Dense Temporal Convolution Network for Sign Language Translation

Dan Guo[1] , **Shuo Wang**[1] , **Qi Tian**[2] and **Meng Wang**[1]
[1]School of Computer Science and Information Engineering, Hefei University of Technology
[2]Huawei Noah's Ark Lab Department of Computer Science, University of Texas at San Antonio
guodan@hfut.edu.cn, shuowang.hfut@gmail.com, tian.qi1@huawei.com, eric.mengwang@gmail.com

Abstract

The sign language translation (SLT) which aims at translating a sign language video into natural language is a weakly supervised task, given that there is no exact mapping relationship between visual actions and textual words in a sentence label. To align the sign language actions and translate them into the respective words automatically, this paper proposes a dense temporal convolution network, termed *DenseTCN* which captures the actions in hierarchical views. Within this network, a temporal convolution (TC) is designed to learn the short-term correlation among adjacent features and further extended to a dense hierarchical structure. In the k^{th} TC layer, we integrate the outputs of all preceding layers together: (1) The TC in a deeper layer essentially has larger receptive fields, which captures long-term temporal context by the hierarchical content transition. (2) The integration addresses the SLT problem by different views, including embedded short-term and extended long-term sequential learning. Finally, we adopt the CTC loss and a fusion strategy to learn the feature-wise classification and generate the translated sentence. The experimental results on two popular sign language benchmarks, *i.e.* PHOENIX and USTC-ConSents, demonstrate the effectiveness of our proposed method in terms of various measurements.

1 Introduction

There is a communication gap between deaf-mutes and normal people. People with no knowledge and experience of sign language always have difficulty in comprehending the expressions performed by the signers. Recently, the vision-based SLT system which recognizes and translates complicated variations of hand gestures, skeleton movements and facial expressions of singers provided an applicable communication platform to these people [Futane *et al.*, 2012]. However, there exist challenges in the sign language (SL) capturing and translating. Therefore, SL research has become significant in reality.

The SL task can be divided into two categories, namely the sign language recognition (SLR) and the SLT. The SLR is considered as a classification task [Guo *et al.*, 2017]. Different from the SLR task which uses dynamic visual SL action to represent a single word or a phrase, the video of SLT task usually consists of several SL actions and composes a continuous semantic description [Pu *et al.*, 2018].

In this paper, we proposed a model to learn the representation of actions in the video without any artificial alignment, and translate the representations into linguistic words. Recently, the 3D convolutional neural network (3D-CNN) which captures both sequential and spatial pixel distribution in continuous frames has shown better effectiveness to express the video and was used in many video-based tasks [Huang *et al.*, 2018; Pu *et al.*, 2018]. Therefore, we use the 3D-CNN for visual expression in our translation process. The recurrent neural network (RNN) was widely adopted in many sequential learning tasks, such as visual caption and textual translation [Guo *et al.*, 2018], however In [Cui *et al.*, 2017], however, the authors indicated that the RNN-based methods are more inclined to end in over-fitting when training on a limited amount of data. Meanwhile, RNN converged at a low speed while training [Pu *et al.*, 2018]. René *et al.* proposed a convolution-based architecture named encoder-decoder TCN (ED-TCN) for action segmentation and detection [René and Hager, 2017]. Their model avoided the problem of long-term dependencies ignoring and achieved better expressions than the RNN. Considering advances in ED-TCN for the sequential task, we design a temporal convolutional operation which calculates n adjacent features to capture the local-pattern contents in each convolution layer. We then extend the TC to a dense hierarchical structure and use it to learn the global-pattern contexts.

With the increasing of depth in the convolution network, although the more visual information have been viewed by the subsequent convolution layers, the details of contents have been weakened. To solve this problem, Huang *et al.* proposed a brilliant and effective connectionist structure called DenseNet which connects each layer to every other layer in a feed-forward fashion [Huang *et al.*, 2017]. Inspired by DenseNet, we employ the dense connectionist mechanism to keep and deliver the sequential contents in a multi-layered structure. Compared with the DenseNet which concerns the relations in the pixels, the DenseTCN focuses on the successive features across the temporal dimension. After that, we optimize the parameters of our network with the connection-

