UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Face reenactment with GANs using landmark representation of a face

Author:
Ivan KOSAREVYCH

Supervisor:
Volodymyr KARPIV

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

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2020

Declaration of Authorship

I, Ivan KOSAREVYCH, declare that this thesis titled, “Face reenactment with GANs using landmark representation of a face” 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:

"Generative Adversarial Networks is the most interesting idea in the last ten years in machine learning"

Yann LeCun, Director at Facebook AI

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Face reenactment with GANs using landmark representation of a face

by Ivan KOSAREVYCH

Abstract

Face reenactment is an emerging technology that attracts high interest in recent years. It aims at generating face with the identity of one person (known as target) and facial expression from another (source). Many existing methods are limited to reenact a predefined personality of either source or target. In this study, we present the approach that is agnostic to the identity of source and target and observes only a single image of each of them. Our method is based on recently introduced Generative adversarial networks (GANs). We experimentally find a proper GAN loss for our system. An accurate expression transfer from a source person is essential for face reenactment. In this study, we examine different approaches to achieve it and design a landmark loss function based on our novel landmark detector.

Acknowledgements

Firstly, I am very grateful to my supervisor Volodymyr Karpiv, who was opened to a discussion even very late in the evening, provided me with interesting ideas, and encouraged me throughout the work. Then, I am thankful to Marian Petruk, who has run related research, which highly supported me with mine. I express my gratitude to SoftServe R&D team and Mykola Maksymenko, in particular, for the big support. I am thankful to those 32 people who found time to participate in my user study, which gave me valuable information. Finally, I want to thank Oles Dobosevych, who was always opened to questions.

Contents

List of Figures

List of Tables

List of Abbreviations

GAN	**G**enerative-**A**dversarial **N**etwork
CNN	**C**onvolutional-**N**eural **N**etwork
FR	**F**ace **R**eenactment
MLE	**M**aximum-**L**ikelihood **E**stimation

To my family

Chapter 1

Introduction

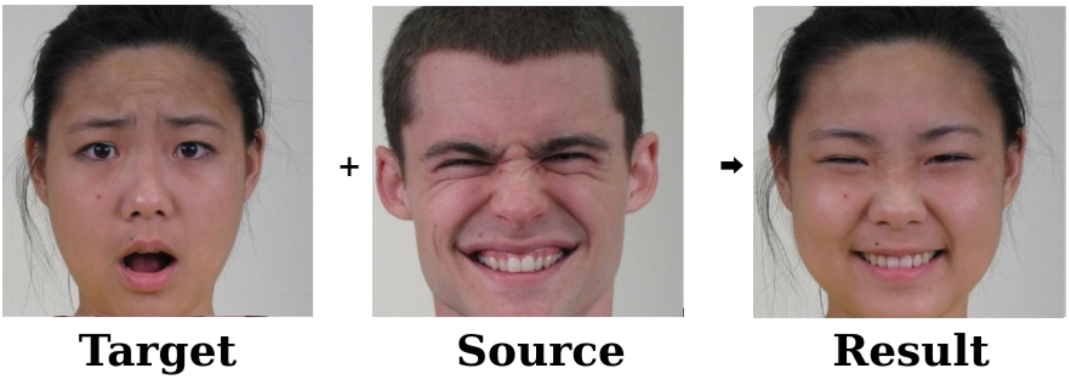


FIGURE 1.1: Example of face reenactment generated with our model

Interest in digital image processing has increased exponentially in the last decades. It finds applications in the industries, where a photograph is an essential piece, such as entertainment, multimedia systems. Apart from these, it is used in every industry, where a picture can bring any additional information, such as agriculture, medicine, automotive, security systems, to name just a few. The most common subareas of digital image processing include image analysis, image compression, image restoration, and image enhancement.

