Natural audio-to-video generation with representation learning

Chih-Yu Lai
Electrical Engineering and Computer Science, MIT
Cambridge, Massachusetts
chihyul@mit.edu

1. Abstract

This project is dedicated to investigate the difficult audio-to-video generation with representation learning. Audio to video generation is an interesting problem that has abundant application across several industry fields. Here, we propose a novel training flow consisting of pre-trained models (StyleGAN3, Wav2Vec2, MTCNN networks), newly trained models (variational autoencoders and transformers), and an adversarial learning algorithm. To the best of the author's knowledge, this is the first implementation of audio-to-video generation using a pre-trained StyleGAN3. The input is a speech audio sequence and an image of a face. Our model will learn to "animate" the face by predicting the facial expressions and lip movement. We find that the latent code of our generative model can be encoded 16-fold into a 96-dim vector that retains the information of the talking face. By using this method, audio-to-video generation can be realized without training any generative models, and only latent codes should be predicted from audio. This minimizes our requirement for dataset size and training time. (The reconstructed videos can be found here.)

2. Introduction

Audio to video generation can be regarded as an extension of audio to image in the time domain [1]. In 2017, Chung et al used encoder-decoder CNNs to develop a model capable of generating talking faces. The method runs in real time and is applicable to faces and audio not seen at training time. They use a known image or video to modify frames and generate the video sequence [2]. Another research proposed in CVPR 2020 used an algorithm OneShotA2V, which leverages curriculum learning to learn movements of expressive facial components. This paper uses spatially adaptive normalization and a generative adversarial network, which adapts to any given unseen selfie by applying fewshot learning with only a few output update epochs [3]. The most recent work on audio to video proposed a novel discrete variational autoencoder with adversarial loss, dVAE-Adv, which learns a new discrete latent representation which they called Memcodes [4].

Most of the audio-to-video literature train generative adversarial networks (GANs) for generating image sequences. In particular, the GANs are used for generating facial images that have different expressions. They focus mainly on lip movement the most, since it is most correlated with speech. There are also other GANs that can synthesize talking faces. For instance, StyleGAN is a generator architecture that can automatically learn unsupervised separation of high-level attributes such as pose and identity, and also enables scale-specific control of the synthesis [5]. The latent codes in StyleGAN are disentangled so that it can be easier to interpolate different properties of human faces. In its most recent update, the StyleGAN3 is specifically designed to improve on its previous version StyleGAN2 to guarantee equivariant translation and rotation at pixel scale details [6], making it suitable for generation of video and animations. Different high-level attributes are encoded in each layer of the latent code, so one can target different layers for modifying targeted high-level attributes.

From the other end, prediction of frame sequences from audio can be treated as a Seq2seq learning problem. Lin et al. applied seq2seq learning to solve a similar audio-visual event localization problem [7]. Since one frame results from not only its corresponding section in the audio sequence but also its context, one needs to transform the audio sequence into some contextual representation that can be further used for predicting the output frames. The Wav2Vec2 model learns powerful representations of given an audio sequence by masking the input into a latent space and minimizes a Connectionist Temporal Classification (CTC) loss defined over a quantization of the latent representations which are jointly learned [8]. The output of the Wav2Vec2 model contains abundant contextual representation that can be further used for several downstream tasks.

In this project, we make use of pre-trained StyleGAN3 and Wav2Vec2 models, and propose a novel training flow that can predict animated talking face images from one image and audio sequence (Fig. 1). First, we encode cropped

