Audio-visual generative adversarial network (AVGAN)

Dillig Marlene (MIM), Felger Max (DIM),
Père Valentin (IMT Mines Albi), Weiß Markus (MIM)*

Advisor: Prof. Dr. Ruxandra Lasowski, Prof. Dr. Norbert Schnell

July 23, 2019

Abstract

During the master research project at the Furtwangen University we worked on creating a special generative adversarial network (GAN) for audio-visual forms. For that basics of seperate networks which generates audio or images were used. Our own Audio-visual Generative Adversarial Network (AVGAN) is based on two existing networks - the DCGAN [1] and the architecture of the "Lip Movements Generation at a Glance" (LMGAN) by Lele Chen et. al. [2]. The work is still in progress so we have little results so far and expect the completion and more results of AVGAN in the end of august this year.

INTRODUCTION

We asked ourselves how to combine audio and image generation in only one single network. For that we set ourselves two tasks: The one task of creating a new database with different funny faces with a suitable sound to that. The other task is developing an audio-visual GAN in which audio + image is combined and trained to get a new funny face with a suitable sound as an output. There are already good working GANs for image generation such like deep convolutional network (DCN) which is working with the CelebFaces dataset [5]. This one creates new, not existing faces looking like real persons. But there are also networks which are trying to generate sounds and human voices like the WaveGan/SpecGan and GANSynth by Chris Donahue et. al. [3] [4]. For our work we focused on three existing networks - DCGAN, LMGAN and WaveGan/GANSynth - which will be explained in the next chapter and are the basics of the architecture of AVGAN.

RELATED WORK

For AVGAN we used some features of DCGAN, LMGAN and WaveGan/GANSynth. The basics of the discriminator and generator, the loss and optimization and also the training loop we took from the DCGAN [1]. The idea of the concatenation of audio and image we got from the LMGAN. Two streams - sound and image - are going separately through convolution, are getting concatenated and after the concatenation both streams are going together again through convolution [2]. We were working a long time on the WavGan and analysed the procedure of how sound is going into the network and what preprocessing like PhaseShuffle [3] is done to get qualitative better sound results after the training. There are some different ways how to feed in sound-files into a network and we decided to do it like GANSynth, where we had to transform the .wav files into two pictures [4]. One shows the log magnitude, the other displays the phase (figure 1).

[*] Code: https://github.com/markus-weiss/AVGAN

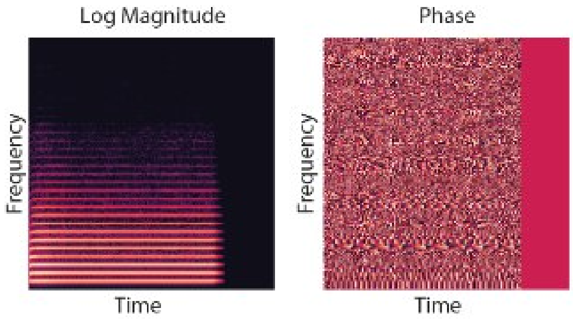

Figure 1: Log Magnitude and Phase Image of a Sound Data from GANSynth [4]

1 STRATEGY

1.1 DATAcollection

DATAcollection Pictures of the website ? lottery?

1.2 Experiments

1.2.1 WaveGan

During the research project we started with some experiments on different codes to get more information for our own architecture. While working with WaveGan we found out, that the code doesn't work with raw sound data and that we have to transform them into other file formats. For the training we tried to use existing checkpoints in order to not train from scratch, but somehow the training failed. Another thing was, that our time for the project was limited and after calculating the time the network needs only for preprocessing with the PhaseShuffle, we decided to stop working on WaveGan and we switched to GANSynth.

1.2.2 GANsynth

We tried to train the GANSynth network, but it didn't work, because of wrong file formats. So we learned more about the file transformation by transforming the raw sound wave into two images like it is shown in figure 1. The relevant data is displayed on one hand in the image of the phase and on the other hand in the image of the log magnitude.

1.2.3 DCGAN

We came back to DCGAN and used our own collected data as the dataset input and got our first generated results. In the first time we trained with very less images:

- 8 images
- incl. 2 very similar images
- 5000 epochs

The generated result (figure 2) was very similar to the two very similar input images (figure 3).

