Text Recognition from Silent Lip Movement Video

Youda Wei
Xi'an Institute of Optics and Precision Mechanics
Chinese Academy of Sciences
Xi'an, China
University of Chinese Academy of Sciences
Beijing, China
e-mail: weiyouda2016@opt.cn

Xiaodong Hu*
Xi'an Institute of Optics and Precision Mechanics
Chinese Academy of Sciences
Xi'an, China
e-mail: hxd@opt.ac.cn

***Abstract*—Lip reading, the ability to recognize text information from the movement of a speaker's mouth, is a difficult and challenging task. Recently, the end-to-end model that maps a variable-length sequence of video frames to text performs poorly in real life situation where people unintentionally move the lips instead of speaking. The goal of this work is to improve the performance of lip reading task in real life. The model proposed in this article consists of two networks that are visual to audio feature network and audio feature to text network. Our experiments showed that the model proposed in this article can achieve 92.76% accuracy in lip reading task on the dataset that the unintentional lips movement was added.**

Keywords-lip reading; convolutional neutral networks; deep learning;

I. INTRODUCTION

Lip reading plays a significant role in human communication. The technical can help the hearing loss people and can be applied in audio-visual speech recognition system.

However, lip reading is a notoriously difficult task for humans. The visual information from different speakers are different due to various appearances of lips, various backgrounds, and various talking ways even the content of the conversation is the same. For example, Fisher [1] gives 5 categories of visual phonemes, out of a list of 23 initial consonant phonemes, that are confused by tested people when viewing a speaker's mouth. Nevertheless, machine lip reading is difficult because it requires extracting spatiotemporal features from the video.

Recently, the recognition task can be solved by using deep neural network models and large-scale datasets for training. Most existing work, have viewed lip reading as a video to text task where the final output is a set of textual sentences corresponding to the lip movements of the speaker. However, these networks perform poorly in real life situation where people unintentionally move the lips instead of speaking.

In this paper, we developed three stages for the lip recognition task. First, we presented a new data preparation method by adding noise data (unintentional lips movement) in real life. Second, we presented a new 3D convolutional neural networks architecture, which is to the best of our knowledge, the first deep learning models maps a variable-length sequence of video frames to the auditory features. Third, we employed a neural network to recognize the text from the auditory features that are the output of the first stage. The network works well in real life situation. The words and the noise can be recognized from the silent lip movement video. Our experiments show that the model we presented achieves 92.76% accuracy in word recognition and 91.39% accuracy in detecting unintentional lips movement on the GRID audio-visual corpus [2].

II. RELATED WORKS

Lip reading has traditionally been posed as a classification task where words or short phrases from a limited dictionary are classified based on features extracted from lip movements. In this section, we outline various existing approaches to lip reading.

Notably, Goldschen [3] was the first to do visual-only sentence-level lip reading using hidden Markov models (HMMs) in a limited dataset, using hand-segmented phones. Later, Neti [4] were the first to do sentence-level audiovisual speech recognition using an HMM combined with hand-engineered features, on the IBM ViaVoice [4] dataset. The authors improve speech recognition performance in noisy environments by fusing visual features with audio ones.

Recently, there has been a surge in end-to-end deep learning approaches for lip reading. Wand [5], Assael [1], Chung and Zisserman [6] focused on either word level or sentence-level prediction using a combination of convolutional and recurrent networks. Furthermore, Gergen [7] used speaker-dependent training on an LDA-transformed version of the Discrete Cosine Transforms of the mouth regions in an HMM/GMM system.

III. DATA PREPARATION

Data preparation plays a crucial part. This section describes the data preprocessing for lip reading task. First, a public audio-visual dataset is used in this paper. Then, the visual and audio part was processed as input and label for the visual to audio feature network in training stage. In the end, the unintentional lips movement that is a common behavior to people was added in the data preparation part.

978-1-5386-6396-7/18/$31.00 ©2018 IEEE

TABLE I. GRID VOCABULARY

Command	Color	Preposition	letter	Digit	Adverb
Bin	Blue	At	a-z	0-9	Again
Lay	Green	By			Now
Place	Red	In			Please
Set	White	With			Soon

A. Dataset

The dataset used for training the network was the GRID audio-visual corpus which consists of audio and video recordings of 34 different speakers (male and female). For each speaker, there are 1000 utterances and each utterance is a combination of six words from a 48-word vocabulary which is shown in Table I. In this paper, we used digit from "one" to "nine". Video and audios are both 3 seconds long with a sampling rate of 25fps and 44kHz respectively.

