Speech-Driven Facial Animation by CNN-RNN-GAN Model

Fengru Li, Xiaoyi Tang, Fei Tao, Ke Xu
Department of Electrical and Computer Engineering
Rutgers University

Abstract

The speech-to-face translation is a class of problems combined speech analysis and face modeling. In this project, we study the problem of transferring a speech sample in one domain to an image sample. Given two related domains but with different dimensions, S and I, our goal is to learn a mapping G that can translate a speech with to a facial response expression.

The neural network we present employs a combination of the CNN, GRU and DCGAN to extract voice features and form a simple expression model and then convert it to a face animation with emotion and expressions with a general face.

In the first stage we generate landmarks of the speaker by CNN and GRU based on audio features and the corresponding video features. In the second stage, faces are synthesized conditioned on the landmarks generated from the first stage.

1. Introduction

In this project, we will predict facial expressions of a person when speaking. This can be regarded as a speech-to-face translation. Facial expressions are an essential part of the nonverbal behaviors.Analyzing the facial expressions can help people understand how they feel and think. Facial expressions are influencing our life. For example, it is important for an interviewee to imagine the facial expressions of the interviewer when attending a phone interview. Another such application is the improvement of talking agent and digital household robots, such as Apple Siri, Amazon Alexa and Google Home. These virtual agents are currently primarily speech driven and inherently limited in the ability to interact with humans considering that human can convey meaning through non-verbal communication patterns including facial expressions and and gestures. For face-to-face interaction, the computer-assisted voice agent can adjust the facial expressions when manipulating the recorded or saved speech. [1]

In this work, we are trying to create a model which will give us a face animation when we input a the audio of a speaker. We exploit landmarks to generate faces because landmarks can describe and preserve the information of facial movement. So the first stage is to translate speech into landmarks. Our CNN and GRU model can find the relationship between landmarks and the audio. The landmarks are the input for the second stage where we employ Deep Convolution Generated Adversarial Networks (DCGAN) [2] to synthesize faces in an attempt to accentuate micro changes of facial expressions. Our goal is not simply mapping landmarks to faces but creating meaningful facial movements.

2. Prior work

Speech-driven facial animation.Various approaches have been developed and improved to generate a face model driven by speech. One latest approach is LSTM-RNN model for real-time facial animation. Voice features including Mel-scaled spectrogram and Mel frequency coefficients are extracted.

Sketch-to-face translation.GAN is the latest and most popular approach for image-style transfer. Then conditional GAN has already been confirmed to work well in pixel to pixel transfer [3].These networks learn a solution to image-to-image translation including reconstructing objects from edge maps and colorizing images. Then sketch-to-face translation is focused on. Conditional GAN model plays an important role because it can learn a mapping from a random noise vector to the output image conditioned on auxiliary information.

Landmarks-to-face synthesis. Facial landmarks contain the most compressed information of face and preserve some information of facial movement. GAN models are improved to reconstruct real faces from landmarks, in hope that more concise information can be displayed. [4]

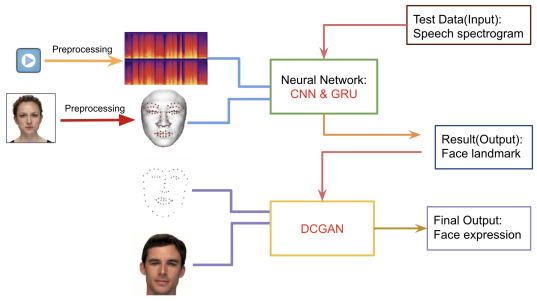

Figure 1. General frame of Design

3. Technical Details

3.1. Feature Representation

For speech to landmarks model, we need to input an audio which can be any speech of any length. The audio is human voice which is not restricted to any particular language. We extract voice features by Fast Fourier Transform (FFT) from the audio signals. Mel-scaled spectrogram are standard acoustic features which are effective to present and preserve contextual information for human voice. [5]

When training CNN and GRU model, we still need to input the corresponding landmarks which are extracted from the corresponding video. For each video, we split it into frames. Then we extract landmarks consisted of 68 coordinates from each frame. In order to reduce the influence that not every face is of same size, all landmarks are normalized to 0 between 1. For each frame, the segment of audio will include additional samples from previous and next frame. Then the input audio feature for each video frame is a 128 by 23 matrix. The feature of every video frame has 136 dimensions.

