LIPREADING MODEL BASED ON WHOLE-PART COLLABORATIVE LEARNING

Weidong Tian[1,2,3] *Housen Zhang*[1] *Chen Peng*[1] *Zhong-Qiu Zhao*[1,2,3,⋆]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, China
[2] Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology)
[3] Intelligent Manufacturing Institute of HFUT

ABSTRACT

Lipreading is a task to recognize speech content from visual information of the speaker's lips movements. Recently, some work has focused more on how to adequately extract temporal information, and spatial information is simply used after extraction. In this paper, we focus on the full use of spatial information in lipreading tasks. The whole lip represents global spatial information, while the parts of the lip contain fine-grained spatial information. We propose the lipreading model based on whole-part collaborative learning (WPCL), which can help this model make full use of both global and fine-grained spatial information of the lip. WPCL contains two branches, which deal with the whole and the part features respectively and are trained jointly by collaborative learning. Further, in order to highlight the different importance of part features when fusing them, we propose an adaptive part features fusion module (APFF) to fusion part features. Finally, we prove our viewpoints and evaluate our WPCL by severed experiments. Experiments on LRW and CAS-VSR-W1k datasets demonstrate that our approach achieves state-of-the-art performance.

Index Terms— Lipreading, Whole-Part, Collaborative learning, Feature fusion

1. INTRODUCTION

Unlike speech recognition, lipreading primarily utilizes visual information. The visual information contains the spatial information of a single frame and the temporal information of the video stream. Some models favor the use of temporal information. Examples include the use of optical flow [3], dynamic flow [1], time-shift module [2], etc. But all these models simply make use of spatial information. As an important part of visual information, spatial information should be paid attention to in the lipreading task.

When using spatial information, most models [4–6] tend to extract features of the whole lip for modeling. Modeling the whole lip allows the model to learn global spatial information of the lip. However, this can cause the model to ignore some fine-grained spatial information due to the large receptive field of the whole lip. As shown in Fig.1, when the speaker reads 'ABOUT', the change in the middle of the lip is more apparent than the corners of the mouth. The word 'GREAT' is pronounced with a distinct change in the corners of the mouth and a smooth change in the middle of the lip. This fine-grained spatial information described may be overlooked because of the large receptive fields when we are modeling the whole. Therefore, we have thought of reducing the receptive field by chunking the lip to solve the above problem. In addition, by looking at Fig.1 we find that chunking the model is missing the perception of the whole lip and also loses spatial location information of parts of the lip. The PBL approach [7] uses a part-level model that focuses on changes in the parts of the lip. But PBL loses knowledge of global spatial information, allowing the model to capture only fine-grained spatial information for each part of the lip. It is beneficial for lipreading to make full use of the global spatial information represented by the lip as a whole, as well as the fine-grained spatial information embedded in parts of the lip.

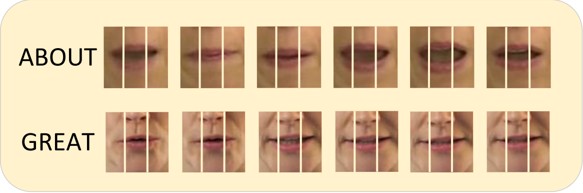

Fig. 1. Two examples of the changes in the corners of the mouth and the middle parts of the lip during the pronunciation of two words: ABOUT and GREAT.

In order to help the model make full use of the global and fine-grained spatial information of lip, we design a two-branch architecture named as the WPCL. These two branches model the front-end encoded features in whole and in part to use the global and fine-grained spatial information of the lip. Two branches interact with each other by collaborative learning [9–11]. Thereby each branch uses the predictions of the other branch as an additional supervisory signal to strengthen its own learning ability.

⋆ is the corresponding author. E-mail address: z.zhao@hfut.edu.cn. This work was supported in part by the National Natural Science Foundation of China under Grant 61976079, and in part by Anhui Key Research and Development Program under Grant 202004a05020039.

Part features are fused in the final stage to produce jointly prediction. In order to measure the importance of each part during feature fusion, we propose APFF, which assigns fusion weights to each part feature based on the difference in affinity [12] between it and the whole feature.

