HMM-based Lip Reading with Stingy Residual 3D Convolution

Qifeng Zeng*, Jun Du* and Zirui Wang†
* University of Science and Technology of China, Hefei, China
E-mail: zqf0722@mail.ustc.edu.cn, ✉ E-mail: jundu@ustc.edu.cn
† Chongqing University of Posts and Telecommunications, Chongqing, China
E-mail: wangzr@cqupt.edu.cn

Abstract—**In this paper, we propose a novel approach for sentence-level lip-reading by using hidden Markov model (HMM) framework. To calculate the posterior probability of HMM states, the architecture of convolutional neural network based visual module followed by multi-headed self-attention Transformers is designed. Recently, 3D convolution for visual module to extract temporal features is popular for lip-reading tasks, which can achieve a higher accuracy at the cost of more computations compared with 2D convolution. This motivates us to invent plug-and-play compact 3D convolution unit called "Stingy Residual 3D" (StiRes3D). We use heterogeneous convolution kernels for different input channels, and apply channel-wise convolutions and point-wise convolutions to make the block compact. Evaluated on Lip Reading Sentence2 (LRS2-BBC) dataset, we first demonstrate that our HMM-based approach outperforms connectionist temporal classification (CTC) based approach with the same visual module and Transformer architecture, yielding a word error rate reduction of 1.9%. Then we empirically show that the proposed approach with StiRes3D based visual module can achieve obvious improvements in terms of both recognition accuracy and model efficiency, over the Pseudo 3D network with a compact 3D convolution design. Our approach also outperforms the current state-of-the-art approach with a word error rate reduction of 1.5%.**

Index Terms: lip-reading, visual speech recognition, compact 3D convolution, hidden Markov model, transformer

I. Introduction

Lip-reading is the task to recognize what people are saying from image alone without audio information. Lip-reading is thought as a challenging task due to the ambiguity introduced by the fact that a visime [1] can be mapped to many different phonemes [2]. Despite the difficulty, a strong lip-reading system can be pretty useful: helping to understand what is being said in a noisy environment [3], [4]; recognizing wake-up word from multi-talker simultaneous speech; and improving mobile interaction with silent command [5].

Conventional approaches usually consist of a spatial feature extractor and followed by a sequential model. More details about these approaches are in [6]. As for deep learning method, a number of works use convolutional neural network (CNN) to predict phonemes [7] or visemes [8] from still images. Long-short term memory recurrent neural networks (LSTMs) with handcrafted features are frequently used to recognise full words and short phrases due to the lack of training data [9], [10]. [11] first proposes a residual network with 3D convolutions to extract more powerful representations. The standard ResNet architecture is modified by changing the first convolutional and pooling blocks from 2D to 3D, and this architecture is widely used in lip-reading tasks [12]. As for the sentence-level lip-reading, [13] designs a lip-reading pipeline that uses a network to output phoneme probabilities. And then convert the phoneme distributions into word sequences with finite state transducers. [3] adopts a network to output character probabilities which is trained with connectionist temporal classification (CTC) [14] loss or sequence-to-sequence (seq2seq) [15] model. Although 3D convolution improves the performance of the network, it brings about huge computational complexity. Simply adding 3D convolutions will make the network too tedious for application, meanwhile it will easily cause overfitting due to the complex structure.

Many researchers work on compressing 2D convolutional block, like Heterogeneous Convolutions(HetConv) [16], Parsimonious Convolutions(ParConv) [17] and Depth-wise Separable Convolutions(DSConv) [18], [19], [20] which propose different compact architectures of convolutional block. As for compressive 3D convolutions, the research efforts are mainly focusing on separating the spatial and temporal convolutions [21], [22], [23]. These methods are useful but restricted. They divide 3D convolutions into spatial and temporal convolutions, so the compression rate is fixed. However in actual applications, we might need to make a trade-off between compression rate and the performance. Another type of compression for convolutional neural networks is pruning, including the work for 2D convolution [24], [25], [26] and 3D convolution [27].