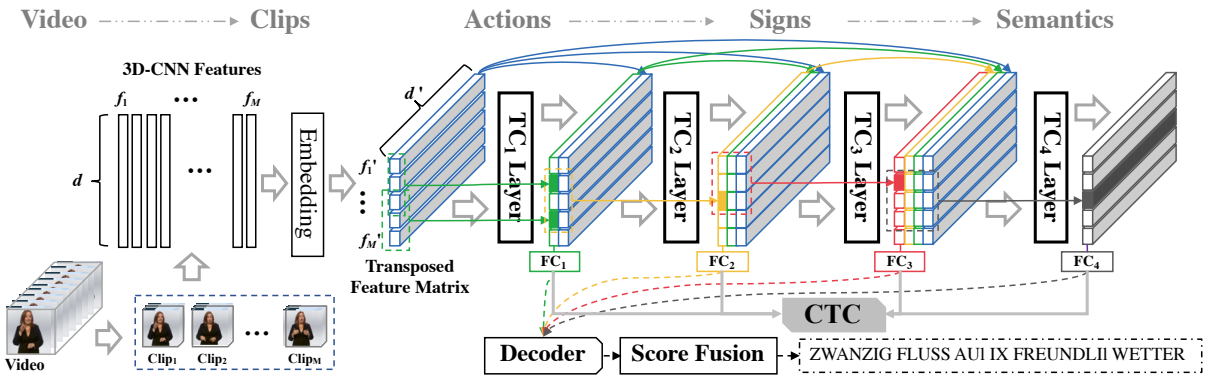

Figure 1: The overview of our $K = 4$ layers sign language translation framework.

ist temporal classification (CTC) [Graves *et al.*, 2006] object function which reliefs the strong alignment in the sequential data to handle the relations between the visual actions and textual words.

As depicted in Figure 1, we first split the video into clips and extract the feature of each clip from the 3D-CNN, which captures the sequential and the spatial information simultaneously from the successive original frames in our datasets. Then, the multi-layered TC structure is developed for calculating adjacent features in different receptive fields. Meanwhile, we concatenate the outputs of all preceding layers and use them as the input of the current calculation layer. In other words, the deeper TC layer not only focuses on the long-term contexts but also contains the contents from other receptive fields. In the training stage, we use the CTC to learn the relationship between the translated and the real sentences in each TC layer. In the testing stage, the greedy decoder and the fusion strategy are used to find a more reliable sentence.

The main contributions of our work are two folds:

- We propose an end-to-end trainable multi-layer structure which is designed in pure temporal convolution for SLT problem. This structure catches the changes in the details of the actions.
- We develop the connectionist mechanism to keep and deliver the contents from different calculation layers. This mechanism fuses the diversity of multi-views observations and improves the expression of the current moment clip.

2 Related Works

The solution of SLT contains two sub-processes: feature extraction and sequence translation. Previous works usually utilize the hand-craft feature as the visual expression to learn the hidden semantic information from the sequential actions. In [Quattoni *et al.*, 2007], the human skeletons which describe the postures trajectory were used to recognize the action from the video. In [Guo *et al.*, 2017], the authors added the histogram of oriented gradient (HOG) descriptor of hand into an adaptive Hidden Markov Model (HMM) model for bettering the sign classification. In addition, Koller *et al.* encoded each frame from the video by a 3D-HOG [Koller *et al.*, 2015] algorithm to solve the SLT problem. As for sequence translation, the traditional methods, such as HMM [Koller *et al.*, 2017] and Hidden Conditional Random Fields (HCRF) [Quattoni *et al.*, 2007] were usually used in early works.

Recently, deep learning approaches has been proved the success in visual information capturing, such as image classification [Krizhevsky *et al.*, 2012], object detection [Girshick *et al.*, 2014], and video tasks [Szegedy *et al.*, 2015; Huang *et al.*, 2017]. As one of the popular techniques, 3D-CNN preserves the strong capacity for sequential visual comprehension and shown higher effectiveness than other traditional methods [Huang *et al.*, 2018; Pu *et al.*, 2018]. In addition, many DL-based sequential learning structures had been proposed and shown the powerful on the translation problem. For example, 2D-CNN features extracted by the GoogleNet [Szegedy *et al.*, 2015] and the VGG [Simonyan and Zisserman, 2015] of frames were fed into a bidirectional RNN model with the long short-term memory (LSTM) to generate words [Cui *et al.*, 2017]. In [Guo *et al.*, 2018], it considered the SLT as a video caption task that learns the visual semantic in the video and decodes the embedding vectors into sequential words by a hierarchical LSTM.

