Woven Artificial Profile (WARP):

Face Video Synthesis from Profile and Audio

A BTP Report

by

Group B21SK05

Nethra Gunti: (S20180010061)

Tarun Teja Obbina: (S20180010120)

Ramancha Nithin: (S20170010120)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

09th December, 2021

Final Report

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the BTP entitled "**Woven Artificial Profile (WARP): Face Video Synthesis from Profile and Audio**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of our own work carried out during the time period from January 2021 to December 2021 under the supervision of Prof. Subu Kandaswamy, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by us for the award of any other degree of this or any other institute.

Nethra.
07/12/2021

O. Taruntej a
06/12/2021

6 December 2020

(Nethra Gunti) **(Tarun Teja Obbina)** **(Ramancha Nithin)**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Prof. Subu Kandaswamy

ABSTRACT

In recent times, Direct Communication between people has become extremely important and rare. Though indirect modes of communication are as convenient as they are, they lack the essence that direct means hold. Taking this as a motivation, we propose a pipeline titled *WARP*, short for *Woven Artificial Profile*. WARP can generate realistic talking videos of a person, give just an image and text as input. Users also have the option to provide a custom face image for the video. Entertainment industries, Educational institutes, Marketing sector, and even the general audience could use WARP to make textual data like messages, emails, and lessons, more realistic and interactive. Even Virtual Assistants can now have a face!

Contents

1. INTRODUCTION

In today's world, communication between people has majorly become indirect, that is via messages and emails. It has become extremely important for people to use direct communication. Indirect means of communication come with great ease. And regardless of how convenient they are, they lack the essence of that direct communication holds. Although, Audio and Video calls are good alternatives, they demands the physical presence of all parties at the same time and other external factors like network, good camera, microphone, speaker etc are all very needed. But with our approach, none of that is needed. Visual talking would be as simple as sending a text message.

Problem Statement: Given a face image and text, WARP is a highly versatile model that can generate realistic talking videos with audio and lip synchronization.

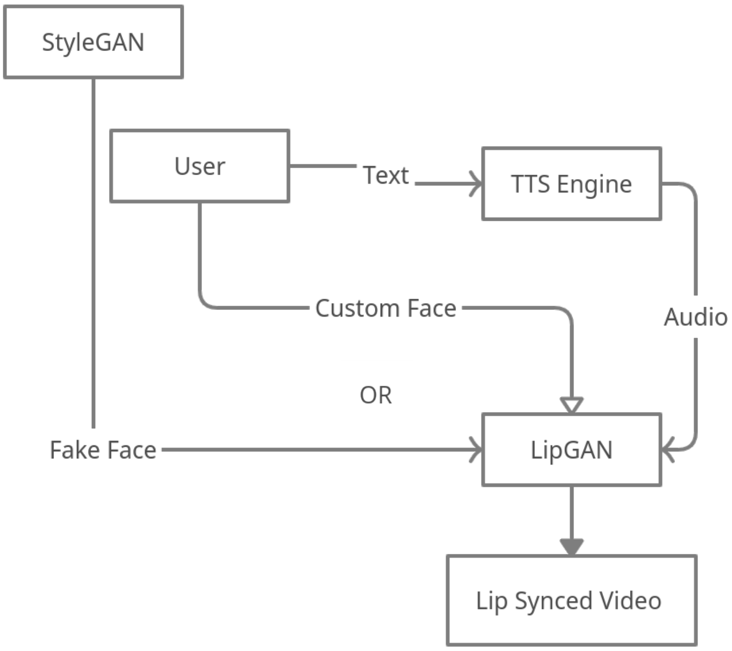


Figure 1: General Use Case

The demographic for this application is pretty wide. There are several commercial and non-commercial uses. Entertainment and Broadcasting industries can, in a way, be revolutionized. Education can be made more interesting by bringing those history and science lessons to life. Marketing can be made more direct. Avatars and Virtual Assistants. like Siri and Alexa can now have a face. It also provides the opportunity for remote officials and general public to communicate across the internet with a face. The pipeline has 4 modules:

- Fake-face generation using StyleGAN
- Speech generation using standard Text-to-Speech engines
- Lip synchronized frame generation using LipGAN
- Video generation using OpenCV

The rest of the report is organised as follows- In section 2, we discuss related works. In section 3 we discuss our approach in detail, followed by the results in section 4. In section 5 we derive at a conclusion for this report and also state the future scope.

