Bi-LSTM) is adopted to model the temporal dependency between the clips. A global temporal pooling layer summarizes the video representation into a fixed-length feature vector, which is fed into a set of classifiers in the WIC module and NGC module simultaneously. In the WIC module, there are a series of classifiers and each performs a word classification task. While in the NGC module, the n-gram classifier conducts a multi-label classification task, where all unigrams, bigrams, and trigrams in the ground truth sentence are the labels. We use the summation of weighted cross-entropy losses to train the architecture, and propose a greedy decoding algorithm to generate a sentence in the testing stage.

A. Video Representation Learning

In this section, we discuss and formulate the video representation with the 3D-ResNet and Bi-LSTM followed by global temporal pooling in detail.

3D-ResNet. Let $\mathbf{X} = \{x_t\}_{t=1}^{T}$ be the given sign video with T frames. First we split the video uniformly into N clips in a sliding-window way with an overlap. Then the video is denoted as $\mathbf{V} = \{v_i\}_{i=1}^{N}$, where v_i is the i-th clip of the video. To better extract the spatio-temporal feature representation for the clips, we extend the 3D CNN with residual connections [40], which is denoted as $\mathcal{C}$. The feature extraction can be represented as follows,

$$\mathbf{F} = \{f_i\}_{i=1}^{N} = \mathcal{C}\left(\{v_i\}_{i=1}^{N}\right), \tag{1}$$

where $\mathbf{F} = \{f_i\}_{i=1}^{N}$ denotes the representations for the clips extracted by the 3D-ResNet, and $f_i \in \mathbb{R}^d$ is the spatio-temporal feature vector of the video clip v_i.

Bi-LSTM. 3D-ResNet learns spatio-temporal representation within a video clip, while Bi-LSTM models contextual relationship across video clips. The Bi-LSTM is stacked by two opposite-directional LSTMs [34], where one for forward information transfer and the other for backward. Taking both previous and future video clips into account facilitates a better understanding of the sign video clip. Let $\mathcal{R}$ denotes the Bi-LSTM. The learning process can be described as follows,

$$\mathbf{H} = \{h_i\}_{i=1}^{N} = \mathcal{R}\left(\{f_i\}_{i=1}^{N}\right), \tag{2}$$

where $h_i \in \mathbb{R}^D$ is the result of sequence learning, corresponding to video clip v_i.

Global temporal pooling. To summarize the content of the sign video, we conduct a global temporal pooling operation on the video representation matrix $\mathbf{H}_{N\times D}$. This operation squeezes the temporal dimension and outputs a feature vector $h_p \in \mathbb{R}^D$. Let $\mathcal{P}$ be the global temporal pooling operation, the pooling process can be represented as:

$$h_p = \mathcal{P}\left(\mathbf{H}\right). \tag{3}$$

There are several alternative pooling strategies: mean pooling, max pooling , first-time pooling and last-time pooling, where $h_p = h_1$ for first-time pooling, and $h_p = h_N$ for last-time pooling. We will evaluate these strategies in the following experimental part.

B. Word-independent Classifiers (WIC) Module

Given a sign video $\mathbf{X}$, our method aims to translate it into a sentence $\mathbf{s} = \langle w_1, w_2, \cdots, w_n\rangle$, where w_i is the i-th word in the sentence. The length of the sentence is defined as $n = |\mathbf{s}|$, indicating the number of words in the sentence. Sign words are from the finite vocabulary V, which is denoted as $w_i \in V$. As the length of a sign sentence is finite, we assume $n \leqslant L$, where L is the largest length of sign sentences.

