Spatio-Temporal Fusion based Convolutional Sequence Learning for Lip Reading

Xingxuan Zhang, Feng Cheng, Shilin Wang*
Shanghai Jiao Tong University
Shanghai, China
xingxuanzhang@hotmail.com, klaus.cheng@qq.com, wsl@sjtu.edu.cn

Abstract

Current state-of-the-art approaches for lip reading are based on sequence-to-sequence architectures that are designed for natural machine translation and audio speech recognition. Hence, these methods do not fully exploit the characteristics of the lip dynamics, causing two main drawbacks. First, the short-range temporal dependencies, which are critical to the mapping from lip images to visemes, receives no extra attention. Second, local spatial information is discarded in the existing sequence models due to the use of global average pooling (GAP). To well solve these drawbacks, we propose a Temporal Focal block to sufficiently describe short-range dependencies and a Spatio-Temporal Fusion Module (STFM) to maintain the local spatial information and to reduce the feature dimensions as well. From the experiment results, it is demonstrated that our method achieves comparable performance with the state-of-the-art approach using much less training data and much lighter Convolutional Feature Extractor. The training time is reduced by 12 days due to the convolutional structure and the local self-attention mechanism.

1. Introduction

lip reading, the ability to recognize the speaker's utterance based on the lip movements, is of great importance for various applications in computer vision, natural language processing and their intersection areas. For example, lip reading can perform as a liveness detector against replay attacks in identity authentication system [9]. Speech recognition performance can be boosted by integrating the visual (lip reading) and audio information, especially in the noisy environment [11]. Moreover, lip reading can also be applied in audio-video synchronization [13], improving hearing aid and silent dictation in public areas or a noisy environment.

Lip reading is also a difficult task for human and machines. Lip movements for different letters are visually similar to each other (e.g. b and p, d and t, etc). Hearing-impaired people can only get an accuracy less than 30% even for a very limited subset of 30 words [34]. Machine lip reading requires extracting spatio-temporal features from the video and mapping such high dimensional features to language, which is also a difficult learning task. Moreover, the complex texture around the lip area, such as teeth, mustache and great variations in the color of the face and lip, brings even more difficulties to lip reading.

Current state-of-the-art lip reading methods can be divided into three categories, RNN-based approaches [11, 12, 36], Transformer self-attention architecture with sequence-to-sequence loss (Transformer-seq2seq) and Transformer with Connectionist Temporal Classification loss (Transformer-CTC) [1]. The first two methods are originally developed for machine translation and the last one is first designed for audio speech recognition [5, 21, 8, 38]. Despite the successes of the methods in the fields they are designed for, directly apply these models in lip reading will not achieve the best performance. In this paper, we present a convolutional sequence-to-sequence model based on a novel temporal connected block and a Spatio-Temporal Fusion Module (STFM), which is specifically designed for lip reading and can well exploit the characteristics of the lip movements.

The first key factor of lip reading is to extract discriminative features to describe the lip movements from the input video. In most current methods, the low dimensional features input into the sequence model are extracted directly with the Convolutional Neural Networks (CNNs) followed by global average pooling [30]. To obtain global activation, global average pooling consumes local spatial information, which is critical to capture the subtle changes in the appearance and state of the lip. Considering the above issue, the global average pooling is replaced by the newly proposed STFM in our approach, which is able to reduce the feature dimension without losing the spatial information.

*Corresponing author

The second key factor is to map the extracted features describing the lip dynamics to the sequences of characters. Note that the mapping from images to sentences is not a one-to-one correspondence, i.e., a single viseme [27] or character may correspond to several input images, and a single word or sentence corresponds to several characters. We propose a novel convolutional block called Temporal Focal block (TF-block in short) to draw more attention to short-range temporal dependencies within the neighboring frames. Then a sequence to sequence model based on TF-blocks is used to map extracted features to sentences. This stacked convolutional structure naturally conforms to learn the multilevel mappings, i.e. the feature-viseme, viseme-word, word-sentence mappings. Moreover, local self-attention is adopted to capture long-range temporal dependencies, which is also important to viseme-word and word-sentence mappings. We find local self-attention is more efficient than global self-attention [10] while maintaining the recognition accuracy.

In addition, due to the network architecture, the optimization procedures in [11] and [1] are very time-consuming (10 days in [11] and 22 days in [1], respectively) and the optimal models are difficult to train. In [1], the Transformer model [40] was directly applied to lip reading. And optimizing the Transformer model requires considerable time and memory costs. Hence, Afouras et al. [1] trained the visual feature extraction CNN network and the Transformer network separately. Moreover, they also designed complex training strategies to train these two networks [11]. The training process costs approximately 22 days for the Transformer model (14 days for training the CNN feature extraction network and 8 days for training the Transformer network) on a single GPU. The slow training speed greatly limits the models transfer learning ability to a new or larger dataset. In contrast, our model is trained end to end and takes only 7 days to train on both LRS2 and LRS3 datasets contributing to the convolution based structure and local self-attention.

