LIP2AUDSPEC: SPEECH RECONSTRUCTION FROM SILENT LIP MOVEMENTS VIDEO

Hassan Akbari, Himani Arora, Liangliang Cao, Nima Mesgarani

Department of Electrical Engineering, Columbia University, New York, NY, USA

ABSTRACT

In this study, we propose a deep neural network for reconstructing intelligible speech from silent lip movement videos. We use auditory spectrogram as spectral representation of speech and its corresponding sound generation method resulting in a more natural sounding reconstructed speech. Our proposed network consists of an autoencoder to extract bottleneck features from the auditory spectrogram which is then used as target to our main lip reading network comprising of CNN, LSTM and fully connected layers. Our experiments show that the autoencoder is able to reconstruct the original auditory spectrogram with a 98% correlation and also improves the quality of reconstructed speech from the main lip reading network. Our model, trained jointly on different speakers is able to extract individual speaker characteristics and gives promising results of reconstructing intelligible speech with superior word recognition accuracy.

Index Terms— Lip Reading, Speech Synthesis, Speech Compression, Neural Networks

1. INTRODUCTION

A phoneme is the smallest detectable unit of a (spoken) language and is produced by a combination of movements of the lips, teeth and tongue of the speaker. However, some of these phonemes are produced from within the mouth and throat and thus, cannot be detected by just looking at a speaker's lips. It is for this reason that the number of visually distinctive units or visemes is much smaller than the number of phoneme making lip reading an inherently difficult task.

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. Some of the early works [1, 2, 3] used a combination of deep learning and hand-crafted features in the first stage followed by a classifier. More recently there has been a surge in end-to-end deep learning approaches for lip reading which focussed on either word level or sentence-level prediction using a combination of convolutional and recurrent networks [4, 5, 6].

Our proposed network also follows a similar structure in the sense that it consists of convolutional layers to extract features from the video followed by an LSTM to encode temporal dependencies. However, we model our output as a generative task over the audio frequency space to directly produce the corresponding speech signal at every time step allowing us to recover not just the information but also the style of articulation. This opens an entirely new world of applications in the audio-visual domain - improving audio in existing videos in the sequences where someone is talking such as blogging videos or news anchoring videos, enabling video-chatting in silent areas like libraries or in noisy environments. We describe below two works that are most closest to ours.

Milner *et al.* [7] reconstructed audio from video by estimating the spectral envelope using a neural network composed solely of fully connected layers and trained on hand-engineered visual features obtained from mouth region. This approach had the limitation of missing certain speech components such as fundamental frequency and aperiodicity which was then determined artificially thereby compromising quality in order to maximize intelligibility. Ephrat *et al.* [8] modified this technique by using an end-to-end CNN to extract visual features from the entire face while applying a similar approach for modeling audio features using 8th order Linear Predictive Coding (LPC) analysis followed by Line Spectrum Pairs (LSP) decomposition. However, it also suffered from the same missing excitation parameters resulting in an unnatural sounding voice.

Our sound generation model differs from these as we use the spectrogram model proposed by Chi *et. al* [9]. Inspired by the sound processing system in human brain, it uses bio-inspired filter-banks and non-linear compressions to calculate spectrogram and gives a higher quality of re-synthesis to speech than traditional spectrograms. However, the spectrograms in itself are highly correlated and usually difficult for the networks to learn accurately. To bypass this issue, we designed a deep autoencoder to extract compressed features of the spectrogram that form the target for our main lip reading network. We show through extensive evaluations that this spectrogram based combination of autoencoder and lip-reading network allows us to generate an acoustic signal that is much more natural sounding than the previous approaches. At the same time, it does not compromise on intelligibility and gives a superior word recognition accuracy in human based evaluations.

The authors would like to thank Xiaodong Cui for his constructive feedback and discussions.