The key idea of the WIC module is to recognize sign words in order with a series of word classifiers. More specifically, the i-th classifier learns to recognize the i-th word w_i. Considering the largest length of sentences is no more than L, we deploy L ordered classifiers in the WIC module. However, in most cases, we have $n < L$, which means the number of words n and the number of word classifiers L are not matched. In order to tackle these cases, we introduce a blank label {'_'} to extend the word vocabulary V, which can be represented as $V' = V \cup \{\text{'\_'}\}$. Thus, it is responsible for the i-th classifier to learn not only whether the i-th word is existent (blank label for non-existent, non-blank label for existent), but also the category of this word (if existent). For instance, for a sign video with the ground truth sentence $\mathbf{s} = \langle$I, have, a, cat$\rangle$, the real length of this sentence is $n = 4$. The maximum sentence length L is assumed to be 5, so there are 5 ordered classifiers in the WIC module. During the training stage, we assign 'I', 'have', 'a', 'cat' to the first four classifiers respectively as the labels. Besides, in order to train the fifth word classifier with an explicit label, we pad the original sentence $\mathbf{s}$ to $\mathbf{s}' = \langle$I, have, a, cat, _$\rangle$ and assign the blank label '_' to the fifth classifier as the reference.

In the WIC module, each classifier with a softmax layer is constrained by a cross-entropy loss and learns a probability distribution of the extended vocabulary V'. The training objective function in the WIC module can be defined as:

$$\mathcal{L}_{WIC} = \sum_{i=1}^{L} \mathcal{L}^{(i)}, \tag{4}$$

where $\mathcal{L}^{(i)}$ is the cross-entropy loss of the i-th classifier, and L is the number of classifiers.

C. N-gram Classifier (NGC) Module

In the WIC module, a classifier only focuses on a single sign word. Actually, in sign language, there are many common phrases, which are expressed in more than one word. It is a great help for continuous SLR to regard and utilize them as additional supervision. To this end, we formulate the recognition of sign sentence as a multi-label classification problem, and each label is a sign word or sign phrase. First of all, according to the definition in the n-gram language model [41], we represent a single word as a unigram (*e.g.*, ‘I’), two adjacent words as a bigram (*e.g.*, ‘I, have’), three consecutive words as a trigram (*e.g.*, ‘I, have, a’). Then we can use the defined n-grams to express the padded sentence s' as an n-gram sentence s''. For example, the n-gram sentence $\mathrm{s}'' = \langle a, b, c, -, ab, bc, c-, abc, bc- \rangle$ is generated from the padded sentence $\mathrm{s}' = \langle a, b, c, - \rangle$.

Based on the statistics of the training set, we regard all unigrams, bigrams, and trigrams as independent categories for the n-gram classifier. All elements in n-gram sentence s'' are the labels. As phrases with more than three words are infrequent, we consider trigram at most, which avoids sparse labels at the same time. Let $\mathcal{L}_{NGC}$ denote the cross-entropy loss of the n-gram classifier, then the joint training objective of WIC module and NGC module is defined as:

$$\mathcal{L} = \mathcal{L}_{WIC} + \lambda \mathcal{L}_{NGC}, \tag{5}$$

where λ is a tunable coefficient balancing the potential significance of the two modules.

D. Greedy Decoding for Sentence Generation

We have proposed two models for continuous SLR. One is the basic model with only the WIC module and trained with (4), while the other contains both WIC and NGC modules and is trained according to (5). In this section, we propose two decoding algorithms to infer a sentence given the test video for the two models.

First, we elaborate a decoding method for the basic continuous SLR model, *i.e.*, the model without NGC. In the WIC module, each word classifier predicts the sign word by selecting the category with the largest confidence. Finally, we obtain a predicted word sequence $seq = \langle w_1, w_2, \cdots, w_L \rangle$, where w_i is the predicted sign word by the i-th classifier. As seq may contain the blank labels $\{‘\_’\}$ that are not in the original vocabulary V, we delete them from seq and obtain sentence $\hat{s}$, which can be represented as:

$$\hat{s} = \mathcal{M}\left(seq\right), \tag{6}$$

where the operation $\mathcal{M}$ deletes all blank labels and is a many-to-one mapping from seq to $\hat{s}$.

