ABSTRACT. Continuous sign language recognition (CSLR) is a challenging research task due to the lack of accurate annotation on the temporal sequence of sign language data. The recent popular usage is a hybrid model based on “CNN + RNN” for CSLR. However, when extracting temporal features in these works, most of the methods using a fixed temporal receptive field and cannot extract the temporal features well for each sign language word. In order to obtain more accurate temporal features, we propose a multiscale temporal network. The network mainly consists of three parts. The ResNet and two fully connected layers constitute the frame-wise feature extraction part. The time-wise feature extraction part performs temporal feature learning by first extracting temporal receptive field features of different scales using the proposed multiscale temporal block (MST-block) to improve the temporal modeling capability, and then further encoding the temporal features of different scales by the transformers module to obtain more accurate temporal features. Finally, the proposed multilevel connectionist temporal classification (CTC) loss part is used for training to obtain recognition results. The multilevel CTC loss enables better learning and updating of the shallow network parameters in CNN, which has no parameter increase and can be flexibly embedded in other models. Experimental results on two publicly available datasets demonstrate that the method can extract sign language features in an end-to-end manner without any prior knowledge effectively, improve the accuracy of CSLR, and achieve competitive results.

1 Introduction

Sign language conveys semantic information through hand movements, gesture appearance, etc. It is the main way of communication between deaf people or between deaf people and normal people. With the increasing number of hearing-impaired people worldwide, sign language recognition occupies an increasingly important position; involves various fields such as computer vision, natural language processing (NLP), and human–computer interaction technology; and has gained extensive attention. Video-based sign language recognition is divided into two categories: one is isolated word recognition, where each video segment represents only a single sign language word, and the other is continuous sign language recognition (CSLR), where each video segment represents a sign language sentence. For real-life applications, CSLR research is of more social value.
At present, CNN is becoming more and more popular in CSLR research due to its powerful feature representation ability and sequence modeling capability. A series of deep learning-based CSLR models continue to emerge and perform well.\textsuperscript{9-11} It is mainly based on CNN to extract frame-wise features and time-wise features of video clips. Gloss is the basic unit in CSLR. Video-based sign language recognition is to translate the video sign language action sequence into a continuous gloss, which in turn is translated into natural language to help hearing impaired people communicate with others.\textsuperscript{12} Due to the lack of strict correspondence between video frames and annotation sequences, that is, only the sign language actions and sequences performed by the sign language presenter in the video are known, but the start and end moments of the specific actions are unknown so CSLR is a weakly supervised learning problem.\textsuperscript{13} This places a high demand on sequence learning, that is, it is crucial to learn the correspondence between video sequences and labeled lexical sequences.

Current sequence learning models include 3D-CNN,\textsuperscript{14} RNN,\textsuperscript{15} and long short-term memory (LSTM).\textsuperscript{16} 3D-CNNs have a strong representation ability for videos and have achieved excellent performance on video action recognition tasks. The 3D convolution operation is able to model not only the spatial information but also consider the temporal information between video frames. However, it suffers from the problem of bulky model and large number of parameters. RNN has the ability to obtain the information of the input sequence and output the sequence, which is widely used in sequence recognition, but its ability to acquire long-distance dependency information is weak. The proposal of LSTM alleviates this problem, but this problem still exists, and the convergence speed is slow. 1DCNN is widely used in CSLR for temporal feature extraction in recent years because of its advantages of simple structure and small number of parameters.\textsuperscript{11,17,18} Most of the CSLR models based on these algorithms in recent years have been extracted from local features of fixed temporal receptive fields.\textsuperscript{18,19} However, in the original video sequence, the length of the video clip sequence corresponding to different glosses is different, and the proficiency of sign language and some other interference caused by different sign language performers in the process of presentation will cause the same gloss to have inconsistent time used, as shown in Fig. 1. In this case, the results obtained using the fixed temporal receptive field for feature extraction will also be different, resulting in the limitations of the extracted time-wise features and affecting the final recognition performance. Meanwhile, connectionist temporal classification (CTC)\textsuperscript{20} is widely used for decoding sequences in CSLR because of its ability to optimize the neural network directly without segmentation and annotation for unsegmented sequence data. However, using a single CTC loss for deep neural networks trained by a backward propagation algorithm based on the chain rule causes the shallow network parameters to be poorly learned and updated, which can affect the model fit.

To address the above issues, this paper proposes an end-to-end CSLR network, which is more in line with the real scene. The network mainly consists of three parts. For the output

---

**Fig. 1** Video frame lengths for different glosses and for the same gloss demonstrated by different people (the first row of images shows the respective video frame lengths for the same gloss demonstrated by different people, the second and third rows of images show the video frame lengths for different glosses).
of the frame-wise feature extraction part composed of ResNet and two fully connected (FC) layers, the proposed time-wise feature extraction part is used for temporal feature learning. Specifically, the proposed MST-block is first used to extract different scale temporal receptive field features and combine them into a candidate space, which is fused by learnable parameters, and then further encoded by the transformers module to obtain more accurate temporal features. Finally, the proposed multilevel CTC loss part is used for training to obtain recognition results. The research in this paper is based on 1DCNN, using temporal receptive fields of different scales to enhance sequence modeling capability, followed by the self-attention mechanism of transformers \cite{21,22} to better access long-distance-dependent information and improve the discriminative power of features. Finally, using the proposed multilevel CTC loss, this not only can better decode the temporal features but also can make the parameters of the shallow network well updated, and then efficiently train the frame-wise feature extraction network and temporal modeling network, further improving the recognition performance. A similar work to this paper is the VAC network proposed by Min et al. \cite{18}, where both we and that paper use CNN and 1DCNN for frame-wise and time-wise feature extractions, respectively, and use CTC loss for decoding. The differences between us and them are: (1) we propose a multiscale temporal module when using 1DCNN for time-wise feature extraction, which improves the accuracy of the extracted temporal features compared to the 1DCNN with fixed receptive field they use. (2) The final visual alignment module in that paper uses two different levels of CTC loss and a self-distillation loss, on the basis of which in this paper, a multilevel CTC loss is proposed to further enhance the training of the shallow layers in a deep network.