In summary, our contributions are as follows:

- WPCL owes a two-branch collaborative learning architecture which allows taking full advantage of the global and fine-grained spatial information of the lip.
- To fuse part features effectively, we propose the APFF module.
- We demonstrate that WPCL achieves state-of-the-art performance on LRW and CAS-VSR-W1k datasets.

2. THE PROPOSED WORK

2.1. Lipreading model based on whole-part collaborative learning

The lip is the main object of lipreading. The whole lip contains global and spatial location information, while the parts of the lip highlight details. Combining the whole and parts provide sufficient information about the mouth and enables high-quality lipreading. As shown in Fig.2 we model the whole feature and part feature separately using the WPCL.

The lipreading models generally consist of a feature encoding front-end and a feature decoding back-end. In this paper, we use a two-branch feature decoding back-end and keep the feature encoding front-end unchanged. The video sequences are pre-processed and fed to a feature encoding front-end consisting of a single layer of C3D and Resnet-18 [8]. The role of this front-end is to capture the short-term temporal dynamics and to extract the spatial features of a single-frame image. The two-branch feature decoding back-end consists of two structurally identical MS-TCNs. The two branches are named as whole-branch and part-branch, according to their different functions respectively.

2.1.1. Whole-branch

The whole-branch takes the features extracted from the front-end directly as input for modeling the whole lip. The prediction result of the whole-branch can be expressed as:

$$P_w = \sigma \left(FC \left(G_2^w \left(G_1 \left(x, \theta_1 \right), \theta_2^w \right) \right) \right) \tag{1}$$

where $FC\left(\cdot\right)$ is the fully connected layer and σ is the Softmax function, $G_1\left(\cdot,\cdot\right)$ is the front-end architecture, θ_1 is the model parameters of the front-end, $G_2^w\left(\cdot,\cdot\right)$ is the whole-branch architecture, θ_2^w is the model parameters of the whole-branch and x is the input data.

2.1.2. Part-branch

For the part-branch, we divide the whole feature into three parts according to the real spatial distribution of the lip, two corners of the mouth and the middle of the lip, and then input them into the network in turn. It should be noted that multiple part features of the same lip share the part-branch, which improves the generalization of the part-branch. This is an important difference between our WPCL and [7]. The prediction result of the part-branch is as follows:

$$P_p = \sigma \left(FC \left(\sum_{i=1}^{N} G_2^p \left(W_i, \theta_2^p \right) \right) \right) \tag{2}$$

where N is the number of blocks of part features, W_i is the i-th block of part features, denoted as $W_i = G_1\left(x, \theta_1\right)_i$, $G_2^p\left(\cdot,\cdot\right)$ is the part-branch architecture and θ_2^p is the model parameters of the part-branch. Here feature fusion uses simple feature summation. The next subsection will describe in detail the adaptive part features fusion module.

2.1.3. Two-branch architecture with collaborative learning

Collaborative learning can provide additional supervised signals for each branch. The two-branch architecture of collaborative learning (TBCL) allows two branches to be trained collaboratively as two classifier heads in a single stage. TBCL has a significant advantage over the traditional two-stage knowledge distillation [13,14] in terms of the training time. Because for collaborative learning, there is no need to have a pre-trained model as a teacher.

WPCL uses the two-branch architecture of collaborative learning described above. In this way, each branch of WPCL can learn both the whole and parts of the lip. Thus, the total loss function of the WPCL is as follows:

$$L = L_{CE}^w + L_{CE}^p + \beta L_{CL} \tag{3}$$

$$L_{CL} = KL\left(P_w \parallel P_p\right) + KL\left(P_p \parallel P_w\right) \tag{4}$$

where L_{CE}^w and L_{CE}^p are the standard cross-entropy losses of the two branches. L_{CL} is the collaborative learning loss, where $KL\left(\cdot \parallel \cdot\right)$ is Kullback-Leibler divergence, and β is a hyper-parameter indicating the weight of L_{CL}.