Next, we propose a greedy decoding algorithm for the complete continuous SLR model. Generally, the sentence is inferred word by word according to the confidence scores in the testing stage. To be specific, for inferring the i-th word, we consider the word confidence score provided by the i-th word classifier in the WIC module, and the sum of n-gram confidence scores provided by the n-gram classifier in the NGC module. Let $C_S^i\left(w\right)$ and $C_N^i\left(w\right)$ denote the confidence functions of the word w provided by the WIC module and NGC module, respectively, where $w \in V'$. We represent the score function of the i-th word w as:

$$S^i\left(w\right) = C_S^i\left(w\right) + C_N^i\left(w\right), \tag{7}$$

where $C_N^i\left(w\right)$ is the sum of n-gram confidence scores, *i.e.*, the confidence scores of all the unigram, bigrams, and trigrams that ends with w. Therefore, $C_N^i\left(w\right)$ can be calculated as (8), where $\hat{w}_{i-2}$ and $\hat{w}_{i-1}$ are the generated sign words in the previous steps before recognizing the i-th word, and C_{uni}, C_{bi}, and C_{tri} are the n-gram confidence scores provided by the NGC module. We infer words with the highest scores one by one, according to $S^i\left(w\right)$ in (7). In this greedy way, we obtain a generated sequence with L words. After conducting the operation defined in (6), we get the final sentence.

$$C_N^i\left(w\right) = \begin{cases} C_{uni}\left(w\right), & i = 1 \\ C_{uni}\left(w\right) + C_{bi}\left(\langle \hat{w}_{i-1}, w \rangle\right), & i = 2, \\ C_{uni}\left(w\right) + C_{bi}\left(\langle \hat{w}_{i-1}, w \rangle\right) + C_{tri}\left(\langle \hat{w}_{i-2}, \hat{w}_{i-1}, w \rangle\right), & i \geq 3 \end{cases} \tag{8}$$

IV. Experiments

In this section, we provide the experimental details, then we evaluate our proposed methods and make the analysis. Furthermore, we discuss other continuous SLR approaches and compare them with ours.

A. Dataset and Implementation Details

We evaluate our methods on the CSL dataset [5], a large-scale video-based Chinese sign language dataset, which covers 100 sentences and each sentence is performed by 50 signers. The CSL dataset contains $100 \times 50 = 5000$ video instances in total and the word vocabulary size is 178. All the sentences have an average length of 5, and the longest sentence contains 7 sign words. Therefore, the proposed WIC module consists of $L = 7$ word classifiers.

Following the work [6], we provide two strategies to split the CSL dataset into the training set and testing set. To be specific, (a) **Split I - signer independent test**: this strategy splits the 50 signers into 40 signers as the training set and 10 signers as the testing set. Both the training and testing sets contain all the 100 sentences. (b) **Split II - unseen sentence**

Table I

EVALUATION OF DIFFERENT POOLING STRATEGIES FOR THE GLOBAL POOLING LAYER. BETTER PERFORMANCE HAS A HIGHER VALUE FOR PRECISION, AND A LOWER VALUE FOR WER.

Pooling strategy	Precision on Split I	WER on Split II
mean pooling	0.929	0.549
max pooling	**0.952**	**0.532**
first-time pooling	0.935	0.577
last-time pooling	0.931	0.581

test: this strategy splits the 100 sentences into 94 sentences as the training set and the remaining sentences as the testing set. Signers in training and testing sets are the same. What's more, the words in the testing set are also contained in the training set, but the sentences in both sets are different from each other. We use the same hyper-parameters on both datasets from different splitting strategies.

