UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

One-shot Facial Expression Reenactment using 3D Morphable Models

Author:
Roman VEI

Supervisors:
Eugene KHVEDCHENYA and
Orest KUPYN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2022

Declaration of Authorship

I, Roman VEI, declare that this thesis titled, "One-shot Facial Expression Reenactment using 3D Morphable Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

One-shot Facial Expression Reenactment using 3D Morphable Models

by Roman VEI

Abstract

The recent advance in generative adversarial networks has shown promising results in solving the problem of head reenactment. It aims to generate novel images with altered poses and emotions while preserving the identity of a human head from a single photo. Current approaches have limitations, making them inapplicable for real-world applications. Specifically, most algorithms are computationally expensive, have no apparent tools for manual image manipulation, require audio or take multiple input images to generate novel images.

Our method addresses the single-shot face reenactment problem with an end-to-end algorithm. The proposed method utilizes head 3D morphable model (3DMM) parameters to encode identity, pose, and expression. With the proposed approach, the pose and emotion of a person on an image is changed by manipulating its 3DMM parameters. Our work consists of a face mesh prediction network and a GAN-based renderer. A predictor is a neural network with simple encoder architecture that regresses 3D mesh parameters. A renderer is a GAN network with warping and rendering submodules that renders images from a single source image and target image 3DMM parameters.

This work proposes a novel head reenactment framework that is computationally efficient and uses 3DMM parameters that are easy to alter, making the proposed method applicable in real-life applications. It is first to our knowledge approach that simultaneously solves two of these problems: 3DMM parameters prediction and face reenactment, and benefits from both.

Acknowledgements

I want to thank my supervisors, Eugene Khvedchenya and Orest Kypyn, for directing and mentoring me during the research and for all their invaluable advices given to me. I want to mention Oleksii Molchanovskyi for his support and consulting me throughout the study. I want to express my gratitude to my colleague Vasyl Borsuk with whom we worked on the related research and who has made an invaluable contribution to this work. Separate thanks to PiñataFarms for providing access to GPU hardware and supporting this research project.

Contents

Declaration of Authorship i

Abstract iii

Acknowledgements iv

1 Introduction 1
1.1 Contributions . . . 2

2 Related Works 3
2.1 First Order Motion Model (FOMM) . . . 4
2.2 3D Morphable Model (3DMM) . . . 5
2.3 DAD-3DNet . . . 6
2.4 Generative Adversarial Networks . . . 6
2.5 Fast Bi-Layer . . . 7
2.6 StyleHEAT . . . 7
2.7 HeadGAN . . . 8
2.8 PiRenderer . . . 9

3 Proposed Methods 11
3.1 Model Architecture . . . 12
3.1.1 PNCC Estimation Pipeline . . . 12
3.1.2 Face Reenactment Pipeline . . . 13
3.1.3 SPADE/AdaIN Blocks . . . 15
3.2 Losses . . . 15
3.2.1 3D-Head Loss ($\mathcal{L}_{3D}$) . . . 15
3.2.2 Reprojection Loss ($\mathcal{L}_{Proj}$) . . . 15
3.2.3 Shape Loss ($\mathcal{L}_{Shape}$) . . . 16
3.2.4 Regularization Loss ($\mathcal{L}_{Reg}$) . . . 16
3.2.5 Warping Network Loss ($\mathcal{L}_{Warped}$) . . . 16
3.2.6 Enhancing Network Loss ($\mathcal{L}_{Enhanced}$) . . . 16
3.2.7 Mouth Loss ($\mathcal{L}_{Mouth}$) . . . 16
3.2.8 Style Loss ($\mathcal{L}_{Style}$) . . . 17
3.2.9 Perceptual Loss ($\mathcal{L}_{Perceptual}$) . . . 17
3.2.10 Person Re-identification Loss ($\mathcal{L}_{ReID}$) . . . 17
3.3 Inference . . . 17
3.3.1 Template Selection . . . 17
3.3.2 Deploy using Triton . . . 18

4 Datasets 20
4.1 Face Mesh Prediction Datasets . . . 20
4.1.1 NoW ("Not quite in-the-Wild") . . . 20
4.1.2 FaceScape . . . 20

4.1.3 DAD-3DHeads . . . 21
4.2 Face Reenactment Datasets . . . 21
4.2.1 Cmbiometrics . . . 21
4.2.2 Radboud Faces Database (RaFD) . . . 22
4.2.3 Compound Facial Expressions of Emotions Database (CFEE) . . . 22
4.2.4 The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) . . . 23

5 Evaluation **24**
5.1 Metrics . . . 24
5.2 Benchmark . . . 25

6 Experiments **26**
6.1 Experiment Setup . . . 26
6.2 Dataset Preperation . . . 26
6.2.1 Preprocessing steps . . . 26
6.2.2 Dataset super-resolution . . . 26
6.3 Results . . . 27
6.4 Ablation Studies . . . 30

7 Conclusions **32**
7.1 Result Summary . . . 32
7.2 Points to improve . . . 32

Bibliography **33**

List of Figures

1.1 Example of Face Reenactment: generating rotated faces from front one using our approach . . . 2

2.1 First Oreder Motion Model architecture from the paper . . . 5
2.2 FLAME [Li et al., 2017] model variations for shape, expression, pose, and appearance. . . . 5
2.3 PNCC image examples . . . 6
2.4 DAD-3DNet [Martyniuk et al., 2022] architecture design from original paper . . . 6
2.5 Fast BiLayer model Zakharov et al., 2020 architecture overview. . . . 7
2.6 StyleHEAT pipeline Yin et al., 2022 overview from authors paper. . . . 8
2.7 Architecture of HeadGAN [Doukas, Zafeiriou, and Sharmanska, 2020]. 9
2.8 HeadGAN Demo [Doukas, Zafeiriou, and Sharmanska, 2020] of image manipulation via 3DMM parameters. . . . 9
2.9 Architecture of PiRenderer [Ren et al., 2021]. . . . 10

3.1 General architecture of our approach. . . . 12
3.2 Overview of 3DMM parameters prediction pipeline. . . . 12
3.3 Overview of face reenactment pipeline. . . . 13
3.4 Generated emotions of Leonardo Di Caprio. . . . 19

4.1 Image samples from [Sanyal et al., 2019] project site. . . . 20
4.2 Side-by-side comparison of image and 3D Face model from [Yang et al., 2020] project site. . . . 20
4.3 Diverse image samples from [Martyniuk et al., 2022] paper. . . . 21
4.4 Samples from Cmbiometrics [Nagrani, Albanie, and Zisserman, 2018] dataset. . . . 21
4.5 A model with different emotions from [Langner et al., 2010] paper. . . . 22
4.6 Multiple people with sad emotion from [Du, Tao, and Martinez, 2014] project. . . . 22
4.7 Different people with different emotions from [Livingstone and Russo, 2018] authors. . . . 23

5.1 Video sample from Voxceleb [Nagrani, Chung, and Zisserman, 2017] test set. . . . 25

6.1 Example of super resolution on dataset. . . . 27
6.2 Visual comparison of methods on self-reenactment task on Cmbiometrics dataset. . . . 28
6.3 Visual comparison of methods on cross-reenactment task on Cmbiometrics dataset. . . . 29
6.4 Images of people with generated emotions on CFEE. dataset. . . . 30

List of Tables

3.1 Warping Network architecture. 14
3.2 Mapping Network architecture. 14
3.3 Enhancing Network architecture. 14
3.4 SPADE Block architecture. 15
3.5 Inference speed of reenacted model in end-to-end manner using Triton Inference Server on Nvidia Titan RTX 19

6.1 Comparison of different method results on VoxCeleb [Nagrani, Chung, and Zisserman, 2017] dataset. 27
6.2 Ablation studies for our method. 30

List of Abbreviations

PNCC	**P**rojected **N**ormalized **C**oordinate **C**ode
3DMM	**3D**-**M**orphable **M**odel
SOTA	**S**tate-**o**f-**t**he-**A**rt
PCA	**P**rincipal **C**omponent **A**nalysis
CNN	**C**onvolution **N**eural **N**etwork
GAN	**G**enerative **A**dversarial **N**etwork
LSTM	**L**ong **S**hort-**t**erm **M**emory
FOMM	**F**irst **O**rder **M**otion **M**odel
FLAME	**F**aces **L**earned with an **A**rticulated **M**odel and **E**xpressions
LPIPS	**L**earned **P**erceptual **I**mage **P**atch **S**imilarity
CSIM	**C**osime **Sim**ilarity
FID	**F**rechet **I**nception **D**istance
MSE	**M**ean **S**quared **E**rror
PSNR	**P**eak **S**ignal-to-**n**oise **R**atio
SSIM	**S**tructure **Sim**ilarity

Dedicated to all Ukrainian fighters in a war for our freedom

Chapter 1

Introduction

Neural networks have become a hot field to research in recent years. The main factor of its growth is increased computational abilities and stored data. In combination with optimized algorithms, these factors make a revolution in our world.

Some impossible tasks a few years ago become realistic now, and authors compete to increase the quality bar even higher. One of these tasks is face emotions and position editing, namely Face Reenactment. In this task, the model inputs one or more images of the same person and additional inputs, which encode emotion or pose transformation in the clear for the model format. These format examples include keypoints, 3DMM parameters, audio, or face boundaries. The model produces an image with the same person but with an altered pose and emotion as described in additional inputs.

Creating personalized head avatars is a trending topic of research. With this technology, everybody can easily create their persona without special tools and wasting time. It is crucial because additional user flow simplification can significantly improve user experience and create a competitive advantage in business.