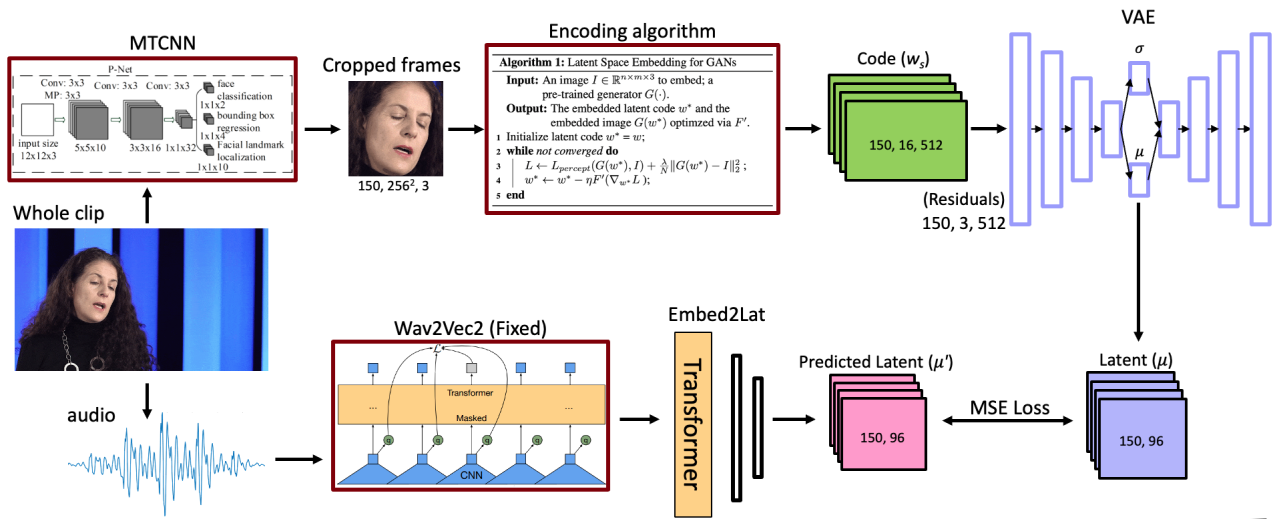

Figure 1. Scheme for speech to video synthesis in this project.

image sequences into latent codes of StyleGAN3. Then we further encode the residuals of certain layers of the latent codes into another latent representation using a variational autoencoder (VAE). The latent representations of the VAE delineate some compact form of the talking face. Then, we train a network to predict the latent representations from the embedded audio sequence of the original video. The results show clear relationship between the movement of lips of the reconstructed frames and the audio sequence.

3. Talking Face Generation from Single Image and Audio

The training flow for speech-driven facial synthesis is shown in Fig. 1. In total, there are 4 pre-trained models and 3 newly-trained models (Table 1). First, the audio and image sequences of the video clips are separated. The images sequences are cropped and resized to 256×256 pixel size that shows only the talking person's face. The cropped image sequences are encoded into latent codes (w_s), which can be used as the input for a pre-trained generative model. The residual of the layers that control general facial expressions of latent codes (Δw_s) are further encoded into another latent representation (μ) using a variational autoencoder (VAE). This latent representation learned by the VAE is our training target. On the other hand, the audio sequences are embedded using Sequence to Sequence (Seq2Seq) model. we use the embedded audio sequence (e) as our training input to train a model that can predict the latent representation (μ') of the VAE, and try to minimize the loss between μ' and μ.

3.1. Face Detection and Frame Cropping

Most of the videos used for training in this project is derived from raw videos, where only a portion of each frame contains the face of the talking person. Here, we used Multi-task Cascaded Convolutional Networks (MTCNN) for detecting the talking face in each frame and cropping [9]. Since the person or the camera might be moving, the size and position of the cropped sections might change. Intuitively, we tried to crop the face in each frame and concatenate them to form the talking face frame sequences. However, we find that this results in a serious jittering. So we only applied face detection in each 10 frames, and linearly interpolated the detection boundaries' position between them. The result is a smooth enough cropped face sequence. The number of frame intervals for detection should depend on the frame rate of each video sequence. MTCNN cannot identify who is talking and who is not if there are multiple faces in one frame. So, we discarded whole sequences in any detected frame, multiple faces are detected. we also discarded whole sequences whenever no faces are detected. The MTCNN is configured so that only faces larger than 200×200 pixel in size count as a valid detection.