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

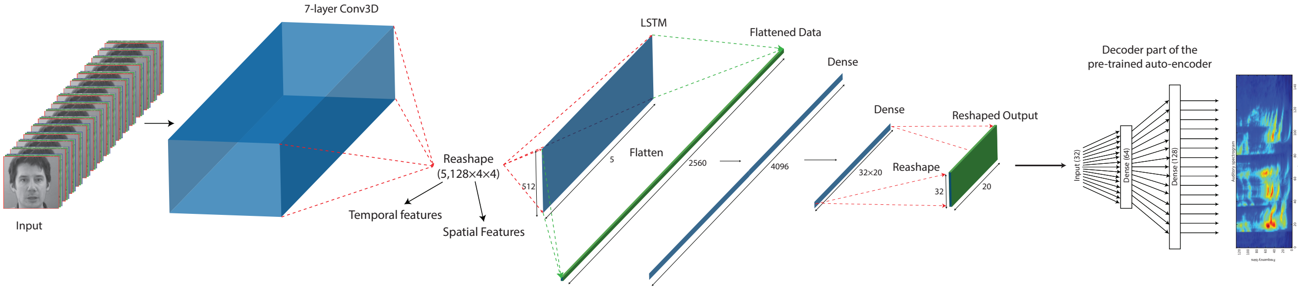

Fig. 1. The overall structure for the proposed network. The network gets a video sequence, captures spatiotemporal features and generates coded features of the audio for that video. These features are then decoded using the pre-trained autoencoder.

2. METHOD

2.1. Data Preparation

Each video frame was converted to grayscale and normalized and the face region was extracted and resized to have dimensions WxH. The videos were then divided into K non-overlapping slices each of length $\mathrm{L_v}$. First and second order temporal derivatives at each frame were stacked along the first dimension to form a 4D tensor of shape (3, H, W, $\mathrm{L_v}$). The target spectrogram was also divided into K slices with length $\mathrm{L_a}$ and no overlap.

2.2. Network I: Autoencoder

In order to compress the auditory features, we designed a deep autoencoder described in Table 1. Both input and output of this network is the 128 frequency bin auditory spectrogram [9] with a bottleneck of size 32 (which was found to be the optimal as shown in the experiments). In addition to this, the output of the activation of the bottleneck is contaminated with Gaussian noise during training to improve robustness of the decoder network.

2.3. Network II: Lip Reading Network

The lip reading network extracts spatiotemporal features from the input video sequence using a 7-layer 3D convolutional network described in Table 1. The output of the CNN block is reshaped to a tensor of shape ($\mathrm{L_v}$, $\mathrm{N_{st}}$) where $\mathrm{N_{st}}$ represents the spatio-temporal features extracted by the CNN. This reshaped tensor is fed into a single-layer LSTM network with 512 units to capture the temporal pattern which is followed by a fully connected layer and then finally the output layer. The output layer has $32\times\mathrm{L_a}$ units to give the 32-bin$\times\mathrm{L_a}$-length bottleneck features. At inference, this is connected to the decoder part of the pre-trained autoencoder to reconstruct the auditory spectrogram. The overall structure of the proposed network can be seen in Figure 1. The audio waveform can then be reconstructed from the output spectrogram using [9].

Table 1. Structure of the proposed networks

Autoencoder		Convolutional network block	
Layers	Size	Layers	Size
Input layer	(None, 128)	Input layer	(None, 3, 128, 128, 5)
Dense (512) LeakyReLU	(None, 512)	Conv3D (32) LeakyReLU MaxPool (2,2,1)	(None, 32, 64, 64, 5)
Dense (128) LeakyReLU	(None, 128)	Conv3D (32) LeakyReLU MaxPool (2,2,1)	(None, 32, 32 32, 5)
Dense (64) LeakyReLU	(None, 64)	Conv3D (32) LeakyReLU MaxPool (2,2,1)	(None, 32, 16 16, 5)
Dense (32) Sigmoid	(None, 32)	Conv3D (64) LeakyReLU	(None, 64, 16, 16, 5)
Gaussian Noise (σ=0.05)	(None, 32)	Conv3D (64) LeakyReLU MaxPool (2,2,1)	(None, 64, 8, 8, 5)
Dense (64) LeakyReLU	(None, 64)	Conv3D (128) LeakyReLU	(None, 128, 8, 8, 5)
Dense (128) LeakyReLU	(None, 128)	Conv3D (128) ELU (alpha=1.0) MaxPool (2,2,1)	(None, 128, 4, 4, 5)

3. EXPERIMENTS

3.1. Dataset

The dataset used for training the network was the GRID audio-visual corpus [10] 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 51-word vocabulary (shown in Table 2). Videos and audios are both 3 seconds long and sampled at 25 fps and 44 kHz, respectively.