2.2. Adaptive part features fusion module

We insert a feature fusion module in the part-branch to combine the part features for joint prediction. After dividing the whole feature by spatial location, the importance degree of the information in various parts is diverse. Part feature, although it contains fine-grained spatial information, is less informative than the whole. Therefore, we assign weights to part features based on the affinity between the part features and the whole feature. Different parts are adaptively assigned with their own fusion weights. The fused features are used

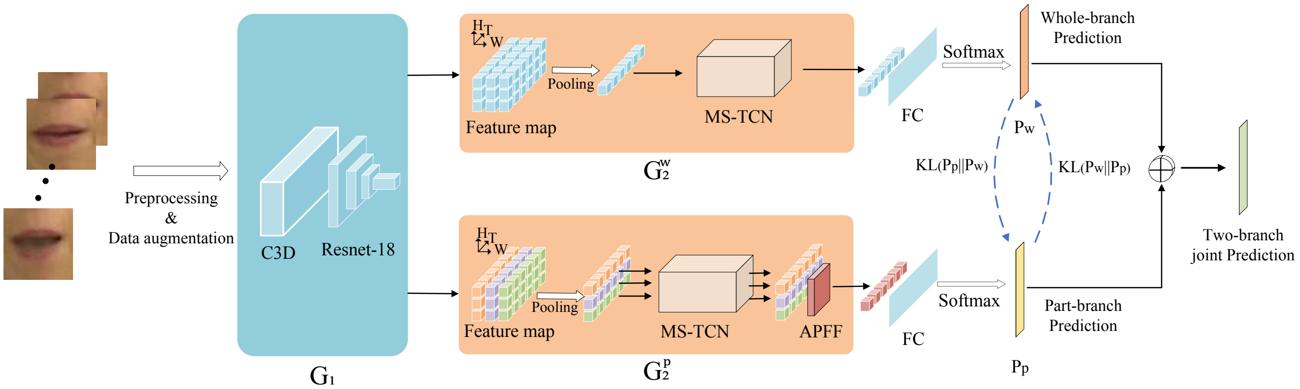

Fig. 2. The pipeline of WPCL. G_1 is the feature encoding front-end, G_2^w is the whole-branch and G_2^p is the part-branch.

as the input of the FC layer. It's notable that the module does not contain any parameters and the computation cost is negligible. The feature of the whole-branch is denoted as $Z^w = G_2^w\left(G_1\left(x, \theta_1\right), \theta_2^w\right) \in \mathbb{R}^{1\times C}$ and the i-th feature of the part-branch is denoted as $Z_i^p = G_2^p\left(W_i, \theta_2^p\right) \in \mathbb{R}^{1\times C}$, where ^{C} is the number of feature channels. The adaptive weights are calculated as follows:

$$a_i = mean\left(Z^w \bigodot Z_i^p\right) \tag{5}$$

where $\bigodot$ is to multiply the elements of the corresponding positions of the two features, $mean\left(\cdot\right)$ is to average over the channel dimensions to obtain $a_i \in \mathbb{R}^{1\times 1}$, and

$$A = \left(\alpha_1, \alpha_2 \cdots \alpha_N\right) = \sigma\left(cat\left(a_1, a_2 \cdots a_N\right)\right) \tag{6}$$

where cat is the connection operation, $A \in \mathbb{R}^{N\times 1}$, and α_i is the fusion weight assigned to the i-th part feature by APFF.

3. EXPERIMENTS

3.1. Datasets

We evaluated our model on the English and Mandarin word-level lipreading datasets.

LRW [15]: This is a very challenging dataset. The dataset containing 500 English words, and every word has 800-1000 training samples, 50 validation samples, and 50 test samples. Each sample consists of 29 frames, and the target word appears in the middle of the sample shot.

CAS-VSR-W1k [16]: This dataset is originally named LRW-1000, and is the largest publicly available Mandarin word-level lipreading dataset. There are 1000 words classes and more than 700,000 samples. Unlike LRW, its sample length is not constant and it does not have the uniform resolution of the samples.

3.2. Data preprocessing

For both LRW and CAS-VSR-W1k datasets, we crop each sample to the ROI of 96×96 fixed pixel size. All images changed to grayscale for reduced computation. During training, each frame of one sample is randomly cropped to a size of 88×88 pixels, and horizontal flip with a probability of 0.5 and Mixup [17] with a weight of 0.4 is used for data augmentation. For the CAS-VSR-W1k dataset, we use the same data preprocessing method as in [6]. We fix the number of video frames of each sample to 29, and the target word is in the middle of every sample shot.

