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

September 11, 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.

*Code: https://github.com/markus-weiss/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).

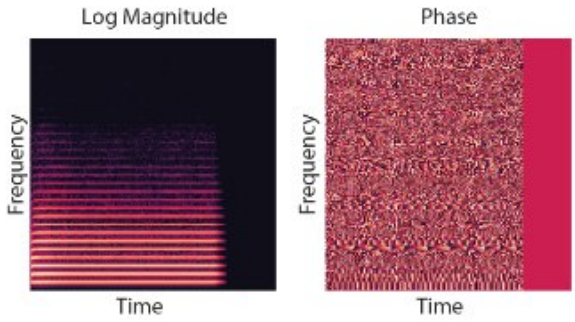

Figure 1: Log Magnitude and Phase Image of a Sound Data from GANSynth [4]

1 STRATEGY

The strategy of the research project in order to achieve the tasks we set was on one hand to create the website for the traing dataset. On the other hand we had to do some experiments with the codes of the related work, to get more information for our own architecture for AVGAN.

1.1 Data collection

Figure 2: Logo of the webpage

For the data collection the website "Make a Face for Science"[1] was developed, where users can record a funny image and a sound. The webpage basically has three main areas: the header with the logo and a short text for explanation, an area with five faces being shown that can also be rated and lastly, the area of the website where the user can take their own faces and sound. The first part of the site contains the logo, as shown in figure 2, and a brief explanation of the project and functions which the site offers to users. Furthermore, there is an area which shows five faces, as well as the respective sounds as hidden html audio tags (figure 3).

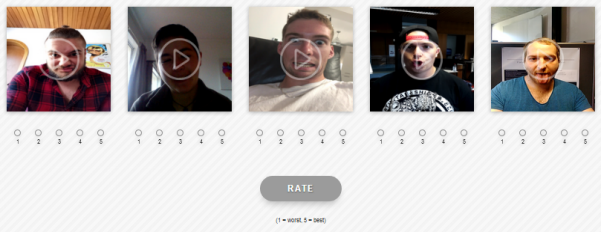

Figure 3: Rating section

For each face there are five buttons for the user to rate the face and a single rate button to save every rating into the database. Last but not least, there is the section of the website that allows users to record their own faces (figure 4). First and foremost, the video stream of the webcam is shown and there is also a silhouette positioned over the video to indicate the optimal position of the user. In addition, the privacy statement of the website is shown here and by clicking on the record button, the countdown (3, 2, 1, Go) can be started. After the both the image and sound of the face has been recorded, the Save button appears instead of the Record button and a note about the competition on the website is shown, in which the user with the best face wins and the user can optionally enter his e-mail address. The user can play the sound of the face with a click on the screenshot of the face now shown where the previously was the video stream and again there are is a little text to explain the functionality of the buttons, so that the user also understands that he must first save the face before he can record new ones.

[1] https://makeafacefor.science/

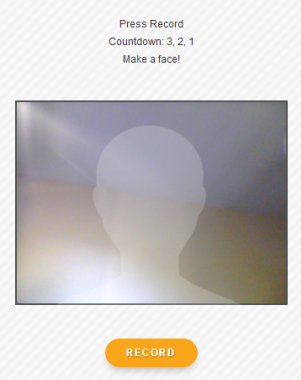

Figure 4: Recording section

The first two collections of data were set to the Day of Media 02/2019 and 07/2019 of the Furtwangen University. After those two days we had around 65 images with sound. After the launching we spread the website across the Furtwangen University and IMT Mines Albi, so the data collection runs continuously to get more and more material for our training dataset.