They were pre-processed as described in section 2.1 with $\mathrm{L_v} = 5$, $\mathrm{L_a} = 20$, $\mathrm{K} = 15$, $W = 128$, $H = 128$. In addition, the original audio waveform was downsampled to 8kHz. We conducted our training and evaluation on videos from two male speakers (S1, S2) and two female speakers (S4, S29) using 80%-10%-10% train-validation-test split.

Table 2. GRID vocabulary

Command	Color	Preposition	Letter	Digit	Adverb
bin	blue	at	A-Z	0-9	again
lay	green	by	minus W		now
place	red	in			please
set	white	with			soon

3.2. Implementation

We used Keras [11] with Tensorflow backend [12] for implementing the network. Initialization of the network weights was done using [13]. We used batch normalization [14] for all layers, dropout [15] of p=0.25 every two layers in convolutional block and L2 penalty multiplier=0.0005 for all convolutional layers. For LSTM and MLPs, we only used dropout of p=0.3. We first trained autoencoder on the 128 frequency bin auditory spectrogram of the training audio samples with a mini-batch size of 128. After training, we extracted the 32-bin bottleneck features which we then provided as target features for the main network. The main lip reading network was trained using a batch size of 32 and data augmentation was performed on the fly by either flipping images horizontally or adding small Gaussian noise. Optimization was performed using Adam [16] with an initial learning rate of 0.0001. The loss function we used for all our networks was a combination of mean squared error (MSE) and correlation. This loss function (that we call it CorrMSE) is given by:

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

in which, λ, which we set as 0.5, is the hyper-parameter for controlling balance between the two loss functions. For the auditory spectrogram generation and audio waveform reconstruction, we used NSRtools [9] with frame length=10, time constant=10, nonlinear factor=-2 and shift=-1.
Code and demo available online:
`https://github.com/hassanhub/LipReading`.

3.3. Network evaluation

For evaluating the networks, we measure accuracy in the frequency domain using 2D correlation between the reconstructed ($\hat{\mathbf{Y}}$) and the actual auditory spectrogram ($\mathbf{Y}$). To assess the quality of the reconstructed audio, we use the standard Perceptual Evaluation of Speech Quality (PESQ) [17]. We also measure intelligibility of the final reconstructions using Spectro Temporal Modulation Index (STMI) [18].

For a baseline comparison, we used the publicly available code of Vid2Speech [8]. We followed their suggestion and trained (and tested) their model individually for each speaker which resulted in 4 different models. Whereas, our autoencoder and lip-reading network was trained *jointly* on all 4 speakers resulting in a single model. Results for the two methods can be found in Table 3. As can be seen from the table, the reconstructed speech using our proposed method have both higher quality and intelligibility compared to the baseline (paired t-test, $p < 0.001$). Our method also gives a higher correlation which indicates a higher accuracy of reconstruction in the frequency space.

Table 3. Quantitative evaluation of our method compared to Vid2Speech

Measure	Method	S1	S2	S4	S29	Average
PESQ	Ours	**2.07**	**2.01**	1.61	**1.84**	**1.88**±0.35
	Vid2Speech	1.90	1.74	**1.79**	1.62	1.76±0.24
Corr2D	Ours	**0.89**	**0.88**	**0.88**	**0.87**	**0.88**±0.03
	Vid2Speech	0.62	0.52	0.64	0.65	0.61±0.06
STMI	Ours	**0.82**	**0.84**	**0.84**	**0.82**	**0.80**±0.04
	Vid2Speech	0.58	0.59	0.46	0.48	0.52±0.08

To shed light on that, we generated spectrograms for a random sample from the test set and compared both reconstructions with the original, which can be seen in Figure 2. It is clear that although both methods fail in retrieving high frequency information which is partly due to the nature of the task, our method is able to recover the original spectrogram including the pitch information to a higher degree of accuracy which is lacking in Vid2Speech. In addition, the model can successfully handle connections between windows and learns the time pattern for continuing a phoneme from one window sample to another.