Cui *et al.*, however, indicated the RNN-based methods usually produce the over-fitting phenomenon when limited training data available [Cui *et al.*, 2017]. Therefore, in [Pu *et al.*, 2018], the authors proposed a structure which combines the 3D Residual Networks (3D-ResNet) [Hara *et al.*, 2017] and dilated convolution [Yu and Koltun, 2016] to translate the video into the sentence. To better the translation process, this combination was limited to use the Expectation-Maximization (EM) strategy which optimizes the feature extraction stage by the pseudo label generated from the sentence decoder stage and then to promote the accuracy of the SLT. And in [Wang *et al.*, 2018], the authors proposed a fusion layer combing the TCOV (short-term) and the BGRU (long-term) information for bettering translation without the EM strategy. In the translation stage, the CTC was proposed to solve the unequal sequence alignment problem, such as textual recognition, and vocal segmentation [Graves *et al.*, 2006].

The methods related to ours are recently proposed by [René and Hager, 2017] and [Huang *et al.*, 2017]. The key differences of ours are two folds. First of all, our model is proposed to solve the sequence-to-sequence SLT problem with

pure convolutional operations. We replace the traditional sequential learning units, *e.g.* RNN, LSTM, BGRU, etc, and tackle the over-fitting problem by focusing on the details with the TC. Second, compared with the DenseNet which aims at learning a good representation for image classification tasks, the proposed DenseTCN targets at temporal sequential learning. In a nutshell, DenseTCN combines short-term and long-term sequential learning on the feature matrix by TC operations. Our model learns the embedding of transposed feature matrix under the multi-granularity TC layers, which is fed into the CTC to learn the translation.

3 Our Proposed Method

We split the video into clips and extract its features by 3D-CNN. Then we will introduce the details of our DenseTCN as follow.

3.1 Dense TC

Compared with the RNN-based units, the TC avoids the problem of long-term dependencies and focuses on the adjacent features in sequential data during the calculation. As a result, we improve the TC to consider both current and preceding views by once calculation.

The operations of the k^{th} TC are shown in Figure 2. We consider a feature matrix contains M temporal features in d' dimension as input $H_k = \{h_i\}_{i=1}^{M} \in \mathbb{R}^{k \times M \times d'}$. Such matrix is concatenated of the outputs from the 0^{th} to $(k-1)^{\text{th}}$ calculation layer, we first pad it across the temporal dimension. Then we employ q TC filters to capture the dynamic visual information from the input by calculating n-item adjacent features. At last, we concatenate the outputs after all filters across the feature dimension into a matrix $\{h'_i\}_{i=1}^{M} \in \mathbb{R}^{M \times q}$ as the output of the k^{th} TC layer. In our method, the number of TC filters q can be set to any value. However, we aim to verify the effectiveness of our structure rather than the setting of parameters and we use the simplest strategy $q = d'$ in our experiments. We use H_k and O_k to represent the input and output of the k^{th} calculation layer, respectively. The entire calculation layers of our deep network are shown as follow.

$$\begin{cases} H_0 = \mathcal{F} & , k = 0 \\ O_0 = \mathbf{\Phi}(H_0) & , k = 0 \end{cases} \tag{1}$$

where $\mathbf{\Phi}$ is a embedding function which wraps the original input visual features $\mathcal{F}$ into transposed feature matrix and finds the appropriate embedding size in our method. We denote the formula (1) as the operations of the 0^{th} calculation layer. Then, the following layers are calculated by formula (2).

$$\begin{cases} H_k = [O_{k-1}, O_{k-2}, ..., O_0] & , k > 0 \\ O_k = TC_k(H_k) & , k > 0 \end{cases} \tag{2}$$

The notation O_k represents the output of the k^{th} TC calculation layer. Similar to the traditional convolutional network, the deeper layer has a larger respective field. Therefore, our TC structure responses the SL in hierarchical design by concatenating the outputs of all preceding calculation layers.

In each calculation layer, we apply the activation function to promote the learning ability of the network. In addition, we adopt the dropout [Krizhevsky *et al.*, 2012] to avoid the over-fitting and improve the generalization of our method.

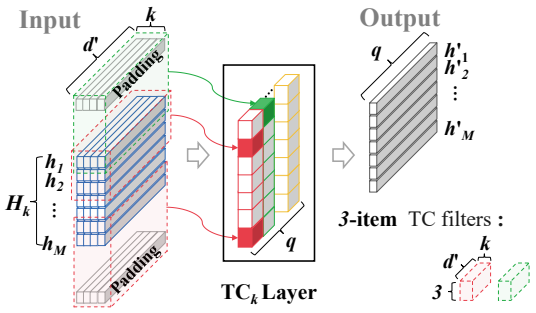

Figure 2: The operations of the TC_k layer (*i.e.* $k = 4$, $n = 3$).