1.1.1 Formats of the image and audio files

The image and sound files recorded on the website should meet the requirements of our GAN training, so that we can continue to work successfully with the files we have collected. Originally, the image file had a width of 800 pixels and a height of 600 pixels, but the creation of the image file only worked on desktop devices, but not when using a mobile device like a smartphone. After recording the image file, a screenshot of the video stream is saved in a html canvas element, which is then converted to a blob via JavaScript. The smartphone obviously has different dimensions in width and height compared to a desktop device and in order to provide a uniform file format for the GAN training, the dimensions of the image file have been set to 320x320 pixels. In order to prevent the image section saved by us from showing too much of the user's background, the screenshot in the html canvas element is automatically adjusted so that a central view of the user can be achieved as effectively as possible. By using the Web Audio API, the sound files can be recorded for one second and also exported and saved into the database as correctly configured wav files.

1.2 Experiments

Experminets were made with WaveGan, GANSynth and DGAN. From this we received various results, which were used in our further work.

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 5) was very similar to the two very similar input images (figure 6).

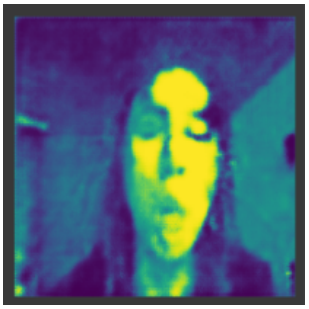

Figure 5: Result after training the DCGAN with 8 images

Figure 6: 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 7 as interim result and saw features from two images from the dataset (figure 8). The result after 2356 epochs (figure 9) was not one of the 2 very similar images from the datasets. So the thesis could not be confirmed.

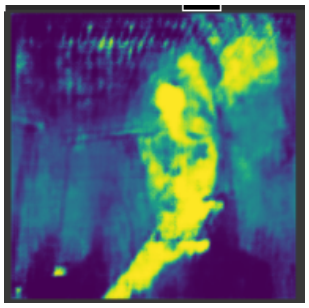

Figure 7: Generated Result after 1454 epochs

Figure 8: Two images out of the training dataset

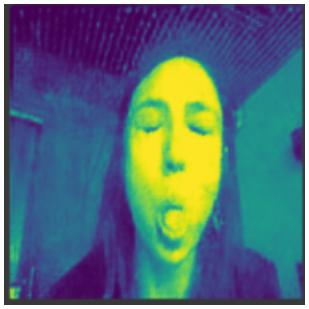

Figure 9: Generated result after 2536 epochs

Another experiment was to train with our complete dataset of 65 images. After 2000 epochs (figure 10) we can recognize the first outlines from a face on our training result. After 3000 epochs (figure 11) 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 10. We also see parts from another picture like an arm. Approximately on 4000 training steps (figure 12) 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 13). In figure 14 and 15 there are the suspected origins of the generated image.

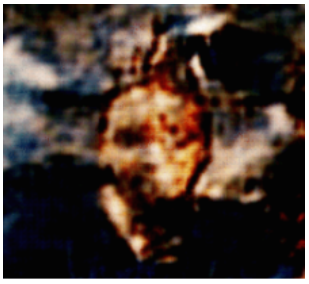

Figure 10: Generated result after 2000 epochs

Figure 11: Generated result after 3000 epochs

Figure 12: Generated result after 4000 epochs

Figure 13: Generated result with more than 4000 epochs

Figure 14: Origin of the generated results

2 AVGAN Architecture

The final AVGAN-architecture is shown in figure 15. At first the raw sound files have to be transformed into two picters and have to be concatenated. Thereafter, the sound images and the funny images go through a convolution separately and are then concatenated. The final discriminator does the convolution with the concatenation of sound and image. The generator does exactly the same way in reverse way, so it starts from a random noise vector till it has the right format to do the deconcatenation of sound and image, afterwards the deconvolution of sound and image separately follows. While training the output of the generator goes back to the discriminator. After the training the output is a generated image and a raw sound file.