The main contributions of this paper are as follows.

1. The proposed end-to-end model multiscale temporal network (MSTNet) improves the accuracy of CSLR on two publicly available datasets and achieves competitive results.
2. The proposed MST-block better exploits the temporal receptive fields of different scales and has a significant improvement in the final recognition performance.
3. The proposed multilevel CTC loss enables the shallow network parameters to be better learned and updated and has a large improvement on the experimental results in the final experiments. And it has no parameter increase and can be flexibly embedded into other models.

2 Related Work

The mainstream models for CSLR mainly consist of frame-level feature extraction, temporal feature modeling, and finally decoding recognition. In recent years, with the rapid development of deep learning, the network structure of “CNN + LSTM/Bi-LSTM + CTC” is mostly used for CSLR.

First of all, the frame-wise feature extraction of sign language is basically based on CNN. Many classic CNN network structures that have been successfully applied in computer vision also play an important role in sign language frame-wise feature extraction. Koller et al. \cite{13} used GoogLeNet for frame-wise feature extraction. Cui et al. \cite{1} used VGG-S and GoogLeNet for frame-wise feature extraction for hand-shaped sequences and full image sequences, respectively. Min et al. \cite{18} used ResNet18 to perform frame-wise feature extraction. Cheng et al. \cite{19} self-constructed a CNN network for feature extraction. Although GoogLeNet is used as a feature extraction network in many papers, it is too complex because of the number of parameters is too large. Considering the great success of the ResNet in the field of image recognition, this paper adopts it to extract frame-wise feature.

Then, since CSLR belongs to weakly supervised learning, it is particularly important to learn the correspondence between sequential representations and lexical labels, that is, temporal features play a very important role in CSLR. A series of algorithm models on sequence learning have been continuously proposed to improve the temporal feature modeling capability. Koller et al. \cite{23} proposed a sequence learning approach that exploits sequence constraints in each individual data stream and combines these constraints by explicitly imposing synchronization points to take advantage of the parallelism shared by all subproblems, and according to this hybrid approach, a powerful LSTM model is embedded in each HMM data stream for recognition. Cui et al. \cite{1} developed a CSLR framework with a DNN and obtained the best results at that time.
in which the sequence learning module was “1DCNN + BiLSTM,” their proposed iterative alignment optimization and multimodality both contributed to the accuracy improvement. Recent models, such as the FCN network proposed by Cheng et al.\(^\text{19}\) and the VAC network proposed by Min et al.\(^\text{18}\) have used 1DCNN for temporal feature extraction and achieved good results. These algorithms have improved the acquisition of time-wise features to a certain extent, but they all extract the local features for a fixed temporal receptive field. The research of Wei et al.\(^\text{11}\) used pooling of different strides for downsampling to generate temporal features of different scales. However, this method is multiscale in the time series length, there is no information fusion between temporal features of different scales, and the pooling itself will lose a certain amount of information. This paper proposes an MST-block using different scales of time receptive fields to improve temporal modeling capabilities. Our MST-block is multiscale on the convolution kernel, using convolution kernels of different scales to calculate temporal features in parallel and then using learnable parameters for feature fusion and output. The temporal features obtained by our method will be richer and more comprehensive.

Finally, for the process of decoding the vectors encoded with temporal features into sequences for recognition, CTC loss is the most popular algorithm nowadays. CTC can handle unsegmented input sequence data and help the model learn the correspondence between input and output sequences. Yang et al.\(^\text{24}\) proposed a structured feature network (SF-Net), extracted features in a structured way, and gradually encoded frame-level, gloss-level, and sentence-level information into feature representations, and finally decoded them using CTC loss. The architecture proposed by Koishybay et al.\(^\text{25}\) involved spatial–temporal feature extraction model to segment useful “gloss” features and BiLSTM with CTC as sequence model. The use of CTC in these models has achieved good results, but basically a single CTC loss is directly used in the final recognition part. Models such as in Ref. 11 and 18 do not use a single CTC loss. Although they all improve the performance of the model to some extent, there is still potential for further improvement. Since deep neural networks use a chain-rule-based backpropagation algorithm to train the network, which can, to some extent, result in poor learning and updating of shallow network parameters. In this paper, a multilevel CTC loss is proposed, which can not only better decode the temporal features but also make the parameters of the shallow network well updated, and then it trains the frame-wise feature extraction network and temporal modeling network efficiently to improve CSLR performance.