3.2 Sentence Learning

In our method, we consider the CTC as our objective function to find a decoded sentence with the maximum sum of probabilities of various alignments $\boldsymbol{\pi}$ between input and target sequences [Graves *et al.*, 2006]. The CTC focuses on the order and correct words of predictions without the strict alignment operation. In other words, it provides an end-to-end training strategy for sequence-to-sequence learning where the unequal length in the different data.

In the learning stage, we introduce a new blank word '_' and add it into vocabulary Voc to construct a new word vocabulary as Voc'=$Voc \cup \{\text{'\_'}\}$, where Voc is a set which contains all words of the training set. Meanwhile, to predict the continuous probable words, we use a fully-connected (FC) layer after each TC layer, which transfers each clip feature into word vocabulary.

$$p_k = FC_k(O_k) = O_k \cdot W_k + b_k \quad , k > 0 \tag{3}$$

where p_k is a set containing prediction words features of the k^{th} TC layer. W_k and b_k are the parameters of the k^{th} FC. The size of these parameters is dependent on the setting of the convolution layer and the scale of word vocabulary in the training set.

Next, we compare the prediction p with the target sentence $\mathcal{Y}$ to optimize the parameters of our proposed network. Firstly, given the prediction p_k of the k^{th} TC layer, we transform it into sequence features as $\{p_k^i\}_{i=1}^{M}$. The probability of a CTC alignment path $\boldsymbol{\pi}_k$ is defined as follow.

$$\Pr(\boldsymbol{\pi}_k|p_k) = \prod_{j=1}^{M} \Pr(\pi_{k,j}|p_k), \forall \pi_{k,j} \in Voc' \tag{4}$$

where $\boldsymbol{\pi}_k$ has the same sequence length as p_k, $\pi_{k,j}$ is the j^{th} element of $\boldsymbol{\pi}_k$.

Then, to transform $\boldsymbol{\pi}_k$ into a variable sentence $\mathcal{Y}$, the CTC applies a many-to-one mapping operation $\mathcal{B}$ which removes the blank words and the repeated words in $\boldsymbol{\pi}_k$, *e.g.* $\mathcal{B}$(_ *a a* _ _ *pencil*)= {*a pencil*}. Therefore, the probability of a labeling $\mathcal{Y} = (y_1, y_2, ..., y_L)$ with L words is calculated as the summation of the probabilities of all word alignments:

$$\Pr(\mathcal{Y}|p_k) = \sum_{\boldsymbol{\pi}_k \in \mathcal{B}^{-1}(\mathcal{Y})} \Pr(\boldsymbol{\pi}_k|p_k) \tag{5}$$

where $\mathcal{B}^{-1}(\mathcal{Y}) = \{\boldsymbol{\pi}_k | \mathcal{B}(\boldsymbol{\pi}_k) = \mathcal{Y}\}$.

	Signers	Sentences	Videos	Words
TRAIN	9	5672	5672	1231
VAL	9	540	540	461
TEST	9	629	629	497

Table 1: PHOENIX Dataset

		Signers	Sentences	Videos	Words
Split I	TRAIN	40	100	4000	178
	TEST	10	100	1000	178
Split II	TRAIN	50	94	4700	178
	TEST	50	6	300	20

Table 2: USTC-ConSents Dataset

To learn the hierarchical views of the visual sequence, we combine the outputs of all FC layers with the CTC. Then we use the combination to optimize the whole parameters of the network. Therefore, assume that $P = \{p_k\}_{k=1}^{K}$ is the inputs of the CTC, where K is the depth of the DenseTCN, the CTC loss is defined as follow

$$\mathcal{L}_{CTC} = -\log \Pr(\mathcal{Y}|P) = -\sum_{k=1}^{K} \log \Pr(\mathcal{Y}|p_k) \tag{6}$$

This operation transforms the sequence translation problem to a hierarchical words translation task.

Score Fusion and Translation

Given the prediction set $P = \{p_k\}_{k=1}^{K}$ in the translation stage, where $p_k \in \mathbb{R}^{M\times w}$ and w are the sizes of word vocabulary. We use the *softmax* operation to normalize each prediction score vector, and sum different normalized score vectors,

$$p^{i}_{fusion,j} = \frac{1}{K}\sum_{k=1}^{K} \frac{e^{p^{i}_{k,j}}}{\sum_{j'=1}^{w} e^{p^{i}_{k,j'}}} \tag{7}$$

Then, we use the function *argmax* on p^{i}_{fusion} and output the i^{th} word classification label with the maximum score value. Finally, the greedy strategy is applied to delete the blank words '_' and merge the repetition in nearby words, *e.g. I _ I _ have _ a a _ pencil → I I have a a pencil → I have a pencil.*

4 Experiments

4.1 Datasets

We evaluate our method on two benchmarks: German continuous sign language dataset (PHOENIX)[1] and Chinese sign language dataset (USTC-ConSents)[2].