3.2. StyleGAN3 Latent Code Construction

The StyleGAN3 we used is pre-trained on the Flickr-Faces-HQ Dataset (FFHQ) [5], previously used for training its previous versions StyleGAN and StyleGAN2. Specifically, we use the *stylegan3-r-ffhqu-256x256* model, which is equivariant to rotation, and outputs a 256×256 image given a 16×512 latent code (w).

We make the assumption that for each cropped frame in the video sequence, there exists at least one corresponding latent code (w^*) that has a reconstruction loss (L) less than a

Table 1. Models used in this project.

Model	Usage	Input	Output
MTCNN (pre-trained)	Face detection and cropping	Raw image	256×256 image
StyleGAN3 (pre-trained)	Face image generation	w	256×256 image
VAE encoder	Encode latent code residual to latent representation	$\Delta w^*_{5,6,7}$	μ^*
VAE decoder	Decode latent representation to latent code residual	μ	$\Delta w_{5,6,7}$
Wav2Vec2 (pre-trained)	Embed audio sequence	Raw Audio	e
Embed2Lat	Predict latent representation from audio embedding	e	μ
pSp encoder (pre-trained)	Encode face image to latent code	256×256 image	w^*

certain threshold (L_{thres}). we used the encoding algorithm described in [10] to find the latent code, which is an adversarial method that uses a weighted sum of Mean Squared Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) loss function to find w^*. we tried several different weights between MSE and LPIPS loss, the final loss is defined as:

$$L = L_{MSE} + 10L_{LPIPS} \tag{1}$$

where the learning rate (lr) is set to 0.02, and decays to 0.01 half-way through training. For the first frame, a random w^* is initialized, and 1000 steps are taken. For every subsequent frame, we assume that there will not be a lot of difference compared to the previous frame, and a reasonable w^* can be found near the encoded w^* of the previous frame. Therefore, only 5 steps are taken for each subsequent frame, with $lr = 0.01$.

3.3. VAE Latent Representation Encoder

The latent codes (w^*) have a dimension of 16×512. Despite the abundant information being encoded, it will become an under-defined problem if we tried to predict w^* from the audio sequence. Hence, we used a variational autoencoder (VAE) for encoding the latent codes. We apply a residual learning method, where w^*_i is the latent code of the i^{th} frame, and the target is

$$\Delta w^*_i = w^*_i - w^*_0 \ (i \in \mathbb{N}) \tag{2}$$

We chose to encode the 5^{th} to 7^{th} layer of the latent code ($\Delta w^*_i[5, 6, 7]$) into another latent representation μ^* that has a dimension of 96. These three layers contain high-level information regarding facial expressions, and most importantly, the shape of lips [6]. There are four linear layers in the encoder and decoder of the VAE, where the input dimensions for the encoder are 1536, 768, 384, and 192, respectively. The structure of the decoder is the exact reverse of the encoder. Gaussian Error Linear Units (GELU) are used as the activation function of each layer. We use a batch size of 10000, learning rate of 0.001, and train the network for 500 epochs. A combined reconstruction MSE and KL divergence loss is used during training.

Since we are only encoding three layers, we expect the pose of angle of the reconstructed face look similar to only the first frame, and not the subsequent ones. There may be movement and rotation of the face from the original cropped sequence, but we assume it does not have a strong relationship between the input audio sequence. Therefore, by using only the 5^{th} to 7^{th} layer, we can actually reduce some noise during training.

3.4. Audio Embedding using Wav2Vec2

Raw audio sequences are preprocessed to be in a 16-bit, 16kHz, mono channel format. Then they are fed into the pre-trained Wav2Vec2 base model [8] to produce embedded vectors (e) that have a shape of $50T \times 768$, where T is the duration of the audio in seconds, and 768 is the output size of the transformer of Wav2Vec2. According to [8], e contains contextualized information at each time step, which is a good starting form of representation of the whole sequence, and suitable for downstream training tasks.

