Voice2Lip: a novel approach to convert voice clips into motion of lips

Yunyang Li
School of Life Science and Technology
ShanghaiTech University
Shanghai, China
liyy2@shanghaitech.edu.cn

Ziye Liu
School of Informance Science and Technology
ShanghaiTech University
Shanghai, China
liuzy3@shanghaitech.edu.cn

Shaoting Peng
School of Informance Science and Technology
ShanghaiTech University
Shanghai, China
pengsht@shanghaitech.edu.cn

Cangli Yao
School of Informance Science and Technology
ShanghaiTech University
Shanghai, China
yaocl@shanghaitech.edu.cn

***Abstract*—Lip movements generation (LMG) can be defined as a pipeline to turn voice clips into lip motions. LMG can be applicable in many settings. However, little effort has been done in this sphere. This paper proposed a novel approach, Vioce2Lip, to ground-break the goldmine of LMG task. Our model is built upon canonical Seq2Seq model with GRU (Gated Recurrent Unit), and integrate audio and video feature-engineering workflow. We applied facial landmark detection and Mel-Frequency Cepstral Coefficients (MFCC) to extract the feature of lip movements and audio sequences. Our model has reached 0.02688 in RMSE (Root Mean Squared Error) in test set, and exhibited to be effective in practical predicting tasks.**

***Index Terms*—Lip Movement, MFCC, Face Landmarks**

I. Introduction

Given a slice of audio, it is desirable to revive the lip movements of the speaker. Since the task does not have to be spoken by one specific target identity, and neither the speech nor the image of target identity is required to be present in the training set, Lip Movements Generation (LMG) can be applied to a spectrum of settings. People with hearing impairment could incrementally pick up the way to pronounce via learning from generated lip movements. Lip Movements Generation can as well aid in generating animation videos and enhancing the comprehensiveness of highly confidential conversation and protecting the privacy of the speaker.

Lip Movements Generation resembles Automatic Speech Recognition (ASR) in that they share common inputs. Since a lot efforts have been put into ASR research, whereas LMG remains to be an inventive task. It is appealing to build models upon the state-of-the-art ASR models to extract audio embeddings. Mel-frequency cepstral coefficients (MFCC) are widely used to learn speech feature vectors in order to distinguish between different voices [1]. Listen, Attend, and Spell (LAS) model has also been implemented to extract video features, which adopts a seq-to-seq paradigm and exhibits to be easy-to-train and outperforms deep neural network-based Hidden Markov Model (DNN-HMM) [2].

To synthesize the video of a speaking mouth, various methods have been applied. Suwajanakorn et al. [3] developed a recurrent neural network (LSTM) to learn the mapping from raw audio features to mouth shape. However, it requires successive images to choose from in order to generate a video. Bo et al. [4] utilized bidirectional LSTM (Bi-LSTM) for audio/visual modelling. However, since they treat phoneme as the smallest unit, it might not be capable of capturing useful information. Combined with the progress in the field of speech recognition and the current stagnation in the field of lip movement, this paper proposes Voice2Lip model which is a seq-to-seq model which takes MFCC feature and Lip-related facial landmark as the input.

This work was supported by CS181 Teaching Team at ShanghaiTech University

II. Proposed Method

A. Problem Formulation

Lip Movements Generation (LMG): Given an audio sequence A_i from a random speaker and a corresponding static image F_i, the objective is to generate a video V_i which generates the lip movements of the given speaker.

Facial Landmarks: Coordinates of the fiducial facial landmark points around facial components and facial contour [5]. Facial landmarks capture the rigid and non-rigid facial deformations due to head movements and facial expressions.

Lip Landmarks: A subset of facial landmarks which capture the characteristics of lips, and act as key players in generating the targeting video sequences. Lip landmarks of a given frame is organized into a matrix $\Sigma^{20\times 2}$.

Gated Recurrent Unit (GRU): Gated recurrent unit (GRU) was proposed by Cho et al. [6]. Each recurrent unit adaptively capture dependencies of different time scales. Similarly to the LSTM unit, GRU has gating units which modulate information

flow within the unit, however, without having a separate memory cells. To capture the two-sided contextual information, bi-directional GRU (Bi-GRU) could be used where it stack two layer of vanilla GRUs, of which respectively pass information in opposite directions.

B. Overview of Voice2Lip

Voice2Lip comprises three parts: an encoder layer, an intermediate attention layers and a decoder layer.