During the face reenactment, the model, firstly, extracts identity, pose, and expression metadata. After that, it applies another set of poses and expressions to produce a novel image with the same person from a different viewpoint. Using this, we can animate people in photos and edit their appearance.

In recent years, multiple attempts have been made to develop algorithms that can generate custom emotions. Some can model avatars with high realism but need multiple photos and manual input. Another group can create a virtual head from a single image but does not pretend on photorealism. Moreover, they are not applicable in real-life scenarios because of slow processing run-time, lack of inputs from a user, or low-quality predictions.

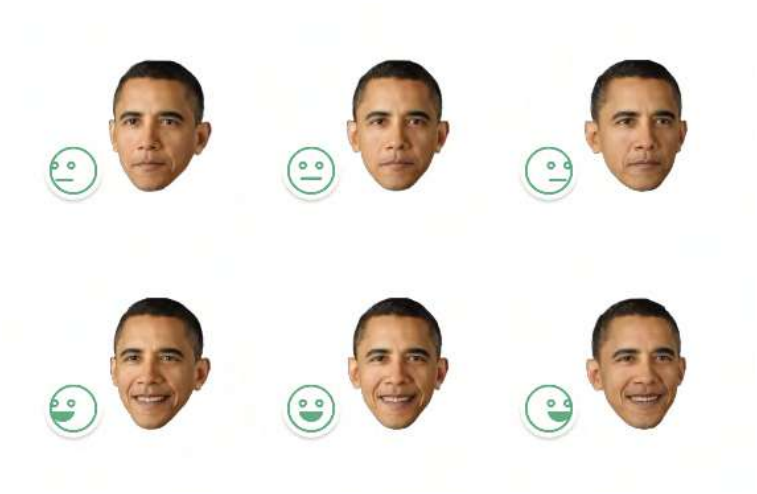

FIGURE 1.1: Example of Face Reenactment: generating rotated faces from front one using our approach

1.1 Contributions

Our work addresses the single-shot face reenactment problem end-to-end from a single image input. We will use 3DMM [Blanz and Vetter, 1999; Booth et al., 2018b; Booth et al., 2018a] parameters to encode identity, pose, and expression. The proposed approach edits the pose and emotion of a real person on an image by manipulating its 3DMM parameters. As a result, our method is easy-to-use and requires only one ordinary RGB image as an input.

In summary, the main contributions of this work are:

- Combining face mesh estimation and face reenactment tasks
- Discovering architecture changes that take away main visual disadvantages of previous approaches: mouth discriminator, residual connections, PNCC conditioning during training
- Develops robust data preprocessing pipeline, taking into account prior knowledge about human faces
- Optimizing model performance to make it applicable in real-life applications
- Providing a method for emotion manipulation from an arbitrary image

Chapter 2

Related Works

Many approaches can generate talking head videos only for specific identities [Garrido et al., 2015; Bregler, Covell, and Slaney, 1997; Chang and Ezzat, 2005; Liu and Ostermann, 2011]. For example, [Suwajanakorn, Seitz, and Kemelmacher-Shlizerman, 2017] synthesizes only the mouth region and combines it with a frame from a large video corpus of that person. Despite the high accuracy and photorealism, this method is hard to use in real-world scenarios because of the large video corpus. Moreover, it is not extendable for new identities. Other algorithms need additional metadata like audio to work [Chen et al., 2020; Doukas, Zafeiriou, and Sharmanska, 2020; K R et al., 2019]. There are multiple criteria to categorize all these methods for image editing.

Firstly, avatar synthesis algorithms can be divided into two groups based on the number of images needed to work. In the first group, we have algorithms that need multiple frames; in another group - there are one-shot algorithms.

Number of algorithms [Cao et al., 2021; Wu et al., 2018; Kim et al., 2018; Thies et al., 2020; K R et al., 2019; Wang et al., 2019; Doukas et al., 2020] are working in a many-to-many manner. ReenactGAN [Wu et al., 2018] uses CycleGAN [Zhu et al., 2017] to convert the facial boundary heatmaps between different persons to increase the quality of the decoder. The method presented by [Kim et al., 2018] can synthesize high-resolution and realistic facial images with GAN. [Thies et al., 2020] animates the expression of the source video by swapping the source face with the rendered image. The problem with all these approaches is that they require many pictures of the specific identity for training and only work on them.

[Doukas, Zafeiriou, and Sharmanska, 2020; Zakharov et al., 2020; Yao et al., 2021; Wang, Mallya, and Liu, 2020; Siarohin et al., 2020; Ren et al., 2021; Yin et al., 2022; Zhang et al., 2021; Bounareli, Argyriou, and Tzimiropoulos, 2022] need only a single frame as an input. These approaches are more challenging but provide more flexibility in real applications.

Another way to divide most face reenactment approaches is based on which inputs they use for conditioning source to driving image transformation:

- landmark-based
- motion-based
- 3D-based
- audio-based
- explicit

Landmark-based methods utilize facial landmarks as conditions to render animated source image. [Wang et al., 2019] injects landmarks as a conditional input

through AdaIN [Huang and Belongie, 2017] blocks. [Zakharov et al., 2020] uses SPADE [Park et al., 2019] blocks to build rendering network based on the landmark skeleton.

Motion-based methods model a relative motion field from source to driving images [Wiles, Koepke, and Zisserman, 2018; Siarohin et al., 2020; Wang, Mallya, and Liu, 2020]. X2Face directly estimates a dense motion field. First Order Motion Model (FOMM) [Siarohin et al., 2020] uses self-supervised 2D keypoints to estimate multiple local sparse motions and then aggregates them into one dense motion flow to wrap the initial image. One-Shot Talking Heads [Wang, Mallya, and Liu, 2020] from NVIDIA researchers go even further and estimate 3D self-supervised keypoints.

In recent year the most popular approach is to use 3DMM parameters as a conditional input:

- HeadGAN [Doukas, Zafeiriou, and Sharmanska, 2020] uses PNCC (rendered from 3DMM parameters) images for guiding reenactment network.
- GIF [Ghosh et al., 2020] and StyleRig [Tewari et al., 2020] use pre-trained StyleGAN [Karras, Laine, and Aila, 2018] and 3DMM parameters for conditioning.
- PiRenderer [Ren et al., 2021] maps original 3DMM parameters through the Mapping Network to get internal representation.

Audio-based methods use the audio signal for conditioning. Some of them directly use audio: [Song et al., 2018; Zhou et al., 2019]. Another part - map audio to the middle representations, such as landmarks [Suwajanakorn, Seitz, and Kemelmacher-Shlizerman, 2017] or 3DMM parameters [Yi et al., 2020]. However, these methods require multiple frames, and audio conditioning is not deterministic and does not generalize well.

Explicit methods for source to driving image transformation use such approaches as [Yin et al., 2022; Bounareli, Argyriou, and Tzimiropoulos, 2022]. They use StyleGAN [Karras, Laine, and Aila, 2018] under the hood and train it to learn pose and expressions transformation explicitly from images.

2.1 First Order Motion Model (FOMM)

First Order Motion Model (FOMM) [Siarohin et al., 2020] is one of the first papers describing image animation using neural networks. It is not designed explicitly for face animation. Still, this approach is widely used as a baseline for this task because of its generalization.

Their method assumes a source image and a frame of a driving video frame as inputs. The unsupervised keypoint detector extracts motion representation (sparse keypoints and local affine transformations). The dense motion network uses the motion representation to generate optical flow and the occlusion map. The generator uses the source image and the outputs of the dense motion network to render the target image.

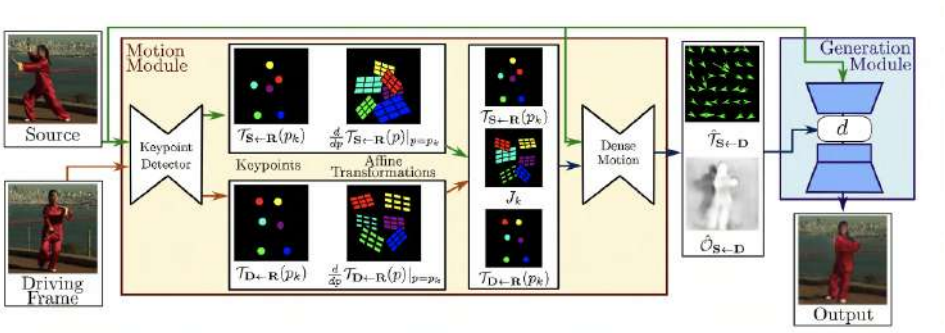

FIGURE 2.1: First Oreder Motion Model architecture from the paper

Despite the popularity of this approach, it has significant disadvantages, like low-quality rendering and the number of artifacts (especially near the eyes and mouth). Moreover, it is hard to specify concrete emotion and finetune it.

2.2 3D Morphable Model (3DMM)

3D morphable model (3DMM) [Blanz and Vetter, 1999; Booth et al., 2018b; Booth et al., 2018a] is a statistical model that inputs the shape and texture of a face and outputs a vector representation. The first versions of 3DMMs are based on principal component analysis (PCA). A face is represented by a linear combination of those orthogonal bases with the largest eigenvalues. The current state-of-the-art model is FLAME (Faces Learned with an Articulated Model and Expressions) [Li et al., 2017]. This model uses a linear shape space trained from 3800 scans of human heads. FLAME combines this linear shape space with an articulated jaw, neck, eyeballs, pose-dependent corrective blend shapes, and other global expression blend shapes. As a result, this approach could be used for face shape and expression encoding.