The second stage is to train GAN model which has a discriminator network and a generator network. For the generator network,the input should be landmarks and the real images, video frames are the input for the discriminator network. To reduce the complexity of training network, we transform frame image size from 1080×720 to 256×256 as input and output size.

3.2. Framework Architecture

The architecture of our model is illustrated in the Figure 1. Our project mainly contains three parts: feature extraction for audio and video, mapping from speech to facial feature and face feature translation to images.

In the first stage of our project, a neural network will be trained which can learn both facial emotional movements and spontaneous facial actions from speech sequence and then translate the integration of facial expressions into 2D images. The neural network we use is a combination of CNN, RNN and GRU. Then in the next stage for test, the existing network will predict the facial expression for our speech input.

In our model, the input of CNN network is raw time-frequency spectrograms of audio signals, which is in the form of 2D array. In CNN network, we apply convolutions on frequency and time separately. This helps a lot to reduce overfitting and speed up the computation process. Pooling layers are also applied for downsampling. For each neural network, there is a dense layer before the output. The dense layer after RNN can reduce over smoothing tendency of RNN. [5, 1]

Then, after CNN and RNN, GRU is followed to generate the image. The inputs of the DCGAN model are series of landmarks which are outputs of the CNN and RNN network. And the model will synthesize a face animation. GAN is now primarily applied in modeling images. The core of DCGAN is to use a CNN architecture.

3.3. CNN + RNN Architecture and Training Details

The CNN and RNN network mainly consists of 8 convolution layers and before the output there is a GRU network. The details can be shown in the table.

Table 1. CNN and GRU Details

Name	Filter Size	Stride	Hidden Layer Size	Activation
Input			128∗23	
Conv1	[3,1]	[2,1]	64∗23∗32	ReLu
Pool1	[2,1]	[2,1]	32∗23∗32	
Conv2	[3,1]	[2,1]	16∗23∗32	ReLu
Pool2	[2,1]	[2,1]	8∗23∗32	
Conv3	[2,1]	[2,1]	4∗23∗64	ReLu
Conv4	[3,1]	[2,1]	2∗23∗64	ReLu
Conv5	[3,1]	[2,1]	1∗23∗128	ReLu
Pool3	[1,2]	[1,2]	1∗11∗128	
Conv6	[1,3]	[1,2]	1∗6∗128	ReLu
Conv7	[1,3]	[1,2]	1∗3∗136	ReLu
Conv8	[1,4]	[1,4]	1∗1∗136	ReLu
Dense1			136	Tanh
RNN			136	
Dense2			136	Tanh
Output			136	Sigmoid

The input of this framework is sequences of spectrograms $S_i, i = 1, 2, ..., T$ and the output is sequences of landmarks L_i. For the output is a vector with 168 entries, we used MSE (Mean Squared Error) to be the loss function of CNN

and RNN. The loss function is shown as following.

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}[Y_i - \hat{Y}_i]^2. \quad (1)$$

We train the model by minimizing MSE. We use tensorflow to implement our neural network. We choose training parameters as follows: minibatch size is 300, epoch size is 100 and learning rate is 0.001. The network parameters are learned by Adam optimizer in 100 epochs.

3.4. DCGAN Architecture and Training Details

Inspired by the success of DCGAN, we develop an adversarial network for synthesizing facial animation from landmark information extracted from audio. DCGAN, similarly to other GANs, consists two competing networks: generator G and discriminator D. The goal of GAN is to train G to spawn samples, and simultaneously to train D to distinguish produced samples from real inputs. We focus on conditional GAN, a variant where G and D is conditioned on additional variables, because we need landmark information as the conditions to reproduce facial animation. The core of the DCGAN architecture is to use a CNN architecture.For the generator, convolutions are replaced with upconvolutions.DCGAN replaces pooling layers with strided convolutions in discriminator networks and fractional-strided convolutions in generator networks.Different from CNN, we need to remove fully connected hidden layers. And we use LeakyReLU with slope 0.2 and Batch normal as regularization on all convolution layers. We also use tanh operation on the output layer to smoothly regularize the result on [0,1] space. The conditioner is attached after the output of first convolution and pooling layer in discriminator and before the input of last unpooling and deconvolution layer in generator,as a modification of traditional CGAN, for better converging performance. The details are shown in tables.