Image enhancement incorporates techniques of modifying images so that the viewer can extract useful information out of it. An image can be seen as a function of two variables. However, in processing time, it is represented as a matrix of integer numbers. Therefore different matrix operations come into a hand for image manipulations, including enhancement. The matrices we apply to images in order to get the desired result can be figured out in two ways: manual and automatic. They do not give the same results. The manual way is rigorous. Approaches that are used to calculate matrices manually produce the precise result. Automatic approaches exploit neural networks that give an approximate result. Depending on the problem complexity, either manual or automatic approach is used. Usually, if it is time-consuming to figure out the solution manually, neural networks come into hand.

Neural networks have developed significantly in recent years. In many image processing problems, such as image classification, segmentation, enhancement, reconstruction, face recognition, they are new state-of-the-art. They can model complex distributions of data, which makes them applicable to a wide range of nontrivial problems, such as image generation.

Image generation appeared in recent years thanks to the development of neural nets. Studies show the ability to generate faces, transfer style from one image to another, modify an image concerning text description, perform multimodal generation, including image generation from the text, and audio. One of the notable fields of research in recent years appeared to be face expression transfer from one person (source) to another (target) known in mass-culture as "Deepfakes". This term encompasses different variations regarding motions that are transferred, individual attributes that are preserved. All the variations have two common attributes: a person, from which some motion is extracted, and a person, that receives that extracted motion. It is not necessary should be a person, as shown in Siarohin et al., 2018. Motion transfer can be applied to the full body or a single face. The face of a source without any modification can be inpainted in-place of the target face, which is called face swap (FS). On the contrary to FS, only face expression can be transferred onto a target, preserving identity characteristics of the later.

Recently a vast amount of works provided their approaches in face reenactment (Ha et al., 2019; Zakharov et al., 2019; Zhang et al., 2019; Nirkin, Keller, and Hassner, 2019; Wu et al., 2018; Pumarola et al., 2018; Tripathy, Kannala, and Rahtu, 2019; Thies et al., 2016 to cite a few). A solid overview is provided in Sec. 2.3. We propose our solution to the aforementioned problem using GANs and examine several sides of it as well as tackle a related problem of facial landmark detection. In Chapter 2 we provide our overview of studies on face reenactment and its "parent" area – Image-to-image translation (Sec. 2.2), Generative adversarial nets (Sec. (2.1). In Chapter 3, we define the problem of face reenactment and related to it, that we study in this work. In Chapter 4, we describe our GAN-based (Goodfellow et al., 2014) approaches to face reenactment with landmark representation of a face and the problem of landmark detection. We propose a landmark detector based on U-Net (Ronneberger, Fischer, and Brox, 2015) and landmark loss for face reenactment, that uses this detector. In Chapter 5, we describe experiments that we conduct, providing information about metrics for quantitative comparison, datasets, and some details of program implementation. We compare our landmark detector with a DLib detector. In Chapter 6, we show the advantages of our landmark detector as well as exciting findings for face reenactment using GANs. Finally, in Chapter 7 we summarize our work and provide the next steps of this study.

Chapter 2

Literature overview

2.1 Generative Adversarial Networks

Generative adversarial networks (Goodfellow et al., 2014) are an example of generative models. Term *generative model* refers to any model (generator) that takes a training set, consisting of samples drawn from a distribution p_{data}, and learns to represent an estimate of that distribution somehow. The result is a probability distribution p_{model}. (Goodfellow, 2016). As an example, generative classifiers learn a model of joint probability $p(x, y)$ of the inputs x and labels y and following Bayes rule calculate $p(y|x)$. Generative models tend to model data distribution explicitly. In comparison, discriminative classifiers learn a model of conditional probability $p(x|y)$. They find a boundary line between classes rather than reproduce the original distribution of data. Higher simplicity in such estimation compared to maximum likelihood estimation in generative models alongside with backpropagation and dropout algorithms has made discriminative models superior in many tasks.

The main difficulty of generative models is to approximate many intractable probabilistic computations that are connected to MLE 'inside' of the Generator network. GANs tackle this problem. They introduce another network called Discriminator, that is responsible for classifying real and fake samples. Generator $G(z, \theta_g)$ produces fake samples from prior noise z, which is commonly sampled from Gaussian or Uniform distribution. Discriminator $D(x, \theta_d)$ outputs a single scalar $D(x)$ – probability that x came from real data. $D(x, \theta_d)$ is trained to maximize (x), while $G(z, \theta_g)$ tends to minimize $\log(1 - D(G(z)))$.