3.5. Embed2Lat Prediction

We construct the Embed2Lat model to predict μ^* from e. Embed2Lat consists of a transformer that has the same structure with the transformer in Wav2Vec2, an adaptive average 1D pooling layer, and a linear layer. The input shape of Embed2Lat is $50T \times 768$, while in the pooling layer, the first dimension is pooled to $r_{fps}T$, where r_{fps} is the frame rate of the image sequences. The second dimension is 96, which corresponds to the output size of the linear layer and the size of μ^*. The learning rate is set to 10^{-5}, and a batch size of 256 is used. We minimize the MSE loss between μ^* and our predicted μ.

4. Experiments

The computational power requirement is heavy due to the requirement for video processing, usage of large-scale GANs, and seq2seq training on a large dataset. A NVIDIA GeForce 3090X GPU is used along with AMD Ryzen 9 5950X 16-Core Processor running on ArchLinux operating

Figure 2. Reconstructed (left) and original (right) cropped image using the encoding algorithm in [10].

Figure 3. Reconstructed (left) and original (right) cropped image using the VAE.

system. Two 32GB DDR4 DRAMs running at 2666MHz are used.

4.1. Dataset

The raw dataset used for training is AVSpeech - a large-scale audio-visual dataset comprising speech video clips with no interfering background noises [11]. The AVSpeech dataset consists of roughly 4700 hours of videos segments, each 3-10 seconds long, with audible sound belonging to a singe speaking person (although more than one person might appear in the clip). We used the testing dataset which consists of 133,286 valid clips. All the clips are separated into audio and image sequences. After extracting the cropped sequences using the method in sec. refsec:mtcnn, a total of 9,996 sequences are constructed, and the frame rate is tuned to 25FPS.

4.2. StyleGAN3 Encoding

Fig. 2 shows the original and reconstructed cropped image from the learned w^* of each frame. Despite the subtle differences between the images, the reconstructed image sequence look mostly the same as the original image sequence. However, this is only true for a clear enough cropped face, and the background should mostly be in plain color or should differ a lot from the color of the face. Also, if the talking face is turning or moving around rapidly, the loss will increase, and the reconstruction image would be blurry.

One drawback using the encoding algorithm in sec. 3.2 is it takes approximately 2 minutes to encode one image from a randomized w^*, and another 2 minutes to encode all the subsequent frames. If only running on one CPU thread, it will take around one month to encode all 9,996 sequences. Thus we run the encoding algorithm on multiple threads, and tried to maximize the efficiency of our GPU. We are able to speed up the encoding rate to around 45 seconds per clip, with the whole encoding duration lasting around 5 days.

During this process, we tried to encode the images using a pre-trained encoder - ReStyle-pSp [12]. This encoder is trained on the FFHQ dataset over the StyleGAN3 generator, and enables image inversion back to w^*. A whole sequence can be encoded just in a few seconds. However, the reconstructed images generally differ a lot from the original images (data not shown). This might be due to the distributional difference between the AVSpeech and FFHQ, and also the cropped images generally do not have a clean enough background and high enough quality to be properly encoded.

4.3. Code2Lat Training Results

After training the VAE for encoding $\Delta w_i^*[5,6,7]$ to μ^*, we tested on our validation dataset, and visualized the reconstructed sequence. The reconstructed i^{th} frame ($w_i'^*$) can be formalized as:

$$w_i'^* = w_0^* + \mathbf{D}(\mathbf{E}(\Delta w_i^*[5,6,7]))\sigma_{code} + \mu_{code} \quad (3)$$

where $\mathbf{D}$ is the VAE decoder, $\mathbf{E}$ is the VAE encoder, σ_{code} and μ_{code} are global constants calculated from the standard deviation and mean of all Δw^*. Fig. 3 shows the reconstructed and original image sequence from w'^* and w^*. We see that the reconstructed image retains the facial expressions including lip movement, but the movements are weakened compared to the original image. Also, it does not retain the rotation and translation of the face, which the information is mostly included in the other layers of w^*. Overall, the reserved lip movements should be a good qualitative indicator that the encoded μ^* do contain the information related to speech signal.