Table 2. Discriminator Networks Details

Name	Filter Size	Stride	Hidden Layer Size	Activation
Input			256∗256∗3	
Conditioner			136	
Conv1	[5,5]	[2,2]	128∗128∗64	LReLu+BN
Pool1	[2,2]	[2,2]	64∗64∗64	
Comb1			64∗64∗200	
Conv2	[5,5]	[2,2]	32∗32∗128	LReLu+BN
Pool2	[2,2]	[2,2]	16∗16∗128	
Conv3	[5,5]	[2,2]	8∗8∗256	LReLu+BN
Conv4	[5,5]	[2,2]	4∗4∗1024	LReLu+BN
Projection			100	
Output			100	Tanh

Table 3. Generator Networks Details

Name	Filter Size	Stride	Hidden Layer Size	Activation
Input			100	
Conditioner			136	
Projection			4∗4∗1024	
DeConv1	[5,5]	[2,2]	8∗8∗256	ReLu+BN
DeConv2	[5,5]	[2,2]	16∗16∗128	ReLu+BN
Unpool1	[2,2]	[2,2]	32∗32∗128	
DeConv3	[5,5]	[2,2]	64∗64∗64	ReLu+BN
Comb1			64∗64∗200	
Unpool2	[2,2]	[2,2]	128∗128∗200	
DeConv4	[5,5]	[2,2]	256∗256∗3	ReLu+BN
Output			256∗256∗3	Tanh

The loss function of our DCGAN model divides into two ones corresponding to the step of optimizing generator and the step of optimizing discriminator. During training phase, DCGAN keeps minimizing loss functions in an intersecting way. The functions are as follows.

$$\begin{aligned} Loss(D) =& E[logD(x,y)]+ \\ & E[log(1-D(G(z,y),y))]. \end{aligned} \quad (2)$$

$$Loss(G) = E[logD(G(z,y),y))]. \quad (3)$$

where x is the real input, y is the landmark information, both x and y are sampled from distribution $Pdata(x,y)$. z is generated by randomly sampling from uniform distribution.

4. Evaluation

4.1. Datasets

We use the RAVDESS [6] datasets to train and test our model. It consists of 24 different actors singing or speaking records with different emotions, which will be good resources for emotion recognition of speeches and generation of facial expressions. In our model, we used both video and audio files from the datasets. Every audio is corresponding to a video. Each audio lasts for 3 to 4 minutes and every video can be split to around 100 frames.

4.2. Evaluation Strategy

We evaluate our model separately in two parts. For the part of CNN + RNN, we directly use the loss function to estimate the accuracy. It measures mean squared difference between the real facial landmark and the predicted facial landmark.

For the part of DCGAN, We use mean squared error(MSE) estimator to evaluate the accuracy of our output samples from our DCGAN model. The mean squared error from generated sample image $\hat{X}$ to real image X can be ex-

pressed as

$$MSE = \frac{1}{D_X D_Y} \sum_{i \in Dim_x} \sum_{j \in Dim_y} (X_{i,j} - \hat{X}_{i,j})^2. \quad (4)$$

where D_X and D_Y represent dimensions of x-axis and y-axis of images.

5. Experiment

5.1. Results

The corresponding outputs of the inputs in first stage are a series of landmarks. The comparison of prediction and original landmarks is shown in the figure 2. The landmarks can be reconstructed to real faces from second stage. The results of DCGAN are shown in the figure3.

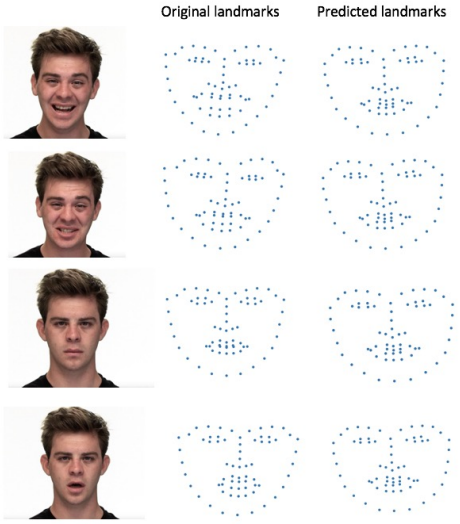

Figure 2. Audio to facial landmark training result. 1st column are test inputs; 2nd column are real landmarks; 3rd column are predicted landmarks.

Table 4. Error between generated landmarks and original landmarks (for CNN+RNN), the smaller the better. We compute the average MSE between 25 audio files and corresponding video files with given training size. The batch size is 300.