In terms of Game Theory Generator and Discriminator are playing two-player minimax game aiming at reaching Nash equilibrium. The equilibrium is the state, in which no player will not benefit from changing his strategy, knowing strategies of his rivals. According to Nash's Existence Theorem, if mixed strategies are allowed, then every game with a finite number of players and strategies has at least one Nash equilibrium (Nash, 1951). Such equilibrium exists for GANs. GAN objective is as follows:

$$min_D max_G V(D, G) = \mathrm{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathrm{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.1}$$

Generator and Discriminator are trained simultaneously. Gradients from Discriminator flow to the Generator, updating its weights. Throughout the training, Generator and Discriminator should be approximately equal. Meaning none of these players should significantly outperform the other one considering loss function. Speaking in terms of game theory, they should be in equilibrium. Generator and Discriminator have separate optimizers. Usually, these optimizers do their steps simultaneously, which is more computationally prohibitive.

GANs should be capable of generating some amount of modes. However, control over these modes had not been present until Mirza and Osindero, 2014 proposed conditional GANs, that appeared quite recently after the original paper of GANs. Conditional setting modifies GAN objective to appear as follows:

$$min_D max_G V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathrm{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{2.2}$$

where y is some property we want to control (e.g. in MNIST dataset (LeCun and Cortes, 2010) such property would be some number). Such conditioning allows to generate samples of specific mode in controllable manner. Conditional GANs are now applied to a wide range of problems and are not limited to specific tasks (*e.g.* Isola et al., 2016).

2.1.1 Improved techniques for training GANs

Training GANs to completion is a non-trivial task. Salimans et al., 2016 in their work on improving GANs show major problems typical for GANs and tackle them. Firstly, SGD optimizers (Robbins and Monro, 1951) find a minimum of the loss function rather than equilibrium. When these algorithms are used to seek equilibrium, they may fail to converge. Minimizing the error of Generator may usually increase error in Discriminator and vice versa. The absence of coordination between gradients of Generator and Discriminator makes persuading both algorithms to converge simultaneously a complex task. Secondly, mode collapse is a well-known problem of the GAN training procedure. It means that instead of producing different modes (e.g. different numbers in MNIST dataset LeCun and Cortes, 2010) Generator produces only a small number of them. After the collapse happened, the Discriminator can not distinguish real examples from fake ones. Thus gradients that flow to Generator are small. The training process should be started from the beginning.

Salimans *et al.* propose several solutions to stabilize training and improve convergence. They show that matching feature vectors of some intermediate layers of Discriminator instead of matching probability scalars in the Generator objective prevents Generator from overtraining on the current Discriminator. New Generator objective is defined as follows:

$$C(G) = \mathrm{E}_{x\sim p_{data}(x)} f(x) - \mathrm{E}_{z\sim p_z(z)} f(G(z)) \tag{2.3}$$

Another improvement proposed by Salimans *et al.* tackles mode collapse problem. They make a Discriminator look at several examples of data in combination rather than isolation. They provide a specific algorithm for such a combination. The main idea is in calculating the similarity of samples produced by Generator. Minibatch discrimination allows to generate perceptually sharp images very quickly, and regarding this, it is superior to feature matching (Salimans et al., 2016).

Several minor improvements include one-sided label smoothing (prevents Discriminator from overconfidence), historical averaging (improves seek for equilibria), and virtual batch normalization (to minimize dependencies in a batch of samples). Finally, they propose a new metric for measuring image realism called Inception score.

2.2 Image-to-Image Translation

Face-reenactment may be interpreted as an image-to-image translation problem. The image-to-image translation is a class of computer vision problems, where the main goal is to learn transition between two or more domains. Proposed solutions involve deep learning methods, specifically, generative adversarial networks.

Recent studies such as Isola et al., 2016 propose methods for translating images to different domains in a supervised manner, which requires *corresponding* images from both domains involved in translation. The method in Isola et al., 2016 is unimodal, meaning able to learn the transition only between two predefined domains. Zhu et al., 2017 introduce BicycleGAN, which is capable of multimodal translation.