1) Encoder: As shown in Fig 1, the encoder layer consists of 2 fully connected (FC) layers, with 512 and 256 neurons respectively. To prevent potential over-fitting problem, dropout with rate 0.1 was implemented. Outputs of the FC layers were pipped into a bi-directional GRU to capture long term dependencies as well as contextual information between sequences of inputs and to enforce smooth transitions between sequences of output.

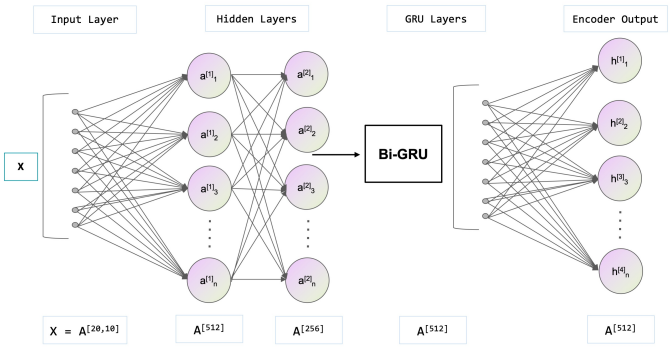

Fig. 1. Overview of Encoder

2) Attention Layer: The attention layer bridges the encoder layer and downstream decoder layer. Previously in encoder module, the output of FC layers were passed into a bi-directional GRU. The subsequent output and hidden representations of GRU at every given time-step was then used for attention layers. Attention weights were calculated via a linear layer which maps the concatenation of encoder outputs and a given hidden representation down to a vector of which size equals the length of the GRU outputs.

3) Decoder: The Decoder outputs one possible sets of lip landmarks per time-step, with aforementioned attention and another GRU as back-end. At a given time-step t, the previous hidden state h_{t-1} will calculate its attention with respect to the encoder outputs and utilize the weighted sum as the current input for GRU. The output of decoder GRU will be passed into a FC layer, which intends to give a plausible prediction of lip landmark matrix at that time step t.

C. Loss and optimization

Since the predicted point sets (represented as $\hat{S}^i$) and ground truth (represented as S^i) are both coordinates in 2d Euclidean space, it is natural to use Mean Square Error (MSE) as our criteria of loss function. Our loss function is defined as follows:

$$J(\hat{S}^i, S^i) = \frac{1}{|S|}\Sigma_{\hat{P}^i_j \in \hat{S}^i, P^i_j \in S}(\hat{P}^i_j - P^i_j)^2$$

With regard to optimization, we adopted Adagrad as our optimizer with learning rate 10^{-3}.

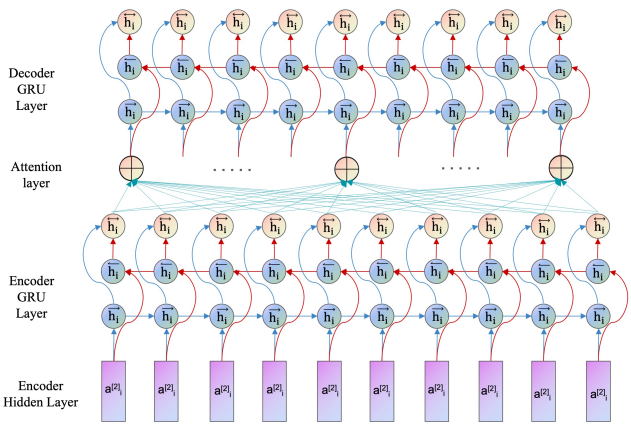

Fig. 2. Intermediate Attention Layer

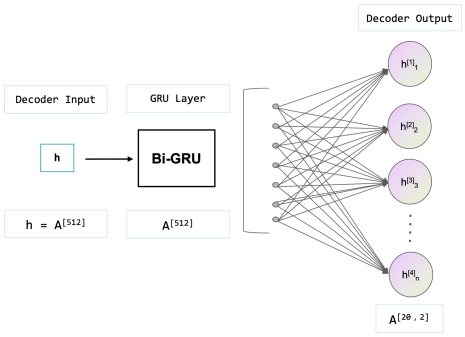

Fig. 3. Overview of Decoder

III. Results

A. Experimental Setup

1) Data Sets: As table I shows, we trained and evaluated our proposed model using CMLR and analysis the generalization capability on LYTalk data set.