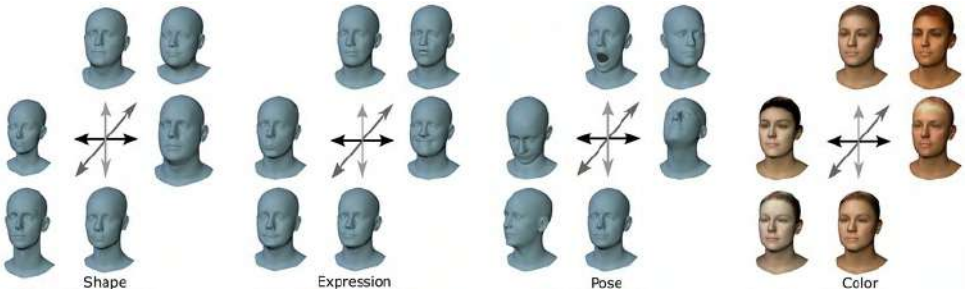

FIGURE 2.2: FLAME [Li et al., 2017] model variations for shape, expression, pose, and appearance.

FLAME model has decoder which renders head from those parameters. One of the rendering output options is **PNCC** (Projected Normalized Coordinate Code) image. It is used with slight modifications inside our network to encode pose, shape, and expressions as input. The main properties of this image are:

- 3D mean face is normalized to 0-1 with the 3D coordinate of each point being called Normalized Coordinate Code
- Project the 3D face with parameter p using Z-buffer
- Encode the depth at each point using RGB values

FIGURE 2.3: PNCC image examples

2.3 DAD-3DNet

DAD-3DNet [Martyniuk et al., 2022] is a recently published model that computes 3D mesh representation consistent with the FLAME [Li et al., 2017] topology. It is only part of the publication, but it was considered the most interesting for our research, along with its large dataset with images and corresponding 3D meshes.

The model is trained to predict 3DMM parameters in an end-to-end manner. Those 3DMM parameters are used to get 3D mesh representation using the FLAME decoder. The architecture of this model is similar to detection models: CNN encoder as a backbone inside the BiFPN [Tan, Pang, and Le, 2019] pyramid. As an output, this model has two branches: first - 3DMM parameters, second - standard 2D keypoints. This additional branch helps improve the prediction quality of this model and is used only during training.

The advantages of this model are that it can be easily trained and fix one of the biggest challenges of face reenactment task: extracting 3DMM parameters from the image.

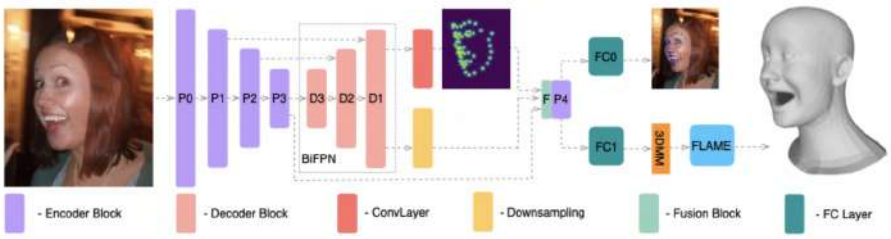

FIGURE 2.4: DAD-3DNet [Martyniuk et al., 2022] architecture design from original paper

2.4 Generative Adversarial Networks

The generative adversarial networks (GANs) [Goodfellow et al., 2014] were introduced in 2014. They propose a framework for estimating generative models via an adversarial process.

The GAN training strategy is a competition between two models. The first *generator* model tries to predict the meaningful output from noise, and the second - *discriminator* tries to distinguish the actual image from the generated. Inside these

models, we use NNs with an encoder-decoder structure. Mathematically, we can describe this process in such a way:

$$\min_G \max_D x \sim \mathbb{P}_r \mathbb{E}[\log(D(x))] + \tilde{x} \sim \mathbb{P}_g \mathbb{E}[\log(1 - D(\tilde{x}))] \tag{2.1}$$

Where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution.

This straightforward training setup became one of the most fundamental works in the recent decade. Not surprising that a lot of SOTA approaches used GANs or their variations for different tasks.

2.5 Fast Bi-Layer

Zakharov [Zakharov et al., 2020] proposes an architecture that renders head avatars from a single photograph. Their approach is decomposed into two blocks. The first block is a pose-dependent low-quality image synthesized by a small neural network. The second block is defined by a pose-independent texture image that contains high-frequency details. The texture image from the first block is generated separately and only once. After that, on each video frame, this texture warped and added to the coarse image to ensure a high effective resolution of the talking head. As input, this model needs image frame person landmarks to initialize and driving landmarks on each step for inference. Figure 2.5 overview of the model architecture.

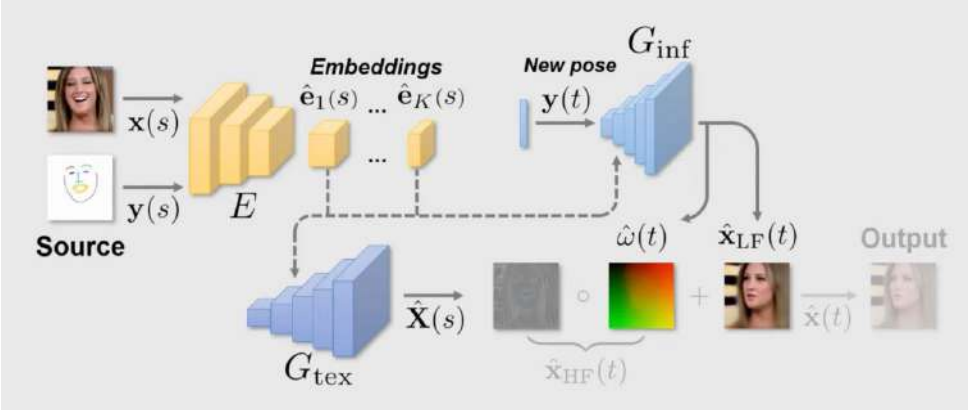

FIGURE 2.5: Fast BiLayer model Zakharov et al., 2020 architecture overview.

Despite its speed, this approach has multiple limitations. The quality of outputs is lower compared to other methods. Another limitation is the usage of landmarks for shape and emotion encoding: it is impossible to get the source face shape and add custom expressions.

2.6 StyleHEAT

StyleHEAT [Yin et al., 2022] is recent approach used for Face Reenactment. Their pipeline consists of four components:

- Pre-trained StyleGAN [Karras, Laine, and Aila, 2018]
- Video-driven Motion Generator

- Audio-driven Motion Generator
- Calibration Network

From a source image, they obtain the feature maps and style vectors via GAN inversion methods. Video or audio is used as additional inputs to predict motion fields using a Video-driven Motion Generator and Audio-driven Motion Generator, respectively. A predicted motion field warps StyleGAN feature maps, and this warped feature map goes through Calibration Network to enhance predictions. This calibrated feature map is then fed into the StyleGAN for the final face generation. The biggest drawback of this approach is that it is hard to use this method on real images. GAN inversion is not very stable and leads to a lack of identity preservation.

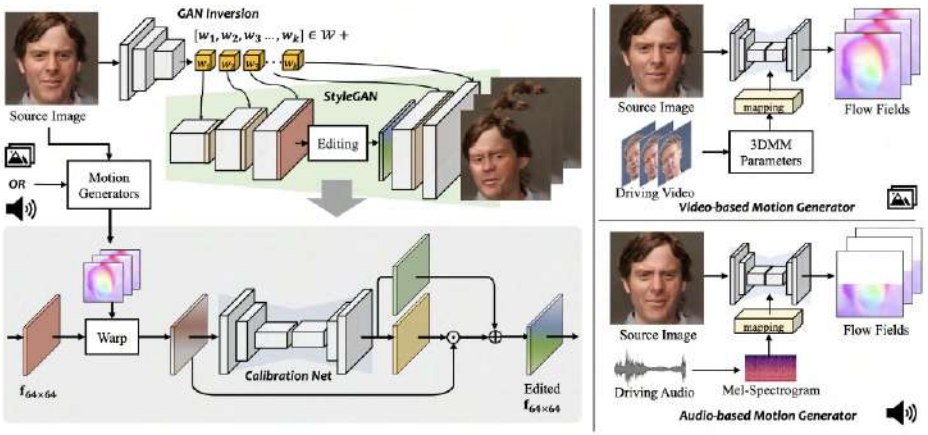

FIGURE 2.6: StyleHEAT pipeline Yin et al., 2022 overview from authors paper.

2.7 HeadGAN

HeadGAN [Doukas, Zafeiriou, and Sharmanska, 2020] is a new model that overcomes issues with 2D landmarks using parameters from the FLAME [Li et al., 2017] 3D head model. Their architecture consists of a dense flow network and a rendering network.

Image concatenated with rendered PNCC used as an input. A dense flow network is based on convolution and SPADE [Park et al., 2019] blocks in which driving PNCCs are injected. This network outputs flow. After that, a source image and feature maps are warped with this output flow. To renderer, inputs are three driving PNCCs. Inside the renderer, they are mixed with warped features and source images from Denseflow inside SPADE blocks and output the final image. Figure 2.7 shows two parts of the model.

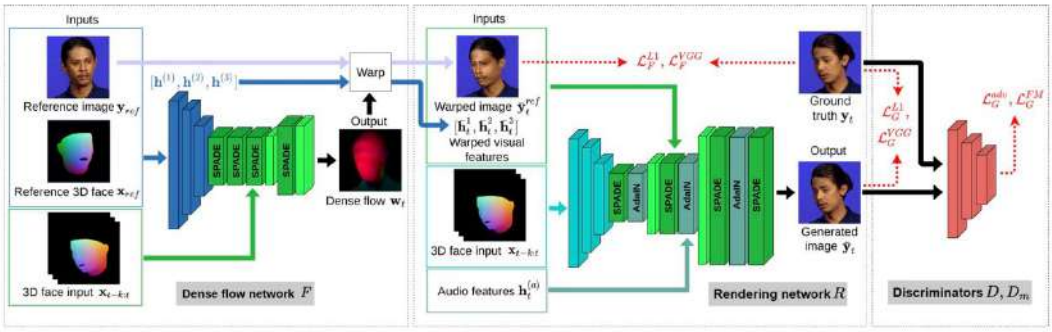

FIGURE 2.7: Architecture of HeadGAN [Doukas, Zafeiriou, and Sharmanska, 2020].

