2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) | 978-1-6654-4989-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICMEW53276.2021.9456014

AN EFFICIENT SOFTWARE FOR BUILDING LIP READING MODELS WITHOUT PAINS

Dalu Feng[1,2], *Shuang Yang*[1,2], *Shiguang Shan*[1,2]

[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China

ABSTRACT

Lip reading is an impressive skill for human beings and has lots of potential applications. However, building an automatic lip-reading system is still a challenging task at present. Previous explorations of building a lip reading model usually contain several implicit operations and training details, which usually brings a big difficulty to the re-implementation for most researchers. In this work, we aim to establish a clear and efficient pipeline, which provides a convenient implementation tool and an easy start point for researchers and any others who want to study the problem of lip reading. We empirically study and introduce several useful training strategies in a clear and unified implementation procedure and compare their effects. Our pipeline got a performance of 88.4%/56.0% on two popular large-scale lipreading datasets, which is based on the basic model but achieves the performance comparable to or even higher than the state-of-the-art results. Our codes and models are available at https://github.com/Fengdalu/learn-an-effective-lip-reading-model-without-pains

***Index Terms*—** Lip reading, Deep Learning, Visual Speech Recognition

1. INTRODUCTION

Lip reading, also known as visual speech recognition, aims to recognize the speech content by watching silent videos. Its robustness against noise can open up lots of potential applications. It has become an emerging topic that has received increased attention in recent years.

However, lip reading remains a challenging task for both humans and machines. Firstly, human lip cues are naturally ambiguous due to the homophones (e.g. ‘p’ and ‘b’), especially in the absence of context. Second, the variation of visual factors, such as lighting conditions, speaker’s accent, viewpoints, and so on, greatly increase the difficulty of accomplishing effective lip reading especially for beginners in the area. So it is in urgent need of convenient software to facilitate both the research and systematic applications of lip reading.

Year	Method	Accuracy	
		LRW	LRW-1000
2017	VGGM	61.10%	25.70%
2017	ResNet + LSTM[1]	83.50%	38.19%
2019	ResNet + DenseNet + GRU	83.30%	36.90%
2020	ResNet + TCN[2]	85.30%	41.40%
2020	Ours+word boundary	88.40%	56.00%

Table 1. Comparison with existing methods.

A common lip-reading system usually consists of three modules: visual feature learning, temporal modeling, and speech content output in the text form. The process involves three areas at the same time: computer vision, speech recognition, and natural language processing, which brings an extra non-negligible difficulty for establishing an effective lip reading system beyond the difficulty of the problem itself.

In this work, we aim to build a simple and high-performance lip-reading pipeline without whistles and bells. We hope to help researchers in this area efficiently build a lip reading system. Based on our implementation, the users can quickly start their works and implement their ideas based on the current start-of-art models. Furthermore, we reach 88.4%/56.0% on two large-scale challenging datasets respectively by using only a simple baseline model. The results are comparable with and even surpass the state-of-the-art results, shown in Table 1.

2. THE PROPOSED SOFTWARE

In this section, we introduce our architecture and highlights features for a better understanding of this work.

2.1. Software Architecture

Our code is all implemented by Python. The architecture of our software is shown in Figure 1, including the data loader, frontend module, backend module, and optimizer respectively. The file structure is shown in 2

Data Loader The role of the data loader is feeding lip images to the model. We provide an online multi-process data loader for both training and testing, as well as offline scripts