### 3 Approach

The overall framework of the end-to-end CSLR proposed in this paper is shown in Fig. 2. The model consists of three parts: frame-wise feature extraction part, time-wise feature extraction part, and multilevel CTC loss training part. There are four levels of gloss features defined in our model. For the input sign language video, the frame-wise features, i.e., the first-level gloss features, are first obtained by the ResNet and two FC layers. After that, the time-wise features are...
obtained by the proposed MST-block. Specifically, the frame-wise features are passed through the first MST-block to get the second-level gloss features, and the second-level gloss features are sequentially used as the input of the second MST-block to get the third-level gloss features, which are then defined as the fourth-level gloss features after transformers timing coding, i.e., the final timing features. Finally, for the four gloss features obtained, they are summed and trained to optimize model using multilevel CTC loss, and the final sign language recognition results are obtained using the fourth-level gloss features. The details of the proposed MST-block will be described in Sec. 3.2.

3.1 Frame-Wise Feature Extraction

The frame-wise feature extraction consists of a main network feature extractor and two FC layers. For an input sign language video \( \tilde{V} = (x_1, x_2, \ldots, x_T) = \{x_t\}_{t \in \mathbb{R}^{T \times c \times h \times w}} \) containing \( T \) frames, where \( x_t \) is the \( t \)'th frame image in the video, \( h \times w \) is the size of \( x_t \), and \( c \) is the number of channels, here \( c = 3 \) for RGB video. \( \tilde{V} \) is the input into the ResNet feature extractor \( F_r \) to obtain feature expression \( f_1 = F_r(\tilde{V}) \in \mathbb{R}^{T \times c_1} \), and then after two FC layers to obtain feature expression \( f_2 = F_{fc}(f_1) \in \mathbb{R}^{T \times c_2} \), which is the final frame-wise feature vector containing spatial information with fixed dimensions. We define it as a first-level gloss feature in this paper. The sizes of \( c_1 \) and \( c_2 \) here are 512 and 1024, respectively. The function of adding two FC layers after the main network is to integrate the features in the image feature maps that have passed through multiple convolutional layers and pooling layers to obtain the high-level meaning of the image features. For the consideration of GPU memory, stochastic gradient stopping\(^{26} \) is used between the feature extraction of the main network and the FC layer to reduce the memory usage and improve the training speed.

3.2 Time-Wise Feature Extraction

Sequence modeling is performed after the frame-wise features extracted in Sec. 3.2. Here the time-wise feature extraction part proposed includes the MST-block and transformers. After MST-block, the second-level gloss features and the third-level gloss features are obtained, respectively, and finally the fourth-level gloss features are obtained after transformers.

3.2.1 MST-block

In recent years, excellent CSLR models have emerged one after another, but most of them extract local features of fixed temporal receptive fields. In the real scene, the sequence lengths of the video clips corresponding to different glosses in CSLR are different, and the proficiency of sign language by different sign language performers and some other disturbances during the demonstration process cause inconsistency in the time used for the same gloss. In this case, the results obtained using the fixed temporal receptive field for feature extraction will also be different, which will affect the performance of temporal modeling. In this paper, we first propose MST-block, as shown in Fig. 3, which uses different scales of temporal receptive fields to improve the capability of temporal modeling.

![Fig. 3 Details of the multiscale temporal block (two MST-blocks are used in this paper, and the timing length is downsampled for each passing MST-block).](image-url)
The MST-block is mainly composed of multiscale feature extraction and feature fusion. Multiple 1D convolutional layers with different convolution kernels are connected in parallel to form the multiscale feature extraction part. The formula of 1DCNN is as follows:

\[ y(t) = f_2(t) \times w(t) = \sum_{i=0}^{K} f_2(t - i)w(i), \]  

where \( w(t) \) is the weight, \( y(t) \) is the output data, \( K \) is the size of the convolution kernel, \( t \in [1, T] \), and \( T \) is the timing length.

For the first-level gloss feature obtained in Sec. 3.1, the dimension of the feature is first transformed, that is, \( f_2 \in \mathbb{R}^{T \times c_2} \) becomes \( f_2^T \in \mathbb{R}^{c_2 \times T} \). After that, it goes through the multiscale feature extraction part. The multiscale 1DCNN’s convolutional kernels have different sizes and the same number of channel dimensions, and the size of the timing and the number of features does not change during the feature extraction process. The size of the first convolution kernel is 3, the maximum convolution kernel size is \( M \), and the step size is 2, that is, the size of each convolution kernel is increased by 2 on the size of the previous convolution kernel:

\[ f_2^i = \text{cat}(y_n(t)), \]  

where \( f_2^i \) is the output after multiscale 1DCNN and \( n \) is the number of 1DCNNs.

After a 2DCNN for feature fusion and downsampling by 2 times, \( f_3 \in \mathbb{R}^{c_2 \times T_1} \) is obtained, that is, the second-level gloss feature, where \( T_1 = T/2 \), then the process is repeated to obtain the third-level gloss feature \( f_4 \in \mathbb{R}^{c_2 \times T_2} \), \( T_2 = T_1/2 \).

### 3.2.2 Transformers encoding

Transformers model is a classical model of NLP proposed by a team at Google in 2017, which uses the self-attention mechanism and does not use the sequential structure of RNNs, allowing the model to be trained in parallel and to have global information. The transformers encoding module used in this paper is shown in Fig. 4. For the temporal feature vector obtained by MST-block, the temporal sequence is further encoded using the transformers encoding module, which results in more accurate temporal features.