Without bells and whistles, our method achieves better results than the previous state-of-the-art approaches [11, 36], on GRID and LRW datasets. Using only part of the training samples, our method achieves comparable results on LRS2 and LRS3 datasets with [1].

2. Related work

In this section, we briefly review the previous works and related techniques in the literature as follows.

2.1. Automatic lip reading

Automatic lip reading mainly focus on two tasks: 1) the design of comprehensive and discriminative visual features and 2) the model design to map the visual features to the natural language. For the first task, the feature extraction can be varied in model-based features and image-based features. The model-based features, including active contour model [25], active shape model (ASM) [15] and active appearance model (AAM) [23], are robust to the variations of the environment illumination, speakers pose and distance towards the camera. The image-based features, including 2D discrete cosine transform feature(DCT) [17], articulatory features (AFs) [37], etc., contain more abundant information depicting the lip and its neighbouring regions, but are more vulnerable to the environment noises. For the second task, methods such as the hidden markov model (HMM) [6], support vector machine (SVM) [16], dynamic Bayesian network (DBN) [24], Temporal gradient-descent boosting (TGD-Boosting) [33], were adopted. Considering the success of deep learning, these two tasks can be integrated into one task based on convolutional neural networks (CNNs). The feature extraction and language prediction networks are trained jointly and influenced mutually. The powerful expressive ability of CNN and joint training bring a giant leap to the automatic lip reading. These deep learning methods include RNN-CTC [20], rnn-seq2seq [38] and, Transformer [40] which have been discussed in section 1.

2.2. Sequence-to-sequence model

The sequence-to-sequence (seq2seq in short) model is first proposed in [38] for natural machine translation (NMT). Seq2seq model follows an encoder-decoder structure. The encoder and decoder are usually formed with stacked recurrent neural networks. The encoder maps the input signal into latent hidden vectors and then propagate to the decoder. The decoder predicts the character at time t based on the encoder's output and character predicted at time $t-1$. Attention mechanism is brought by [5] to calculate attention weights of the encoder output, i.e. the decoders hidden state. The attention mechanism helps the decoder to draw attention to different time-step of the encoder output at different decoder time-step and thus produces a better result. Chung et al. [11] proposed a standard attention-based seq2seq model for sentence-level lip reading. Transformer proposed by [40] for NMT is also a seq2seq model. However, they only use residual dense layers whose temporal relations are learned by self-attention and vallina-attention. Triantafyllos et al. [1] evaluated Transformer for sentence-level lip reading. In contrast to the previous seq2seq models, we use a convolutional model to learn the spatial and temporal features from the video simultaneously and in the experiments, it is shown that the proposed model outperforms the state-of-the-art methods in lip reading.

2.3. Convolutional sequence-to-sequence model

The convolutional sequence-to-sequence model is firstly proposed by [19]. The conv-seq2seq model also follows an

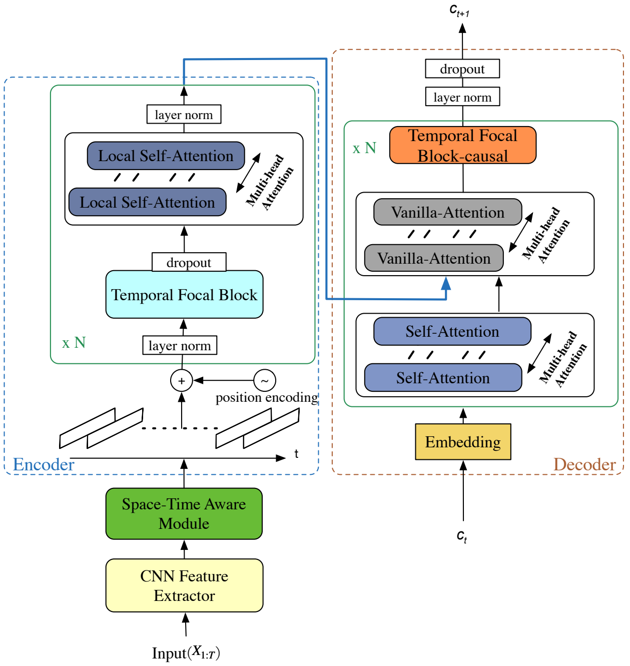


Figure 1: Architecture of our model. The model takes lip image sequences as input and outputs sequences of characters. The position encodings are added to features at the bottom of the encoder. During training, the label c_t at time step t is fed into the decoder to predict the output c'_{t+1}. The decoder utilizes the predicted characters $S'_{1:t} = [c'_1, c'_2, ..., c'_t]$ to predict the next character c'_{t+1} in the inference phase.

encoder-decoder structure with only convolutional layers in both the encoder and decoder. Conv-seq2seq has many advantages than RNN-seq2seq, including better parallelism, more stable gradient, more flexible reception field and lower memory requirement for training [19]. Recently, conv-seq2seq model has been widely investigated in many areas and achieved state-of-the-art performance, such as abstractive text generation [29], human dynamics prediction [28] and, text recognition [18]. Inspired by these works, a new conv-seq2seq model is proposed in this paper and will be discussed in the next section.