B. Visual stream

The goal of this part was to clipped the lip region by a fixed bounding box from the video. We first normalized each lip movement video to have zero mean and unit standard deviation. The face region was then extracted by Supervised Descent Method (SDM) [8] from each frame and resized to have dimensions W×H. It was then divided into K non-overlapping slices each of length L_v. Each frame are calculated to form a 3D tensor of shape H×W×L_v. The 3D tensor of shape is the input of the 3D convolutional neural networks. The processing pipeline of visual stream is shown in Fig. 1.

C. Audio stream

The label of the 3D convolutional neural networks is Mel-frequency cepstral coefficients (MFCC) [9] values from the audio stream. This is a representation of the short-term power spectrum of a second on a non-linear Mel scale of frequency. 13 Mel-frequency bands are used at each time step. The relation between Mel-frequency and frequency can be concluded by:

$$Mel(f) = 2595 \ln(1 + \frac{f}{700}) \quad (1)$$

In which, f is the frequency. The Mel-frequency band is shown in Fig. 2. The feature was calculated to form a 2D tensor of shape 13×L_f, where 13 is the number of Mel-frequency bands. Each input feature map for audio stream has a dimensionality of 13×L_f.

Figure 1. The Processing pipeline of visual stream.

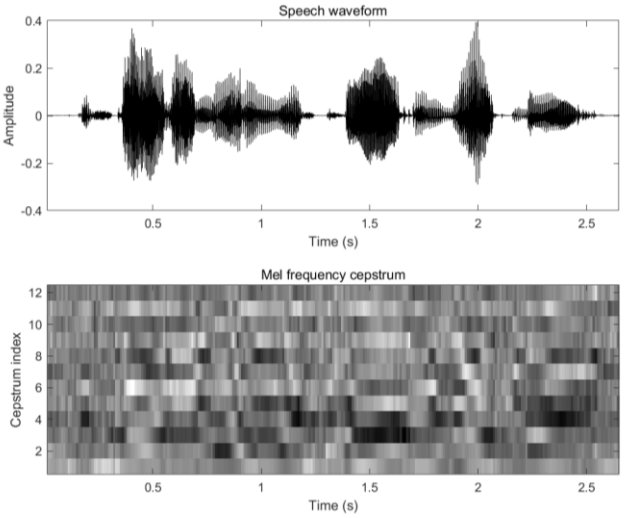

Figure 2. The processing pipeline of audio stream.

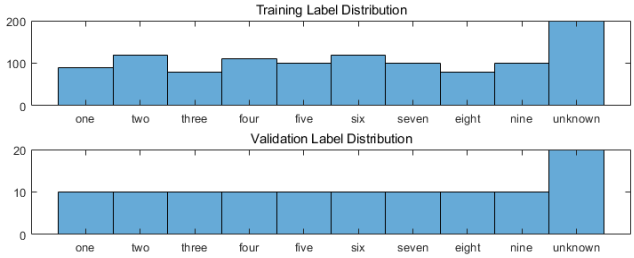

Figure 3. Distribution of the different class labels in the training and validation sets.

D. Data Augmentation And Add Noise Data

We performed data augmentation in each epoch by randomly insert pictures from the dataset video and either flipping them horizontally or adding small Gaussian noise to improve robustness of the network. A new label named "unknown" is added that means unintentional lips movement. The distribution of the different class labels in training and validation sets are shown in Fig. 3. The words "one" to "nine" is the vocabulary from the GRID audio-visual corpus. The "unknown" label means the unintentional lips movement which was added in data augmentation stage.

IV. ARCHITECTURE

This section describes two network architectures for lip reading. First, the visual to audio feature architecture maps a variable-length sequence of video frames to the auditory MFCC features. Second, the audio feature to text architecture distinguishes the text information from the audio feature.

A. Visual to Audio Feature Network

The input to the visual to audio feature architecture network is the pre-processed video slice reshaped as a tensor of shape 1×H×W×L_v where H and W is the height and width of the cropped lip region images respectively and L_v is the number of video slices.