PHOENIX records the daily news and weather forecast airings of German sign language interpretation. It contains 6841 videos performed by 9 signers and each video is displayed by one related sentence. The statistic details are available in the Table 1.

[1]https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/

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

USTC-ConSents is a collection of videos covering 100 Chinese daily sentences played by 50 signers. To evaluate the effectiveness of our proposed method, we split the dataset by two strategies same to [Guo *et al.*, 2018] as "Split I" and "Split II" task in Table 2. (a) "Split I" is a signer independent test. The TRAIN set contains samples of 40 signers and the remaining of 10 signers as the TEST set. The sentences of the TEST set are existing in the TRAIN set. (b) "Split II" is an unseen sentences test which selects video samples of 94 sentences as the TRAIN set and the remaining 6 as the TEST set. Although sentences of the TEST set are completely different from the TRAIN set, the words in the TEST sentences exist in the TRAIN. Moreover, the glosses in the 6 test sentences sparsely occur in the training set and have much more difficult textual semantics.

4.2 Evaluation Metrics

Word error rate (WER) is a widely used metric which evaluates the similarity between two sentences. For each generated sentence, referenced to the ground truth sentence, such measurement counts the least operations of substitution (S), deletion (D), and insertion (I). Then, we denote the number of words in the ground truth sentence as G and the WER can be calculated as

$$\text{WER} = (S + D + I)/G \times 100\% \tag{8}$$

Lower WER means the higher accuracy of the translation process. In addition, there are two auxiliary evaluations *del* and *ins*, which represent the proportions of deletion and insertion operations calculated as follows.

$$del = D/G \times 100\% \quad , \quad ins = I/G \times 100\% \tag{9}$$

4.3 Network Setting

To fairly compare with other methods, we adopt the different 3D-CNN models according to [Pu *et al.*, 2018] and [Guo *et al.*, 2018] on two datasets. Assume a video with N frames as $\mathcal{V} = \{v_i\}_{i=1}^{N}$, we first split it into M clips as $\mathcal{C} = \{c_i\}_{i=1}^{M}$ with s-frames and overlapped by o-frames, where $M = \lfloor \frac{N-o}{s-o} \rfloor$ and $\lfloor z \rfloor$ returns the max integer that is less than z. Then, all clips are represented as fixed-length vectors $\mathcal{F} = \{f_1^d, f_2^d, ..., f_M^d\} = \{\mathbf{\Omega}_\theta(c_i)\}_{i=1}^{M}$ by passing through the 3D-CNN $\mathbf{\Omega}$ with parameters θ, where f_i^d is the d-dimensional vector of i^{th} clip. In the PHOENIX dataset, we split each video into 8-frames and overlapped by 4-frames. Therefore, we acquire 190536 / 17908 / 21349 clips from TRAIN / VAL / TEST sets, respectively. Through by the 18-layer 3D-ResNet [Hara *et al.*, 2017] with initializing parameters by the pre-trained model which trained on an SLR dataset [Zhang *et al.*, 2016], each clip is represented in the $d = 512$-dimensional vector. As for USTC-ConSents dataset, we set $s = 16$ and $o = 8$. Therefore, we achieve 111864 / 29080 and 131892 / 9052 clips in "Split I" task and "Split II" task of TRAIN / TEST set, respectively. Then the C3D [Huang *et al.*, 2015] is used to embed each clip into $d = 4096$-dimensional feature.

In order to find the suitable feature size in the training stage and reduce the parameters when the input contains the high-dimensional feature, we propose a linear embedding function

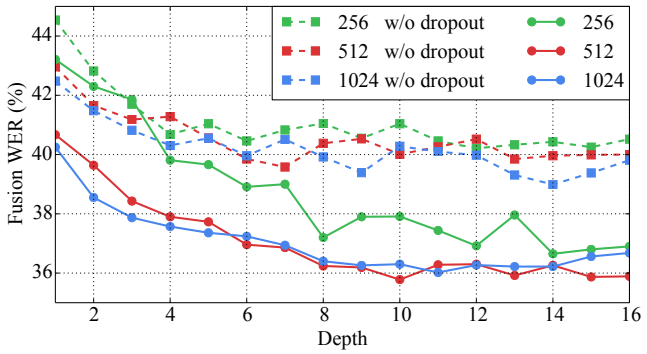

Figure 3: The fusion performance of DenseTCN with different strategies under PHOENIX VAL set.

