

The NPU-ASLP-LiAuto System Description for Visual Speech Recognition in CNVSRC 2023

He Wang¹, Pengcheng Guo¹, Wei Chen², Pan Zhou², Lei Xie^{1*}

¹Audio, Speech and Language Processing Group (ASLP@NPU), School of Computer Science, Northwestern Polytechnical University, Xian, China

²Space AI, Li Auto

hwang2001@mail.nwpu.edu.cn

Abstract

This paper delineates the visual speech recognition (VSR) system introduced by the NPU-ASLP-LiAuto (Team 237) in the first Chinese Continuous Visual Speech Recognition Challenge (CNVSRC) 2023, engaging in the fixed and open tracks of Single-Speaker VSR Task, and the open track of Multi-Speaker VSR Task. In terms of data processing, we leverage the lip motion extractor from the baseline¹ to produce multi-scale video data. Besides, various augmentation techniques are applied during training, encompassing speed perturbation, random rotation, horizontal flipping, and color transformation. The VSR model adopts an end-to-end architecture with joint CTC/attention loss, comprising a ResNet3D visual frontend, an E-Branchformer encoder, and a Transformer decoder. Experiments show that our system achieves 34.76% CER for the Single-Speaker Task and 41.06% CER for the Multi-Speaker Task after multi-system fusion, ranking first place in all three tracks we participate.

Index terms— Visual Speech Recognition, Lip Reading

1. Introduction

Leveraging the strides in deep learning, automatic speech recognition (ASR) has made substantial advancements, reaching parity with human performance on certain benchmarks. Nevertheless, not all scenarios boast access to high-quality speech audio. In response to the challenges posed by such scenarios, there has been a surge of interest among researchers in visual speech recognition (VSR).

To further advance research in VSR, the first Chinese Continuous Visual Speech Recognition Challenge² (CNVSRC) 2023 is initiated, aiming to probe the performance of large vocabulary continuous visual speech recognition (LVCVSR) in two scenarios: reading in a recording studio and speech on the Internet. Furthermore, CNVSRC 2023 employs the CN-CVS [1] dataset as its training set and defines two tasks: Single-Speaker VSR Task (T1) and Multi-Speaker VSR Task (T2). Independent development and evaluation sets accompany each task, denoted as CNVSRC-Single.Dev/Eval (T1.Dev/Eval) and CNVSRC-Multi.Dev/Eval (T2.Dev/Eval), respectively. Moreover, each task includes a fixed track, which restricts the use of data solely to that released by the challenge, and an open track without constraints on data usage.

This study describes our system in CNVSRC 2023. As for the data processing, we extract lip motion video data by



Figure 1: The flow chart of data processing. $\times n$ refers to n times the data of CN-CVS and CNVSRC-Single/Multi datasets. $(N \times N)$ refers to multi-scale video data.

crop sizes of 48, 64, 96, and 112 to build multi-scale systems. The VSR model adopts an end-to-end architecture with joint CTC/attention loss, consisting of dynamic augmentation, visual frontend, encoder, and decoder. Moreover, we diversify our systems by incorporating distinct encoders, including recently proposed E-Branchformer [2], Branchformer [3], and Conformer [4]. After using Recognizer Output Voting Error Reduction [5] (ROVER) for post-fusion, we attain CERs of 34.47% and 34.76% on the T1.Dev and T1.Eval datasets, alongside 41.39% and 41.06% on the T2.Dev and T2.Eval datasets. Our systems rank first place in the open tracks of both tasks and the fixed track of the Single-Speaker VSR Task.

2. Proposed system

2.1. Data processing

Fig. 1 shows the data processing flow of our system. The initial training videos, sized 224×224 , are processed via the lip extractor to obtain $N \times N$ (where N can be 48, 64, 96, or 112) sized lip motion video data. Subsequently, the moviepy³ tool is employed to apply speed perturbation with rates of 0.9, 1.0, and 1.1, which results in three times training data in total.

2.2. Visual speech recognition

Fig. 2(b) shows the overall architecture of our VSR system, comprising four main components: dynamic augmentation, visual frontend, encoder, and decoder. The training video data is first augmented by random rotation, horizontal flipping, and color transformation through the dynamic augmentation module implemented by the kornia⁴ tool. Subsequently, it passes through the ResNet3D Visual Frontend module for visual feature extraction, with detailed information shown in Fig. 2(a).

The design of the visual frontend module draws inspiration from the classic ResNet, where 2D convolutions are replaced with 3D ones. To reduce the computational cost, the network structure is simplified. Specifically, the dynamically augmented video data passes through a 3D convolution input

¹<https://github.com/MKT-Dataoceanai/CNVSRC2023Baseline>

²<http://cnceleb.org/competition>

³<https://pypi.org/project/moviepy>

⁴<https://github.com/kornia/kornia>

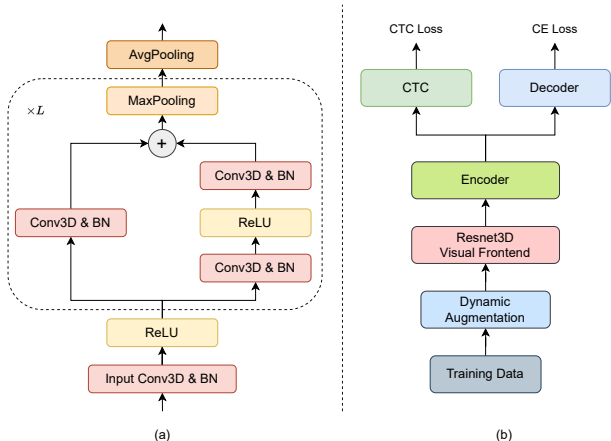


Figure 2: (a) Detail structure of the ResNet3D Visual Frontend; (b) System overview of our VSR system.

layer to project the channel to a higher dimension. It then goes through L (where $L = 5$) ResNet blocks. In each block, the input visual features are modeled by independent convolution modules to achieve a higher and equal dimension of channels before the residual connection. Subsequently, MaxPooling is applied to perform two-fold downsampling on the height and width dimensions. Finally, AvgPooling is used to average the height and width dimensions, completing the process of visual dimension reduction and feature extraction.