2. LITERATURE SURVEY

In this section, several papers are discussed that are related to at least one module of the project, if not the entire project. First to be mentioned is *Neural Voice Puppetry [5].* It is an audio driven network that takes an audio as input and uses a intermediate 3D face model that represents the target person speaking and checks the level of synchronization.

Temporal GAN [6] is a generative model that can learn a semantic representation of unlabeled videos, and is capable of generating videos. Its generator has 2 parts- a temporal generator and an image generator.The temporal generator takes a single latent variable and outputs a set of latent variables, each corresponding to a frame in the video. The image generator then uses such latent variables to create the video that has the number of frames equal to the number of latent variables.

DualLip [7] is a a system that jointly improves lip reading and generation by leveraging the task duality and using unlabeled text and lip video data. Its key ideas include lip generation model to sync with the unlabeled input text and text generation model from unlabeled lip video. The lip generation model is further extended to talking face generation.

3. METHODOLOGY

In this section, we will discuss the approach used in detail. Section 3.1 will include the methodology used for fake face generation. Section 3.2 will include methodology used for video generation.

3.1 Fake Faces Generation

StyleGAN[2] is a style based generative adversarial network which is used to generate fake faces. This architecture has the ability to automatically learn, separate high-level attributes- like pose and identity- in an unsupervised manner, and create stochastic variations in the generated images- like hair, freckles etc.

Dataset: The dataset used for this purpose is CelebA-HQ[3]. It has a total of 30k face images with 28k for training and 2k for validation.

Architecture: Generator consists of a Mapping Network and a Synthesis Network. Mapping network is a multi-layer perceptron of 8 linear layers with LeakyReLU with a slope of negative 0.2 as activation function. Mapping Network takes a noise vector Z as input and outputs an intermediate vector W. Synthesis Network is divided into four parts, Gen block with 2 convolutional layers, AdaIN (two layers with LeakyReLU and Upsampling), Noise (two layers) and Constant Input. Figure2 shows the architecture of the GAN.

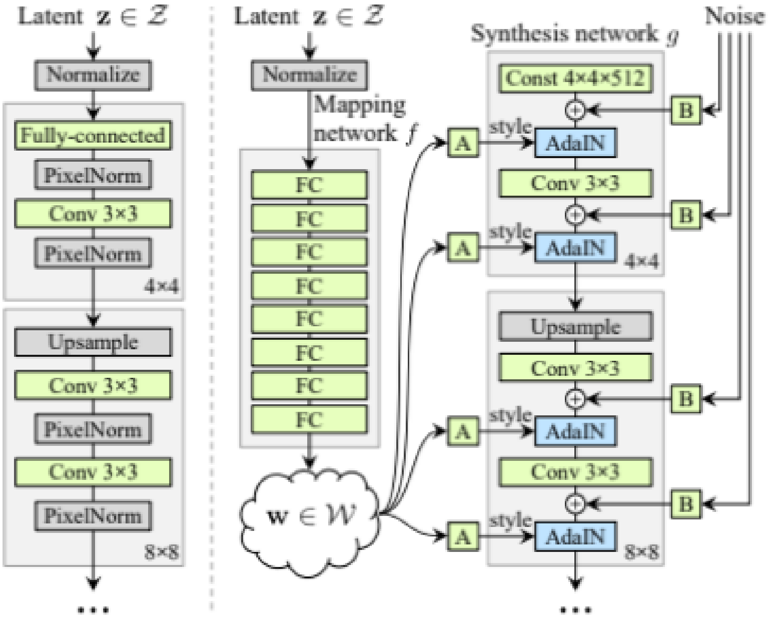


Figure 2: StyleGAN Architecture
Ref: [2]

AdaIN (Adaptive Instance Normalization) transfers the style from intermediate noise vector- W, onto the general image. Instance normalization is essentially normalizing each instance. While the adaptive part in AdaIN is able to ally the different styles from W onto the image or the intermediate feature map. It is done by passing W to a linear layer which gives a style vector that is twice the size of input. This output has two parts- scale and bias, which are used to normalize the image.