However, samples of paired images are not usually available. Therefore cycle consistency loss is introduced in CycleGAN (Zhu et al., 2017), DiscoGAN (Kim et al., 2017) and UNIT (Liu, Breuel, and Kautz, 2017). These approaches are limited by unimodality. Unlike them, StarGAN (Choi et al., 2018), MUNIT (Huang et al., 2018) and DRIT (Lee et al., 2018) can handle more than two domains.

Wang et al., 2017 in pix2pixHD and Karras et al., 2017 in ProGAN have introduced a possibility to generate high-resolution images. Their approaches are coarse-to-fine, meaning that high-quality images are synthesized out of low-resolution ones following several stages of improvements. Corresponding generators and discriminators on each stage are responsible for increasing image resolution.

These approaches have caught the eye by translating semantic maps into city views Isola et al., 2016, clothes(Lassner, Pons-Moll, and Gehler, 2017), nature(Park et al., 2019), dance (Wang et al., 2019), synthesizing between paintings and photographs, chairs and cars (Kim et al., 2017) and transferring objects from one material to another (Yi et al., 2017).

While these methods can generate plausible images, they are limited in the possibility of providing stable face reenactment.

2.3 Face Reenactment

This field of study has emerged in the recent years and is not yet well studied. Face reenactment aims at transferring of facial expression from source person to target one, preserving target identity. The facial expression, captured from the source, works as a driver for the target person. The initial expression of the target person is modified to be similar to the captured expression. Such transfer should not modify the facial geometry and colors of the target person. Moreover, the image background should not be changed. Concluding, face-reenactment has the following main objectives: 1) facial expression transfer; 2) identity preservation; 3) background and illumination retention. Apart from these, others may be considered, such as head pose or eye gaze transfer Kim et al., 2018; Thies et al., 2016; Thies et al., 2015.

Face reenactment may be conducted in different scenarios. The most simple is a one-to-one scenario when we map one person into himself with a different expression. Formally speaking, source and target identities are the same in this case. The one-to-many scenario is mostly well-studied in recent worksThies et al., 2016; Wu et al., 2018; Zakharov et al., 2019; Tripathy, Kannala, and Rahtu, 2019; Pumarola et al., 2018. In this scenario, either target person (Thies et al., 2016; Wu et al., 2018; Zakharov et al., 2019; Wang et al., 2017) or source person (Tripathy, Kannala, and Rahtu, 2019; Pumarola et al., 2018) is predefined, while another one may be arbitrary. Such an approach is not quite challenging and has plenty of real-world applications, such as cinematography, media, video games, to name just a few. Nevertheless, the most challenging one is a many-to-many scenario, when the source and target persons are entirely arbitrary and previously unseen by the model. Most recent works show optimistic results (Kosarevych et al., 2020; Nirkin, Keller, and Hassner, 2019; Zhang et al., 2019) in images as well as in video sequences.

Similar to reenactment is face-swapping. In the face-swapping identity of a target person is not preserved, meaning the face of the target is completely or partially replaced with the source person's face. The most common is a partial replacement. Usually, the skin color of the target person is not changed. However, eyes, eyebrows, nose, and mouth are those of source person. The focus of this study is a face-reenactment problem.

History of face swapping and reenactment goes back for as long as two decades. Such approaches were needed mostly to resolve privacy issues (Blanz et al., 2004). In the next sections, we discuss different approaches to face modeling and corresponding face modification methods.

2.3.1 3D based approaches

Initially, the way to perform face manipulation given an image was fitting 3D morphable face model (3DMM)in a supervised manner and then adjusting estimated parameters (shown in Blanz and Vetter, 2002). That was a starting point for later approaches. They used more information than a single image were able to learn high-level details or inferred 3DMM parameters directly from RGB data without a need in labels.