In this work, we focus on the sentence-level lip-reading and conduct experiments on Lip Reading Sentence2 BBC (**LRS2-BBC**) dataset. We introduce a new pipeline for lip-reading. First we use a CNN to extract the visual features of the input video. Then we predict the posteriori probabilities of hidden states by 6-layer multi-head self-attention Transformer together with a fully connected layer. We use hidden-Markov-model (HMM) [28] and an external language model to get the sentence with highest probability. Using this pipeline, we compare how different types of CNN influence the performance and size of the networks. We use the CNN consisting of a 3D convolutional layer followed by a 18-layer residual network(ResNet-18) as the baseline which is also the baseline of many proposed architectures for lip-reading [29], [30]. This pipeline outperforms CTC-Transformer approach [3] which

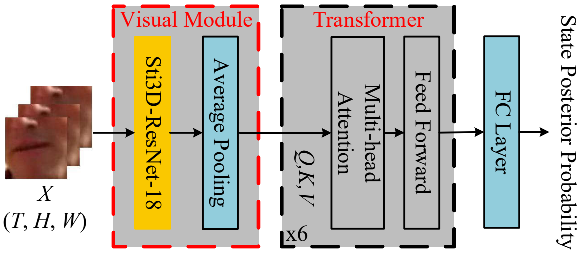


Fig. 1. Proposed network architecture.

has the same visual module and Transformer as ours. In order to get better performance and smaller model, we introduce a novel compact 3D convolution architecture called Sti3D which is the main contribution of this work. We first use channel-shuffle [31] to let the information flow in different channels and sent half of the input channels to the lower half to go through point-wise convolution. The other half is sent to a point-wise convolution with $\omega\times$ more feature maps before sent to channel-wise convolution [32], [33] followed by a point-wise convolution, so that it provides control over the complexity of the model. We also make innovations on adding a residual shortcut (StiRes3D) within the convolutional block to make up for the information loss when we use separable convolutions to approximate the standard 3D convolutions to get better performance. We further demonstrate the merits of the proposed StiRes3D by comparing the performance of different visual module based on whole 3D CNN (All3D) and pseudo 3D CNN (P3D).

II. Our Proposed Approach

The pipeline we proposed for sentence-level lip-reading task uses the visual component as the input and the audio component to create the transcript. We adopt the transcript, the utterance of the audio stream, a dictionary, and the acoustic model to transform the words to triphones. More specifically, there are 6831 triphones with each modeling by a 3-state HMM. We adopt the triphone states via a neural network as shown in Figure 1 and the state alignments to calculate the cross-entropy (CE) loss for training. In the decoding state, we output the final prediction of the sentence with an external language model and lip-reading HMMs. The best performing model using this pipeline achieve a WER of 46.8% on **LRS2-BBC**.

As shown in Figure 1, our proposed network architecture for HMM-based lip-reading can be divided into two main parts. The visual module takes image sequence around the mouth area $\mathbf{X} \in \mathbb{R}^{T\times H\times W}$ as input to extract features $F \in \mathbb{R}^{T\times 512}$ where T denotes the number of frames of the input sequence, and H, W denote the height and width of the input images, respectively. The essential part which is highlighted with yellow color is used to replace the standard ResNet18 to get better performance and smaller model. In the second part we use 6 multi-head self-attention Transformer layers where the features we extract by visual module serve as key, query, and value [35]. The Transformer takes the feature to generate the state posterior probabilities with a fully connected layer and HMMs. We get $P = \{p(s_t|\mathbf{x}_t)\}$ where P denotes the probabilities of hidden states for each frame. Finally, we adopt an external 4-gram language model together with cascading 3-states HMMs each representing a triphone to compute the sentence $\hat{\mathbf{W}}$ with highest probability, which can be formulated as the Bayesian decision problem:

$$\begin{aligned}\hat{\mathbf{W}} &= \arg\max_{\mathbf{W}} p(\mathbf{W}|\mathbf{X}) \\ &= \arg\max_{\mathbf{W}} p(\mathbf{X}|\mathbf{W})p(\mathbf{W})\end{aligned} \tag{1}$$

where $\hat{\mathbf{W}}$ denotes the sentence we recognize from the T-frames input image sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, each $\mathbf{x}_t \in \mathbb{R}^{H\times W}$ is an image, and $\mathbf{W} = \{W_1, W_2, ..., W_n\}$ is the possible word sequence. We can represent the formula in Eq 1 with the mathematical principle of HMM:

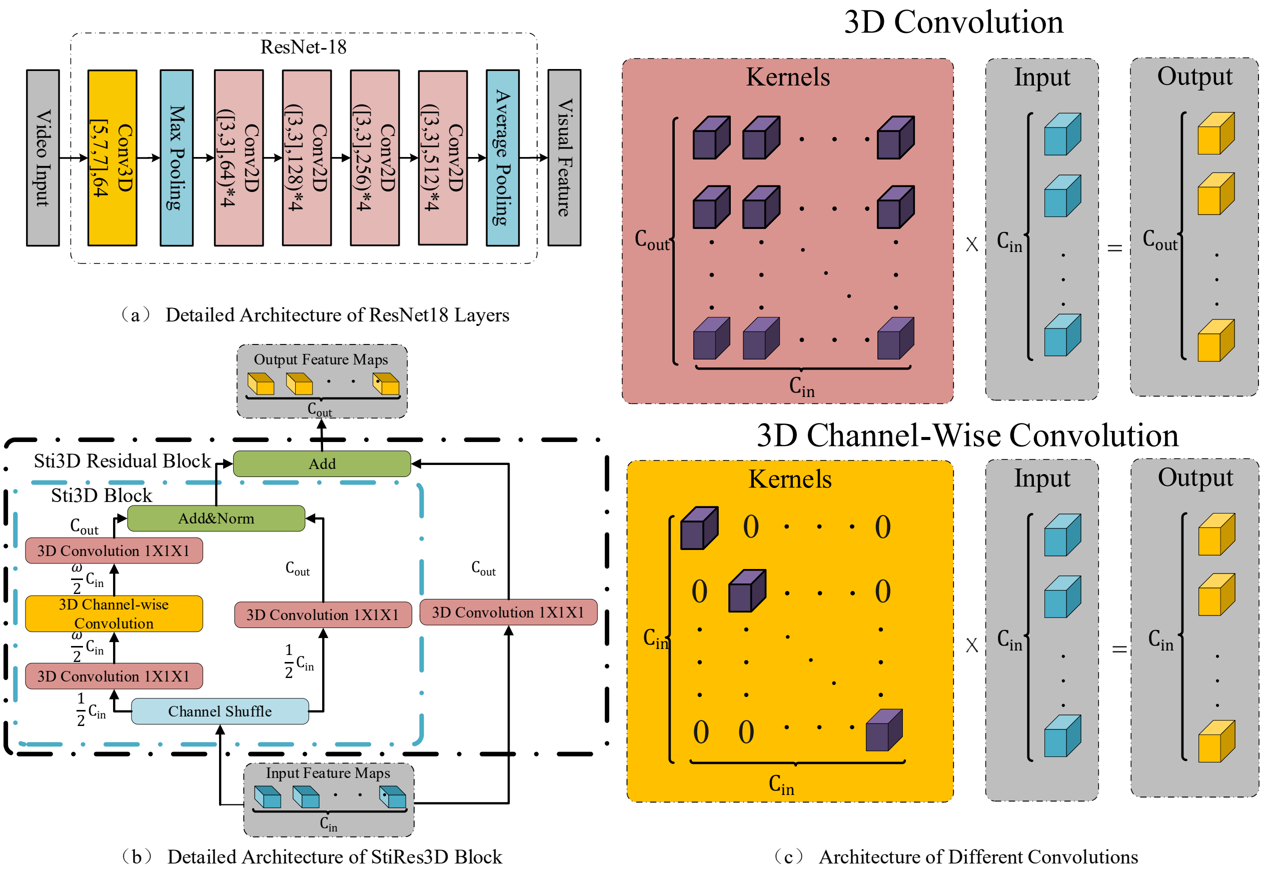
$$\begin{aligned}p(\mathbf{X}|\mathbf{W})p(\mathbf{W}) = \sum_S &\left[\prod_{t=2}^{T} a_{s_{t-1}s_t} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t)\right] \\ &\prod_{i=1}^{n} p(W_i|W_{i-1}, W_{i-2}, ..., W_1)\end{aligned} \tag{2}$$