Noise layer contains a learnable weight parameter, each of which is applied on each channel of input image. Constant Input is where image generation begins. For every image the input is a constant vector with all the values as 1. Altogether, the Synthesis Network contains a total of 9 Gen blocks with the output dimension increasing from 4x4 to 1024x1024 and 9 RGB layers which are used to convert the output to RGB scale.

Discriminator is a single network which contains 9 Disc block layers and 9 RGB layers which are similar to RGB layers in Generator architecture. Each Disc block consists of a downsample layer, two convolutional layers with LeakyReLU of negative slope 0.2 as activation function.

Training: The training has been done by increasing the output size gradually from 4x4. When the output size is increased from 4x4 to 8x8, it initially add less weight to the upsampled image with convolutional layer than the upsampled image without convolutional layer. This weight will be decreased for single upsample layer and increased for upsample layer with convolutional layer. This weight is controlled by the hyperparameter alpha. We used the Wasserstein GAN gradient penalty as the objective function for this GAN as suggested in [2] for the CelebA-HQ dataset[3]. Currently 112 epochs of training is done with the output size being 128x128.

3.2 Video Generation

LipGAN is a GAN architecture that was developed as part of the paper [1] with the main objective of automatic video-to-video translation. This input of this architecture is MFCC features and a pair of source-destination images. That is, given a source image A, destination image B, and an audio, a video starting from A and ending at B is generated with lip synchronization to the audio.

Dataset: The dataset we're using is LRS2[4] which contains $\sim$46,000 short videos of people talking. The length of the videos range anywhere from 1 second to 10 seconds. These videos also contain their corresponding annotations of the text that is being spoken.

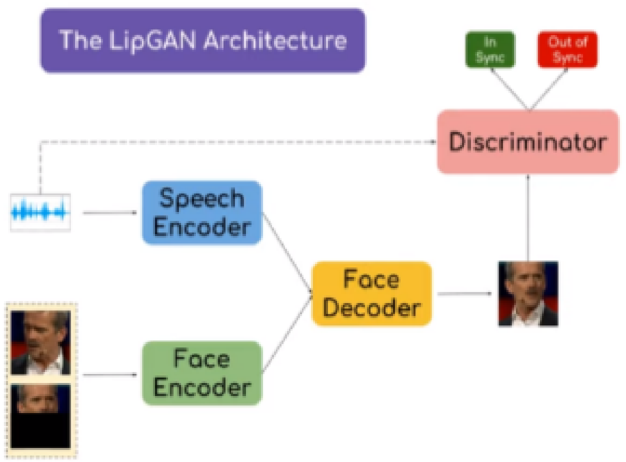


Figure 3: LipGAN Architecture Overview
Ref: [1]

Preprocessing: The original dataset is just a set of annotated videos. This however needs to be preprocessed to be used for training the model. In the preprocessing stage, we extract the following from the videos:

1. Audio: We extract the audio using *ffmpeg* and save it as a .wav file

2. Mel Spectrogram: We generate the Mel Spectrogram with the help of *Librosa* and save it as compressed numpy array (.npz). Figure.7 visualises the Mel Spectrogram for a sample video from the dataset.

3. Valid Frames: We extract those frames from the video which have a valid frontal face view. A frontal face view is not considered valid if it is oddly positioned. We do this using *DLIB*'s frontal face detector. For each valid frame, we generate a set of possible views and the view maximizing the bounding box area is finally chosen and saved as an image. Although DLIB is completely capable of extracting such views directly from a video, due to limited computational resources, we split the video into frames using OpenCV and apply the frontal face detector on the mid frames of these frames. Figure.6 shows the extracted valid frames of a video that is ~2 seconds long where the lady says *" This is very exciting "*.

Model Architecture: Like any other GAN architecture, the training network consists of a Generator and a Discriminator. As seen in Figure.3, the generator consists of an audio encoder, a face encoder, and a face decoder. The discriminator on the other hand, consists of the audio encoder and face encoder. Each encoder is a combination of convolution blocks and residual blocks, while the decoder is a combination of transpose convolution blocks and residual blocks.