![Fig. 4 Details of the transformers encoding module (this paper uses the two-layer transformers encoding module to further encode the temporal features).](image-url)
Two identical transformers encoding modules are used in our proposed model. The multihead self-attention (MHSA) and the FC feedforward (FF) part constitute the transformers encoding module. The third-level gloss feature \( f_4 \in \mathbb{R}^{c_3 \times T_2} \) plus the corresponding position information PE is used as the input of MHSA, and then through the temporal feature \( f'_4 \in \mathbb{R}^{c_3 \times T_2} \) obtained by FF, the same operation is repeated to obtain the final temporal feature \( f_5 \in \mathbb{R}^{c_3 \times T_2} \), that is, the fourth-level gloss feature. MHSA not only extends the ability of the model to focus on different positions but also enhances the ability of the attention mechanism to express the roles between words within the sentences of interest. Compared with a single-head self-attention, each head in MHSA maintains a \( Q, K, \) and \( V \) matrix of its own to achieve different linear transformations so that each head also has its own special expressive information. FF, on the other hand, strengthens the representation in a nonlinear way, making the features more expressive.

### 3.2.3 Multilevel CTC loss

CSLR belongs to weakly supervised learning. The input video is an unsegmented sequence and lacks a strict correspondence between video frames and labeled sequences. After encoding the input video sequence, it is very appropriate to use CTC as a decoder. CTC is originally designed for speech recognition, mainly to perform end-to-end temporal classification of unsegmented data to address the problem of mismatched lengths of input and output sequences. In recent years, it is often used in CSLR. CTC introduces a blank label \(-\) to mark unclassified labels during decoding, that is, any word in the input video clip that does not belong to the sign language vocabulary so that the input and output sequences can be matched, and the dynamic programming method is used for decoding.\(^{20}\)

For the input video \( \tilde{V} \) of \( T \) frames, the label of each frame is represented by \( \pi = (\pi_1, \pi_2, \ldots, \pi_T) \), where \( \pi_i \in \nu \cup \{-\} \), and \( \nu \) is sign language vocabulary, then the posterior probability of the label is

\[
p(\pi|\tilde{V}) = \prod_{t=1}^{T} p(\pi_t|\tilde{V}) = \prod_{t=1}^{T} Y_{t,\pi_t}.
\]

(3)

For a given sentence-level label \( s = (s_1, s_2, \ldots, s_L) \), where \( L \) is the number of words in the sentence. CTC defines a many-to-one mapping \( B \), whose operation is to remove blank labels and duplicate labels [for example, \( B(-dd - ag - g-) = B(d-d - ag - g) = dog \)] in the path, then the conditional probability of label \( s \) is the sum of the occurrence probabilities of all corresponding paths:

\[
p(s|\tilde{V}) = \sum_{\pi \in B^{-1}(s)} p(\pi|\tilde{V}),
\]

(4)

where \( B^{-1}(s) = \{\pi|B(\pi) = s\} \) is the inverse mapping of \( B \). CTC loss is defined as the negative log-likelihood of the conditional probability of \( s \):

\[
L_{\text{CTC}} = - \ln p(s|\tilde{V}).
\]

(5)

Then the multilevel CTC loss can be expressed as

\[
L_{\text{sum}} = - \ln \prod_{i=1}^{n} p(s|V_i)
\]

\[
= - \ln(p(s|V_1)p(s|V_2)\ldots p(s|V_{n-1})p(s|V_n)),
\]

(6)

where \( n \) is the number of CTC.

For the four-level gloss feature obtained after transformers, after a FC layer, softmax is used for normalization, and then the normalized result is decoded by CTC to obtain \( L_{\text{CTC}_4} \). Likewise, corresponding \( L_{\text{CTC}_1}, L_{\text{CTC}_2}, \) and \( L_{\text{CTC}_3} \) are obtained for the primary, secondary, and tertiary gloss features, respectively. The four CTC losses are added to get the final loss for training.
The final sign language recognition result is obtained by CTC decoding after softmax for the fourth-level gloss feature only.

4 Experiments

In this section, we conduct experiments on two widely used sign language recognition datasets. We compare our model with advanced methods and perform ablation studies to demonstrate the effectiveness of each part of our model.

4.1 Datasets

The RWTH-PHOENIX-Weather-2014 (RWTH) dataset. The RWTH is recorded by a public weather radio and television station in Germany. All presenters are dressed in dark clothes and perform sign language in front of a clean background. The videos in this dataset are recorded by 9 different presenters with a total of 6841 different sign language sentences (of which the number of sign language word instances is 77,321, and the number of words is 1232). All videos are preprocessed to a resolution of 210 × 260 and a frame rate of 25 frames per second (FPS). The dataset is officially divided into 5672 training samples, 540 validation samples, and 629 test samples.

Chinese sign language (CSL) dataset. The CSL is captured using a Microsoft Kinect camera and contains 100 Chinese everyday phrases, each of which is demonstrated 5 times by 50 presenters with a vocabulary size of 178. The video resolution is 1280 × 720 and the frame rate is 30 FPS. The performance diversity of the dataset is richer because the demonstrators wear different clothes and demonstrate at different speeds and magnitudes of movement. Without giving an official segmentation, we divide the CSL into a training set and a test set according to the 8:2 rule, with 80% of the training set and 20% of the test set, i.e., into a training set of 20,000 samples and a test set of 5000 samples, and ensure that the sentences in the training and test sets are the same, but the presenters are different.