3.4. Human evaluations

We conducted a survey on Amazon Mechanical Turk to evaluate the intelligibility and quality of reconstructed speech by our method as well as Vid2Speech. The task was to transcribe each audio sample (selected randomly from the test set) and answer questions on its quality. These questions asked workers to rate how natural each sample sounded on a scale of 1-5 (1:unnatural, 5:very natural) and to guess the gender of the speaker from 3 choices (male, female, hard to say). Each audio sample was evaluated by 20 workers who were provided with the GRID vocabulary and allowed to replay the audio unlimited times. Table 4 shows the result of this evaluation.

Our model achieves a higher word recognition accuracy for 3 out of 4 speakers thereby improving upon the baseline by average 5% and leading us to conclude that the reconstructed speech by our method is more intelligible. The accuracy for random guessing is 19% here. The fact that accuracy for S4 is more for Vid2Speech and the emphasis on this speaker in the main paper rises this question that their model might be fine-tuned on this specific speaker. We also observed that the accuracy for the baseline is not as high as reported in the main paper (audio-only accuracy) which might be because of differences in human evaluation implementation (biased tasks or subjects). In terms of quality of speech, our method consis-

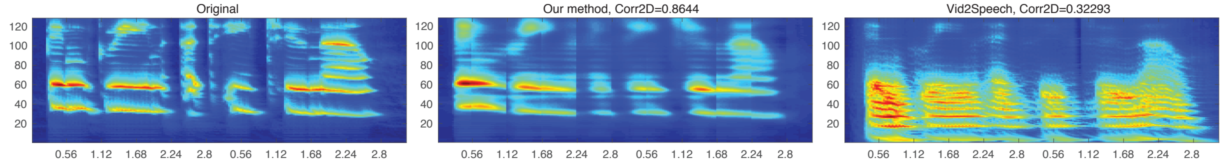

Fig. 2. Reconstructed audio spectrograms

Table 4. Speaker-wise assessment from Human evaluations

Measure	Method	S1	S2	S4	S29	Avg
Accuracy	Vid2Speech	35.2	51.2	**57.7**	59.6	50.9
(%)	Ours	**49.3**	**56.1**	54.9	**63.7**	**55.8**
Natural	Vid2Speech	1.13	1.45	1.44	1.37	1.35
sound (1-5)	Ours	**1.69**	**1.48**	**1.67**	**1.67**	**1.63**
Correct	Vid2Speech	58.0	77.0	21.0	17.0	43.2
Gender (%)	Ours	**85.83**	**79.2**	**83.3**	**92.0**	**85.1**
Hard to say	Vid2Speech	36.0	16.0	42.0	32.0	31.5
(%)	Ours	**9.16**	**12.5**	**10.0**	**2.0**	**8.4**

tently outperforms the baseline. Not only is our reconstructed speech more natural, it also retains speaker dependent characteristics such as gender which is due to correct pitch information retrieval that is missing in Vid2Speech.

3.5. Ablation study

We conducted ablation experiments to examine:
1. The effect of the number of the bottleneck nodes
2. The effect of dropout and additive noise to bottleneck
3. The necessity of autoencoder and CorrMSE loss function.

For 1 and 2, we varied the number of bottleneck nodes and trained the autoencoder both with and without dropout and additive noise to bottleneck. We then conducted evaluations both for the autoencoder output and the lip-reading network reconstruction, the results of which are presented in Table 5.