3. Approach

We first formalize the lip reading task. Given a lip-centered video $X_{1:T} = [x_1, x_2, ..., x_T]$, where $x_i \in R^{H \times W \times 3}$ is the image frame, the goal of lip reading is to generate the sentence that the speaker said $S_{1:L} = [c_1, c_2, ..., c_L]$, where $c_j \in D^v$ is the *j-th* character in the dictionary D of size v. Instead of directly mapping $X_{1:T}$ to $S_{1:L}$, we encode the spatio-temporal input $X_{1:T}$ into a hidden temporal feature $Z_{1:T} \in R^{T \times C}$ with the Convolutional Feature Extractor and Spatio-Temporal Fusion Module, and then map $Z_{1:T}$ to $S_{1:L}$ with the conv-seq2seq model as shown in Figure 1. In the following sections, we will elaborate the details of our Convolutional Feature Extractor, STFM and conv-seq2seq model.

3.1. Convolutional Feature Extractor

To extract the visual features $Y_{1:T}$ from the input image sequence $X_{1:T}$, a Convolutional Feature Extractor (CFE) is used as the front end. To capture the spatio-temporal characteristics of the lip dynamics, we adopt two layers of *3D* convolutions with kernel size of 5 on the input sequence. As for the following *2D* convolution, ResNet-18 instead of ResNet-50 structure is adopted in consideration of the memory and computational costs. To further accelerate the training, we decrease the spatial dimension by using a max-pooling layer after each *3D* convolution layer and removing some of the 2-stride operations in ResNet-18.

3.2. Spatio-temporal fusion module

Since the output of CNNs is of high dimensionality and cannot be directly used by the sequence model, most current lip reading methods adopt global average pooling to reduce the feature dimension. Global pooling, which can be regarded as a structure regularizer that explicitly enforces feature maps to be confidence maps of categories [30], is originally proposed in various classification tasks. It is common to use global pooling to average the class scores across the spatial dimensions. Global average pooling is shown to be effective in object localization [41, 31]. However, we find this localizing ability, only indicating the class activation and attention map of CNNs, is not capable of capturing continuous, subtle changes in the appearance of the lip. This is because activation in different spatial locations, which correspond to different visemes, may contribute the same to the final features generated by global pooling.

To fuse high-dimensional spatio-temporal features into low-dimensional temporal features while keeping the important local spatial information, we propose a Spatio-Temporal Fusion Module (STFM), as shown in Figure 2. To remove the fixed-size constraint of the features, STFM applies a SpatialPooling operation, similar to RoIPooling [22] but operating across the entire spatial dimension. Spatial Pooling extracts a small feature map $l_i \in R^{C \times n \times n}$ with a fixed size from each spatial feature $y_i \in R^{C \times W \times H}$, and then reshape the features $L_{1:T} = [l_1, l_2, ..., l_T] \in R^{T \times C \times n \times n}$ to $Z_{1:T} \in R^{T \times C'}$, where C is the number of input channels and C' is the number of output channels. Then $Z_{1:T}$ is fed into a stack of temporal convolutions to enhance communication between time steps and control the number of output channels. Note that if the spatial size of the input is fixed, the SpatialPooling can be generalized as max pooling.

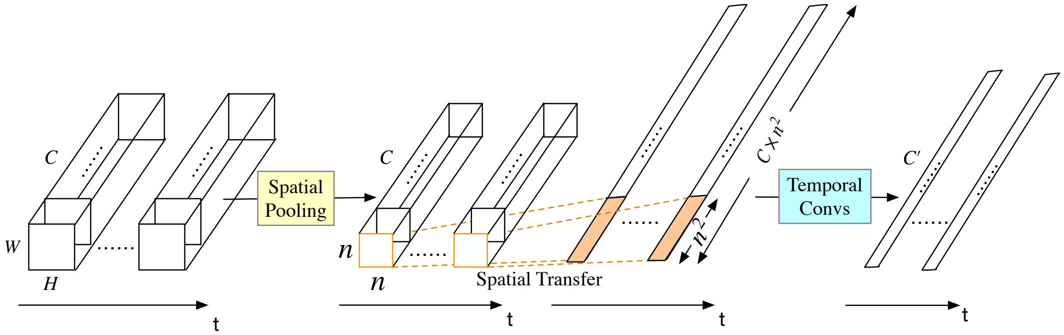


Figure 2: Spatio-Temporal Fusion Module(STFM). The input feature $Y_{1:T} \in R^{T\times C\times W\times H}$ is of high dimension and the output feature $Z_{1:T} \in R^{T\times C'}$ is of low dimension. C and C' donate the number of the channels of the input feature and output feature, respectively. The spatial pooling outputs feature $L_{1:T} = [l_1, l_2, ..., l_T] \in R^{T\times C\times n\times n}$ and n is the spatial dimension. The local spatial information of feature $Y_{1:T}$ is remained in feature $Z_{1:T}$.