CMLR: The first part is the Chinese Mandarin Reading (CMLR) Data Set, which was used as a training set and was originally designed to facilitate researches on automatic lip-reading. It contains 102,072 spoken sentences from 11 speakers, recorded between June 2009 and June 2018 from the national news program "News Broadcast".

LYTalk: The LYTalk Dataset (Luo & Yang Talk slices Dataset) is used mainly for evaluation of model generalization in multiple settings. It contains videos and audio of Prof.Luo from China University of Political Science and Law as well as famous TalkShow slices scraped from Bilibili.com. The second part was built to test the model's generalization capabilities.

B. Data-Preprocessing

1) Video: The objective of video data pre-processing is to encode the lip features of every data frame of videos into Numpy [7] array organized by date. Firstly we trimmed videos frame by frame and reorganized them by date. Dlib [8] facial landmarks detection were then implemented on every data frame, which can in turn yield the 68 landmarks of each face, while we selectively picked the 20 points on the lip (from 49 to 68) of interest. To ensure model generalization capability, we rotated the lip points with the leftmost (point 48) and rightmost point (point 54) on the lips as the rotation center to enforce this line to become horizontal. As a consequence, the y-coordinate of point 48 and 54 are the same. Finally we do normalization on the data points by subtracting each points by the coordinates of the previous midpoint and scaling the x-coordinate into [-1, 1]. We reorganized the data of each resulting point set into a numpy array of size $Number\ of\ Frames \times 20 \times 2$.

TABLE I
STATISTICS OF OUR DATASETS:

Dataset	CMLR	LYTalk
# of Speakers	11	2
# of Spoken Sentences	102, 072	NA
Language	Mandarin	Mandarin

2) Audio: The aim of audio data pre-processing is to encode spectral features of each segment of audios into Numpy array organized by date. Initially, each audio sequence is re-sampled to 48kHz to achieve consistency. Subsequently, a pre-emphasis filter was utilized to amplify high frequencies, which can further balance the frequency spectrum since high frequencies tend to have smaller magnitudes compared to lower frequencies [9]. Time domain sequences were normalized to -3dB by the formula $x_{normalized} = \frac{x}{\sqrt{2}\max(|x|)}$, after which we get $48kHz \times 100ms = 4800$ samples per second. Additionally, it is necessary to split the signal into short-time frames and add hamming window of 100ms to it to avoid the lose of frequency contours of the signal, as well to lower the risk of spectral leakage and maintain the number of frames as video. All the above steps finished, librosa could be applied to subtract Mel-Frequency Cepstral Coefficients (MFCC) features from the audio. The MFCC features are finally normalized by subtracting the mean with the function $x_{normalized} = x - \overline{x} + 0.8$ by Sutskever et al. [10]. Two of the aforementioned MFCC features is demonstrated in Fig 4 and Fig 5.

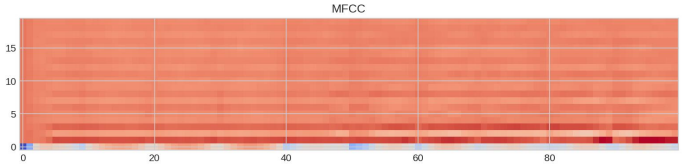

Fig. 4. Features Extracted from Speech Signals_1

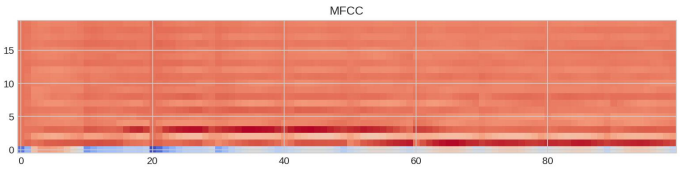

Fig. 5. Features Extracted from Speech Signals_2

C. Performance Evaluation

Initially, we used the difference between ground truth and prediction as an evaluation criteria. Under most circumstances, our model was capable of delineating the contours of lips, and describing the motion of lips correctly. Fig 6 and Fig 7 were two exemplary illustrations of the predicted motions and ground truth movements of lips. In the outer rim of the visualized lips, our predicted points co-localized with the ground truth with minor errors. In the inner rim of the lips, however, some points tended to be fluctuating, our model was capable of capturing that volatility at most time, but failed in some rare cases.