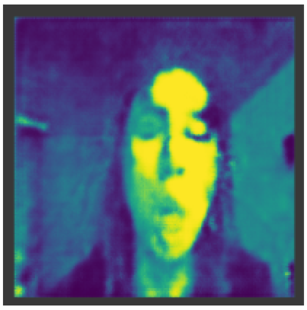

Figure 2: Result after training the DCGAN with 8 images

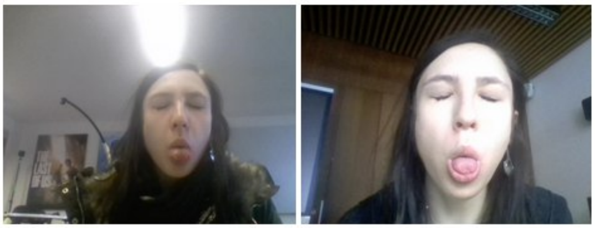

Figure 3: Two very similar images from the training datasets

We set up the thesis that the network concentrates on the most similar images and tries to generate that. For the next experiment in order to test the thesis we trained as followed:

- 7 images
- incl. 2 other very similar images
- 5000 epochs

For epoch 1454 we got figure 4 as interim result and saw features from two images from the dataset (figure 5). The result after 2356 epochs (figure 6) was not one of the 2 very similar images from the datasets. So the thesis could not be confirmed.

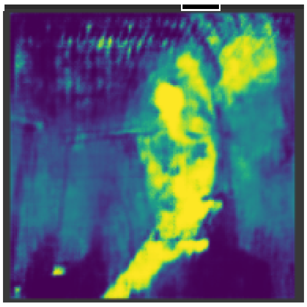

Figure 4: Generated Result after 1454 epochs

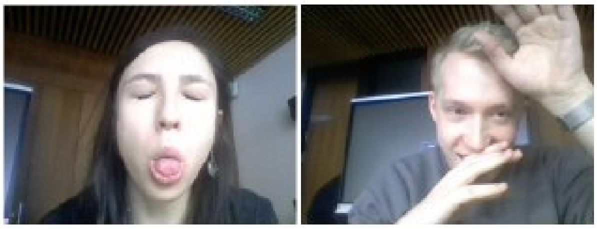

Figure 5: Two images out of the training dataset

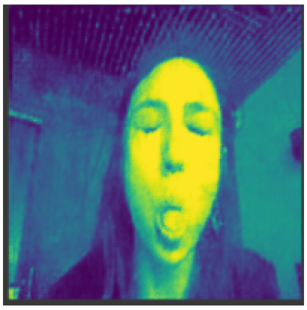

Figure 6: Generated result after 2536 epochs

Another experiment was to train with our complete dataset of 65 images. After 2000 epochs (figure 7) we can recognize the first outlines from a face on our training result. After 3000 epochs (figure 8) we can see similarity between the result and a picture from the training set. On the left from the middle we see a much clearer picture than in figure 7. We also see parts from another picture like an arm. Approximately on 4000 training steps (figure 9) on the left side of the image we can see a really clear face. On the top from the face now we see some fingers. But it's hard to say where the origin lies. When we trained more than 4000 epochs the image quality gets worse again (figure 10). In figure 11 and 12 there are the suspected origins of the generated image.

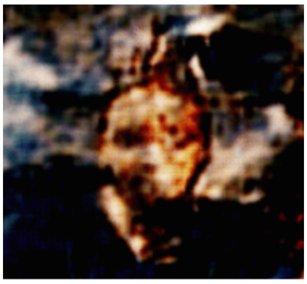

Figure 7: Generated result after 2000 epochs

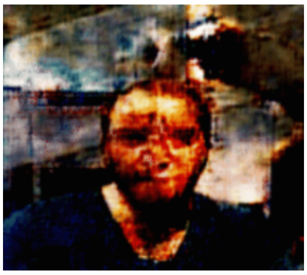

Figure 8: Generated result after 3000 epochs

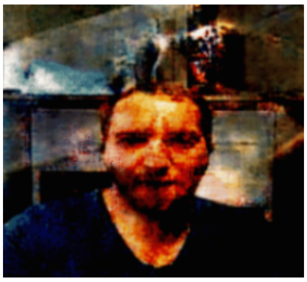

Figure 9: Generated result after 4000 epochs

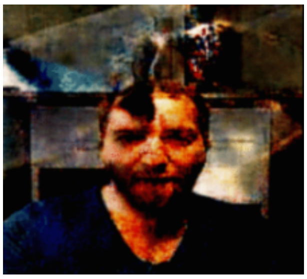

Figure 10: Generated result with more than 4000 epochs

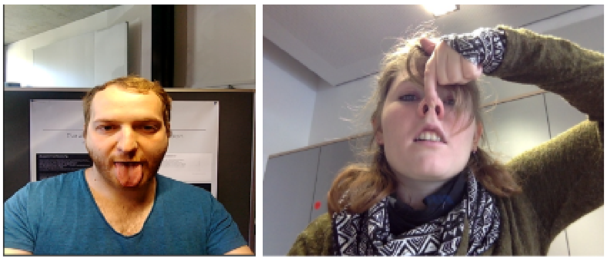

Figure 11: Origin of the generated results

2 AVGAN Architecture

The final AVGAN-architecture is shown in figure 12. To combine the sound and image input streams the concatenation was used, after the two transformed sound images were concatenated, as well.