To evaluate the performance quantitatively, a series of performance metrics are introduced. For CSL Split I, the sentences in test videos are seen in the training stage. So we use a strict metric called precision, which is essentially a sentence-level accuracy rate. For CSL Split II, as the sentences in the testing set are all different from the training set, generating completely correct sentences is too hard. Instead of precision, the word error rate (WER) is used as the evaluation metric. It is a kind of word-level error rate and has been widely used in continuous SLR [18], [21]. The definition of WER is as follows,

$$WER = \frac{\#\text{ins} + \#\text{del} + \#\text{sub}}{\#\text{reference}}, \tag{9}$$

where #ins, #del and #sub denote the number of insertions, deletions and substitutions, respectively, to transform the predicted sentence to the reference sentence. In addition, we also employ other semantic evaluation metrics from natural language processing and neural machine translation, including BLEU, METEOR, ROUGE-L, and CIDEr. Note that better performance has a lower value for WER and the higher values for the rest of the metrics.

We use a sliding window with a size of 8 and a stride of 4 to split sign videos into clips. Therefore, each clip contains 8 frames and there are 50% overlapped frames between two adjacent clips. Actually, in the CSL dataset, consecutive 8 frames can describe a Chinese sign word on the average. First of all, an 18-layer 3D-ResNet is trained following the work [27] on an isolated SLR dataset [2] that covers all words in vocabulary V. We use a SGD optimizer with a learning rate of 0.001, a momentum of 0.9, and a weight decay of 5×10^{-5}. The batch size is 5. All the following experiments are conducted with the clip representation extracted by the convergent 3D-ResNet. The Bi-LSTM module has one layer and both the forward and backward LSTM have a hidden layer of 512 nodes, *i.e.*, $D = 512 \times 2 = 1024$. We employ a 50% dropout operation in the input layer of Bi-LSTM. The proposed deep network is trained using a learning rate of 0.00005 with a batch size of 50.

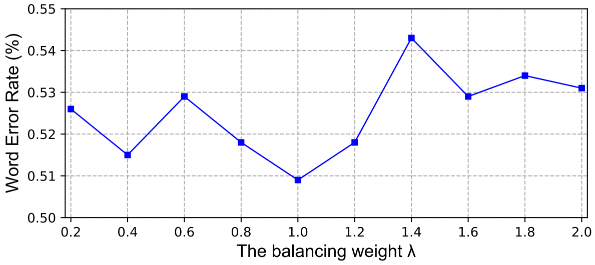

Figure 2. The sensitivity analysis of parameter λ, which reflects the relative significance of WIC and NGC, in (5). The experiments are conducted on CSL Split II dataset.

B. Ablation Study

In this section, we first compare different pooling strategies for the global temporal pooling layer, based on the basic model without NGC. Then we analyze the sensitivity of balancing parameter λ in (5) with the complete model.

Table I illustrates the performance on both datasets with the following pooling strategies: mean pooling, max pooling, first-time pooling, and last-time pooling, which are defined in Section III-A. The max-pooling operation has a strong capacity in selecting the recognizable feature, and it has been widely used in classification tasks [43], [44]. Compared to max-pooling, the mean pooling operation takes all temporal channels into account, which is more like feature fusion. First-time pooling and last-time pooling have comparable performance. They all mainly consider unidirectional context modelling. According to the results, the max pooling strategy achieves the best performance, with the highest precision of 0.952 on CSL Split I and the lowest WER of 0.532 on CSL Split II. Hence, all following experiments employ the max pooling layer as the global temporal pooling module.

In (5), λ adjusts the weight ratio of the loss functions in WIC and NGC. With the increase of λ, the n-gram constraint plays a more important role in the training objective. Fig. 2 shows the experimental results on CSL Split II dataset. λ ranges from 0.2 to 2.0, and we perform an experiment for every 0.2 increase in λ. Adopting $\lambda = 1.0$, our method achieves the best performance with the lowest WER=0.509, which demonstrates the equal significance of constraints from word classifiers and the n-gram classifier. Therefore, we select $\lambda = 1.0$ for the following experiments.

C. Comparison with Other Methods

In this section, we discuss our two methods and other existing approaches for comparison on both CSL datasets. Besides, more metrics are adopted for detailed evaluation, including BLEU, CIDEr, ROUGE-L, and METTOR.