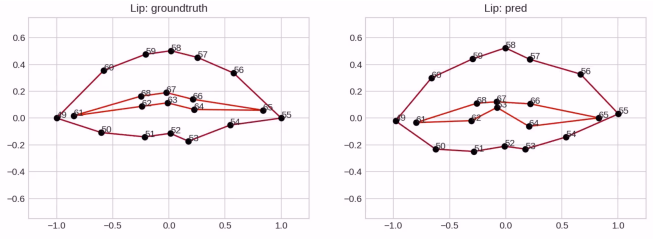

Fig. 6. Comparison between Ground Truth and our Prediction_1

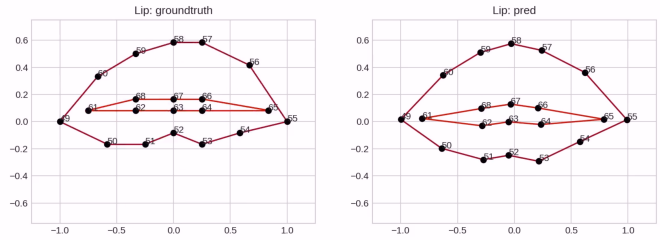

Fig. 7. Comparison between Ground Truth and our Prediction_2

D. Analysis

1) Convergence Analysis: As shown in Fig 8, we find that the training loss falls rapidly within the first 25 epochs, and begins to converge gradually in about 100 epochs.

GRU suffers from gradient explosion problem, and our model will only be able to converge if we do gradient clipping in the final training process.

IV. Conclusion and Future Work

As stated before, our current method to encode lip features simply rotate and normalize the 20 points given by dlib facial landmark detection, which is simply a vector of 20 dimensions

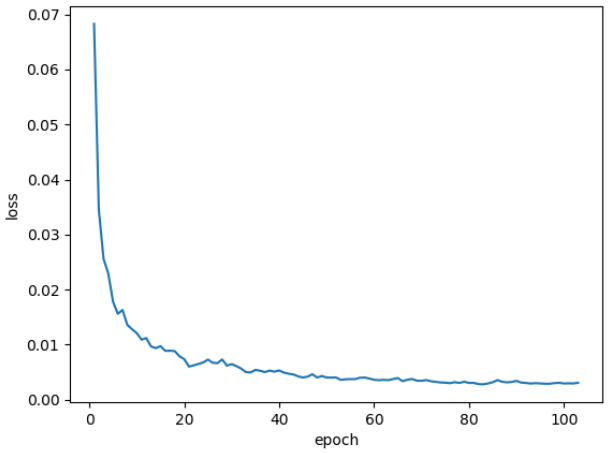

Fig. 8. Loss of Training Set

for every data frame of one video. However, in this method the lip features we get is a 'standard lip', which means we didn't take the variability of different people's lips into consideration. To reach state-of-the-art, we intent to make our lip features more general by implementing auto-encoder on the points given above, and hopefully, we can get 2 types of output features: one is the encoded lip features, the other is the features of this specific lip model. In this way we are able to suit the new lips better on different people. Besides, the performance of the abstraction of lip features can also be improved by operating on the entire video sequence instead of on every data frame of a given video, and for the reason that our datasets are all in Chinese, we don't support multi-language currently.

V. Acknowledgment

The research is originally supported by CS181 (Introduction to Artificial Intelligence) teaching team at ShanghaiTech University, with computational resources and constructive suggestions.

VI. Related Resource and Miscellaneous

In our project, we utilized following external library:

- **Numpy**: We use Numpy which support powerful usage of high dimensional arrays to store vectors when pre-processing data and load stored features before training.
- **PyTorch**: PyTorch was used as a backend to train our network.
- **Librosa**: Librosa was used in audio pre-processing progress. It is handy to get MFCC features.
- **Dlib**: Dlib was used in in mfcc_process.py to show face landmarks in video pre-processing progress.

Faceio.py was an auxiliary function and played as part of our dataloader. In Network.py, we implemented the architeture of our nerwork, which is built on PyTorch. In video_preprocessing.ipynb, video was pre-processed to the desirable features. In mfcc_process.py, audio was pre-processed to the desirable features.

References

[1] H. Yeganeh, S. M. Ahadi, and A. Ziaei, "A new mfcc improvement method for robust asr," pp. 643–646, 2008.

[2] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," pp. 4774–4778, 2018.

[3] B. Fan, L. Wang, F. K. Soong, and L. Xie, "Photo-real talking head with deep bidirectional lstm," pp. 4884–4888, 2015.

[4] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, July 2017.

[5] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, 2019.

[6] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[8] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[9] H. M. Fayek, "Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between," 2016.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, pp. 3104–3112, 2014.