$\mathbf{\Phi}$ which embeds the d-dimensional features into dimension d'. Then, defining the parameters format of $k^{\text{th}}(k > 0)$ TC as [number of filters, k, n, d, $padding$, $stride$], we set the parameters of k^{th} TC as $[d', k, 3, d', 1, 1]$, where $n = 3$ means that each calculation only concerns 3 adjacent features across the temporal dimension. The discussions of d' are analyzed in section 4.4. In the PHOENIX dataset, the feature of input is in $d = 512$ dimension, and the parameters W and b of each FC are the matrix in $\mathbb{R}^{d' \times 1232}$ and the vector in 1232-dimension, respectively, where 1232 is the size of vocabulary containing the blank label '_'. As for the USTC-ConSents dataset, the original input features are with dimension $d = 4096$, and the size of vocabulary w is 179.

In the training stage, we use ReLU as activation function [Krizhevsky *et al.*, 2012] and the parameter of the dropout is $\rho = 0.5$. Then, we train our network by CTC object function in ADAM optimization starting with the learning rate of 10^{-4}, beats range from 0.5 to 0.999, and weight decay is 10^{-5}. We reduce the learning rate by 0.1 after each 30 training epoch and stop the training stage when the learning rate is lower than 10^{-6}. In the testing stage, we remove all dropout layers.

4.4 Depth Discussion

We set the depth of network K range from 1 to 16, meanwhile, we use different size with or without (w/o) dropout to find the appropriate embedding vectors. Experiments on the PHOENIX VAL set, as depicted in Figure 3, it is easy to find the network with dropout is better. And with the increasing of the depth in our network, the performance of translation is better and more stability. Therefore, we set $K = 10$ and embedding size $d' = 512$ on PHOENIX dataset referred to Figure 3. However, there is no validation set in the USTC-ConSents dataset. Thus, we observe the statistics from two datasets: the linguistic word responses to the 3.5 and 5.5 clips on average in the PHOENIX and the USTC-ConSents dataset, respectively. Thus, we choose the deeper structure on the USTC-ConSents dataset to obtain a larger receptive field. In our experiments, we set $K = 16$ and $d' = 512$ which is the maximum value under the up-limit of calculation capacity of GPU as the network setting on this datasets.

Methods	VAL		TEST	
	del / *ins*	WER	*del* / *ins*	WER
HOG-3D ▲	25.8 / 4.2	60.9	23.2 / 4.1	58.1
CMLLR ▲	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1M-Hands ▲ △	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid ▲ △	12.6 / 5.1	38.3	11.1 / 5.7	38.8
Staged-Opt ▲ △	13.7 / 7.3	39.4	12.2 / 7.5	38.7
SubuNets ▲	14.6 / 4.0	40.8	14.3 / 4.0	40.7
Dilated-CNN △	8.3 / 4.8	38.0	7.6 / 4.8	37.3
LS-HAN	-	-	-	38.3
CTF-SLT	12.8 / 5.2	37.9	11.9 / 5.6	37.8
DenseNet*	-	49.7	-	49.2
Our DenseTCN	10.7 / 5.1	**35.9**	10.5 / 5.5	**36.5**

Table 3: Evaluations under PHOENIX (▲: Other modality, △: Extra supervision)

4.5 Comparison

PHOENIX

As is shown in Table 3, the notation ▲ denotes that the models were trained with other feature descriptors, such as "hand image", "trajectory motion", and "face image". △ means that the models utilized an additional off-line optimization, such as using multiple EM iterations on a hybrid CNN-HMM (CNN-Hybrid) framework for weak supervision [Koller *et al.*, 2016b]. Here we analyze the differences among these models. Both HOG-3D and CMLLR belonged to traditional HMM-based model learning with different hand-craft features [Koller *et al.*, 2015]. Then turning to deep features, Cui *et al.* proposed a three-step training optimization named Staged-Opt [Cui *et al.*, 2017]. In 1M-Hands [Koller *et al.*, 2016a] and SubuNets [Camgoz *et al.*, 2017], both hand features and global image features were used to solve the SLT problem. LS-HAN introduced the attention mechanism to measure the influences of all input sources to the current decoding position [Huang *et al.*, 2018]. And the model in Dilated-CNN [Pu *et al.*, 2018] was trained five times by the EM optimization procedure. In the CTF-SLT [Wang *et al.*, 2018], it combined the BGRU with the TCOV to focus both on the long-term and short-term contents from the video by joint CTC-based fusion.