After the sound and image concatenation the GAN network starts the training while the generator tries to generate better images and the discriminator judges the output. After the training the network has to do the reverse way to display deconvoluted images. The last step is de deconcatenation of the sound images and the retransformation into a .wav file, so sound can be played.

2.1 Results

First results after training the AVGAN displayed only black and gray noise (figure 13). During the training only little changes within the noise can be recognized, but no faces or sound waves could be achieved. So the output was very similar to the first training steps of the network.

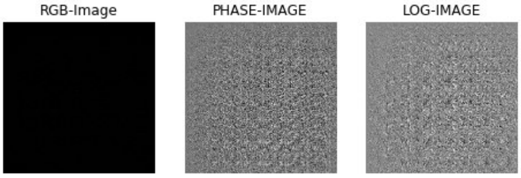

Figure 13: First results after training the AVGAN

CONCLUSIONS

AVGAN is related to existing networks, which had to be analysed in detail and with that some experiments had to be done in the beginning. The relevant parts, such as file transformation and the GAN architecture, were selected. In the end the architecture of AVGAN is part of a mapping of DCGAN, GANSynth and LMGAN with own ideas and extensions in order to get the result we set in our research assignment. AVGAN is still in progress, so there are currently only a few results and no evaluation. Probably there has to be some fixes in the convolutions and in the training, to get a person-like image output and a human sound. The data collection runs continuously. A complete evaluation can be done when the network is finished.

References

[1] S. Chintala A. Radford, L. Metz. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015 (last revised 7 Jan 2016 (v2).

[2] Lele Chen, Zhiheng Li, Ross K. Maddox, Zhiyao Duan, and Chenliang Xu. Lip movements generation at a glance. *CoRR*, abs/1803.10404, 2018.

[3] Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018.

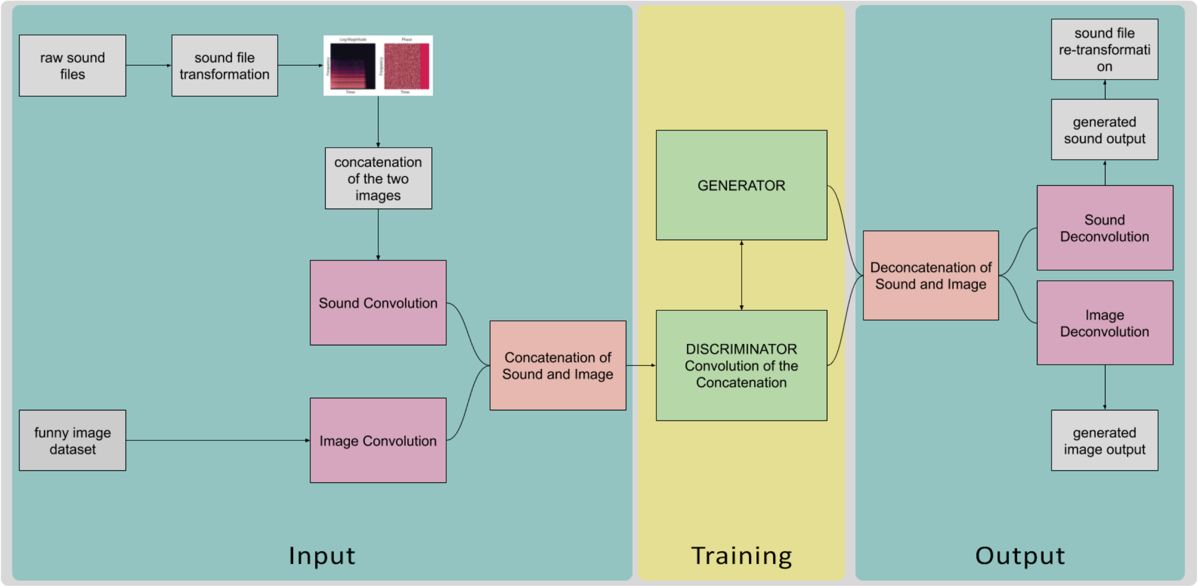
raw sound files
sound file transformation
concatenation of the two images
Sound Convolution
funny image dataset
Image Convolution
Concatenation of Sound and Image
Input
GENERATOR
DISCRIMINATOR Convolution of the Concatenation
Training
sound file re-transformati on
generated sound output
Sound Deconvolution
Deconcatenation of Sound and Image
Image Deconvolution
generated image output
Output

Figure 12: Architeture of AVGAN

[4] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *CoRR*, abs/1902.08710, 2019.

[5] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. *CoRR*, abs/1509.06451, 2015.