2.1 Results

First results after training the AVGAN displayed only black and gray noise (figure 16). During the training only little changes within the noise can

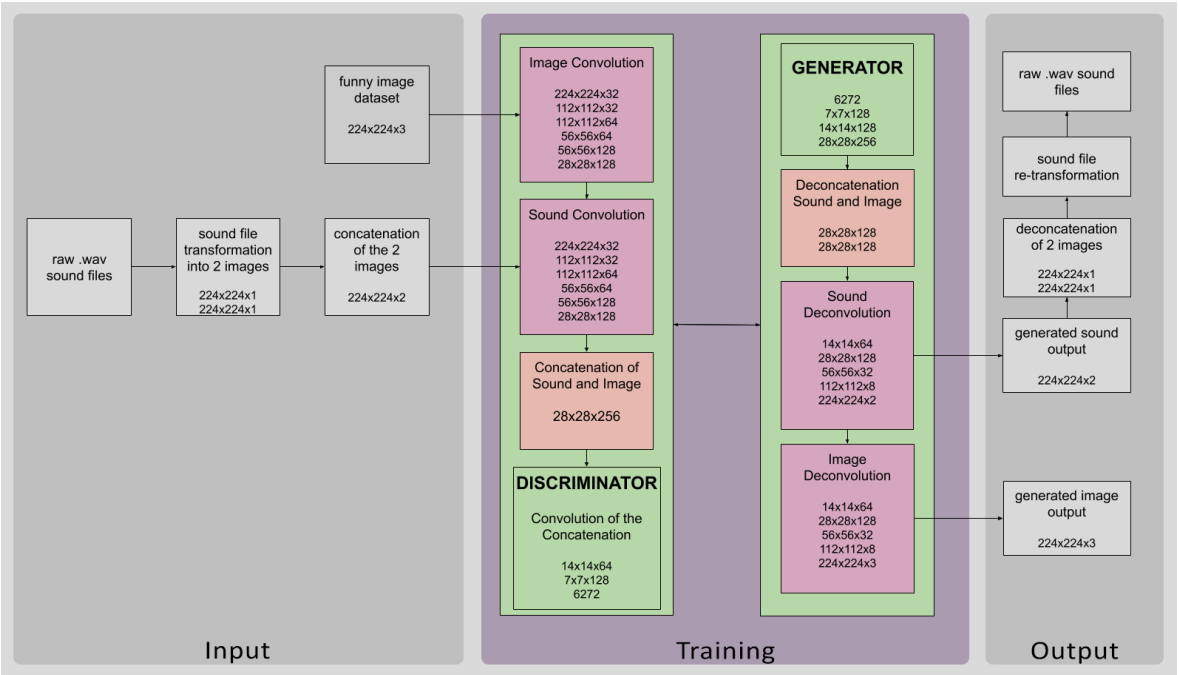

Figure 15: Architeture of AVGAN

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.

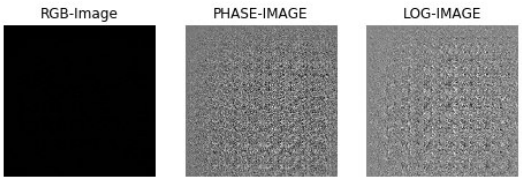

Figure 16: First results after training the AVGAN

To this moment many experiments with the code were made in order to get only more than black and gray noise. In one case we feeded in twice the image-dataset, because we are still not sure if the concatenation and all the convolutions and deconvolutions are working in the right way as well with the sound-images. We thought if we have twice an image of a person, the network could nothing other do, than generate a human-like silhouette. But the result was disillusioning and again we only recieved a random image with black and RGB noise as our output (figure 17), which changed during the training, but did not reach our desired goal.

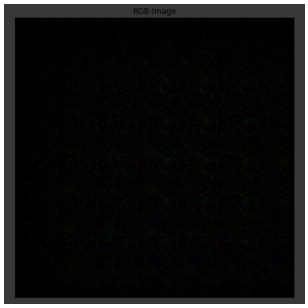

Figure 17: First results after training the AVGAN

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.

[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.