Our model has CTC learning optimization and training the network in hierarchical views in one-step learning. Compared with the other methods based on the sequential calculation, our structure only uses the temporal convolution to focus on the short-term to the long-term contents. Our results outperform the state-of-the-art methods with only once end-to-end training. Furthermore, we replace our TC layers with the dense blocks introduced in DenseNet as DenseNet*, and the results are displayed Table 3 - 5 show that the operations of DenseNet are unsuitable for the sequential learning. It is because that DenseNet ignores the temporal relationship in the sequences. Here is a translation process of our $K = 10$ layers DenseTCN in Figure 4. For a video with 38 clips, we show the translated sentences of each TC layer. Referenced to "ground-truth", the values of the evaluation metric WER are range from 50% to 8%, and the fusion results show the correct translation sentence. In a nutshell, DenseTCN regains the deletion (D) words: "MOEGLICH" and "VERSCHIEDEN", picks the substitution (S) words:

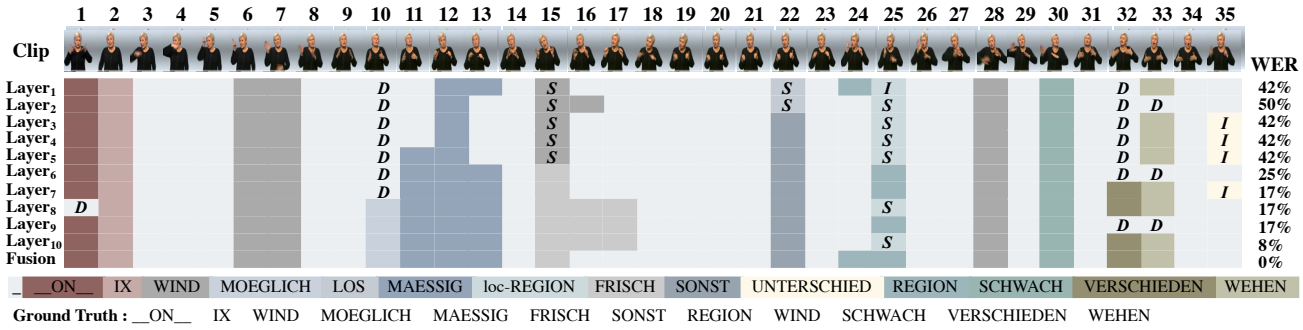

Figure 4: The predictions of the DenseTCN with depth $K = 10$.

Methods	WER
DTW-HMM [Zhang *et al.*, 2014]	28.4
LSTM [Venugopalan *et al.*, 2015b]	26.4
S2VT [Venugopalan *et al.*, 2015a]	25.5
LSTM-A [Yao *et al.*, 2015]	24.3
LSTM-E [Pan *et al.*, 2016]	23.2
HAN [Yang *et al.*, 2016]	20.7
LS-HAN [Huang *et al.*, 2018]	17.3
HLSTM-atten [Guo *et al.*, 2018]	**10.2**
CTF-SLT [Wang *et al.*, 2018]	11.2
DenseNet* [Huang *et al.*, 2017]	38.3
Our DenseTCN	14.3

Table 4: Evaluation under USTC-ConSents Split I

"WIND" and "loc-REGION", and excludes the insertion (I) word: "UNTERSCHIED". It demonstrates the effectiveness of our DenseTCN model.

USTC-ConSents

"Split I" task is designed to evaluate the translation process that already learns the sentence of the dataset and only changes the sign language users. Experimental results are shown in the Table 4. The DTW-HMM required segmentation in recognition process, and encoder-decoder RNN-based methods, such as LSTM, S2VT, LSTM-A, LSTM-E, HAN, LS-HAN and HLSTM-atten, are calculated the whole sequential data and generate sequential words. CTF-SLT also contains a long-term memory to remember the existing sentence. These methods perform better on "Split I" task, it is because that the RNN-based methods were easier into overfitting which tends to translate the sentence existing in the TRAIN set. But we achieve a comparable performance on this task.