3.3. Training details

We use AdamW optimizer [18] with the initial learning rate 3e-4 and the weight decay 1e-2. Cosine annealing algorithm is used to attenuate the learning rate. The model is trained end-to-end for 80 epochs, using a mini-batch of 32. We use the model initialization method in [5], to speed up the convergence of the model. In addition, we also use the variable-length augmentation strategy in [5] to enhance the robustness of the model for temporal sequences.

3.4. Evaluation of two-branch architecture with collaborative learning

In this experiment, we evaluate the effectiveness of a two-branch architecture with collaborative learning. The baseline is MSTCN [5], which uses the whole lip as the input. The two branches of TBCL (branch1, branch2) both adopt the same structure as MSTCN, and also uses the whole lip as the input. From Table 1, we can find each branch has a large improvement of prediction accuracy over the baseline, and the two-branch joint prediction improved over baseline by 2.3%(LRW) and 7.3%(CAS-VSR-W1k) on the two datasets, respectively.

Experimental results also show that two branches, which own the same structure, have different performances. This

Table 1. Evaluation of the two-branch architecture with collaborative learning (TBCL) by TOP-1 Acc.

Method		LRW(%)	CAS-VSR-W1k(%)
MSTCN (baseline) [5]		85.3	41.4
TBCL	Branch1	87.3	47.6
	Branch2	87.2	47.8
	Two-branch joint	87.6	48.7

should be caused by their different initial parameters.

3.5. Evaluation of lipreading model based on whole-part collaborative learning

In this section, we evaluate the effectiveness of WPCL. Here, one branch in TBCL learns the whole feature and the other branch learns the part features. The part-branch uses simple summation for feature fusion. From Table 2 we can find that these two branches achieve 87.6%, 87.6% on the LRW dataset and their joint prediction achieves 88.1%. The accuracy of these two branches on the CAS-VSR-W1k dataset reaches 47.7% and 47.9% respectively, and their joint prediction reaches 48.9%. This experiment indicates that when two branches model the whole and parts separately, the model can make full use of global spatial information and fine-grained spatial information.

The performance is further improved when APFF is used to fusion features. The accuracy of part-branch improved by 0.3% (LRW) and 0.6% (CAS-VSR-W1k), and their joint predictions reached 88.3% and 49.4% for LRW and CAS-VSR-W1k respectively. The reason lies in that the simple summation of part features tends to ignore the different importance of various parts, while APFF assigns weights to part features based on their similarity to the whole feature. Part features can capture fine-grained spatial information, which, in conjunction with the APFF method enables efficient lipreading. From Table 2, we can find that making full use of the global and fine-grained spatial information of the lip is beneficial for lipreading, our approach achieves state-of-the-art performance.

3.6. Evaluation of computational complexity of our model against the state-of-the-art

The MSTCN multi-stage distillation (MSTCN-MSD [19]) uses the same strategy as [14, 20]. Its disadvantage is that achieving a good student requires multiple knowledge distillation processes. The MSTCN-MSD experiences 4 times and 2 times iterations on the two datasets respectively, which is a huge waste of training time. In contrast, our two-branch model only performs collaborative training in a single stage, and these two branches enable parallel computation [11]. As shown in Table 2 and Table 3, each branch of our model is with almost the same number of parameters and FLOPs as MSTCN-MSD. On the LRW dataset, each branch has achieved the same accuracy as the MSTCN-MSD. Excitingly, on the CAS-VSR-W1k dataset, each branch has definitely improved over MSTCN-MSD by 2.8% and 3.2% respectively. Compared with MSTCN-Ensemble, which is integrated by multiple models and has a large number of parameters and high computational costs, our two-branch joint prediction decreases the accuracy by 0.2% on LRW, while requiring 2.4× fewer parameters and 3× fewer FLOPs. For CAS-VSR-W1k, our two-branch joint prediction accuracy is improved by 2.8% over MSTCN-Ensemble while the number of parameters and FLOPs are reduced. Compared with other work, each branch of our model achieves significant improvement on accuracy with slightly higher parameters and FLOPs.

Table 2. Comparison between our whole-part collaborative learning and other methods.