4.4. Embed2Lat Training Results

The MSE loss can reach a value of around 0.273 for the validation set, and 0.269 for the training set. After training, we constructed the whole complete flow from raw audio, preprocessed audio, e, μ, $\Delta w[5,6,7]$, to reconstructed image. Note that we should also input one image as the first frame, and the learned residuals ($\Delta w[5,6,7]$) are added to the 5^{th} to 7^{th} layer of that encoded first frame. Originally, the reconstructed sequence does not move at all. Then, we magnified the output of μ 500 times, and we can see observable correlation between the audio and the reconstructed image sequence.

Figure 4. Reconstructed (left) and original (right) image using the complete flow.

Fig. 4 shows the reconstructed and original image of four arbitrary frames from three sequences. Generally, as the speech goes, the lip move accordingly. However, we do not see a clear relationship between the pronunciation of a syllable and the shape of the lips. They seem to be simply opening and closing along with the sound. This means the network cannot clearly differentiate between phonations. Also, the head does not move in the reconstructed sequence. This is reasonable and can be inferred from the result in sec. 4.3. Most often, the eye does not blink and stays the same for the whole sequence. However, there are some circumstances where the eye blinks as the lips close. We find that the background flickers a lot as the lips move. This may be caused by the magnified noises. In a nutshell, the audio-to-video prediction is generally in accordance with each other, though subtle differences between the syllable pronunciations cannot be well displayed.

4.5. Video Generation from Different Image/Audio Source

We can even generate talking face videos from different image and audio sources. The results are mostly the same as for using image and audio from the same video. The face *speaks* in another language, accent, tone, and even for different genders. There is also no decaying effect as time goes on in the sequence. All of animated results above can be found in the link here.

5. Conclusion

In conclusion, a speech-to-video model is proposed for generating talking face sequences from raw audio and the first frame. The cropping of raw image sequences, encoding of images to latent code, encoding of latent code residuals to latent representation, audio embedding, and embedding to latent representation prediction are implemented using 5 different models. We find that by using StyleGAN3 as the face generator, it prevents the output from becoming blurry given the noise of the prediction. To the best of the author's knowledge, this is the first implementation of audio-to-video generation using StyleGAN3. We hope that this study can shed some light on relationships between audio and image encoded in the latent space of StyleGAN3 and also between the Wav2Vec2 embeddings and frame sequence, further taking a step towards natural audio-to-video generation with representation learning.

References

[1] Chia-Hung Wan, Shun-Po Chuang, and Hung-Yi Lee. Towards audio to scene image synthesis using generative adversarial network. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 496–500. IEEE, 2019. 1

[2] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017. 1

[3] Neeraj Kumar, Srishti Goel, Ankur Narang, and Mujtaba Hasan. Robust one shot audio to video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 770–771, 2020. 1

[4] Rayhane Mama, Marc S Tyndel, Hashiam Kadhim, Cole Clifford, and Ragavan Thurairatnam. Nwt: Towards natural audio-to-video generation with representation learning. *arXiv preprint arXiv:2106.04283*, 2021. 1

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1, 2

[6] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[7] Yan-Bo Lin, Yu-Jhe Li, and Yu-Chiang Frank Wang. Dual-modality seq2seq network for audio-visual event localization. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2002–2006. IEEE, 2019. 1

[8] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020. 1, 3

[9] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 2

[10] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 3, 4

[11] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018. 4

[12] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *arXiv preprint arXiv:2201.13433*, 2022. 4