$$p(\mathbf{x}_t|s_t) = \frac{p(s_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(s_t)} \tag{3}$$

where $S = \{s_1, ..., s_T\}$ denotes the hidden states sequence corresponding to the given $\mathbf{W}$. $p(s_1)$ is the initial state probability, $a_{s_{t-1}s_t}$ is the state transition probability from state at frame $t-1$ to state at frame t estimated, $p(s_t|\mathbf{x}_t)$ is the posteriori probability which is also the output of the network, $p(s_t)$ is the prior probability of s_t estimated from the training set, and $p(\mathbf{x}_t)$ is independent of the given sentence $\mathbf{W}$.

A. Stingy 3D Convolution

The detailed architecture of a Stingy 3D convolution block is shown in Figure 2(b). 3D convolution can be decomposed to spatial convolution and temporal convolution [21], [22]. In addition, the coupling between channel and spatial can be decoupled [19], [20]. We apply this principle to the three dimensions. As shown in Figure 2(c), the idea of different types of 3D convolutions can be presented as matrix multiplication where elements in the matrix are 3D arrays and the operations between elements are convolution instead of multiplication. In three dimensions, the channel-wise convolution can compress more because of the temporal dimension, we will further explain it later. Since not all the channels' information is needed in convolution, we can divide the input channels into two parts. Half of those will go through the channel-wise convolution block, and other half will go through point-wise convolution. In other words, we use heterogeneous convolution kernels, which overcomes the limitation of the existing approaches that are based on efficient architecture search and model compression [16]. Also, from the point of



（a） Detailed Architecture of ResNet18 Layers

（b） Detailed Architecture of StiRes3D Block

（c） Architecture of Different Convolutions

Fig. 2. Detailed Structure

view that letting the information adequately flow within all the channels, we add channel-shuffle to the input. Moreover, in order to provide control over the complexity of the convolution block, we add a point-wise convolution with $\omega\times$ feature maps. [36] states that the set of layer activations forms a “manifold of interest” from the input set, and the information encoded in all individual channels actually lie in some manifolds. Only parts of the channels to go through channel-wise convolution would result in information loss in some channels, so we make it up by adding a residual short cut to the Sti3D block. The modified block is called StiRes3D.

Now we compare the theoretical compression rate between our StiRes3D with a strong compression method P3D [22]. Assume that the input size to the 3D convolutional block is $T \times D \times D \times C_{\text{in}}$, D is the size of the image, T is the number of frames, and C_{in} is the number of input channels. The kernel size of the 3D convolutional block is $T_k \times K \times K \times C_{\text{in}}$, and the number of such kernels is C_{out}. The computational complexity for 3D convolutional block, it is

$$FL_{\text{3D}} = TT_kD^2K^2C_{\text{in}}C_{\text{out}} \tag{4}$$

TABLE I
COMPRESSION RATE OF DIFFERENT COMPACT 3D CONVOLUTIONS

Type	**Compression Rate**
P3D	$1/T_k + 1/K^2$
StiRes3D	$(6+3\omega)/(4T_kK^2)$

For P3D convolutional block, it is

$$FL_{\text{P3D}} = TD^2K^2C_{\text{in}}C_{\text{out}} + TT_kD^2C_{\text{out}}^2 \tag{5}$$

For StiRes3D convolutional block, it is

$$FL_{\text{StiRes3D}} = TD^2C_{\text{in}}(\frac{3+\omega}{2}C_{\text{out}} + \frac{w}{4}C_{\text{in}} + \frac{w}{2}T_kK^2) \tag{6}$$

In most situation, we can make a reasonable assumption that $C_{\text{in}} = C_{\text{out}}$ and $C_{\text{out}} \gg \omega$. Then we can have the compression rate comparing to the standard 3D convolutions of different compact 3D convolutions in Table I. We observe that StiRes3D has two main merits comparing with P3D. First, StiRes3D can have better compression rate. Second, StiRes3D's compression rate can be controlled by adjusting ω.