This method allows simple face emotion manipulation of the source image using 3DMM parameters. Despite that, the most significant limitation is also related to them. We need to solve the optimization process to get these parameters, which takes a while. Because of that, we cannot use this model in real applications.

FIGURE 2.8: HeadGAN Demo [Doukas, Zafeiriou, and Sharmanska, 2020] of image manipulation via 3DMM parameters.

2.8 PiRenderer

Ren et al. [Ren et al., 2021] proposed the end of the 2021 year a neural rendering model PiRenderer. Given a source image and target 3DMM parameters, this model generates realistic results with the accurate motion of the person from a source image.

The proposed model has three parts:

- the Mapping Network - maps 3DMM parameters into latent space
- the Warping Network - generates warping flow, which preserves movement
- the Editing Network - enhance predictions and generate missing parts

The Mapping Network maps motion descriptors into latent vectors. These latent vectors guide the Warping Network, which produces a coarse image by predicting warping flow and using it on the source image. Although Warping Network is efficient at spatial transformations, it introduces artifacts and cannot fill missing parts

of the content. Therefore, for this purpose, the authors add the Editing Network, which modifies coarse warped images and generates a final prediction.

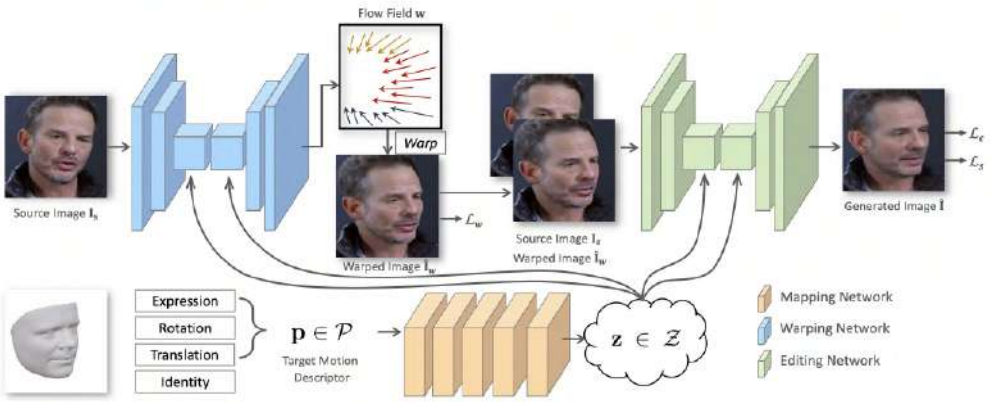

FIGURE 2.9: Architecture of PiRenderer [Ren et al., 2021].

As an extension, the authors provide an audio-driven facial reenactment model. This model is a recurrent network similar to LSTM [Hochreiter and Schmidhuber, 1997] that inputs previously generated k motions and audios as conditional information.

Chapter 3

Proposed Methods

Despite many papers in this field, all described above methods have several limitations.

Firstly, head reenactment should generate parts of the human face that are invisible or occluded in the reference image. Moreover, there is no tolerance for small algorithm mistakes. Multiple approaches [Ren et al., 2021; Doukas, Zafeiriou, and Sharmanska, 2020] use two-stage pipelines to tackle this challenge. The first model predicts warped images, and the second one enhances predictions and fills the gaps.

The second problem is the difficulty of efficiently encoding driving pose and emotion into the pipeline. For example, most state-of-the-art methods [Zakharov et al., 2020; Zakharov et al., 2019; Siarohin et al., 2020; Wang et al., 2019] use facial landmarks to guide the synthesis. They usually lead to identity preservation problems during the reenactment as keypoints encode identity information. It is visible when the head geometry of the source person differs from that of the person in the driving video.

The last and most significant problem is related to the end-to-end deploying of such models. Even recent papers, like [Doukas, Zafeiriou, and Sharmanska, 2020; Ren et al., 2021; Yin et al., 2022], don't solve the problem of retrieving 3DMM parameters for the unseen images in the wild.

We propose a new approach that combines advantages of face reenactment and 3D-mesh prediction fields. It is a two stage algorithm consisting of the following building blocks:

- **3DMM Parameters Predictor** - predicts 3DMM parameters from an image
- **FLAME Decoder** - decodes 3DMM parameters into vertices
- **External C++ library** - renders PNCC image from vertices
- **Warping Network** - outputs warped source image
- **Enhancing Network** - enhances output and generates missing parts

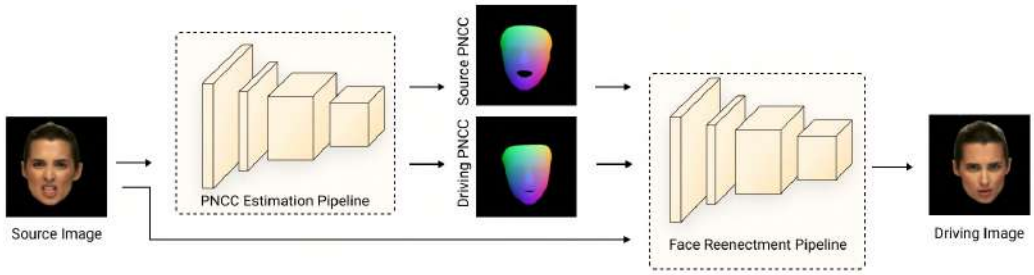

FIGURE 3.1: General architecture of our approach.

The first stage is the **PNCC Estimation Pipeline**. This part has 3DMM Parameters Predictor, which inputs RGB images and outputs 3DMM parameters. These 3DMM parameters can be intuitively modified to get the target 3DMM parameters. After that, these parameters are decoded through FLAME Decoder and generated PNCC images using an external C++ library.

Original image, source PNCC image, and target PNCC images are fed to the second stage - **Face Reenactment Pipeline**. Firstly, it goes through the Warping Network to output the warped source image to look like a target. Secondly, this warped image is enhanced by Enhancing Network.

This network outputs the person from the source image, but emotions and pose for them are obtained from the target PNCC image. This architecture solution overcame external time-consuming parameters calculation [Zakharov et al., 2020; Doukas, Zafeiriou, and Sharmanska, 2020; Ren et al., 2021] or not an intuitive representation of the source and driving poses [Siarohin et al., 2020; Zakharov et al., 2020]. As a result, we can use our method everywhere without additional preprocessing, and it is a much easier solution for deployment in real-life scenarios. As a bonus, we wrap all these stages into Triton Inference Server. Because of that, our method is used end-to-end from the arbitrary image.

3.1 Model Architecture

3.1.1 PNCC Estimation Pipeline

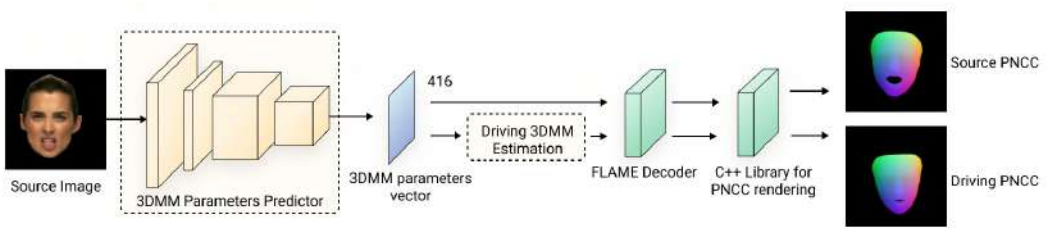

FIGURE 3.2: Overview of 3DMM parameters prediction pipeline.

The main block of the PNCC Estimation Pipeline is Face Mesh Predictor, which predicts 3DMM parameters from a single photo. This network consists of a backbone

(feature extractor) inside a BiFPN [Tan, Pang, and Le, 2019] and a Regression Module that directly predicts the 3DMM parameters vector. Deterministic FLAME Layer takes 3DMM vector as an input and outputs 3D head model vertices. Because of this light setup, predictions of 3DMMs would be near real-time and easily incorporated into the face reenactment module. This pipeline also uses an external C++ library to generate PNCCs from the given 3DMM parameters. The process of getting PNCC images is shown in Fig. 3.2.

3.1.2 Face Reenactment Pipeline

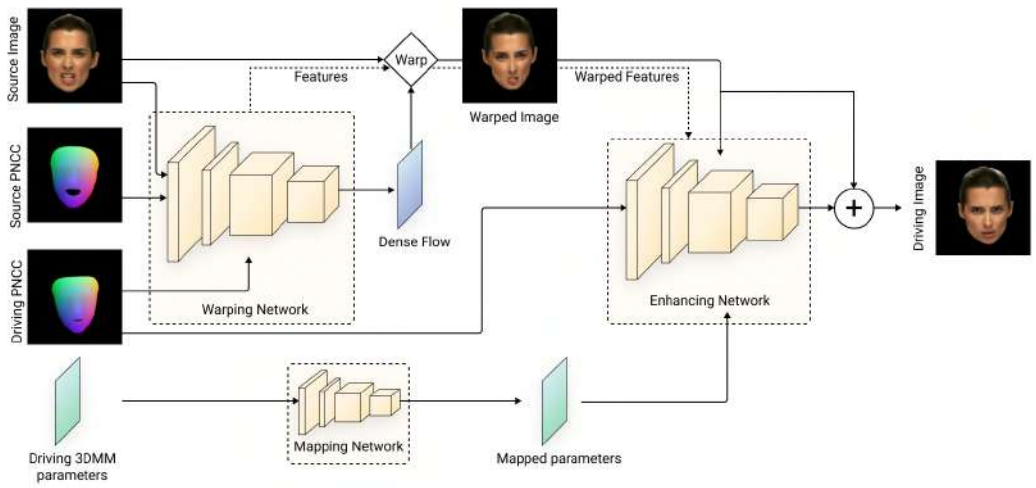

FIGURE 3.3: Overview of face reenactment pipeline.