*Training:*The model takes a masked image input (of a frame) and the corresponding masked audio segment for the image and audio encoders respectively. Their concatenated output is fed as an embedding to the decoder, which then classifies the resultant image as in sync or out of sync with the audio segment. The model is trained using Adam optimizer for 20 epochs with a training objective to reduce the Mean Absolute Error (MAE).

Video Generation: For a given test image, we first detect the valid face view and generate the bounding boxes. Simultaneously, we generate the audio from the input text. We first initialize a video with the input image to the length of the audio. For each segment of the audio, we replace the existing frame with the output of the model. Finally, we stitch all the frames and render the video. The resultant video has no audio, so we use ffmpeg to lap the audio over the video. This way, we are able to generate realistic talking videos with minimal audio to video latency.

4. RESULTS

In this section, any module wise results are reported along with the final ensemble application.

4.1 StyleGAN

After training the model for 112 epochs, Figure.4 shows the loss plot, and Figure.5 shows some sample images from 128x128 to 4x4.

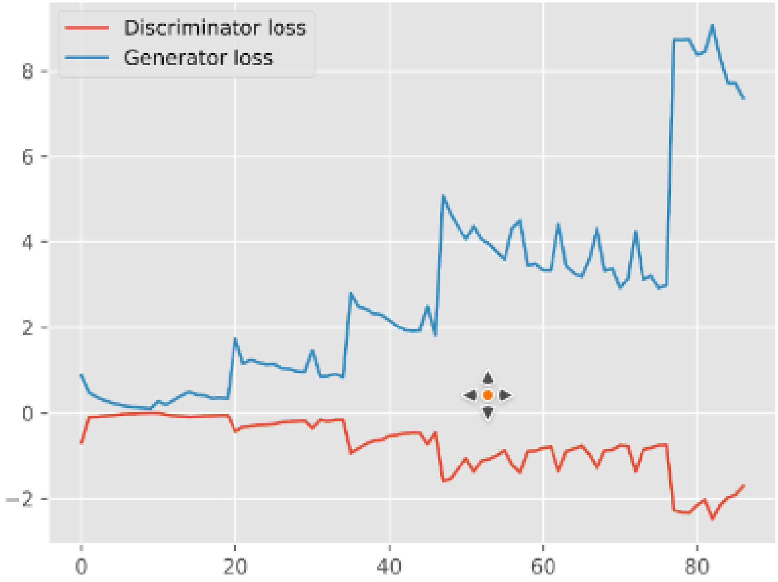


Figure 4: Loss Plot for 88 Epochs

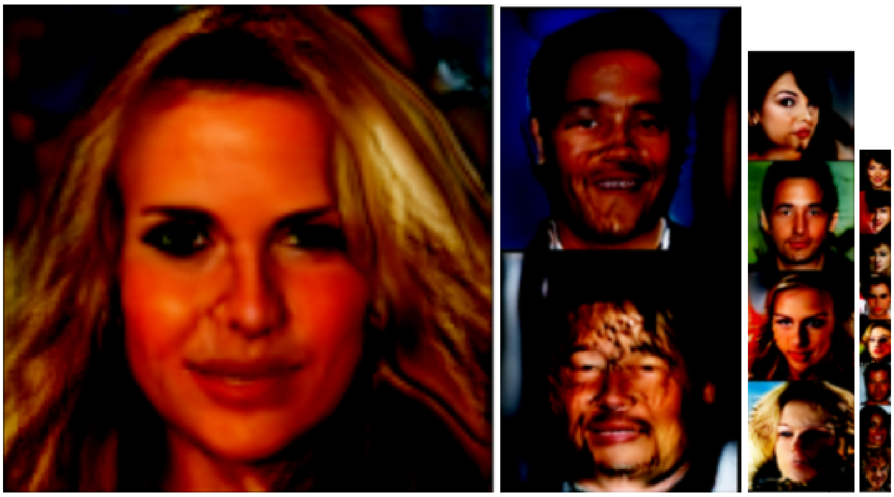


Figure 5: Samples of Generated Images

4.2 LipGAN

Following is the visualization of the output from the preprocessing stage of LipGAN implementation. Figure.6 shows the visualization of valid frames extracted using the aforementioned method, and Figure.7 is the visualization of the Mel Spectrogram the same audio (from the video).