TABLE II: Structure of the Visual to Audio Feature Network

Convolutional network block	
layers	Size
Input layer	1×128×128×5
Conv3D1+ReLU+Pooling	32×64×64 ×5
Conv3D1+ReLU+Pooling	32×64×64 ×5
Conv3D1+ReLU+Pooling	32×32×32 ×5
Conv3D1+ReLU+Pooling	32×16×16 ×5
Conv3D1+ReLU+Pooling	64×16×16 ×5
Conv3D1+ReLU+Pooling	1×64×8 ×10
Conv3D1+ReLU+Pooling	1×1×13×128

Spatiotemporal features of the video sequence are extracted using a 7-layer 3D convolutional network (CNN) [10] described in Table II. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. The output of the each convolution layer has the same size of the next layer of input and is followed by a 3D max pooling layer which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. The pooling layer helps the network increase robustness to the moving lip effect. The 3D convolutional operations can find the correlation between temporal and spatial information in pictures.

B. *Audio feature to text Network*

The network architecture used for training on audio to text streams is described in Table III. As with Table II, the architecture of the audio feature to text network is similar to the structure of the visual to audio feature network. However, the input of the audio feature can be seen as a picture. The architecture is simpler than the visual to audio one, due to the low dimension of the input. The last two layers are full connected networks. In a fully connected network, all nodes are interconnected. This kind of topology does not trip and affect other nodes in the network. The final output of the network is a 48-dimensional vector which means the 48-word vocabulary in dataset.

TABLE III: Structure of the Audio to Text Network

Convolutional network block	
layers	Size
Input layer	1×1×13×128
Conv3D1+ReLU+Pooling	32×64×64 ×5
Conv3D1+ReLU+Pooling	32×64×64 ×5
Conv3D1+ReLU+Pooling	32×32×32 ×5
Conv3D1+ReLU+Pooling	32×32×32 ×5
FC1	256
FC2	10

V. EXPERIMENT

In this section, we elaborate the details about implementation of the network and provide detailed information regarding the evaluation.

A. *Implementation*

As mentioned before, the pre-processed video was used as input to the network. We fixed the length of each video slice L_v to 5 (which is equivalent to 200 ms), and length of each audio slice $L_f = 128$ was also set accordingly such that the number of audio and video slices K are equal. We found that both width W and height H of cropped face images when set to 128 was sufficient to extract enough features.

We shuffled all the utterances from all 34 speakers and selected 9 words from "one" to "nine" from the vocabulary. We used Keras with Tensorflow [11] for implementing the network. Initialization of the network weights was performed using the proposed method by He [12]. We used batch normalization [13] for all layers, dropout [14] of p=0.25 every two layers in convolutional block and L2 penalty multiplier set to 0.0005 for all convolutional layers.

The audio feature to text network was trained using a batch size of 32 and the parameter α for ELU non-linearity was set as 1.

Optimization was performed using Adam [15] with an initial learning rate of 0.0001, which was reduced by a factor of 5 if validation loss was not improving in 4 consecutive epochs.

The loss function we used for all our networks was a combination of mean squared error (MSE) and correlation as given by:

$$\lambda \frac{1}{n}\sum_i (y_i-\hat{y}_i)^2 - \frac{\sum_i (y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{(\sum_i (y_i-\bar{y})^2)(\sum_i (y_i-\bar{\hat{y}})^2)}} \qquad (2)$$

in which, λ is the hyper-parameter for controlling balance between the two loss functions. We show that this loss function, with λ fixed to 1, performs better than both MSE and correlation.

The specific training process is shown in Fig. 4. I use single GPU (NVIDIA GeForce 1080Ti) which runs at around 7min for the 4875 iterations.

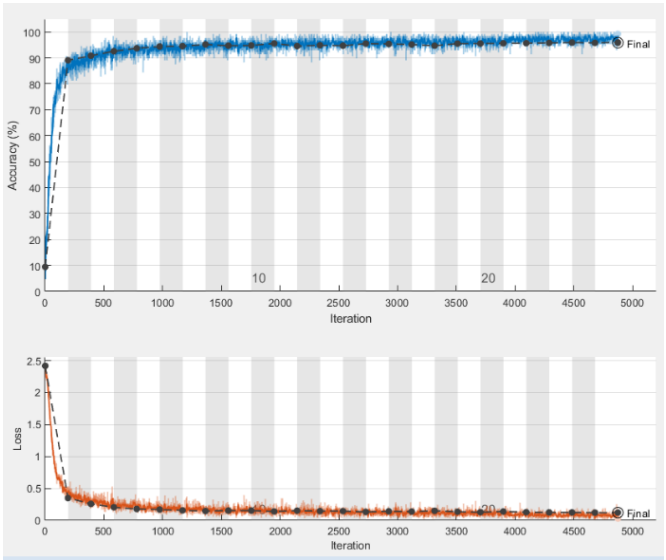

Figure 4. The accuracy and loss value in training process.