"Split II" task is similar to the PHOENIX dataset in some aspects that each sentence never exists in the training set, but all words in the sentence have been contained in. Compared with "Split I" task, "Split II" task has meaningful in the practical application. As is shown in the Table 5, S2VT, HLSTM and other various structure all used the RNN-based methods that calculate or remember all sequential data by shared weights cells. Therefore, in "Split I" task, the RNN-based methods shown the better performance, but in "Split II" task, the new sentence never exists the training stage which means the remember strategy is invalid in the testing stage. But in our method, the hierarchical views on the temporal information are designed to learn the word-level representation and capture the SL actions in detail. It avoids the problem of long-term dependencies and relieves the influences of the words which were translated before. Therefore, our method achieves the best performance on "Split II" task, which performs much better than others by 19.4 - 22.3 WER improvement.

Methods	WER
S2VT [Venugopalan *et al.*, 2015a]	67.0
S2VT(3-layer) [Yao *et al.*, 2015]	65.2
HLSTM (SYS sampling) [Guo *et al.*, 2018]	66.3
HLSTM [Guo *et al.*, 2018]	66.2
HLSTM-atten [Guo *et al.*, 2018]	64.1
DenseNet* [Huang *et al.*, 2017]	52.1
Our DenseTCN	**44.7**

Table 5: Evaluation under USTC-ConSents Split II

5 Conclusion

This paper proposes a hierarchical structure which captures the visual contents from short-term to long-term transition to address the problem of SLT. In detail, with the increasing of depth in our DenseTCN, the translated words are observed from actions, signs, and semantics. It improves the performance of word-level translation and relieves the over-fitting phenomenon in the limited training dataset. Experiments on two popular SLT benchmarks have shown the effective performance of DenseTCN on different sides.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grants 61725203, 61732008, and 61876058.

References

[Camgoz *et al.*, 2017] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *ICCV*, pages 3075–3084, 2017.

[Cui *et al.*, 2017] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *CVPR*, pages 1610–1618, 2017.

[Futane *et al.*, 2012] PR Futane, RV Dharaskar, and VM Thakare. A comparative study for approaches for hand sign language. In *NCIPET*, pages 36–39, 2012.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[Graves *et al.*, 2006] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[Guo *et al.*, 2017] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Online early-late fusion based on adaptive hmm for sign language recognition. *TOCCAP*, 14(1):1–18, 2017.

[Guo *et al.*, 2018] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical lstm for sign language translation. In *AAAI*, 2018.

[Hara *et al.*, 2017] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV*, volume 2, page 4, 2017.

[Huang *et al.*, 2015] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Sign language recognition using 3d convolutional neural networks. In *ICME*, pages 1–6, 2015.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[Huang *et al.*, 2018] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, 2018.

[Koller *et al.*, 2015] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *CVIU*, 141:108–125, 2015.

[Koller *et al.*, 2016a] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *CVPR*, pages 3793–3802, 2016.

[Koller *et al.*, 2016b] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *BMVC*, page 12, 2016.

[Koller *et al.*, 2017] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *CVPR*, pages 4297–4305, 2017.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[Pan *et al.*, 2016] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016.

[Pu *et al.*, 2018] Junfu Pu, Wengang Zhou, and Houqiang Li. Dilated convolutional network with iterative optimization for coutinuous sign language recognition. In *IJCAI*, pages 885–891, 2018.

[Quattoni *et al.*, 2007] Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Morency Collins, and Trevor Darrell. Hidden conditional random fields. *TPAMI*, 29(10):1848–1852, 2007.

[René and Hager, 2017] Colin Lea Michael D Flynn René and Vidal Austin Reiter Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *ICCV*, pages 1003–1012, 2017.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. In *CVPR*, 2015.

[Venugopalan *et al.*, 2015a] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015.

[Venugopalan *et al.*, 2015b] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015.

[Wang *et al.*, 2018] Shuo Wang, Dan Guo, Wengang Zhou, ZhengJun Zha, and Meng Wang. Connectionist temporal fusion for sign language translation. In *MM*, pages 1483–1491, 2018.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NAACL HLT*, pages 1480–1489, 2016.

[Yao *et al.*, 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[Yu and Koltun, 2016] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

[Zhang *et al.*, 2014] Jihai Zhang, Wengang Zhou, and Houqiang Li. A threshold-based hmm-dtw approach for continuous sign language recognition. In *ICIMCS*, page 237, 2014.

[Zhang *et al.*, 2016] Jihai Zhang, Wengang Zhou, Chao Xie, Junfu Pu, and Houqiang Li. Chinese sign language recognition with adaptive hmm. In *ICME*, pages 1–6, 2016.