3.3. Conv-seq2seq model

The conv-seq2seq model aims to map the extracted feature vector $Z_{1:T}$ to the natural language $S_{1:L}$. It should be noted that several continual frames of feature Z_t are corresponding to one viseme, while several continual visemes are corresponding to one word, and the sequential words compose the sentence $S_{1:L}$. The 'continual' characteristic can be perfectly learned via convolutional layers. The convolution operation uses a small kernel sliding across the whole sequence and can naturally learn the feature-viseme, viseme-word mappings. Therefore, we proposed a temporal-connection block called Temporal Focal block to look around each feature frame and focus on local dependencies to learn the 'continual' characteristic. Moreover, to look further and map words to sentences we use a local self-attention mechanism to capture long dependencies in temporal sequences.

Temporal focal block. The Temporal Focal block (TF-block) is proposed to help the features to look around their neighbours and capture short-range temporal dependencies. We introduce the TF-block starting from the one-dimensional convolution:

$$output_t^{c_o} = \sum_{c^i=1}^{C_i} \sum_{i=1}^{k} kernel_i^{c_i c_o} * input_{t-k/2-1}^{c_i} \tag{1}$$

where $kernel \in R^{k\times C_i\times C_o}$ is the convolution kernel, C_i is the number of input channels and C_o is the number of output channels, $output_t^{c_o}$ is the convolutional result at time t, channel c_o. The $output_t^{c_o}$ are learned from $input_t^{1:C_o}$ and its neighbouring k features. By employing convolution operation, $output_t^{c_o}$ focuses not only on the input feature at time-step t but also the neighbours and fuses these features together. As shown in Figure 3a, TF-block-a is a simple implementation consists of a branch with two convolution layers. Each convolution layer is followed by layer normalization [4] and Relu activation.

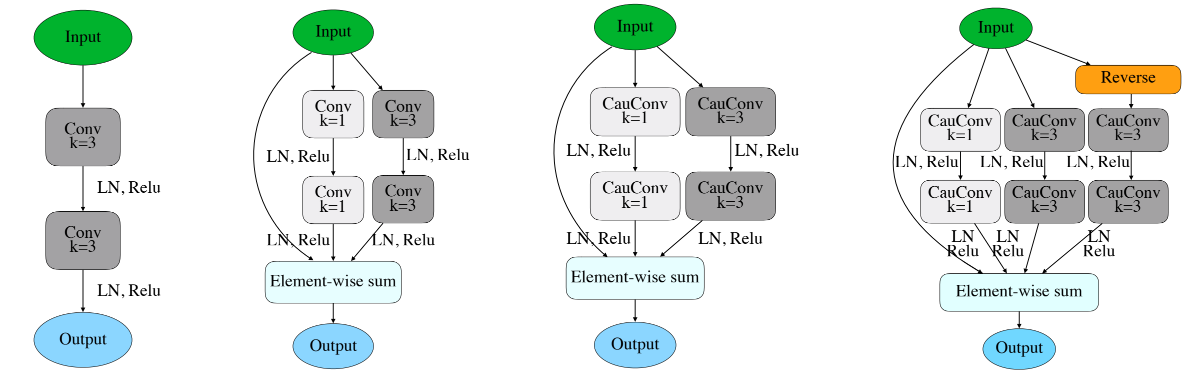
Moreover, TF-block should also be capable to learn more robust representations with speech rate invariance, which is the ability to extract correct semantic information regardless of the speech rate. So filters of different sizes are used to fuse the features at multiple scales. Here we simply add a convolution with kernel size of 1 and a shortcut connection as new branches to TF-block-a.

In a seq2seq model, the decoder should be future-blind, so normal convolution is not applicable to the decoder. We adopt causal convolution [39] to split the local fusion of features into two directions: forward and backward. The encoder can perform both forward and backward fusion just like BiLSTM [42] while the decoder only performs forward fusion. The formula of causal convolution is given in Equation 2. All the future information will be blocked by causal convolution. The TF-blocks based on unidirectional and bidirectional causal convolution are shown in Figure 3c, Figure 3d, respectively.

$$output_t^{c_o} = \sum_{c_i=1}^{C_i} \sum_{i=1}^{k} kernel_i^{c_i c_o} * input_{t-k+1}^{c_i} \tag{2}$$

Local self-attention. As discussed in subsection 3.2, much semantic information is implicitly contained in the whole sequence. Considering the whole sequence at each position can help learn the semantics contained in long-range temporal dependencies.

The self-attention mechanism is adopted to learn the semantics contained in long-range dependencies. Unlike the widely used vanilla-attention [5] whose attention weights are derived from the decoder hidden states and all the encoder output states, self-attention derives attention weights by comparing the features with its neighbours.