B. *Results*

In this part, we calculate the final accuracy on the training set and validation set. The confusion matrix is

shown in Fig. 5. The network is very accurate on this data set. The average validation accuracy is 92.76%. As for the recognition rate of "unknown" label, which means the unintentional lips movement for the speaker, can be up to 91.39%.

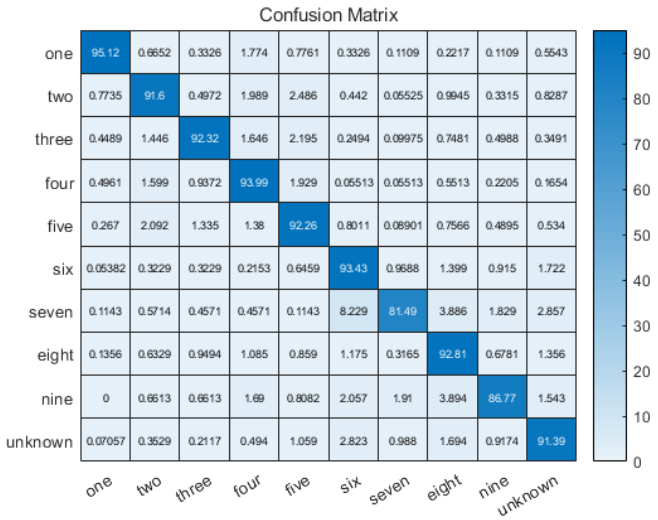

Figure 5. The confusion matrix in validation accuracy.

VI. CONCLUTION

In this paper, two network architectures are proposed for lip reading task. First, the visual to audio feature architecture maps a variable-length sequence of video frames to the auditory MFCC features. Second, the audio feature to text architecture distinguishes the text information from the audio feature. For the dataset, a new way of data augmentation is applied in each epoch by randomly insert pictures from the dataset video and adding small Gaussian noise to simulate the situation in real life. Our experiments showed that such a combination improves both quality and accuracy in real life situation. The result of the validation accuracy is 92.76%.

Due to the size of the GRID dataset, the vocabulary is relatively small. The future work is to collect more train data and to propose a more robust and accurate structure to detect the lips movement in real life situation for large vocabulary dataset.

ACKNOWLEDGMENT

This work was supported by my supervisor, Mr. Xiaodong Hu, for his invaluable advice, constant encouragement and precise modification, and I admire his knowledge and his personality. We would also like to thank: Lijie Qi for helping us evaluate the result; Meng Zhu for her phonetics guidance; Wenjiang Wu and Junming Tao for helpful comments.

REFERENCES

[1] Assael, Yannis M, et al. "LipNet: End-to-End Sentence-level Lipreading." (2016).

[2] Cooke, M, et al. "An audio-visual corpus for speech perception and automatic speech recognition. " Journal of the Acoustical Society of America 120.1(2006):2421-2424.

[3] I Goldschen, Alan J., O. N. Garcia, and E. D. Petajan. Continuous Automatic Speech Recognition by Lipreading. George Washington University, 1993.

[4] Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000.

[5] Wand, Michael, J. Koutník, and J. Schmidhuber. "Lipreading with long short-term memory." IEEE International Conference on Acoustics, Speech and Signal Processing IEEE, 2016:6115-6119.

[6] Chung, Joon Son, and A. Zisserman. "Lip Reading in the Wild." Asian Conference on Computer Vision Springer, Cham, 2016:87-103.

[7] Gergen, Sebastian, et al. "Dynamic Stream Weighting for Turbo-Decoding-Based Audiovisual ASR." INTERSPEECH 2016:2135-2139.

[8] Xiong, Xuehan, and F. D. L. Torre. "Supervised Descent Method and Its Applications to Face Alignment." Computer Vision and Pattern Recognition IEEE, 2013:532-539.

[9] Martin, A., D. Charlet, and L. Mauuary. "Robust speech/non-speech detection using LDA applied to MFCC." IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings IEEE, 2001:237-240 vol.1.

[10] Akbari, Hassan, et al. "Lip2AudSpec: Speech reconstruction from silent lip movements video." (2017).

[11] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[12] He, Kaiming, et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." (2015):1026-1034.

[13] Ioffe, Sergey, and C. Szegedy. "Batch normalization: accelerating deep network training by reducing internal covariate shift." (2015):448-456.

[14] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1(2014):1929-1958.

[15] Kingma, Diederik P, and J. Ba. "Adam: A Method for Stochastic Optimization." Computer Science (2014).