The Face Reenactment pipeline is responsible for actual face rendering. Our architecture has warping and enhancing modules similarly to [Doukas, Zafeiriou, and Sharmanska, 2020; Ren et al., 2021; Zakharov et al., 2020]. The general overview of the pipeline is shown in Fig. 3.3.

The **Warping Network** is used to predict warping flow which makes an initial "rough" transformation between the source image and driving PNCC. This network inputs concatenated RGB source image with source PNCC and has three convolutions to downscale the initial image shape. After that model has multiple SPADE blocks followed by Pixel Shuffle blocks to return to the initial scale. We eject information about driving PNCC into SPADE blocks. Because of that, our network learns warping flow, which can make source image or encoder features aligned with the target PNCC face position.

The **Mapping Network** maps target 3DMM parameters into intermediate space. This type of network is used in [Karras, Laine, and Aila, 2018; Ren et al., 2021] and primarily consists of sequential linear layers followed by Relu activation.

The **Enhancing Network** enhances warped image and fills missing parts. It consists of three convolutions and multiple SPADE-AdaIN blocks in different resolutions to encode driving PNCC into the final image output. Into the SPADE blocks, we eject warped source image or warped source features, and into the AdaIN block - mapped parameters of the target 3DMM parameters. This network learns how to map the driving PNCC image into the reenacted RGB image. The 3DMM parameters before injecting into AdaIN blocks are processed through the mapping network similarly to [Karras, Laine, and Aila, 2018; Ren et al., 2021]. We add a residual connection between warped image and enhanced output to boost predictions' quality.

Block	Output size
Input	(6, 512, 384)
7x7 Conv2D (32 ch.) + Inst. Norm. + Relu	(32, 512, 384)
3x3 Conv2D (128 ch.) + Inst. Norm. + Relu	(128, 256, 192)
3x3 Conv2D (256 ch.) + Inst. Norm. + Relu	(512, 128, 96)
SPADE Block	(512, 128, 96)
Pixel Shuffle	(128, 256, 192)
SPADE Block	(128, 256, 192)
Pixel Shuffle	(32, 512, 384)
7x7 Conv2D (2 ch.)	(2, 512, 384)

TABLE 3.1: Warping Network architecture.

Block	Output size
Input	(106)
Linear + Relu	(128)
Linear + Relu	(128)
Linear + Relu	(128)
Linear + Relu	(128)
Linear	(128)

TABLE 3.2: Mapping Network architecture.

The Enhancing Network predicts only changes needed to add to a warped image, but not the whole image from scratch, like in [Doukas, Zafeiriou, and Sharmanska, 2020].

Block	Output size
Input	(3, 512, 384)
7x7 Conv2D (32 ch.) + Inst. Norm. + Relu	(32, 512, 384)
3x3 Conv2D (128 ch.) + Inst. Norm. + Relu	(128, 256, 192)
3x3 Conv2D (256 ch.) + Inst. Norm. + Relu	(512, 128, 96)
SPADE Block	(512, 128, 96)
AdaIN Block	(512, 128, 96)
Pixel Shuffle	(128, 256, 192)
SPADE Block	(128, 256, 192)
AdaIN Block	(128, 256, 192)
Pixel Shuffle	(32, 512, 384)
SPADE Block	(32, 512, 384)
AdaIN Block	(32, 512, 384)
SPADE Block	(32, 512, 384)
7x7 Conv2D (3 ch.)	(3, 512, 384)
Residual Connection	(3, 512, 384)
Clamp (-1, 1)	(3, 512, 384)

TABLE 3.3: Enhancing Network architecture.

3.1.3 SPADE/AdaIN Blocks

The SPADE/AdaIN blocks similarly to [Doukas, Zafeiriou, and Sharmanska, 2020] consists of SPADE layer introduced by Nvidia [Park et al., 2019] or AdaIN layer [Huang and Belongie, 2017], convolutions and leaky relu activations.

SPADE/AdaIN Block
SPADE/AdaIN Layer
Leaky Relu + 3x3 Conv2D
SPADE/AdaIN Layer
Leaky Relu + 3x3 Conv2D

TABLE 3.4: SPADE Block architecture.

3.2 Losses

In our pipeline we train model on two stages. The face mesh predictor is trained by minimizing the following total loss:

$$\mathcal{L}_{MeshPrediction} = \lambda_1 \mathcal{L}_{3D} + \lambda_2 \mathcal{L}_{Proj} + \lambda_3 \mathcal{L}_{Shape} + \lambda_4 \mathcal{L}_{Reg}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ is a coefficients with values 50., 0.1, 1. and 1. respectively. Those values are taken from [Martyniuk et al., 2022] authors.

During second stage we trained face reenactment model using this combined loss:

$$\mathcal{L}_{FaceReenactment} = \lambda_5 \mathcal{L}_{Warped} + \lambda_6 \mathcal{L}_{Enhanced} + \lambda_7 \mathcal{L}_{Mouth} + \lambda_8 \mathcal{L}_{Style} + \lambda_9 \mathcal{L}_{Perceptual} + \lambda_1 0 \mathcal{L}_{ReID}$$

where $\lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_1 0$ is a coefficients with values 1., 1., 1., 50., 10. and 10. respectively. These losses are proved their importance in such papers as [Ren et al., 2021; Doukas, Zafeiriou, and Sharmanska, 2020; Yin et al., 2022]. These lambdas are taken from [Doukas, Zafeiriou, and Sharmanska, 2020] authors. The only ReID loss was added for better face preservation (lambda chosen empirically).

3.2.1 3D-Head Loss ($\mathcal{L}_{3D}$)

This loss is used to measure how good our 3DMM parameters are by comparing them to ground truth in a 3D space. The predicted parameters from the Face Mesh Predictor go through FLAME Decoder to obtain vertices V_{Pred}, the same applied to ground truth 3DMM parameters to get V_{GT}.

To get more representative results, we use only vertices that are mentioned as "head" (v_{Pred}, v_{GT}) and normalize ϕ them to fit into the unit cube. More formally, this loss can be described below:

$$\mathcal{L}_{3D} = |\phi(v_{Pred}) - \phi(v_{GT})|_2$$

3.2.2 Reprojection Loss ($\mathcal{L}_{Proj}$)

This loss measures how well our 3DMM parameters can be reprojected into a 2D image. We project 3D vertices of head mesh onto the image to compute them. Such

pipeline is repeated on target 3DMM parameters to get ground truth. After that, these reprojected vertices were compared via L1 distance.

3.2.3 Shape Loss ($\mathcal{L}_{Shape}$)

Our Face Mesh Predictor must be consistent and output the same shape 3DMM parameters on the same person. To achieve this, shape loss compares shape parameters on a set of images with the same person via L1 loss.

3.2.4 Regularization Loss ($\mathcal{L}_{Reg}$)

We add additional regularization loss to prevent parameter explosion to ensure that our 3DMM parameters remain small. This loss is calculated as a mean value of normalized shape and expression values.

3.2.5 Warping Network Loss ($\mathcal{L}_{Warped}$)

Similar to [Doukas, Zafeiriou, and Sharmanska, 2020; Ren et al., 2021], one of the face reenactment losses is a GAN loss between warped and target images. This loss guides our Warping network to predict warping flow correctly. We derive that Hinge loss has better visual quality than Wasserstein variations during multiple experiments.

$$\mathcal{L}_D = -\mathbb{E}\left[\min\left(0, -1 + D\left(I_{Warped}||I_{TgtPNCC}\right)\right)\right] - \mathbb{E}\left[\min\left(0, -1 - D\left(I_{Tgt}||I_{TgtPNCC}\right)\right)\right]$$

$$\mathcal{L}_G = -\mathbb{E}\left[D\left(I_{Warped}||I_{TgtPNCC}\right)\right]$$

The additional step we found necessary is a **PNCC conditioning**. We stack our target image with target PNCC and make it 6-channel. This extra stacking provides additional information for the discriminator to focus more on the face.