For the encoder of the VSR system, we adopt the recently proposed E-Branchformer. Moreover, we build diverse VSR systems featuring different encoders such as Branchformer and Conformer, facilitating multi-system fusion. Regarding the decoder, a common Transformer decoder is utilized. The final model loss is composed of the CTC loss derived from the encoder and the cross-entropy (CE) loss computed on the decoder.

3. Experiment

3.1. Setup

All systems are implemented with the ESPnet⁵ toolkit, only using the CN-CVS and the development set of CNVSR- Single/Multi datasets. For the encoder of the VSR system, we use a 12-layer E-Branchformer block, each with 256 attention units, 4 attention heads, and 1024 feed-forward units. Additionally, the decoder contains 6 Transformer layers, each with 4 attention heads and 2048 feed-forward units. The visual frontend is a 5-layer ResNet3D module, whose channels are 32, 64, 64, 128, 256, and kernel size is 3. To achieve better results by post-fusion, we also build Conformer and Branchformer VSR systems with similar param sizes to the E-Branchformer one.

3.2. Inference Procedure

We build 24-layer Transformer language models (LM) using the text from CN-CVS and the CNVSR- Single/Multi datasets for the shallow fusion in Single-Speaker and Multi-Speaker Tasks, respectively. The embedding and attention dimensions are set to 512, with 8 attention heads and 2048 feed-forward units. During decoding, the beam size is set to 48 and both CTC and LM contribute to scoring with weights of 0.5 and 0.4, respectively.

⁵<https://github.com/espnet/espnet>

Table 1: The CER(%) results of our VSR systems on Dev and Eval sets in T1 and T2 tasks. Conf, Branch and E-Branch represent Conformer, Branchformer, and E-Branchformer encoder, respectively. Crop refers to the size of training lip motion video data. SP means speed perturbation is applied or not.

System	Encoder	Crop	SP	T1.Dev	T1.Eval	T2.Dev	T2.Eval
Baseline ⁶	Conf	96	✗	48.57	48.60	58.77	58.37
M1	Conf	96	✓	39.43	39.99	46.08	45.73
M2	Branch	96	✓	39.00	39.36	46.63	46.37
M3	E-Branch	96	✓	38.59	38.61	46.26	45.80
M4	E-Branch	48	✗	46.88	45.81	55.58	55.51
M5	E-Branch	64	✗	44.40	43.59	53.64	52.98
M6	E-Branch	80	✗	42.95	42.26	50.77	50.38
M7	E-Branch	96	✗	40.56	40.42	47.16	46.53
M8	E-Branch	112	✗	38.46	38.95	45.17	44.87
ROVER	-	-	-	34.47	34.76	41.39	41.06

3.3. Results

Table 1 shows the outcomes of our VSR systems (M1~M8) and the system fusion result obtained by the ROVER. Specifically, in terms of the encoder, the performance of the E-Branchformer surpasses that of the Branchformer and Conformer (M1~M3). Moreover, as the crop size increases from 40 to 112, the performance of our VSR system improves (M4~M8). Using three times speed perturbation also yields a certain improvement in CER (M3 and M7). After multi-system fusion by ROVER, we achieve 34.47% and 34.76% CERs on the Dev and Eval sets of T1, alongside 41.39% and 41.06% CERs of T2, respectively.

4. Conclusion

This paper describes the visual speech recognition (VSR) system introduced by the NPU-ASLP-LiAuto (Team 237) in the Chinese Continuous Visual Speech Recognition Challenge (CNVSR) 2023, engaging in the fixed and open tracks of Single-Speaker VSR Task, and the open track of Multi-Speaker VSR Task. In detail, we extract lip motion video at multiple scales and employ different mainstream encoders to build diverse VSR systems for multi-system fusion. Finally, we achieve CERs of 34.76% and 41.06% on the final evaluation sets of Single-Speaker VSR Task and Multi-Speaker VSR Task, respectively, ranking first place in all three tracks we participate.

5. References

- [1] Chen Chen, Dong Wang, and Thomas Fang Zheng, "Cn-cvs: A mandarin audio-visual dataset for large vocabulary continuous visual to speech synthesis," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.
- [2] Kwangyoung Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, Kyu J Han, and Shinji Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. SLT*. IEEE, 2023, pp. 84–91.
- [3] Yifan Peng, Siddharth Dalmia, Ian Lane, et al., "Branchformer: Parallel MLP-attention architectures to capture local and global context for speech recognition and understanding," in *Proc. ICML*. PMLR, 2022, pp. 17627–17643.
- [4] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, et al., "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*. ISCA, 2020, pp. 5036–5040.
- [5] Jonathan G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU*. IEEE, 1997, pp. 347–354.