4.2 Implementation Details

The model experiments in this paper are trained using the Adam optimizer, with both the initial learning rate and weight factor set to $10^{-4}$. The batch size used is 2. A total of 60 epochs are used, and the learning rate is reduced by 80% at the 40th and 50th epochs. Data augmentation is performed using random cropping and random flipping. For random cropping, the frame size of each video sequence is first resized to 256 × 256, and then randomly cropped to 224 × 224 to fit the shape of the input. For random flips, its flip probability is 0.5. Flip and crop processing is performed on video sequences. We also do temporal augmentation, where the length of the video sequence grows or shrinks randomly within ±20%. In addition, the GPU dedicated memory is limited and the amount of video data is too large. To reduce the memory footprint and improve the training speed, we use mixed precision computation with acceptable precision loss. In the final CTC decoding stage, we used a beam search algorithm for decoding with a beam width of 10. For the CSL dataset, we used 15 epochs, and the learning rate is reduced by 90% at the eighth epoch, limited by GPU memory to downsample the original data by a factor of 2 as input, and used only two levels of CTC loss. The graphics card used in this experiment is RTX2080Ti, the GPU dedicated memory size is 12 G, the CPU memory is 8 G, and the number of cores is 4. The proposed network model consists of three parts, each of which is detailed as follows.

Part I: Frame-wise feature extraction. We use ResNet34 as the backbone, remove the last FC classification layer, and use the weight parameters trained on the ImageNet dataset as the initialization parameters. To further reduce memory usage and improve computational efficiency, a random gradient stop with a random stop probability of 0.5 is used during training. Two FC layers are connected after ResNet34 to obtain frame-wise features, and the number of channels is boosted from 512 to 1024.
Part II: Time-wise feature extraction. This part consists of the MST-block and the transformers encoding module. Each MST-block consists of multiple 1DCNNs with different kernels, and then the features are fused using 2DCNNs. We use four different kernels with kernel sizes of 3, 5, 7, and 9, padding sizes of 1–4, and stride of 1. 2DCNN has kernel = (4, 2), padding = 0, and stride = 2. A total of 2 MST-blocks are used. Since the input sequence will be downsample by a factor of 4 in this process, the size of the original video sequence needs to be converted to an integer multiple of 4 when it is input. In the transformers module, the number of input and output features is 1024, the number of layers is 2, and the number of multiheads is 8.

Part III: Multilevel CTC Loss. We use four CTC losses to calculate the losses of each of the four gloss features and sum them and then use the summed results for training. Finally, the fourth level of gloss features are used as a decoding input for sign language recognition.

4.3 Judgment Criteria
We use the word error rate (WER) to measure the performance of the model, a criterion that is widely used in CSLR.\textsuperscript{27,31} WER is a Levenshtein distance, which is the sum of the minimum insertion operations, substitution operations, and deletion operations required to convert a recognition sequence into a standard reference sequence. Lower WER means better recognition performance, and its definition is as follows:

\[
\text{WER} = 100\% \times \frac{\text{ins} + \text{del} + \text{sub}}{\text{sum}},
\]

where “ins” represents the number of words to be inserted, “del” represents the number of words to be deleted, “sub” represents the number of words to be replaced, and “sum” represents the total number of words in the label. Because “ins + del + sub” may be greater than sum, the result of WER may be > 100\%. When experimenting with the CSL dataset, we treat a single Chinese character as a word.

4.4 Experimental Results
The experimental results on the RWTH dataset and the CSL dataset are shown in Tables 1 and 2, respectively. The WER curves generated for each epoch in the training process are shown in Figs. 5(a) and 5(b).

| Methods        | Backbone   | Dev  | Test |
|----------------|------------|------|------|
| Align-iOpt\textsuperscript{32} | 3D-ResNet   | 37.1 | 36.7 |
| Re-Sign\textsuperscript{13} | GoogLeNet  | 27.1 | 26.8 |
| SFL\textsuperscript{26} | ResNet18   | 26.2 | 26.8 |
| CNN + LSTM + HMM\textsuperscript{23} | GoogLeNet  | 26.0 | 26.0 |
| CrossModal\textsuperscript{25} | BN-Inception | 23.9 | 24.0 |
| FCN\textsuperscript{19} | Custom     | 23.7 | 23.9 |
| SLRGAN\textsuperscript{34} | BN-Inception | 23.7 | 23.4 |
| DNF\textsuperscript{1} | GoogLeNet  | 23.1 | 22.9 |
| CMA\textsuperscript{35} | GoogLeNet  | 21.3 | 21.9 |
| VAC\textsuperscript{18} | ResNet18   | 21.2 | 22.3 |
| STMC\textsuperscript{26} | VGG11      | 21.1 | 20.7 |
| SEN\textsuperscript{5} | ResNet18   | 19.5 | 21.0 |
| Our methods    | ResNet34   | 20.3 | 21.4 |
As can be seen from Tables 1 and 2, the proposed model achieves better performance compared to other advanced models on both datasets. For the RWTH dataset, our model achieves 20.3% and 21.4% on the validation and test sets, respectively. On the CSL dataset, the best result reaches 0.7%. The effectiveness of the proposed model is demonstrated. It can be seen from the curve that WER decreases with the increase of epoch. WER decreases significantly when the first learning rate changes. RWTH reaches the minimum value at the 57th epoch, and CSL reaches the minimum value at the 13th epoch.