(a) TF-block-a (b) TF-block-b (c) TF-block-c (d) TF-block-d

Figure 3: Temporal Focal blocks (TF-blocks). Each convolution layer is followed by a layer normalization (donated by LN) and a Relu activation. ***k*** donates the kernel size of the convolution. ***Reverse*** is the same as the reverse operation described in [35], which is to flip the sequence along the time axis as the input to the new branch.

Define source states $H_s = [h_1, h_2, ..., h_{T_s}] \in R^{T_s \times C_s}$ and target states $H_t = [h_1, h_2, ..., h_{T_t}] \in R^{T_t \times C_t}$, where T_s and T_t, C_s and C_t are the source and target states temporal length and feature length at each time step respectively. In self-attention, H_s and H_t are the same, i.e. the learned features from look-around TC-blocks. We adopt Luong's multiplicative attention [32] as shown in Equation 3 to calculate the attention output.

$$\begin{gathered}\Lambda = softmax(\frac{H_t W_t (H_s W_s)^T}{\sqrt{C_s}}) \\ AttentionOutput = \Lambda H_s\end{gathered} \tag{3}$$

where $\Lambda \in R^{T_t \times T_s}$ is the attention weights, $AttentionOutput \in R^{T_t \times C_t}$ is the output of the attention, W_s and W_t are learnable parameters.

Since the original self-attention, referred to as global self-attention, calculates attention weights with the entire input sequence, the model complexity increases with the growth of the sequence length. However, we find it is not necessary to set the scope of dependencies the same as the sequence length in our experiments. A local self-attention mechanism is then proposed to capture the fixed range dependencies both in our encoder and decoder as Equation 4 shows.

$$\lambda = softmax(\frac{H_t W_t (H_s W_s)^T * W_m}{\sqrt{C_s}}) \tag{4}$$

where $W_m \in R^{T_t \times T_s}$ is a mask matrix. As our experiments will show, the local self-attention can accelerate the training procedure while maintaining the recognition accuracy.

Furthermore, to allow the model to jointly attend to information from different representation subspaces at different positions and reduce the computational complexity, we adopt multi-head attention to the attention layer [40].

Encoder-decoder. The structures of the encoder and decoder are shown in Figure 1. The encoder takes the feature Z extracted by CFE as input. The decoder takes the encoder outputs and previously predicted labels $S_{1:t} = [c_1, c_2, ..., c_t]$ to predict the next lable c_{t+1}. The encoder consists of N encoder-module which is formed by one TC-block and one self-attention layer. The decoder is composed of N decoder-module which is formed by one TC-block, one local self-attention layer and one vanilla-attention layer. The vanilla attention is of the same formula of self-attention but takes the outputs of the encoder as source states and decoder hidden states as target states.

3.4. Implementation details

For the Convolutional Feature Extractor, we set the dropout rate to 0.5. In STFM, the spatial size after SpatialPooling is set as $n = 5$ and the number of output channels is set to 512. In the conv-seq2seq model, the hidden size and the dropout rate are set to 512 and 0.1, respectively. The encoder-module and decoder-module stacked six times. The number of multi-head attention split is set to 8. The decoder output dictionary D, including the 26 letters $a - z$, 10 digits $0 - 9$, one punctuation mark " ' " and three tokens for $[PAD]$, $[EOS]$ and $[SPACE]$, is of size 40.

3.5. Training

In the training phase, the Adam [26] is employed as the optimizer with the default parameters. The model is trained end-to-end with the learning rate schedule strategy in [40], shown as Equation 5.

$$\begin{gathered}lr = d_{model}^{-0.5} * \min((factor * step)^{-0.5}, \\ (factor * step) * warmupSteps^{-1.5})\end{gathered} \tag{5}$$

where d_{model} is the hidden size in conv-seq2seq model. $step$ is the training iteration numbers. The lr will lin-

Dataset Method	LRW	LRS2-BBC	LRS3-TED
PW-FFN	22.0	59.2	70.5
TF-block-a	19.3	58.5	70.5
TF-block-b	**18.7**	**55.6**	**65.5**
TF-block-c	20.1	57.3	69.3
TF-block-d	18.8	55.9	66.0
TM-CTC	35.2	72.3	83.1
TM-seq2seq	22.1	60.5	70.8

Table 1: Word Error Rate (WER%) of different TF-blocks and Transformer based models. ***PW-FFL*** donates models with stacked position-wise feed-forward layers instead of TF-blocks. ***TF-block-a, b, c, d*** are the TF-blocks described in subsection 3.3. ***TM-CTC*** and ***TM-seq2seq*** are short for Transformer with sequence-to-sequence loss and Connectionist Temporal Classification loss, respectively [1]. Note that the results of TM-CTC and TM-seq2seq are reproduced with the datasets which are available to us. The training strategy, settings and datasets of all the methods are exactly the same.