3.2.6 Enhancing Network Loss ($\mathcal{L}_{Enhanced}$)

This loss makes sure that Enhancing Network outputs as real images as possible. It is similar to Warping Network Loss, with the only difference that here we measure how good $I_{Enhanced}$ enhanced image is.

$$\mathcal{L}_D = -\mathbb{E}\left[\min\left(0, -1 + D\left(I_{Enhanced}||I_{TgtPNCC}\right)\right)\right] - \mathbb{E}\left[\min\left(0, -1 - D\left(I_{Tgt}||I_{TgtPNCC}\right)\right)\right]$$

$$\mathcal{L}_G = -\mathbb{E}\left[D\left(I_{Enhanced}||I_{TgtPNCC}\right)\right]$$

3.2.7 Mouth Loss ($\mathcal{L}_{Mouth}$)

During experiments, we found out that the eyes and mouth have the most significant problems with quality because of their complexity. To improve the quality of generated mouth region, we add additional GAN Loss over those pixels. After preprocessing, we heuristically estimate where the mouth is located and create a corresponding m mask to determine this region. After that, we use the same Enhancing Network Loss, but only over masked pixels of enhanced $(I_{Enhanced}|m)$ and target $(I_{Tgt}|m)$ images.

$$\mathcal{L}_D = -\mathbb{E}\left[\min\left(0, -1 + D\left((I_{Enhanced}|m)||(I_{TgtPNCC}|m)\right)\right)\right]$$

$$-\mathbb{E}\left[\min\left(0, -1 - D\left((I_{Tgt}|m)||(I_{TgtPNCC}|m)\right)\right)\right]$$
$$\mathcal{L}_G = -\mathbb{E}\left[D\left((I_{Enhanced|m})||(I_{TgtPNCC}|m)\right)\right]$$

3.2.8 Style Loss ($\mathcal{L}_{Style}$)

Sometimes only GAN loss is not enough to produce high-quality images. Such losses tend to make predictions more "blurry." We use additional L1 losses between warped/real and enhanced/real images to handle this issue and make output images sharper.

$$\mathcal{L}_{Style} = |I_{Enhanced} - I_{Tgt}|_1 + |I_{Warped} - I_{Tgt}|_1$$

3.2.9 Perceptual Loss ($\mathcal{L}_{Perceptual}$)

To increase the quality of images even more, we use widespread perceptual loss over the research community [Doukas, Zafeiriou, and Sharmanska, 2020; Ren et al., 2021; Yin et al., 2022; Wang, Mallya, and Liu, 2020; Zakharov et al., 2020]. This loss is calculated as an L1 distance between activation maps of the pre-trained VGG-19 network.

$$\mathcal{L}_{Perceptual} = \sum_i |\phi_i\left(\mathbf{I}_t\right) - \phi_i\left(\hat{\mathbf{I}}_w\right)|_1$$

3.2.10 Person Re-identification Loss ($\mathcal{L}_{ReID}$)

To better preserve a reenacted person's identity, we use auxiliary Reid loss. We use the pre-trained ρ Reid model [Deng et al., 2019] to calculate face embeddings inside this loss. The loss is a cosine similarity between embeddings derived from predicted and target faces.

$$\mathcal{L}_{ReID} = \frac{\rho(I_{Enhanced}) \cdot \rho(I_{Tgt})}{||\rho(I_{Enhanced})||_2 \cdot ||\rho(I_{Enhanced})||_2}$$

3.3 Inference

3.3.1 Template Selection

Another important topic is how to generate driving 3DMMs from source parameters. A simple solution is to get parameters responsible for expressions from a set of images with different emotions. The negative side of this approach is the problem with the disalignment of shape and expression parameters. Sometimes Face Mesh Predictor cannot distinguish those parameters well. So, your final output would not have as preserved identity as possible because of this additional face shape information introduced in driving 3DMMs. To tackle this challenge, we generate not one driving expression parameters per emotion but a set of them. We preprocess the CFEE [Du, Tao, and Martinez, 2014] dataset with labeled emotions and save corresponding 3DMM parameters into a small database. During inference, we find the most similar shape parameters to our image and use their expression parameters to get driving 3DMMs

3.3.2 Deploy using Triton

The proposed method is further integrated into the Triton Inference Server to test its throughput and efficiency. Triton is a framework from Nvidia that makes it possible to run different types of ML models and process as many requests as possible by the provided hardware. It supports different backends:

- PyTorch backend - to run PyTorch models
- Python backend - custom Python functions
- ONNX backend - ONNX models
- Tensorflow backend - Tensorflow models

In addition, it has built-in features to handle concurrent requests, run multiple instances of the same model, model management, efficient GPU utilization, and more. To deploy our model on Triton, we split our model into the following parts inside the model registry:

- **Face Mesh Predictor Prepossessing Model** (Python Backend) - model, which resizes the initial image.
- **PyTorch Face Mesh Predictor Model** (PyTorch Backend) - the traced PyTorch model itself which outputs 3DMM parameters.
- **Face Mesh Predictor Postprocessing Model** (Python Backend) - model, which rescale 3DMM parameters.
- **Face Mesh Predictor Ensemble Model** (Ensemble Backend) - a particular type of Triton model that describes how multiple models can interact. We configure all our models above into one pipeline and use it in the future end-to-end, extracting 3DMM parameters from an arbitrary image.
- **PyTorch Face Reenactment Model** (PyTorch Backend) - the traced PyTorch model itself which outputs reenacted images.
- **Face Reenactment Model** (Python Backend) - model works end-to-end and outputs reenacted images from a single photo. This model calls a Face Mesh Predictor Ensemble Model and PyTorch Face Reenactment Model using Business Logic Scripting to get 3DMM parameters and finally reenacted images.

With the following setup, we make our model dockerizable and achieve impressive results on reenacting side:

Our model can generate set of 15 different person emotions and need up to 1.1 second to handle one request. Here is an example of final results:

Concurrency	Throughput (infer/sec)	Latency (sec)
1 Request	0.8	1.266
2 Requests	0.92	2.163
3 Requests	1	3.053
4 Requests	0.98	4.079
5 Requests	1	5.113
6 Requests	0.98	6.107

TABLE 3.5: Inference speed of reenacted model in end-to-end manner using Triton Inference Server on Nvidia Titan RTX

FIGURE 3.4: Generated emotions of Leonardo Di Caprio.

Chapter 4

Datasets

4.1 Face Mesh Prediction Datasets

4.1.1 NoW ("Not quite in-the-Wild")

This dataset is used as a benchmark for the RingNet [Sanyal et al., 2019] face reconstruction pipeline, containing high-resolution 3D face scans of 100 different subjects and their corresponding images.

FIGURE 4.1: Image samples from [Sanyal et al., 2019] project site.

4.1.2 FaceScape

This large-scale detailed 3D face dataset [Yang et al., 2020] contains 18,760 3D faces taken from 938 people performing 20 different expressions. These images are also taken in laboratory conditions.

FIGURE 4.2: Side-by-side comparison of image and 3D Face model from [Yang et al., 2020] project site.

4.1.3 DAD-3DHeads

It is a newly presented dataset with images and corresponding 3D Meshes taken in the wild. It has 44,898 images collected from various sources.

For each image, they provide 5,023 vertices of the FLAME mesh. This dataset has 39% front, 52%, and 9% atypical poses, where half of them are with expressions. We utilize this dataset the most because of its size, diversity, and accuracy. These photos are labeled with a custom labeling tool that ensures the quality of the annotations.

FIGURE 4.3: Diverse image samples from [Martyniuk et al., 2022] paper.

4.2 Face Reenactment Datasets

4.2.1 Cmbiometrics

It is a dataset with cropped face images from [Nagrani, Albanie, and Zisserman, 2018] researchers based on the VoxCeleb [Nagrani, Chung, and Zisserman, 2017] dataset, which contains nearly 2.4 million images of 1.2k persons on 20.4k scenes. This dataset is used as a baseline for face reenactment tasks because of the many photos per person. The VoxCeleb dataset was constructed by extracting shots from YouTube where celebrities are speaking. The length of each video varies from 4 to 20 seconds. On the negative side, the images' resolution is usually low-quality.

FIGURE 4.4: Samples from Cmbiometrics [Nagrani, Albanie, and Zisserman, 2018] dataset.

4.2.2 Radboud Faces Database (RaFD)

A Radboud Faces Database [Langner et al., 2010] is a collection of pictures of 67 models (both adults and children, males and females) displaying eight emotional expressions. These pictures are taken in laboratory conditions and used for validation because of their high quality.

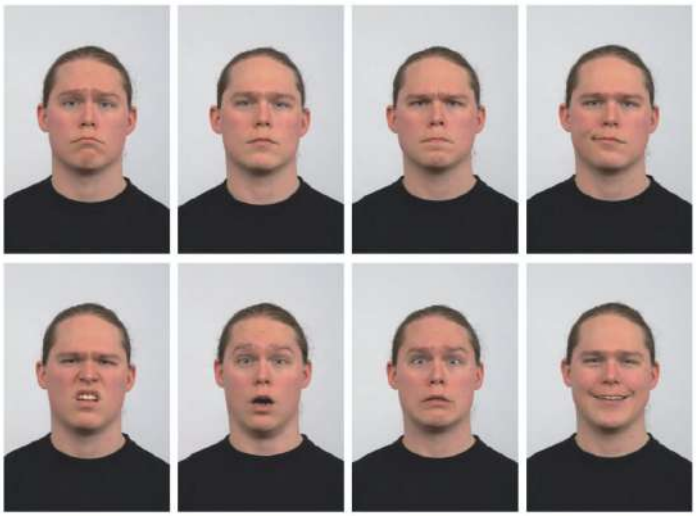

FIGURE 4.5: A model with different emotions from [Langner et al., 2010] paper.

4.2.3 Compound Facial Expressions of Emotions Database (CFEE)

The CFEE [Du, Tao, and Martinez, 2014] dataset contains 5,060 facial images labeled with seven primary emotions and 15 compound emotions for 230 subjects. All images are pictured in laboratory conditions.

FIGURE 4.6: Multiple people with sad emotion from [Du, Tao, and Martinez, 2014] project.

4.2.4 The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

The Ryerson Audio-Visual Database of Emotional Speech and Song [Livingstone and Russo, 2018] is another dataset made in laboratory conditions. It contains 7,356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male). Images include calm, happy, sad, angry, fearful, surprised, and disgusted expressions. Each expression is produced at two levels of emotional intensity (normal and strong), with an additional neutral expression.

FIGURE 4.7: Different people with different emotions from [Livingstone and Russo, 2018] authors.