Table II

Method comparison on CSL Split I dataset for seen sentence recognition. For all the metrics in this table, a higher value indicates a better performance.

Method	Precision	BLEU-1	CIDEr	ROUGE-L	METEOR
LSTM-CTC [14], [34]	0.858	0.936	8.632	0.940	0.646
S2VT [42]	0.897	0.902	8.512	0.904	0.642
S2VT (3-layer) [42]	0.903	0.911	8.592	0.911	0.648
HLSTM (SYS sampling) [6]	0.910	0.935	8.907	0.938	0.683
HLSTM [6]	0.924	0.942	9.019	0.944	0.699
HLSTM-attn [6]	0.929	0.948	9.084	0.951	0.703
Ours (WIC)	**0.952**	**0.982**	**9.420**	**0.980**	**0.729**
Ours (WIC-NGC)	**0.949**	**0.979**	**9.416**	**0.979**	**0.725**

Table III

Method comparison on CSL Split II dataset for unseen sentence recognition. Note that better performance has a lower value for WER and higher values for other metrics.

Method	WER	BLEU-1	CIDEr	ROUGE-L	METEOR
LSTM-CTC [14], [34]	0.757	0.343	0.241	0.362	0.111
S2VT [42]	0.670	0.466	0.479	0.461	0.189
S2VT (3-layer) [42]	0.652	0.475	0.477	0.465	0.186
HLSTM (SYS sampling) [6]	0.630	0.463	0.476	0.462	0.173
HLSTM [6]	0.662	0.487	0.561	0.481	0.193
HLSTM-attn [6]	0.641	**0.508**	0.605	0.503	0.205
Ours (WIC)	**0.532**	0.483	**0.760**	**0.514**	**0.219**
Ours (WIC-NGC)	**0.509**	0.505	**0.641**	**0.537**	**0.223**

For continuous SLR, CTC based approaches typically conform to RNN-CTC structure after feature representation by CNNs. In this paper, LSTM-CTC is used as a comparative method. S2VT [42] is a successful method in encoder-decoder framework for video to text task. Its encoder and decoder are both multilayer LSTMs. HLSTM [6] is also an encoder-decoder based method especially for continuous SLR. HLSTM (SYS sampling) conduct a systematic sampling operation for a fixed-length vector. Additionally, HLSTM-attn adopts attention mechanism on the basic HLSTM network. Compared to the above architectures, our proposed methods (*i.e.*, without NGC and with NGC) are novel and effective. We explicitly recognize every word and n-gram, instead of learning the whole sentence in CTC based methods. The latter may be confused to tackle the complicated word relationships inside sentences. Moreover, our methods are rarely disturbed by error accumulation, which is a common problem in encoder-decoder based methods.

The comparison results on CSL Split I are shown in Table II. For all metrics in the table, a higher value indicates a better performance. Apparently, our two methods are comparable and outperform other methods significantly, which demonstrates the superiority of our novel architecture for continuous SLR. Table III depicts the method comparison on CSL Split II. Note that better performance has a lower value for WER and higher values for the rest of the metrics. For CSL Split II, the sentences in the training set and testing set are different, but the specific phrases appear in both sets, which is more in line with reality and challenging. With this setting, the NGC module has greater potential to learn phrases and help recognize the sign sentences. It can be seen from the results that NGC plays a significant role in performance improvement, compared to the basic method with only the WIC module.

V. Conclusion

In this paper, a novel deep architecture with multiple grammatical-rule-based classifiers is proposed for continuous sign language recognition. We formulate the sentence recognition task as a classification task for word and n-grams by introducing the WIC module and NGC module. We train the network by minimizing the joint classification loss and use a greedy algorithm for sentence inference. Extensive experiments are conducted on a large-scale benchmark, and the results show the effectiveness of our proposed methods.

Acknowledgment

This work was supported by NSFC under contract No.61632019 and Youth Innovation Promotion Association CAS under Grant 2018497.