Models	*DR	LRW	LRS2-BBC	LRS3-TED
TM-CTC	*GAP	35.2	72.3	83.1
	*STFM-s	34.7	70.4	81.6
	STFM	33.7	68.2	79.4
TM-seq2seq	*GAP	22.1	59.5	69.8
	*STFM-s	21.3	57.6	67.6
	STFM	20.5	55.0	65.0
Conv-seq2seq	*GAP	18.7	55.6	65.5
	*STFM-s	16.8	52.4	62.5
	STFM	**16.3**	**51.7**	**60.1**

*RD: The way to reduce the dimension of features.
*GAP: Global average pooling.
*STFM-s: STFM-simple.

Table 2: WER of the models with different modules to reduce the dimension of features before the sequence model. ***STFM-simple*** is a simplified STFM whose temporal convolution layers are replaced by convolution layers with kernel size of 1.

Method	Results			
	GRID	LRW	LRS2-BBC	LRS3-TED
WAS	3.0	23.8	70.4	-
Bi-LSTM	-	17.0	-	-
TM-CTC	-	-	54.7	66.3
TM-seq2seq	-	-	**48.3**	**58.9**
Ours	**1.3**	**16.3**	51.7	60.1

Table 3: Comparison with state-of-the-art approaches. ***WAS*** is short for "Watch, Attend and Spell" [11]. All the results (WER) except ours here are the results reported in [11, 36, 1].

early increase for the first $warmupSteps/factor$ iterations and decrease thereafter with the inverse square root decay. $factor$ is a normalizing parameter and is set to 1 on GRID, 0.1 on LRW, LRS2-BBC, LRS3-TED. When training on LRS2-BBC and LRS3-TED, the curriculum learning strategy used in [11] is adopted. We start training on single word examples and then let the sequence length grow as the network trains. The model is first trained on LRW dataset and the pre-train sets of LRS2-BBC and LRS3-TED. Then it is fine-tuned on the train-val set of LRS2-BBC and LRS3-TED separately. The training batch size is 50 on GRID and LRW, 12 on LRS2-BBC and LRS3-TED. The model is pre-trained on ImageNet and trained on a single GeForce Titan X GPU with 12GB memory.

In the test phase, the beam search decoder is applied to the decoder and the beam width is set to 5.

4. Experiments

In this section, we evaluate our method in comparison to the state-of-the-art. Ablation study is performed to show the effectiveness of each module of our method. We train the model as subsection 3.4 described and evaluate on the test set of GRID [14], LRW [12], LRS2-BBC [1], LRS3-TED [2] datasets with the corresponding trained model.

We adopt the Word Error Rate (WER) as our evaluation protocol, which compares the reference to the hypothesis and the calculation formula is: $WER = (S+D+I)/NUM$, where S, D and I are the numbers of substitutions, deletions and insertions respectively and NUM is the number of words in the reference.

4.1. Datasets and preprocessing

GRID dataset. The GRID dataset contains 34000 sentences uttered by 34 speakers. We followed the dataset division in [3], i.e., the data is randomly divided into the train, validation and test sets, where the latter contains 255 utterances for each speaker.

LRW dataset. The Lip Reading in the Wild (LRW) dataset [12] consists of 450,000 utterances each containing a single word out of a vocabulary of 500. The length of each video lasts 1.16 seconds (29 frames), and one word is uttered in the middle of the video.

Method	Training details on LRS2 and LRS3				
	CFE	Training Data	Video Duration	Training Time	End-to-end
WAS	VGG-M	LRW+ LRS2+ LRS3+ MV-LRS	1637.4	10d	✓
TM-CTC	3DCNNs+ ResNet50	LRW+ LRS2+ LRS3+ MV-LRS	1637.4	19d	
TM-seq2seq	3DCNNs+ ResNet50	LRW+ LRS2+ LRS3+ MV-LRS	1637.4	22d	
Ours	3DCNNs+ ResNet18	LRW+ LRS2+ LRS3	863	7d	✓

Table 4: Training details on LRS2-BBC and LRS3-TED. ***CFE*** is short for Convolutional Feature Extractor. The MV-LRS dataset is not public available. ***d*** is short for days. ***Video Duration*** is the total video duration of the datasets used to train the model. ***End-to-end*** donates whether the model is trained end-to-end.

LRS2-BBC dataset. The Lip Reading Sentences BBC dataset (LRS2) is a large-scale lip reading dataset composed of 143,000 utterances from BBC television. Each utterance contains a sentence with variable length. It contains over 2.3 million words with a vocabulary size of 41,000.

LRS3-TED dataset. The LRS3-TED dataset consists of about 150,000 utterances from TED and TEDx videos. It contains over 4.2 million words and the vocabulary size is 51,000.

Preprocessing. For all the datasets, we use dlib face and landmark detector [7] to detect the 68 facial key-points in all the video frames. Then affine transformation is performed based on three points, specifically two points at the outside eyes and the middle lowest point of the nose, to align the faces. Finally, a lip-centered image of size 112x112 is cropped from the aligned face. The width of the lip is normalized to occupy 1/3 of the image width.