Chapter 5

Evaluation

5.1 Metrics

Our experiments focus on getting fast and accurate image face reenacting. For these purposes, we compared the rendered image with the ground-truth one using well-known visual metrics:

- **LPIPS** (Learned Perceptual Image Patch Similarity) - this metric compares deep network activations on different images. The authors of this metric [Zhang et al., 2018] found that this approach worked surprisingly well as a perceptual similarity metric, which was true across different network architectures. Also, they slightly improved scores by linearly "calibrating" networks - adding a linear layer on top of off-the-shelf classification networks. We use variation with AlexNet [Krizhevsky, Sutskever, and Hinton, 2017] model inside in our experiments.

- **CSIM** (Cosime Similarity) - this is another model-based metric that measures the dot-product of face embeddings. We use the already pre-trained RetinaFace [Deng et al., 2019] model to get those embeddings. When predicted, the score is high, and target faces look similar and return similar embeddings.

- **FID** (Frechet Inception Distance) - is a metric that calculates the distance between feature vectors calculated for predicted and target images. Lower scores indicate that the two groups of images are more similar, with a perfect score of 0.0, meaning that the two are identical.

- **PSNR** (Peak Signal-to-noise Ratio) - is the ratio between the maximum possible value of a signal and the power of distorting noise that affects the quality of its representation. The main difference between **MSE** and **PSNR** that it measures not *absolute error*, but tries to incorporate luminance and contrast terms inside.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{5.1}$$

  where I, K are original and corrupted images and MAX_I - maximum value on image. For RGB images used sum of $PSNR$ across all channels.

- **SSIM** (Structure Similarity) - is a perceptual metric that quantifies the image quality degradation that is caused by processing such as data compression or by losses in data transmission.

$$SSIM(x, y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{5.2}$$

where μ_x the average over x, μ_y the average over y, σ_x the variance over x, σ_y the variance over y, σ_{xy} the covariance of x and y, c_1, c_2 - constants to prevent division by zero.

5.2 Benchmark

We contact the authors of [Doukas, Zafeiriou, and Sharmanska, 2020] to get a benchmark protocol. We reenacted every video in the VoxCeleb [Nagrani, Chung, and Zisserman, 2017] test set. Firstly, we use the first video frame as a source image and all subsequent frames as driving photos for reenactment independently. After that, we calculate the metric for each pair of generated and driving image and average them to get the final metric.

FIGURE 5.1: Video sample from Voxceleb [Nagrani, Chung, and Zisserman, 2017] test set.

For visual quality comparison, we randomly sample images from VoxCeleb [Nagrani, Chung, and Zisserman, 2017] and generate a table of those images to see the differences in approaches. We use the same person images as a source and drive photos for self-reenactment, and images of different identities for cross-reenactment. Another crucial part is that we use segmented images only because our focus is on face generation. Because of that, during evaluation, all approach results are segmented.

For visual judgment, we generate a set of images with the same person and different emotions. To achieve this, we randomly select faces from a collection of pictures with neutral emotions from the CFEE [Du, Tao, and Martinez, 2014] dataset and generate multiple reenacted images of them with different expressions.

Chapter 6

Experiments

6.1 Experiment Setup

All proposed networks were trained on 3 NVIDIA RTX A6000 GPUs with 48Gb of VRAM and 64 CPU cores. We pre-train the Face Mesh Predictor model during the first stage using batch size 64 and training for 200 epochs. With this pre-trained model, we preprocess all our face reenactment datasets. It takes 2 weeks to train the final Face Reenactment model with batch size 8 on each GPU. The primary validation metrics were SSIM and PSNR during the training. Another important part was visual validation because those metrics do not always show the real quality of the models.

6.2 Dataset Preperation

6.2.1 Preprocessing steps

A significant limitation of a big part of methods is the usage of easy but limited preprocessing steps from a [Siarohin et al., 2020] paper. Firstly, the long hair with this approach might be cropped. Secondly, 256x256 and all other rectangular sizes are not very efficient for storing human faces with long hair. Finally, 256x256 is not enough to preserve the quality applicable in real-life projects. To tackle those problems, we decided to increase the training image size to 512x384. This size stores more useful information inside and has higher image quality. In this image, the person's head is slightly higher than the center to ensure we can handle long hair and some head accessories. Finally, this image is segmented, preserving all hair, and saved on disk. The same approach was used for rendered PNCC.

6.2.2 Dataset super-resolution

Face reenactment datasets need to be additionally processed. First, 3DMM parameters are calculated and generated PNCC based on these parameters. Secondly, the segmented head from the image removes the background and makes training more stable. PNCCs are rendered using C++ code and Python wrapper with slight modifications from [Guo et al., 2020; Guo, Zhu, and Lei, 2018]. After these steps, an additional step was applied - super-resolution of all images using Wang et al., 2021. This step increases the quality of generated images without additional artifacts because current versions of academic datasets have low-resolution images, so our approach benefits more from this.

FIGURE 6.1: Example of super resolution on dataset.

6.3 Results

To evaluate our results we compare them to [Siarohin et al., 2020] (FOMM), [Wang, Mallya, and Liu, 2020] (Head Synthesis) and [Zakharov et al., 2020] (Bilayer) models.

Model	PSNR	SSIM	LPIPS	FID	CSIM
BiLayer	14.009	0.571	0.311	39.146	0.412
Head Synthesis	22.955	0.779	0.136	14.382	0.523
FOMM	23.615	0.797	0.139	14.541	0.583
Ours	**24.077**	**0.803**	**0.118**	**12.655**	**0.598**

TABLE 6.1: Comparison of different method results on VoxCeleb [Nagrani, Chung, and Zisserman, 2017] dataset.

It is clear from the table that BiLayer has the lowest results compared to all other models. In contrast, our model is comparable to FOMM and HeadSynthesys, with only slightly better results. Our method mainly focused on 3DMM manipulation and has no significant advantages in reenacting an image task. Although, our approach provides excellent visual results on emotion manipulation tasks.

For better visual understanding, we provide two large tables with a comparison of self-reenactment (source and driving images have the same person) and cross-reenactment (source and driving images have a different people) tasks. In the last figure of this section, we provide pictures of people with generated emotions from a single neutral photo - the primary purpose of our algorithm. To summarize, we achieved a high emotion manipulation quality and optimized our end-to-end approach.

FIGURE 6.2: Visual comparison of methods on self-reenactment task on Cmbiometrics dataset.

FIGURE 6.3: Visual comparison of methods on cross-reenactment task on Cmbiometrics dataset.

FIGURE 6.4: Images of people with generated emotions on CFEE. dataset.

6.4 Ablation Studies

Experiment	SSIM	PSNR	FID	LPIPS
- perceptual loss	0.8428	26.073	7.401	0.0746
default	0.8518	26.678	8.17	0.0632
+ 3DMM params injection	0.8588	26.974	7.309	0.0603
+ PNCC conditioning	0.8653	27.266	7.008	0.0583
+ mouth discriminator	0.8634	27.295	5.328	0.0581
+ residual connections	0.8642	27.24	6.526	0.0585

TABLE 6.2: Ablation studies for our method.

We perform multiple experiments with minor changes to validate our hypotheses to understand how those tweaks influence the final results. Our ablation studies could be described as follows:

- Default setup without Perceptual Loss
- Default setup
- Default setup with additional Mouth Discriminator
- Default setup with additional PNCC Conditioning inside losses
- Default setup with residual connections between warped image and enhanced output
- Default setup with additional 3DMM parameters injection into a model

From a table, we see that all our changes increase the scores, so all of them we add as a contribution. As an additional experiment, we validate that Perceptual VGG Loss indeed improves the quality of predictions.

Chapter 7

Conclusions

7.1 Result Summary

This work presents a novel two-stage algorithm to solve the problem of face reenactment. We increase the overall speed of the proposed method by avoiding time-consuming direct optimization of 3DMM parameters from 2D landmarks. To our knowledge, we are the first to combine Face Reenactment and Face Mesh Prediction task and inference model in an end-to-end manner using 3DMM parameters. The essential benefit of these parameters is easy-to-use face manipulation and no need for additional steps, like face boundary generation or keypoints reprojection. In addition, we managed to increase the quality of rendered images through multi-step data preprocessing and architecture tweaks (residual connections, proper image size). Finally, we demonstrate the computational efficiency of the proposed method, making it applicable for real-world applications.

7.2 Points to improve

The next major things which could be improved are:

- Experimenting with model replacing it with StyleGAN-like architectures
- Gathering a dataset with higher quality images
- Proposing benchmark in a face reenactment field

Bibliography

Blanz, Volker and Thomas Vetter (1999). "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 187–194. ISBN: 0201485605. DOI: 10.1145/311535.311556. URL: https://doi.org/10.1145/311535.311556.

Booth, James et al. (2018a). "3D Reconstruction of "In-the-Wild" Faces in Images and Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.11, pp. 2638–2652. DOI: 10.1109/TPAMI.2018.2832138.

Booth, James et al. (Apr. 2018b). "Large Scale 3D Morphable Models". In: *Int. J. Comput. Vision* 126.2–4, 233–254. ISSN: 0920-5691. DOI: 10.1007/s11263-017-1009-7. URL: https://doi.org/10.1007/s11263-017-1009-7.

Bounareli, Stella, Vasileios Argyriou, and Georgios Tzimiropoulos (2022). "Finding Directions in GAN's Latent Space for Neural Face Reenactment". In: *arXiv*. eprint: 2202.00046.

Bregler, Christoph, Michele Covell, and Malcolm Slaney (1997). "Video Rewrite: Driving Visual Speech with Audio". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 353–360. ISBN: 0897918967. DOI: 10.1145/258734.258880. URL: https://doi.org/10.1145/258734.258880.