To further demonstrate the effectiveness of the proposed model, we present some sample outputs in Fig. 6, showing the recognition results on the RWTH validation set. Where “label” represents the real sentence corresponding to the sequence, “GT” represents the correspondence between the real sentence and the sequence, the correspondence between the sentence and the sequence in “GT” is the result of manual alignment, “Ours” represents the recognition result of the model in this paper, and “red” represents the wrong one recognition. The video frame sequence in Fig. 6 is the result of downsampling the original video frame sequence. It can be seen from Fig. 6: for the first sample, the sentence can be accurately recognized; in the second

| Methods   | WER (%) |
|-----------|---------|
| Align-iOpt32 | 6.1     |
| DPD37     | 4.7     |
| SF-Net24  | 3.8     |
| FCN19     | 3.0     |
| CrossModal32 | 2.4   |
| STMC36    | 2.1     |
| SLRGAN34  | 2.1     |
| VAC18     | 1.6     |
| CAS-GCN38 | 1.2     |
| STENet17  | 0.9     |
| CorrNet2  | 0.8     |
| Our methods | 0.7    |

Fig. 5 WER variation curves for the validation and test sets of the (a) RWTH dataset and (b) CSL dataset.

As can be seen from Tables 1 and 2, the proposed model achieves better performance compared to other advanced models on both datasets. For the RWTH dataset, our model achieves 20.3% and 21.4% on the validation and test sets, respectively. On the CSL dataset, the best result reaches 0.7%. The effectiveness of the proposed model is demonstrated. It can be seen from the curve that WER decreases with the increase of epoch. WER decreases significantly when the first learning rate changes. RWTH reaches the minimum value at the 57th epoch, and CSL reaches the minimum value at the 13th epoch.

To further demonstrate the effectiveness of the proposed model, we present some sample outputs in Fig. 6, showing the recognition results on the RWTH validation set. Where “label” represents the real sentence corresponding to the sequence, “GT” represents the correspondence between the real sentence and the sequence, the correspondence between the sentence and the sequence in “GT” is the result of manual alignment, “Ours” represents the recognition result of the model in this paper, and “red” represents the wrong one recognition. The video frame sequence in Fig. 6 is the result of downsampling the original video frame sequence. It can be seen from Fig. 6: for the first sample, the sentence can be accurately recognized; in the second
sample, the third word is misidentified from “NICHT-KAUM” to “KAUM,” and the word “_Pu_” is misidentified between the third and fourth words.

4.5 Ablation Studies

In this section, we conduct ablation experiments on the RWTH dataset to further verify the effectiveness of the model. All ablation experiments use WER as the metric.

(1) Experiment 1: The number of 1DCNNs in MST-block. The MST-block proposed in this paper utilizes different scales of temporal receptive fields to improve the ability of sequence modeling. It mainly consists of multiple 1DCNNs with different convolution kernels in parallel to form a multiscale feature extraction part. The influence of the number of 1DCNNs used on the experimental results is shown in Table 3. The kernel size of the starting 1DCNN is 3, and each additional 1DCNN increases the kernel size by 2.

It can be seen that with the increase of the number of 1DCNNs, the WER in this table has a tendency to decrease first and then increase. When the number of 1DCNNs increases to 4, the recognition effect is the best, and the WER decreases to 20.3%, and the WER starts to rise slightly when it increases to 5 1DCNNs.

(2) Experiment 2: The number of FC layers used after backbone. In this paper, the function of adding two FC layers after backbone is to integrate the features in the image feature map after multiple convolution layers and pooling layers to obtain the high-level meaning of image features. The influence of the number of FC layers used on the experimental results is shown in Table 4.

It can be seen that the value of WER is 21.0% when no FC layer is added. As the number of FC layers increases, the WER first decreases and then increases. When the number of FC layers is 2, it reaches a minimum value of 20.3%.

(3) Experiment 3: Enhancement coding method. In this paper, for the temporal feature vector obtained by MST-block, the temporal features are further encoded using transformers encoding module so as to make the obtained temporal features more accurate. For the effect of using different enhanced coding methods on the experimental results, it is shown

Table 3 Experimental results of different numbers of 1DCNNs in MST-block on RWTH.

| Kernel | Dev (%) | Test (%) |
|--------|---------|----------|
| 1      | 21.5    | 22.8     |
| 2      | 21.4    | 22.0     |
| 3      | 20.7    | 21.0     |
| 4      | 20.3    | 21.4     |
| 5      | 20.4    | 21.6     |

Note: bold values indicate the optimal values for each ablation experiment.
in Table 5. We use BiLstm and transformers for further encoding of temporal features for comparison.

It can be seen that using the transformers module can better encode the temporal features compared to BiLstm. BiLstm is widely used for temporal feature modeling in CSLR, and it works well. Here the reason why BiLstm is not as good as transformers are that when further encoding the temporal features extracted by MST-block, it is necessary to pay more attention to global features, whereas BiLstm pays more attention to local features.

(4) Experiment 4: The number of CTC losses for multilevel CTC loss. In this paper, to address the problem that the deep neural network uses the backward propagation algorithm based on the chain rule to train the network, which to a certain extent causes the shallow network parameters to not be well learned and updated, multilevel CTC loss is proposed to decode the temporal features and then efficiently train the feature extraction network and temporal modeling network to further improve the recognition performance. For the effect of using the number of CTC losses on the experimental results, it is shown in Table 6.