Method		LRW(%)	CAS-VSR-W1k(%)
E2E [4]		82.0	-
PBL [7]		82.8	-
STFL [3]		84.1	-
DFTN [1]		84.1	41.9
EFFECTIVE LR [6]		85.0	48.0
MSTCN(baseline) [5]		85.3	41.4
TSM [2]		86.2	44.6
MSTCN-MSD [19]		87.9	45.3
MSTCN-Ensemble [19]		**88.5**	46.6
Ours(WPCL)	Whole-branch	87.6	47.7
	Part-branch	87.6	47.9
	Two-branch joint	88.1	48.9
Ours(WPCL+APFF)	Whole-branch	87.8	48.1
	Part-branch	87.9	48.5
	Two-branch joint	88.3	**49.4**

Table 3. Comparison with the state-of-the-art in terms of computational complexity on the LRW (D1) and CAS-VSR-W1k (D2) datasets. We use a sequence of 29-frame with a size of 88 by 88 to report FLOPs.

Method		Params $\times 10^6$		FLOPs $\times 10^9$		Number of iterations	
		D1	D2	D1	D2	D1	D2
E2E [4]		29.7	-	18.7	-	×	×
DFTN [1]		48.0	-	32.9	-	×	×
EFFECTIVE LR [6]		59.4	60.5	10.6	10.6	×	×
MSTCN (baseline) [5]		36.4	36.7	10.3	10.3	×	×
MSTCN-MSD [19]		36.4	36.7	10.3	10.3	4	2
MSTCN-Ensemble [19]		146.6	73.4	41.2	20.6	4	2
WPCL +APFF	Whole-branch	36.4	36.7	10.3	10.3	×	×
	Part-branch	36.4	36.7	12.5	12.5	×	×
	Two-branch joint	61.2	61.5	13.7	13.7	×	×

4. CONCLUSION

In this paper, we propose a novel WPCL model which utilizes collaborative learning to make full use of global and fine-grained spatial information of the lip. In WPCL, we adopt an APFF to fuse part features which can further improve the accuracy of our model. Our model also outperforms the existing state-of-the-art in terms of training times, model parameters, and FLOPs. Our model achieves state-of-the-art performance.

5. REFERENCES

[1] J. Xiao, S. Yang, Y. Zhang, S. Shan, and X. Chen, "Deformation flow based two-stream network for lip reading," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. IEEE, 2020, pp. 364–370.

[2] M. Hao, M. Mamut, N. Yadikar, A. Aysa, and K. Ubul, "How to use time information effectively? combining with time shift module for lipreading," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7988–7992.

[3] X. Weng and K. Kitani, "Learning spatio-temporal features with two-stream deep 3d cnns for lipreading," *arXiv preprint arXiv:1905.02540*, 2019.

[4] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018, pp. 6548–6552.

[5] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading using temporal convolutional networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6319–6323.

[6] D. Feng, S. Yang, S. Shan, and X. Chen, "Learn an effective lip reading model without pains," *arXiv e-prints*, pp. arXiv–2011, 2020.

[7] Z. Miao, H. Liu, and B. Yang, "Part-based lipreading for audio-visual speech recognition," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 2722–2726.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[10] G. Song and W. Chai, "Collaborative learning for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1832–1841, 2018.

[11] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo, "Online knowledge distillation via collaborative learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 020–11 029.

[12] Y. Ge, C. L. Choi, X. Zhang, P. Zhao, F. Zhu, R. Zhao, and H. Li, "Self-distillation with batch knowledge ensembling improves imagenet classification," *arXiv e-prints*, pp. arXiv–2104, 2021.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *stat*, vol. 1050, p. 9, 2015.

[14] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1607–1616.

[15] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian conference on computer vision*. Springer, 2016, pp. 87–103.

[16] S. Yang, Y. Zhang, D. Feng, M. Yang, C. Wang, J. Xiao, K. Long, S. Shan, and X. Chen, "Lrw-1000: A naturally-distributed large-scale benchmark for lip reading in the wild," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019, pp. 1–8.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[19] P. Ma, B. Martinez, S. Petridis, and M. Pantic, "Towards practical lipreading with distilled and efficient models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7608–7612.

[20] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5628–5635.