4.2. Ablation experiments

To investigate the behavior of the proposed Temporal Focal blocks and STFM, we conducted several ablation experiments. Unless otherwise noted, all models use Convolutional Feature Extractor described in subsection 3.1 and both the encoder and decoder are composed of a stack of 6 layers. The hidden size is 512 and other settings remain the same in all comparative experiments unless otherwise specified.

Temporal Focal Block. To evaluate the importance of short-range temporal dependencies and local fusion, we first compare TF-block-a with stacked position-wise feed-forward layers(PW-FFL) [40], which can be regarded as two 1D temporal convolutions whose filter size is 1. The only difference between the two models is whether the features from adjacent time steps are fused to generate new features. Note that the PW-FFL based model is not the same as Transformer due to the position of PW-FFL. Two extra PW-FFls are added to the top of the encoder to make sure the PW-FFL based model to gain valid results. In the decoder, the future information cannot be exploited and thus each block in of our paper is used in the Table 1 encoder and the causal version of the corresponding block is used in the decoder. The TF-block-a based model achieves 2.7% lower WER than the position-wise feed-forward layer based model on LRW, which suggests that short-range dependencies are critical to recognition.

Results using different TF-blocks are shown in Table 1. TF-block-b has a 0.6% higher accuracy than TF-block-a on LRW dataset, indicating that feature fusion at multiple scales is important to lip reading. Compared to TF-block-b, TF-block-c which uses causal convolution in the encoder brings a 1.4% drop in accuracy. But with a bi-directional mechanism, the causal convolution based encoder is able to achieve similar accuracy.

Then we compare the convolutional seq2seq models based on different TF blocks with Transformer-CTC and Transformer-seq2seq on LRW and LRS2 datasets. Note that the training dataset MV-LRS used in [1] is not available to us and our Convolutional Feature Extractor is lighter, so the baselines reported here are inconsistent with the results in [1]. Compared to the best Transformer based model, i.e. Transformer-seq2seq, TF-block-b improves performance by 2.4% on LRW, 4.9% on LRS2-BBC and 5.3% on LRS3-TED.

Spatial-Temporal Fusion Module. We simplify the STFM by replacing the temporal convolution with a 1D convolution with kernel size of 1 to analyze the effect of local spatial information which may be consumed by global pooling. For models without STFM, two convolution layers with kernel size of 1 are added before global average pooling to maintain an approximate amount of parameters with STFM. We compare simplified STFM (STFM-simple) with global average pooling on the proposed Conv-seq2seq model, Transformer-CTC and Transformer-seq2seq. The results are given in Table 2. The Conv-seq2seq model is based on TF-block-b and achieves 1.9% higher accuracy on

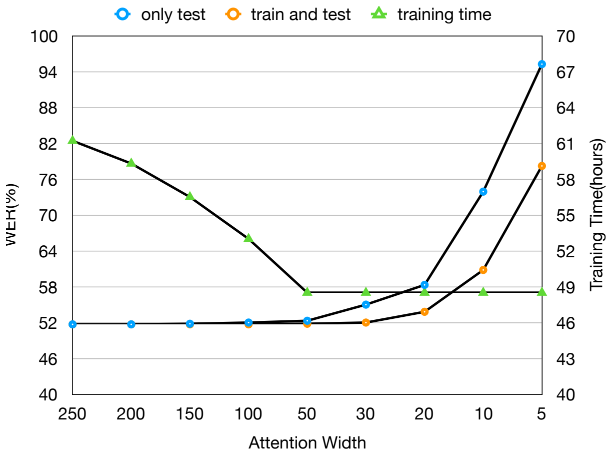


Figure 4: WER and training time of Conv-seq2seq models with different attention widths. ***only test*** indicates the WER of models trained with global self attention but tested with local self attention, while ***train and test*** indicates the WER of models trained and tested with local self attention. ***training time*** indicates the training time of sequence models with different attention width.

LRW using STFM-simple than global average pooling. For Transformer-CTC and Transformer-seq2seq, STFM-simple also leads to an increase in accuracy compared to global average pooling.

Additionally, we add a stack of two temporal convolution layers with kernel size of 3 before the convolution with kernel size of 1. This enhances communication between adjacent time steps and brings further improvement to accuracy, particularly on Transformer-CTC and Transformer-seq2seq because the feed-forward layers used in these model can not fuse adjacent features.

Global Self-attention vs. Local Self-attention. To better understand the effect of the width of attention on sequence model better, we analyze the distribution of the attention weights. For this, we train a Conv-seq2seq model based on TF-block-b with global self-attention, whose width is the same as the length of the input sequence. For each time step, we pick a fixed length area with the current position as the center point from the original weights as the selected range. During testing, we restrict the effective width of attention weights by replacing the weights outside the selected range with zero. Compared to global self-attention, local self-attention with a width larger than 100 achieves the same accuracy. When the attention width is 50, the accuracy only drops 0.8% as shown in Figure 4.