References

[1] F. Yin, X. Chai, and X. Chen, “Iterative reference driven metric learning for signer independent isolated sign language recognition,” in *European Conference on Computer Vision (ECCV)*, 2016.

[2] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, “Chinese sign language recognition with adaptive hmm,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016.

[3] J. Huang, W. Zhou, H. Li, and W. Li, “Attention based 3D-cnns for large-vocabulary sign language recognition,” *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2018.

[4] O. Koller, O. Zargaran, H. Ney, *et al.*, “Deep sign: hybrid cnn-hmm for continuous sign language recognition,” in *British Machine Vision Conference (BMVC)*, 2016.

[5] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, “Video-based sign language recognition without temporal segmentation,” in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[6] D. Guo, W. Zhou, H. Li, and M. Wang, “Hierarchical lstm for sign language translation,” in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[7] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural sign language translation,” in *IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2018.

[8] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, “Sign language production using neural machine translation and generative adversarial networks,” in *British Machine Vision Conference (BMVC)*, 2018.

[9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[10] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” *arXiv preprint arXiv:1511.06732*, 2015.

[11] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, “Multi-digit number recognition from street view imagery using deep convolutional neural networks,” *arXiv preprint arXiv:1312.6082*, 2013.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep structured output learning for unconstrained text recognition,” in *International Conference on Learning Representations (ICLR)*, 2015.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *International Conference on Machine Learning (ICML)*, 2006.

[15] T. Starner, J. Weaver, and A. Pentland, “Real-time american sign language recognition using desk and wearable computer based video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 20, no. 12, pp. 1371–1375, 1998.

[16] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, “Hidden conditional random fields for gesture recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[17] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” in *European Conference on Computer Vision (ECCV)*, 2014.

[18] N. C. Camgöz, S. Hadfield, O. Koller, and R. Bowden, “Subunets: End-to-end hand shape and continuous sign language recognition,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] H. Zhou, W. Zhou, and H. Li, “Dynamic pseudo label decoding for continuous sign language recognition,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2019.

[20] O. Koller, S. Zargaran, and H. Ney, “Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] R. Cui, H. Liu, and C. Zhang, “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] J. Pu, W. Zhou, and H. Li, “Iterative alignment network for continuous sign language recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 1, pp. 221–231, 2013.

[24] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2017.

[25] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3D cnns retrace the history of 2D cnns and imagenet?,” in *IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2018.

[26] J. Huang, W. Zhou, H. Li, and W. Li, “Sign language recognition using 3D convolutional neural networks,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2015.

[27] J. Pu, W. Zhou, and H. Li, “Dilated convolutional network with iterative optimization for continuous sign language recognition,” in *International Joint Conference on Artifcial Intelligence (IJCAI)*, 2018.

[28] S. Wang, D. Guo, W. Zhou, Z. Zha, and M. Wang, “Connectionist temporal fusion for sign language translation,” in *ACM International Conference on Multimedia (ACM MM)*, 2018.

[29] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, “Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video,” *International Journal of Computer Vision (IJCV)*, vol. 126, no. 2-4, pp. 430–439, 2018.

[30] R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuous sign language recognition by iterative training,” *IEEE Transactions on Multimedia (TMM)*, 2019.

[31] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[32] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[33] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[34] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[36] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Transactions on*

Pattern Analysis and Machine Intelligence (TPAMI), vol. 31, no. 5, pp. 855–868, 2009.

[37] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[38] K. Xu, D. Li, N. Cassimatis, and X. Wang, "Lcanet: End-to-end lipreading with cascaded attention-ctc," in *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2018.

[39] M. Faisal and S. Manzoor, "Deep learning for lip reading using audio-visual information for urdu language," *arXiv preprint arXiv:1802.05521*, 2018.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2016.

[41] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[42] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, *et al.*, "Sequence to sequence-video to text," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.