Levels 1–4 CTC loss correspond to levels 1–4 gloss features, and CTC levels are retained in reverse order when performing multilevel CTC ablation experiments. That is, when one CTC is used, the last CTC level is retained, and when two CTCs are used, the last two CTC levels are retained. When only two CTCs are used in the ablation experiments, then the used is the sum. As can be seen in Table 6, the WER is in a downward trend as the number of multilevel CTCs increases.

| The number of CTC losses | Dev (%) | Test (%) |
|-------------------------|---------|----------|
| 1                       | 28.1    | 28.4     |
| 2                       | 22.1    | 23.4     |
| 3                       | 21.0    | 22.1     |
| 4                       | 20.3    | 21.4     |

Note: bold values indicate the optimal values for each ablation experiment.

Table 4 Experimental results of adding different number of FC layers on RWTH after backbone.

| The number of FC layers | Dev (%) | Test (%) |
|-------------------------|---------|----------|
| 0                       | 21.0    | 21.5     |
| 1                       | 20.8    | 21.1     |
| 2                       | 20.3    | 21.4     |
| 3                       | 21.2    | 21.8     |

Note: bold values indicate the optimal values for each ablation experiment.

Table 5 Experimental results of different enhancement coding methods on RWTH.

| Enhancement coding method | Dev (%) | Test (%) |
|---------------------------|---------|----------|
| BiLstm                    | 21.3    | 21.7     |
| Transformers              | 20.3    | 21.4     |

Note: bold values indicate the optimal values for each ablation experiment.

Table 6 Experimental results of different numbers of CTC losses on RWTH.

| The number of CTC losses | Dev (%) | Test (%) |
|--------------------------|---------|----------|
| 1                        | 28.1    | 28.4     |
| 2                        | 22.1    | 23.4     |
| 3                        | 21.0    | 22.1     |
| 4                        | 20.3    | 21.4     |

Note: bold values indicate the optimal values for each ablation experiment.
5 Conclusion

This paper proposes a MSTNet for CSLR. In this work, to address the lack of accurate annotation of data time series in CSLR, we propose a multiscale temporal receptive field approach to obtain more accurate temporal features to improve the accuracy of CSLR. The resulting temporal features will be more accurate compared to the method of extracting temporal features using a fixed temporal receptive field in recent research works.\textsuperscript{1,18,19} It will also be richer and more comprehensive than the method in Ref. 11 that generates temporal features of different scales using different strides of pooling for temporal features. Furthermore, our proposed multilevel CTC loss is able to better learn and update the shallow network parameters in the CNN, effectively train the network and further improve the model recognition performance. This is superior to the multiscale perceptual loss used in Ref. 11 because their proposed multiscale perceptual loss is not helpful for the training of shallow networks. In our proposed model MSTNet, the temporal feature extraction part first extracts temporal receptive field features of different scales through the proposed MST-block and combines them into a candidate space, which is fused through learnable parameters, greatly improving the temporal modeling capability, and then encodes them by the transformers module to better obtain remote dependence information, which further improves the accuracy of the final temporal features. Finally, the network is trained with the proposed multilevel CTC loss to improve the accuracy of sign language recognition. The entire network is trained end-to-end and experimentally validated on two large-scale sign language datasets, and the experimental results demonstrate the effectiveness of the model proposed in this paper.

We believe that a possible future research direction for CSLR is to keep the final recognition rate essentially constant after downsampling the temporal dimension of the input sign language video data. There have been some similar studies, such as video super-resolution,\textsuperscript{39,40} but there is no such study in CSLR yet. The sign language actions in the video are continuous in time, and they are described using fewer frames to reduce redundancy, which in turn can improve network execution efficiency and reduce memory usage. Therefore, downsampling in the temporal dimension is an effective way to improve the real-time performance of CSLR. However, downsampling will lead to accuracy degradation. How to keep the accuracy basically unchanged while downsampling is a problem worth studying.

Disclosures

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and Data Availability

The datasets used in the paper are cited properly. Code is available at https://github.com/woshi-sad159/MSTNet.git.

Acknowledgments

This work was supported in part by the Development Project of Ship Situational Intelligent Awareness System, China (Grant No. MC-201920-X01), the National Natural Science Foundation of China (Grant No. 61673129), and the Young Talent Fund of Association for Science and Technology in Shaanxi, China (Grant No. qsy007).