TABLE II
COMPARISON AMONG DIFFERENT NETWORKS

Network name	**WER**	**Memory(MB)**	**FLOPs** $(*10^8)$
ResNet18	51.8%	56.60	2.7049
All3D	49.7%	155.88	6.6091
P3D	48.5%	88.07	3.4205
StiRes3D($\omega = 2$)	46.8%	44.48	2.2286

III. EXPERIMENTS

In this section, we compare the performance of our proposed HMM-based lip-reading approach with other approach and StiRes3D based with other types of CNNs within HMM framework. First we describe the training strategy. Our implementation is based on the Pytorch library [37] and experiments are conducted on two TeslaV100 GPUs with 16GB memory. The network is trained using the ADAM optimiser [38] with initial learning rate of 10^{-4}, and weight decay of 10^{-4}. We keep training until the CE loss stops decreasing on the validation set, and we keep training for 6 epochs with learning rate reduced by a factor of 2 for each epoch. Decoding is performed with the method we introduced in Section 2. We conduct the experiments on a large-scale English dataset, **LRS2-BBC**, generated and presented in [3]. It contains hundreds of hours of video with talking faces in the middle together with the transcript of the sentences being said. The videos are from a variety of BBC programs. Each video corresponds to a sentence or a phrase which varies in length. The dataset includes nearly 200 hours of videos, and we use the same division of training, validation and test sets as described in [3].

For all the experiments, we adopt the word error rate(WER) as the criterion to evaluate the performance. WER is defined as $\mathrm{WER} = (\mathrm{S} + \mathrm{D} + \mathrm{I})/\mathrm{N}$, where S, D and I are the number of substitutions, deletions, and insertions we get from the reference to the hypothesis, and N is the numebr of all words in the reference. Since we use the same Transformer [35] for all the proposed visual module, we only compare the complexity of the visual modules i.e. the convolutional neural networks. We employ two measures to evaluate the complexity of the network. One measure is floating-point operations (FLOPs) to evaluate the computational complexity of the CNN. As the input size is the same for all the variants, the total FLOPs of the convolutional visual module can represent the computational complexity of the CNN. Another measure is memory used for storage of the visual module to evaluate the space complexity of the network. The CNN's structure is based on [11]. It applies 3D convolutions on the input image with a filter width of 5 frames, followed by a standard ResNet-18 to decrease the spatial dimensions. The detailed architecture is shown in Figure 2(a). The **All3D** denotes the 3D CNN that replaces all the $[3 \times 3]$ 2D kernel with $[3 \times 3 \times 3]$ 3D kernels. **P3D** and **StiRes3D** denote CNN that replaces the $[3 \times 3 \times 3]$ 3D convolution with **P3D-A** [22] and **StiRes3D**, respectively.

A. *Analysis on the performance*

The results of different networks are listed in Table II. The ResNet18 based visual module achieves a WER of 51.8%, and outperforms CTC based approach with the same visual module and Transformer architecture [3], yielding a WER reduction of 1.9%. The results also demonstrates that in lip-reading tasks applying more 3D convolutional block yields an absolute improvement of 2.1% (ResNet18 vs All3D). Since the lip movements and the words in a sentence are temporally relevant, the results are quite reasonable. However, replacing the 2D convolutional blocks with 3D convolutional blocks prominently increases the computational and space complexity of the network. All3D has 2.4× more FLOPs and nearly 3× larger comparing to ResNet18. The huge memory cost and FLOPs make it unacceptable for many applications. Moreover, from Table III we observe that All3D's structure is too complicated and easy to overfit. It has a better training loss but worse validation loss and WER comparing to StiRes3D. The results imply StiRes3D helps to prevent overfitting, because the decomposition of StiRes3D brings about more batch normalization layers.