Train Size (# of batches)	100	200	400	800
MSE	0.0007	0.0006	0.0007	0.0006

To evaluate the accuracy of our DCGAN model, we use MSE to compare DCGAN outputs with real images. To constrain other unrelated variables, we use facial images from same person during training and testing. We extract

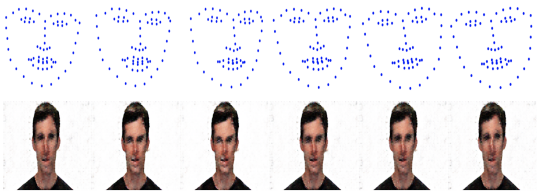

Figure 3. Facial landmark to facial image result. 1st row are input landmarks; 2nd row are predicted faces.

Table 5. Error between generated facial images and real images (for GAN), the smaller the better. We compute the average MSE between 320 generated facial images and their real images with given training size. The batch size is 32.

Train Size (# of batches)	200	400	800	1000
MSE	1.210E-1	1.613E-1	1.082E-1	1.083E-1

1095 frames of facial images and corresponding facial landmarks from 60 video clips, left with 775 images in the training set and 320 images in the testing set. The network is evaluated with different training sizes and the result shows that MSE stops to decrease when training size increases to certain number, which is around 800.

From the experiment, we find DCGAN architecture be capable of generating facial expressions corresponding to landmarks. Most obvious synchronization behaves on movements of head. With certain amount of training, the generator can simulate mouth and eyes movements in subtle way. Not all predictions are correct and the accuracy are heavily affected by distribution of training set. Intense expressions are usually weakened even distinguished, because of the deficients of GAN models.

5.2. Discussions

Our model successfully predicts face expressions from audio sequences. When we input an speech sequence, we can get face animation and landmarks as secondary outputs. Therefore, we can use landmarks for other research.

On one hand, from the perspective of statistical data, the performance of CNN+RNN network is perfect. The error comes to very small value and converge quickly, and the change of facial expression between continuous outputs is obvious. On the other hand, when we see the landmark output from CNN , we may find it still have obvious difference with the original one. This is resulted from several reasons. After training process and normalization, for landmarks are represented by 68 points, they can be more and more similar between each other. Therefore, the face shape can be average for each person in landmarks.

We can see that the results of DCGAN are not very perfect. The reason could be that we train two models separately in order to save time. However, this leads to the ob-

vious flaw that two loss functions may fail to get minimum results at the same time. The future work we need to do is to train our two models simultaneously.

Besides, another commonly failure case for GANs, mode collapse problem, also affects accuracy of our model. Mode collapse happens when the generator reproduces similar outputs which means it loses multimodal property as real-world data. Unfortunately, mode collapse can be triggered in a seemingly random fashion and cannot be completely prevented in nowaday's GAN architectures. Albeit we reduce similar training images and use a random shuffling strategy to prevent highly similar batches, mode collapse problems are still existed when number of training epochs grows.

There are some advantages to use GAN. GAN has been proven to perform better in image-style transfer.Therefore, it is easy to transplant our model. Slight change of the loss function can change the objectives and apply the secondary outputs into other models.

Specifically, for our whole model, we have not used data sets big enough. In our future work, if we use more data for training, we may get better prediction results.

6. Contributions

Generally speaking, our contributions can be divided equally. During the process of whole project, we discussed with each other about all questions all the time. Almost all functions are worked together by all members, so it is hard to specify.
For the code implementation, we divided our main work into two parts: CNN+RNN and GAN and divided all members into two groups (each group has two members). The split reference is as follows:
Feature extraction: Fei Tao and Fengru Li
CNN+RNN: Fengru Li and Xiaoyi Tang
GAN: Ke Xu and Fei Tao
When preparing for the presentation and report, all our members worked together all the time.

References

[1] H. X. Pham, S. Cheung, and V. Pavlovic. Speech-driven 3d facial animation with implicit emotional awareness: A deep learning approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2328–2336, July 2017. 1, 2

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 1

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 1

[4] Xing Di, Vishwanath A. Sindagi, and Vishal M. Patel. GP-GAN: gender preserving GAN for synthesizing faces from landmarks. *CoRR*, abs/1710.00962, 2017. 1

[5] Hai X. Pham, Yuting Wang, and Vladimir Pavlovic. End-to-end learning for 3d facial animation from raw waveforms of speech. 10 2017. 2

[6] Steven R Livingstone, Katlyn Peck, and Frank A Russo. Ravdess: The ryerson audio-visual database of emotional speech and song. In *Annual Meeting of the Canadian Society for Brain, Behaviour and Cognitive Science*, 2012. 3