To further analysis the effect of the attention width in training, we train and test models with the same attention width. In Figure 4, the training of models with local self-attention is 20% faster than the one with global self-attention and causes little drop in accuracy.

4.3. Comparison to state-of-the-art methods

We evaluate our method on GRID, LRW, LRS2-BBC and LRS3-TED datasets and compare results to recent state-of-the-art methods. Results are presented in Table 3. On word-level datasets GRID and LRW, our method achieves 1.7% and 0.7% lower WER than the previous state-of-the-art methods [11, 36] respectively.

For a fair comparison against previous work [1, 11], we train our model on a single GPU. Restricted by the memory, the model can not be trained end-to-end with ResNet-50 as the feature extractor. We adopt the training strategy used in [1], which is training CFE first, and then training the sequence model with extracted features. MV-LRS(w), the dataset used to train CFE in [1] is not available to us, so we use LRS3-TED as an alternative. The performance is not comparable with the results reported in [1] because MV-LRS(w) is essential for training CFE and of the comparable size with the summation of LRS2-BBC and LRS3-TED, as shown in Table 4.

Then we adjust the training strategies by replacing ResNet-50 with ResNet-18 to train our model end-to-end and obtain our best results, as shown in Table 3. Note that our method achieves comparable results with state-of-the-art works but using much less training data and much lighter CFE. The training of WAS model takes approximately 10 days given the structure is simple. Among the three models with comparable results (TM-CTC, TM-seq2seq, ours), our method needs much less time to complete the training. More details about CFE and training data are shown in Table 4.

5. Conclusion

In this paper, we proposed the Spatio-Temporal fusion module (STFM) and a convolutional sequence-to-sequence model based on the temporal focal block (TF-block) for lip reading. Our STFM can be combined with most lip reading models to improve the utilization of local spatial information and the proposed TF-block can extract short-range temporal dependencies which are critical to lip reading. Our method achieves the state-of-the-art results on GRID and LRW datasets and comparable results with state-of-the-art approaches on LRS2-BBC and LRS3-TED datasets using much less training data and training time.

Acknowledgement

This research is fully supported by NSFC Fund (No.61771310).

References

[1] Triantafyllos Afouras, Joon Son Chung, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *CoRR*, abs/1809.02108, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. *CoRR*, abs/1809.00496, 2018.

[3] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. Lipnet: Sentence-level lipreading. *CoRR*, abs/1611.01599, 2016.

[4] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[6] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Statist.*, 37(6):1554–1563, 12 1966.

[7] Donatella Castelli and Pasquale Pagano. Opendlib: A digital library service system. In *European Conference on Research & Advanced Technology for Digital Libraries*, 2002.

[8] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *CoRR*, abs/1508.01211, 2015.

[9] Feng Cheng, Shi Lin Wang, and Wee Chung Liew. Visual speaker authentication with random prompt texts by a dual-task cnn framework. *Pattern Recognition*, 83:340–352, 2018.

[10] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733, 2016.

[11] Joon Son Chung, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. *CoRR*, abs/1611.05358, 2016.

[12] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 87–103, 2016.

[13] Joon Son Chung and Andrew Zisserman. Out of time: Automated lip sync in the wild. In *Computer Vision - ACCV 2016 Workshops - ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 251–263, 2016.

[14] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[15] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[17] Saeed Dabbaghchian, Masoumeh P. Ghaemmaghami, and Ali Aghagolzadeh. Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition*, 43(4):1431–1440, 2010.

[18] Yunze Gao, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Reading scene text with attention convolutional sequence modeling. 2017.

[19] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[20] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.

[21] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[23] Martial Hebert, Katsushi Ikeuchi, and Herve Delingette. A spherical representation for recognition of free-form surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(7):681–690, 1995.

[24] Kar Cherng Hon. Dynamic bayesian networks: representation, inference and learning. 2002.

[25] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[27] Oscar Koller, Hermann Ney, and Richard Bowden. Deep learning of mouth shapes for sign language. In *Workshop on Assistive Computer Vision & Robotics*, 2015.

[28] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. 2018.

[29] Wang Li, Junlin Yao, Yunzhe Tao, Zhong Li, and Du Qiang. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. 2018.

[30] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[31] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. Parsenet: Looking wider to see better. *CoRR*, abs/1506.04579, 2015.

[32] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[33] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. In *International Conference on Neural Information Processing Systems*, 1999.

[34] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. *CoRR*, abs/1501.05396, 2015.

[35] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[36] Themos Stafylakis and Georgios Tzimiropoulos. Combining residual networks with lstms for lipreading. *CoRR*, abs/1703.04105, 2017.

[37] Michael Studdert-Kennedy and Donald Shankweiler. Hemispheric specialization for speech perception. *The Journal of the Acoustical Society of America*, 48(2B):579–594, 1970.

[38] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[39] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[41] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015.

[42] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.