References

1. R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuous sign language recognition by iterative training,” \textit{IEEE Trans. Multimedia} \textbf{21}(7), 1880–1891 (2019).
2. L. Hu et al., “Continuous sign language recognition with correlation network,” in \textit{Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.}, pp. 2529–2539 (2023).
3. P. Xie et al., “Multi-scale local-temporal similarity fusion for continuous sign language recognition,” \textit{Pattern Recognit.} \textbf{136}, 109233 (2023).
4. H. Hu et al., “SignBERT+: hand-model-aware self-supervised pre-training for sign language understanding,” \textit{IEEE Trans. Pattern Anal. Mach. Intell.} \textbf{45}, 11221–11239 (2023).
5. L. Hu et al., “Self-emphasizing network for continuous sign language recognition,” \textit{Proc. AAAI Conf. Artif. Intell.} \textbf{37}(1), 854–862 (2023).
6. Q. Xiao et al., “Multi-information spatial–temporal LSTM fusion continuous sign language neural machine translation,” *IEEE Access* 8, 216718–216728 (2020).
7. M. Hassan, K. Assaleh, and T. Shanableh, “Multiple proposals for continuous Arabic sign language recognition,” *Sens. Imaging* 20(1), 1–23 (2019).
8. A. Choudhury et al., “Movement epenthesis detection for continuous sign language recognition,” *J. Intell. Syst.* 26(3), 471–481 (2017).
9. S. Sharma, R. Gupta, and A. Kumar, “Continuous sign language recognition using isolated signs data and deep transfer learning,” *J. Ambient Intell. Hum. Comput.* 14, 1531–1542 (2021).
10. O. Koller et al., “Deep sign: enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs,” *Int. J. Comput. Vision* 126(12), 1311–1325 (2018).
11. C. Wei et al., “Semantic boundary detection with reinforcement learning for continuous sign language recognition,” *IEEE Trans. Circuits Syst. Video Technol.* 31(3), 1138–1149 (2020).
12. D. Guo et al., “Hierarchical LSTM for sign language translation,” *Proc. AAAI Conf. Artif. Intell.* 32(1), 6845–6852 (2018).
13. 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. Vision and Pattern Recognit.*, pp. 4297–4305 (2017).
14. Y. Hao et al., “Spatiotemporal consistency-enhanced network for video anomaly detection,” *Pattern Recognit.* 121, 108232 (2022).
15. Z. Zhao, S. Zhao, and J. Shen, “Real-time and light-weighted unsupervised video object segmentation network,” *Pattern Recognit.* 120, 108120 (2021).
16. Z. Huang et al., “LSTM based trajectory prediction model for cyclist utilizing multiple interactions with environment,” *Pattern Recognit.* 112, 107800 (2021).
17. W. Yin et al., “Spatial temporal enhanced network for continuous sign language recognition,” *IEEE Trans. Circuits Syst. Video Technol.* (2023).
18. Y. Min et al., “Visual alignment constraint for continuous sign language recognition,” in *Proc. IEEE/CVF Int. Conf. Comput. Vision.*, pp. 11542–11551 (2021).
19. K. L. Cheng et al., “Fully convolutional networks for continuous sign language recognition,” *Lect. Notes Comput. Sci.* 12369, 697–714 (2020).
20. H. Li and W. Wang, “Reinterpreting CTC training as iterative fitting,” *Pattern Recognit.* 105, 107392 (2020).
21. V. Mazzia et al., “Action transformer: a self-attention model for short-time pose-based human action recognition,” *Pattern Recognit.* 124, 108487 (2022).
22. P. Xie, M. Zhao, and X. Hu, “PISLTrc: position-informed sign language transformer with content-aware convolution,” *IEEE Trans. Multimedia* 24, 3908–3919 (2021).
23. O. Koller et al., “Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos,” *IEEE Trans. Pattern Anal. Mach. Intell.* 42(9), 2306–2320 (2019).
24. Z. Yang et al., “SF-Net: structured feature network for continuous sign language recognition,” arXiv:1908.01341 (2019).
25. K. Koishybay, M. Mukushev, and A. Sandygulov, “Continuous sign language recognition with iterative spatiotemporal fine-tuning,” in *25th Int. Conf. Pattern Recognit. (ICPR)*, IEEE, pp. 10211–10218 (2021).
26. Z. Niu and B. Mak, “Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition,” *Lect. Notes Comput. Sci.* 12361, 172–186 (2020).
27. O. Koller, J. Forster, and H. Ney, “Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers,” *Comput. Vision Image Understanding* 141, 108–125 (2015).
28. J. Huang et al., “Video-based sign language recognition without temporal segmentation,” *Proc. AAAI Conf. Artif. Intell.* 32(1), 2257–2264 (2018).
29. K. He et al., “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
30. O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vision* 115(3), 211–252 (2015).
31. D. Guo et al., “Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation,” *IEEE Trans. Image Process.* 29, 1575–1590 (2019).
32. J. Pu, W. Zhou, and H. Li, “Iterative alignment network for continuous sign language recognition,” in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 4165–4174 (2019).
33. I. Papaioannou et al., “Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space,” *IEEE Access* 8, 91170–91180 (2020).
34. I. Papaioannou, K. Dimitropoulos, and P. Daras, “Continuous sign language recognition through a context-aware generative adversarial network,” *Sensors* 21(7), 2437 (2021).
35. J. Pu et al., “Boosting continuous sign language recognition via cross modality augmentation,” in *Proc. 28th ACM Int. Conf. Multimedia*, pp. 1497–1505 (2020).
36. H. Zhou et al., “Spatiotemporal multi-cue network for continuous sign language recognition,” *Proc. AAAI Conf. Artif. Intell.* 34(07), 13009–13016 (2020).
37. H. Zhou, W. Zhou, and H. Li, “Dynamic pseudolabel decoding for continuous sign language recognition,” in *IEEE Int. Conf. Multimedia and Expo (ICME)*, IEEE, pp. 1282–1287 (2019).

38. Q. Guo et al., “Interactive attention and improved GCN for continuous sign language recognition,” *Biomed. Signal Process. Control* **85**, 104931 (2023).

39. K. C. Chan et al., “Understanding deformable alignment in video super-resolution,” *Proc. AAAI Conf. Artif. Intell.* **35**(2), 973–981 (2021).

40. H. Liu et al., “Video super-resolution based on deep learning: a comprehensive survey,” *Artif. Intell. Rev.* **55**, 5981–6035 (2022).

Biographies of the authors are not available.