Table 5. Quantitative evaluation of autoencoder architectures

Measure	16 nodes (w/ noise)	32 nodes (w/ noise)	64 nodes (w/ noise)	No noise (32 nodes)	Dropout (32 nodes)
			Autoencoder output		
PESQ	2.76	2.81	**2.92**	2.88	2.33
Corr2D	0.98	0.98	**0.99**	0.97	0.95
			Lip-reading network reconstruction		
PESQ	1.19	**1.76**	1.26	1.09	1.29
Corr2D	**0.89**	**0.89**	0.87	0.46	0.88

As expected, on increasing the number of nodes both correlation and quality at the autoencoder output improve but it becomes harder for the lip-reading network to reconstruct these large features. Using 32 nodes achieves a balance between this trade-off, still resulting in a 98% correlation. Also, it can be seen that using dropout makes the results worse, and by using noise although the accuracy of reconstruction at the autoencoder output drops slightly, however it significantly improves overall performance of the lip-reading structure. We believe this is because training with Gaussian noise allows autoencoder to handle variations in the input videos to the lip-reading network and the resulting variations in the output. Based on these findings, we used 32 nodes bottleneck with additive noise for all our evaluations.

In order to understand the role of autoencoder (task 3), we trained two variants of the main network - using bottleneck features as target (Bott.) versus directly feeding the spectrogram to the output layer of the main lip-reading network (Spec.). In addition to this, we varied λ to correspond to the loss functions: MSE ($\lambda = 1$), Corr2 ($\lambda = 0$) and CorrMSE ($\lambda = 0.5$). For these experiments, we only trained and evaluated our model on speaker S29. We found experimentally that using CorrMSE as loss function with bottleneck features as target results in the best performance among all the conditions. Table 6 summarizes these findings.

Table 6. Quantitative measures for ablation study

Measure	MSE (Bott.)	Corr (Bott.)	CorrMSE (Bott.)	MSE (Spec.)	Corr (Spec.)	CorrMSE (Spec.)
PESQ	1.54	1.73	**1.76**	1.29	1.69	1.58
Corr2D	0.84	0.88	**0.89**	0.87	**0.89**	0.88

4. CONCLUSION

In this paper, we proposed a structure consisting of a deep autoencoder for coding speech and a deep lip-reading network for extracting speech-related features from the face. We showed that such a combination improves both quality and accuracy of the reconstructed audio. We also conducted different tests for comparing our network with a strong baseline and showed that the proposed structure outperforms the baseline in speech reconstruction. Future work is to collect more train data, include emotions in reconstructed speech, and to propose an end-to-end structure to directly estimate raw waveform from facial speech-related features.

5. ACKNOWLEDGEMENT

This work was funded by a grant from the National Science Foundation CAREER Award, and the Pew Charitable Trusts.

6. REFERENCES

[1] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[2] Stavros Petridis and Maja Pantic, "Deep complementary bottleneck features for visual speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2304–2308.

[3] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[4] Michael Wand, Jan Koutník, and Jürgen Schmidhuber, "Lipreading with long short-term memory," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6115–6119.

[5] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas, "Lipnet: Sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.

[6] Joon Son Chung and Andrew Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 87–103.

[7] Ben Milner and Thomas Le Cornu, "Reconstructing intelligible audio speech from visual speech features," *Interspeech 2015*, 2015.

[8] Ariel Ephrat and Shmuel Peleg, "Vid2speech: Speech reconstruction from silent video," *arXiv preprint arXiv:1701.00495*, 2017.

[9] Taishih Chi, Powen Ru, and Shihab A Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *The Journal of the Acoustical Society of America*, vol. 118, no. 2, pp. 887–906, 2005.

[10] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[11] François Chollet et al., "Keras," `https://github.com/fchollet/keras`, 2015.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[14] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[15] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] AW Rix, J Beerends, M Hollier, and A Hekstra, "Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *ITU-T Recommendation*, vol. 862, 2001.

[18] Mounya Elhilali, Taishih Chi, and Shihab A Shamma, "A spectro-temporal modulation index (stmi) for assessment of speech intelligibility," *Speech communication*, vol. 41, no. 2, pp. 331–348, 2003.