Cao, Meng et al. (2021). "UniFaceGAN: A Unified Framework for Temporally Consistent Facial Video Editing". In: *CoRR* abs/2108.05650. arXiv: 2108.05650. URL: https://arxiv.org/abs/2108.05650.

Chang, Yao-Jen and Tony Ezzat (Jan. 2005). "Transferable video-realistic speech animation". In: pp. 143–151. DOI: 10.1145/1073368.1073388.

Chen, Lele et al. (2020). "Talking-head Generation with Rhythmic Head Motion". In: *ECCV*.

Deng, Jiankang et al. (2019). "RetinaFace: Single-stage Dense Face Localisation in the Wild". In: *CoRR* abs/1905.00641. arXiv: 1905.00641. URL: http://arxiv.org/abs/1905.00641.

Doukas, Michail Christos, Stefanos Zafeiriou, and Viktoriia Sharmanska (2020). "HeadGAN: Video-and-Audio-Driven Talking Head Synthesis". In: *CoRR* abs/2012.08261. arXiv: 2012.08261. URL: https://arxiv.org/abs/2012.08261.

Doukas, Michail Christos et al. (2020). "Head2Head++: Deep Facial Attributes Re-Targeting". In: *CoRR* abs/2006.10199. arXiv: 2006.10199. URL: https://arxiv.org/abs/2006.10199.

Du, Shichuan, Yong Tao, and Aleix M. Martinez (2014). "Compound facial expressions of emotion". In: *Proceedings of the National Academy of Sciences* 111.15, E1454–E1462. DOI: 10.1073/pnas.1322355111. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1322355111. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1322355111.

Garrido, P. et al. (May 2015). "VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track". In: *Comput. Graph. Forum* 34.2,

193–204. ISSN: 0167-7055. DOI: 10.1111/cgf.12552. URL: https://doi.org/10.1111/cgf.12552.

Ghosh, Partha et al. (2020). "GIF: Generative Interpretable Faces". In: *CoRR* abs/2009.00149. arXiv: 2009.00149. URL: https://arxiv.org/abs/2009.00149.

Goodfellow, Ian J. et al. (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].

Guo, Jianzhu, Xiangyu Zhu, and Zhen Lei (2018). *3DDFA*. https://github.com/cleardusk/3DDFA.

Guo, Jianzhu et al. (2020). "Towards Fast, Accurate and Stable 3D Dense Face Alignment". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Huang, Xun and Serge J. Belongie (2017). "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: *CoRR* abs/1703.06868. arXiv: 1703.06868. URL: http://arxiv.org/abs/1703.06868.

K R, Prajwal et al. (2019). "Towards Automatic Face-to-Face Translation". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: Association for Computing Machinery, 1428–1436. ISBN: 9781450368896. DOI: 10.1145/3343031.3351066. URL: https://doi.org/10.1145/3343031.3351066.

Karras, Tero, Samuli Laine, and Timo Aila (2018). "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *CoRR* abs/1812.04948. arXiv: 1812.04948. URL: http://arxiv.org/abs/1812.04948.

Kim, Hyeongwoo et al. (2018). "Deep Video Portraits". In: *ACM Transactions on Graphics (TOG)* 37.4, p. 163.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (May 2017). "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6, pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386.

Langner, Oliver et al. (Dec. 2010). "Presentation and validation of the Radboud Face Database". In: *Cognition Emotion - COGNITION EMOTION* 24, pp. 1377–1388. DOI: 10.1080/02699930903485076.

Li, Tianye et al. (2017). "Learning a model of facial shape and expression from 4D scans". In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36.6, 194:1–194:17. URL: https://doi.org/10.1145/3130800.3130813.

Liu, Kang and Joern Ostermann (2011). "Realistic facial expression synthesis for an image-based talking head". In: *2011 IEEE International Conference on Multimedia and Expo*, pp. 1–6. DOI: 10.1109/ICME.2011.6011835.

Livingstone, Steven R. and Frank A. Russo (May 2018). "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PLOS ONE* 13.5, pp. 1–35. DOI: 10.1371/journal.pone.0196391. URL: https://doi.org/10.1371/journal.pone.0196391.

Martyniuk, Tetiana et al. (2022). "DAD-3DHeads: A Large-scale Dense, Accurate and Diverse Dataset for 3D Head Alignment from a Single Image". In: *arXiv*. eprint: 2204.03688.

Nagrani, Arsha, Samuel Albanie, and Andrew Zisserman (2018). "Seeing Voices and Hearing Faces: Cross-modal biometric matching". In: *CoRR* abs/1804.00326. arXiv: 1804.00326. URL: http://arxiv.org/abs/1804.00326.

Nagrani, Arsha, Joon Son Chung, and Andrew Zisserman (2017). "VoxCeleb: a large-scale speaker identification dataset". In: *CoRR* abs/1706.08612. arXiv: 1706.08612. URL: http://arxiv.org/abs/1706.08612.

Park, Taesung et al. (2019). "Semantic Image Synthesis with Spatially-Adaptive Normalization". In: *CoRR* abs/1903.07291. arXiv: 1903.07291. URL: http://arxiv.org/abs/1903.07291.

Ren, Yurui et al. (2021). "PIRenderer: Controllable Portrait Image Generation via Semantic Neural Rendering". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 00, pp. 13739–13748. DOI: 10.1109/iccv48922.2021.01350.

Sanyal, Soubhik et al. (June 2019). "Learning to Regress 3D Face Shape and Expression from an Image without 3D Supervision". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7763–7772.

Siarohin, Aliaksandr et al. (2020). "First Order Motion Model for Image Animation". In: *CoRR* abs/2003.00196. arXiv: 2003.00196. URL: https://arxiv.org/abs/2003.00196.

Song, Yang et al. (2018). "Talking Face Generation by Conditional Recurrent Adversarial Network". In: *CoRR* abs/1804.04786. arXiv: 1804.04786. URL: http://arxiv.org/abs/1804.04786.

Suwajanakorn, Supasorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman (July 2017). "Synthesizing Obama: Learning Lip Sync from Audio". In: 36.4. ISSN: 0730-0301. DOI: 10.1145/3072959.3073640. URL: https://doi.org/10.1145/3072959.3073640.

Tan, Mingxing, Ruoming Pang, and Quoc V. Le (2019). *EfficientDet: Scalable and Efficient Object Detection*. arXiv: 1911.09070 [cs.CV].

Tewari, Ayush et al. (2020). "StyleRig: Rigging StyleGAN for 3D Control over Portrait Images". In: *CoRR* abs/2004.00121. arXiv: 2004.00121. URL: https://arxiv.org/abs/2004.00121.

Thies, Justus et al. (2020). "Face2Face: Real-time Face Capture and Reenactment of RGB Videos". In: *CoRR* abs/2007.14808. arXiv: 2007.14808. URL: https://arxiv.org/abs/2007.14808.

Wang, Ting-Chun, Arun Mallya, and Ming-Yu Liu (2020). "One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing". In: *CoRR* abs/2011.15126. arXiv: 2011.15126. URL: https://arxiv.org/abs/2011.15126.

Wang, Ting-Chun et al. (2019). "Few-shot Video-to-Video Synthesis". In: *CoRR* abs/1910.12713. arXiv: 1910.12713. URL: http://arxiv.org/abs/1910.12713.

Wang, Xintao et al. (2021). "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data". In: *International Conference on Computer Vision Workshops (ICCVW)*.

Wiles, Olivia, A. Sophia Koepke, and Andrew Zisserman (2018). "X2Face: A network for controlling face generation by using images, audio, and pose codes". In: *CoRR* abs/1807.10550. arXiv: 1807.10550. URL: http://arxiv.org/abs/1807.10550.

Wu, Wayne et al. (2018). "ReenactGAN: Learning to Reenact Faces via Boundary Transfer". In: *CoRR* abs/1807.11079. arXiv: 1807.11079. URL: http://arxiv.org/abs/1807.11079.

Yang, Haotian et al. (2020). "FaceScape: a Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction". In: *CoRR* abs/2003.13989. arXiv: 2003.13989. URL: https://arxiv.org/abs/2003.13989.

Yao, Guangming et al. (2021). "One-shot Face Reenactment Using Appearance Adaptive Normalization". In: *CoRR* abs/2102.03984. arXiv: 2102.03984. URL: https://arxiv.org/abs/2102.03984.

Yi, Ran et al. (2020). "Audio-driven Talking Face Video Generation with Natural Head Pose". In: *CoRR* abs/2002.10137. arXiv: 2002.10137. URL: https://arxiv.org/abs/2002.10137.

Yin, Fei et al. (2022). "StyleHEAT: One-Shot High-Resolution Editable Talking Face Generation via Pre-trained StyleGAN". In: *arXiv*. eprint: 2203.04036.

Zakharov, Egor et al. (2019). "Few-Shot Adversarial Learning of Realistic Neural Talking Head Models". In: *CoRR* abs/1905.08233. arXiv: 1905.08233. URL: http://arxiv.org/abs/1905.08233.

Zakharov, Egor et al. (2020). "Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars". In: *CoRR* abs/2008.10174. arXiv: 2008.10174. URL: https://arxiv.org/abs/2008.10174.

Zhang, Richard et al. (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR*.

Zhang, Zhimeng et al. (2021). "Flow-guided One-shot Talking Face Generation with a High-resolution Audio-visual Dataset". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 00, pp. 3660–3669. DOI: 10.1109/cvpr46437.2021.00366.

Zhou, Hang et al. (2019). "Talking face generation by adversarially disentangled audio-visual representation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 9299–9306.

Zhu, Jun-Yan et al. (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *CoRR* abs/1703.10593. arXiv: 1703.10593. URL: http://arxiv.org/abs/1703.10593.