978-1-6654-4989-2/21/$31.00 ©2021 IEEE

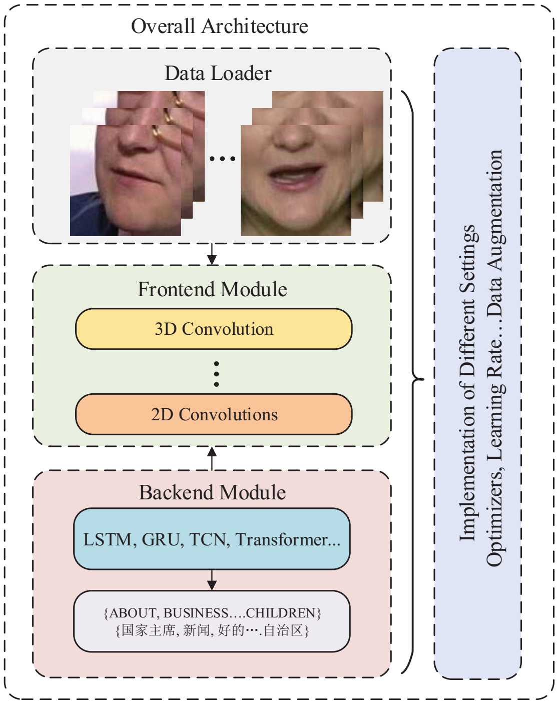
Overall Architecture
Data Loader
…
Frontend Module
3D Convolution
2D Convolutions
Backend Module
LSTM, GRU, TCN, Transformer...
{ABOUT, BUSINESS....CHILDREN}
{国家主席, 新闻, 好的….自治区}
Implementation of Different Settings
Optimizers, Learning Rate...Data Augmentation

Fig. 1. The overall architecture of our software, including data loaders, frontend module, backend module, training and testing module.

for preprocessing other user-specified raw videos.

Frontend Module The frontend module is designed to pay more attention to spatial dynamic patterns. It accepts lip images and converts them into deep features by one 3D convolutional layer together with a Resnet-18 network. We also provide a pluggable SE-Module for users to obtain higher performance.

Backend Module The backend module is often designed to learn the temporal dynamics of the sequence. It outputs the final prediction based on the deep features of the frontend module. We provide configurable backend networks, including the implementation of both the GRU and TCN in our project.

Implementation of Different Settings Beyond the above basic modules, we provide an extra module to offer multiple choices to the users, including several different settings of the optimization process, like the learning rate schedulers, the data augmentation choice.

2.2. Highlight Features

In this section, we share our best practice in building our software and hope to provide a good start point for further applications.

Efficient Data Loading In lip reading applications, the program needs to load and decode a large number of videos at once, which is very time-consuming. Our project tackles the problem by balancing the burden of CPU and I/O loading.

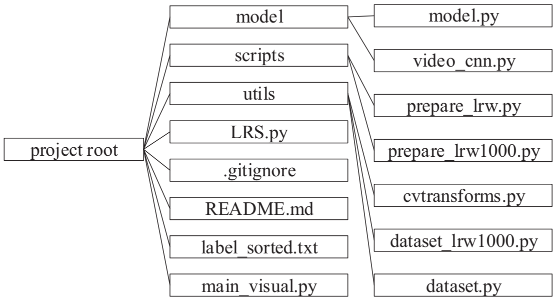

Fig. 2. The file structure of our software.

Configurable Training Strategies Our models are learned in an end-to-end manner. The user only needs to run the program once to get the final result. We also provide several training tweaks as pluggable modules in our code. Different users can choose any ones of them according to their practical case. For example, run SE-ResNet-18 with Label Smoothing in GPU 0-4 for 120 epochs on LRW dataset as:

```
python main_visual .py −−gpus=’0,1,2,3’ −−lr=3e−4 −−
      batch_size=400 −−num_workers=8 −−max_epoch=120
      −−test=False −−save_prefix=’checkpoints/lrw−baseline
      ’ −−n_class=500 −−dataset=’lrw’ −−border=False −−
      mixup=False −−label_smooth=True −−se=True
```

High-Performance Networks We perform with previous network choices to show their effects for lip reading respectively. With a simple architecture, our performance is comparable or even surpasses the best results, shown in Table 1.

Instruction and Pretrained Models We provide detailed step-by-step instructions about how to set up the environment, process data, test and train models, and so on. We also have released the weights of several models for testing and they can be finetuned on smaller datasets.

3. CONCLUSION

This work has provided a complete and efficient tool for training a lip reading system. Besides the basic pipeline, we also provide several alternative modules to adapt to different settings in several aspects. Finally, the experimental results proved the effectiveness of our pipeline.

4. REFERENCES

[1] Themos Stafylakis and Georgios Tzimiropoulos, “Combining residual networks with lstms for lipreading,” in *Interspeech*, 2017.

[2] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic, “Lipreading using temporal convolutional networks.,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.