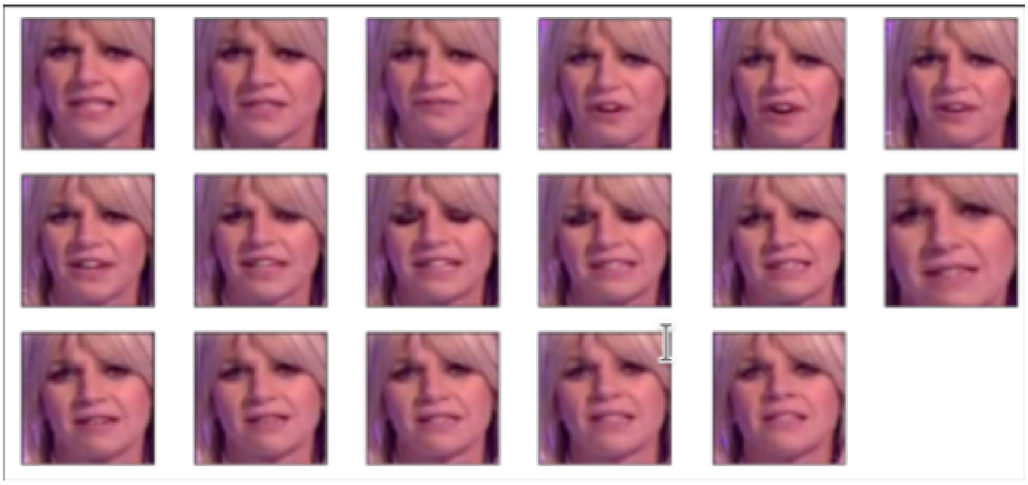


Figure 6: Valid Frames of a Sample Video

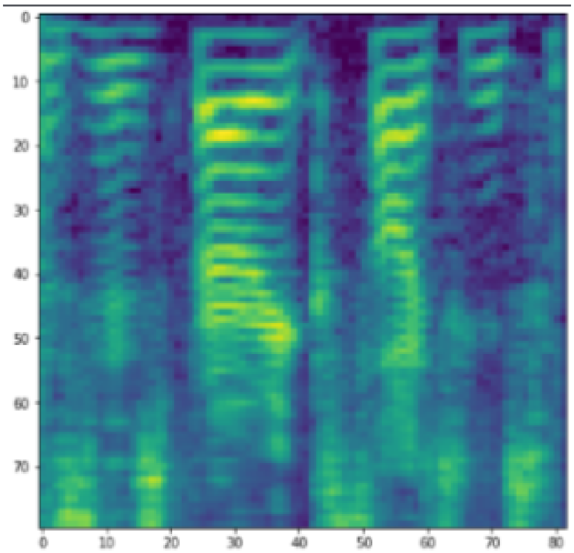
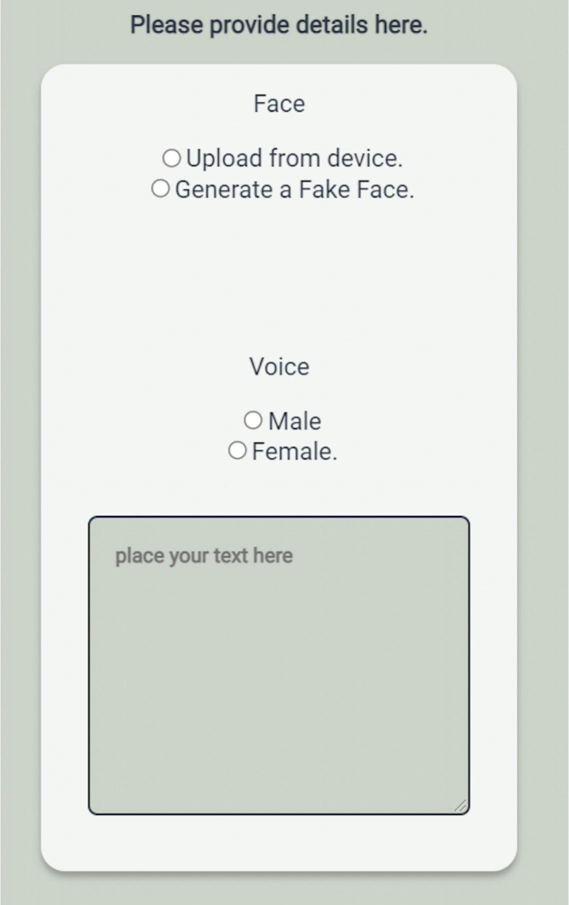


Figure 7: Visualization of a Mel Spectrogram

4.3 Web Application



Please provide details here.