The experiments also show that StiRes3D is fully superior to P3D and All3D in terms of lower WER, less storage, and smaller FLOPs. Comparing to All3D, StiRes3D has about 3.5× and 3.0× compression rate on memory and FLOPs, repectively. StiRes3D achieves great improvement over P3D with 1.8× and 1.9× on memory and FLOPs. Meanwhile StiRes3D has an absolute reduction of 1.7% over P3D on WER.

B. *Comparison with the State-of-the-art*

The StiRes3D also surpasses the previous state-of-the-art [3] which adopts a CNN with 3D layers to extract the visual features and a sequence-to-sequence transformer on LRS2-BBC by a WER reduction of 1.5%. Moreover, the CNN used in our approach which is StiRes3D is 0.78x smaller than the CNN used in [3] which is ResNet with some 3D layers. Another advantage of our proposed approach is that the StiRes3D CNN can fit different application cases by adjusting the parameter ω from 2 to 0.5 which will be further discussed later.

C. *Analysis on the Sti3D*

To further illustrate the merits of the proposed StiRes3D convolutional block, we did some more experiments to verify the effectiveness. First we compare the performance between standard Sti3D and the StiRes3D, as shown in Table IV. We keep the value of $\omega = 2$ in both architectures. It is observed that the residual shortcut will add little burden to

TABLE III
COMPARISON BETWEEN ALL3D AND STIRES3D

Network	**Training Loss**	**Validation Loss**
All3D	3.064	3.695
StiRes3D	3.238	3.594

TABLE IV
COMPARISON BETWEEN STI3D AND STIRES3D

Network name	WER	Memory(MB)	FLOPs $(*10^8)$
Sti3D($\omega = 2$)	48.3%	39.69	2.0117
StiRes3D($\omega = 2$)	46.8%	44.48	2.2286

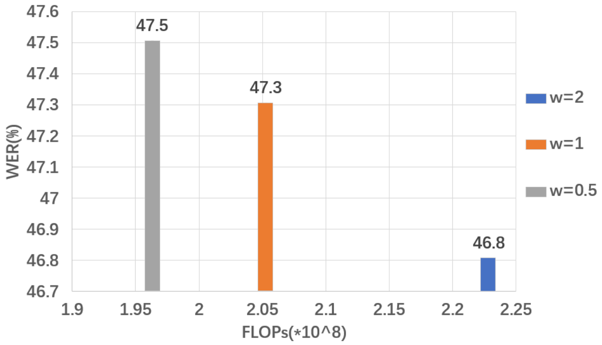


Fig. 3. Performance of StiRes3D with different ω.

the network, specifically 4.79MB and 0.2169×10^8 for space and computation, respectively. However, it can prominently improve the performance on WER by 1.5%.

The complexity and performance of StiRes3D can be controlled by the parameter ω which makes the network flexible to adapt to different application environments. We change the ω of StiRes3D from 2 to 0.5 and compare the performance. In Figure 3. It is clear that we can get better recognition performance by increasing ω, and get smaller networks by decreasing ω.

IV. CONCLUSIONS

In this paper, we introduce a novel HMM pipeline and a new compact 3D convolutional block, Stingy Residual 3D Convolution, and show that our pipeline performs well on lip-reading task. We also verify that adding 3D convolutions to the visual module effectively benefit the performance of the model. Then we use different CNNs on lip-reading tasks to show that our StiRes3D improves the performance with less parameters and FLOPs, meanwhile it is more flexible to adapt to different applications.

V. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants No. 62171427, and Ximalaya Inc.

REFERENCES

[1] S. L. Taylor, M. Mahlerc, B. J. Theobald, and I. Matthews, "Dynamic units of visual speech," In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012.

[2] H. L. Bear, and R. Harvey, "Phoneme-to-viseme mappings: the good, the bad, and the ugly," In *Speech Communication*, 2017.

[3] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[4] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip Reading Sentences in the Wild," In *Internaltional Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.

[5] K. Sun, C. Yu, W. N. Shi, L. Liu, and Y. C. Shi, "Lip-interact: Improving mobile device interaction with silent speech commands," In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp, 581-593.