Face

○ Upload from device.
○ Generate a Fake Face.

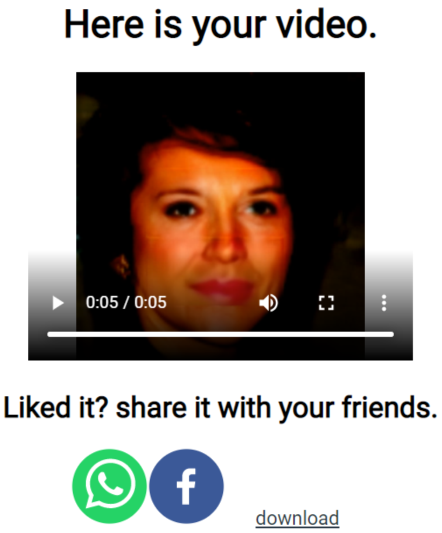
Voice

○ Male
○ Female.

place your text here

Figure 8: User Options in the Web-Application

Following the use case in Figure.1, the final application is an ensemble of all the modules. This application is built using Fast API and Node JS. It has 3 end-points- one each for generating fake faces, uploading an image from the device, and generating the video. As shown in Figure.8 The application provides the users an option to either use the fake image generated by WARP, or an image of their own; and also choices for the audio. The application will throw a interface error if no valid face views are found in the image. Upon successful creation of the video, users can choose from either viewing it or downloading it. Additionally, the users also have an option to share it on their social media with their connections [Figure.9].



Here is your video.

Liked it? share it with your friends.

Figure 9: Final Output

5. CONCLUSION

To summarize this report, the aim of this project is to enable generation of talking face videos using just text as input. The video is generated on a fake face generated using StyleGAN. Alternatively, the user can also use a custom face image. After converting the text to audio, we use it along with the face image to allow LipGAN to generate a series of lip synchronized frames corresponding to an audio segment. Finally, with the help of OpenCV and ffmpeg, we stitch and render the video with audio. The ensemble (web) application allows users to customize their video and either download it or share it via social media. With it's wide range of real-world applications like visualizing education, virtual assistants, revolutionizing entertainment sectors and personalising informal communication, WARP is convenient and can be virtually used by anyone.

List of Figures

List of Abbreviations and Symbols

GAN Generative Adversial Network.

LRS2 Lip Reading Sentences 2.

MFCC Mel Frequency Cepstral Coefficents.

RGB Red Green Blue.

WARP Woven Artificial Profile.

REFERENCES

1. Prajwal K R, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and C V Jawahar. 2019. Towards Automatic Face-to-Face Translation. In proceedings of the 27th ACM International Conference on Multimedia (MM '19).

2. Karras. T., Laine. S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).

3. Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou. 2015, December. Deep Learning Face Attributes in the Wild. In Proceedings of International Conference on Computer Vision (ICCV).

4.T. Afouras, J. S. Chung, A. Senior, O. Vinyals, A. Zisserman, Deep Audio-Visual Speech Recognition, arXiv:1809.02108

5. Thies J., Elgharib, M., Tewari A., Theobalt C., & Nießner M. (2020, August). Neural voice puppetry: Audio-driven facial reenactment. In European Conference on Computer Vision (pp. 716-731).

6. Saito M., Matsumoto. E., & Saito S. (2017). Temporal Generative Adversarial Nets with Singular Value Clipping. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2830-2839).

7. Chen, W., Tan, X., Xia, Y., Qin, T., Wang, Y., & Liu, T. Y. (2020, October). DualLip: A System for Joint Lip Reading and Generation. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 1985-1993).