[6] Z. Zhou, G. Zhao, X. Hong, and M. Pietikainen, "A review of recent advances in visual speech decoding," *Image and vision computing*, 32(9), pp. 590-605, 2014.

[7] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Lipreading using convolutional neural network," In *INTERSPEECH*, 2014, pp, 1149-1153.

[8] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," In *Proceedings of the IEEE Convference on Computer Vision and Pattern Recognition*, 2015, pp, 85-91.

[9] S. Petridis, and M. Pantic, "Deep complementary bottleneck features for visual speech recognition," In *ICASSP*, 2016, pp, 2304-2308.

[10] M. Wand, J. Koutn, and J. Schmidhuber, "Lipreading with long short-term memory," In *ICASSP*, 2016, pp, 6115-6119.

[11] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," In *Interspeech*, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, H. Liao, H. Sak, K. Rao, L. Bennett, M. Mulville, B. Coppin, B. Laurie, A. Senior, and N. de Freitas, "Large-scale visual speech recognition," In *INTERSPEECH*, 2019.

[14] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," In *Proceedings of the International Conference on Machine Learning*, 2006, pp, 369-376.

[15] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," In *Advances in neural information processing systems*, 2014, pp, 3104-3112.

[16] P. Singh, V. K. Verma, and P. Rai, "Hetconv:Heterogeneous kernel-based convolutions for deep cnns[C]," In *CVPR*, 2019, pp. 4835-4844.

[17] Z. Wang and J. Du, "Joint Architecture and Knowledge Distillation in CNN for Chinese Text Recognition," In *arXiv preprint arXiv:1912.07806v3*, 2020.

[18] Francois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," In *Internaltional Conference on Computer Vision and Pattern Recognition*, 2017.

[19] F. Mamalet and C. Garcia, "Simplifying ConvNets for Fast Learning," In *International Conference on Artificial Neural Networks*, 2012, pp. 58-65.

[20] V. Vanhoucke, "Learning visual representations at scale," In *International Conference on Learning Representations(ICLR)*, 2014.

[21] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," In *Internaltional Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," In *IEEE International Conference on Computer Vision*, 2017.

[23] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Claasification," In *Europeon Conference on Computer Vision(ECCV)*, 2018.

[24] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," In *2017 IEEE International Conference on Computer Vision(ICCV)*, 2017.

[25] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," In *2017 IEEE International Conference on Computer Vision(ICCV)*, 2017.

[26] J. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," In *2017 IEEE International Conference on Computer Vision(ICCV)*, 2017.

[27] Z. Xu, T. Ajanthan, V. Vineet, and R. Hartley, "RANP: Resource aware neuron pruning at initialization for 3D CNNs," In *International Conference on 3D Vision*, 2020.

[28] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath, “Deep Neural networks for acoustic modeling in speech recognition,” In *IEEE Signal Processing Magazine*, 2012.

[29] T. Stafylakis, M. H. Khan, and G. Tzimiropoulos, “Pushing the boundaries of audiovisual word recognition using residual networks and lstms,” In *Computer Vision and Image Understanding*, 2018, vol. 176-177, pp. 22-32.

[30] B. Martinez, P.-C. Ma, S. Petridis, and M. Pantic, “Lipreading using temporal convolutional networks,” In *International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, 2020, pp. 6319-6323.

[31] X. Zhang, X. Zhou, and M. Lin, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” In *Internaltional Conference on Computer Vision and Pattern Recogintion*, 2018, pp. 6848-6856.

[32] H. Gao, Z. Wang, and S. Ji, “ChannelNets: Compact and Efficient Convolutional Neural Networks via channel-wise convolutions,” In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” In *arXiv preprint arXiv:1704.04861v1*, 2017.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2015, pp. 4489-4497.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” In *Advances in Neural Information Processing Systems*, 2017.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. -C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[37] A. Paszke, S. Gross, F. Massa, et al, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” In *Neural Information Processing Systems*, 2019.

[38] D. P. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” In *Proceedings of the International Conference on Learning Representations*, 2015.