Some early approaches implied manual involvement (Blanz et al., 2004; Vlasic et al., 2005) Not a long time after automated methods were proposed in Bitouk et al., 2008. However, they all were face-swapping methods, and just recently reenactment approach was introduced by Thies et al., 2016. They fit the 3DMM face model to both source and target and apply expression parts from source person to target one. They can run reenactment in real-time on the source-target video sequence. However, their approach suffers from strong visual artifacts in generating teeth. They select an output frame from the target video sequence, which limits scalability and may lead to inaccurate expressions.

2.3.2 Landmark-based approaches

Landmark-based approaches for FR come with deep learning methods. Deep learning techniques, more specifically – GANs by Goodfellow et al., 2014, improved reenactment results significantly, tackled limitations of hand-crafted techniques (Wu et al., 2018; Zakharov et al., 2019).

Studies on FR exploit recent advances in GANs architecture. For example, Wu et al., 2018 applied cycle consistency loss from CycleGAN (Zhu et al., 2017) to ensure correct expression transfer. They first proposed to use landmarks latent space as a medium between source and target, which is their main contribution. Their system works as follows. Firstly, the source is mapped into the latent space. Then these landmarks are adapted for target, and the synthesized image is reconstructed out of them. Such an approach allows accurately obtain face boundaries under severe poses, diverse expressions, and extreme lighting conditions. Some other shining works include Zakharov et al., 2019; Zhang et al., 2019; Siarohin et al., 2018; Ha et al., 2019; Siarohin et al., 2019.

Chapter 3

Problem formulation

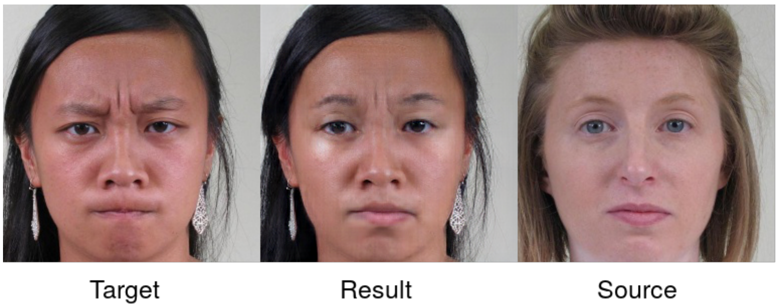


FIGURE 3.1: Example of face reenactment generated with our model

Image editing is a broad field of problems that continually develops. It includes classic problems, such as cropping, noise reduction, colorization, image sharpening, brightening, warping. Most of these problems are solved by applying to an image some filter (*i.e.* convolution) with specific kernel relevant to the problem. Recent problems that include image deblurring, dehazing, image style transfer, face-swapping are more complex. Rule-based approaches often fail here because of a high number of parameters that should be derived manually. On the contrary, neural networks contain a large number of parameters, that model learns to find a solution to the problem approximately.

The emergence of Deep Learning frameworks, such as CNNs and GANs, provided tools for solving practical problems and increased quality. It has become possible to solve problems, which involve non-trivial manipulations with an image. Converting night to day on the image, face generation, face reenactment, image-to-image translation are among them. These problems are getting more attention in the last years. It is caused by the intensive development of generative models, mainly - Generative adversarial networks. GANs are the most powerful among all the generative models. They provide the most visually pleasing results when combined with recent advances in CNNs and rule-based approaches in Computer Vision.

In the scope of this study, we rely on GANs as a primary approach. There exist numerous variations of GAN objective. We study their influence on the quality of face reenactment that we describe below.

Assume we have two images Image 1 and Image 2 with one person on each Person 1 and Person 2. These persons can have a different identity or the same. More important is that they have different facial expressions (e.g smile, amazement) Expression 1 and Expression 2. Now we apply some function to these images, which modifies Person 1 to contain Expression 2 instead of Expression 1. Meanwhile, other properties, such as the identity of Person 1, image background, illumination in Image 1 retain. Concluding, face reenactment aims to enhance Image 1 to make Person

1 contain Expression 2. In other words, we transfer Expression 2 onto Person 1, with identity preservation of Person 1 and image properties of Image 1.

Denote Image 1 as Target image, Image 2 as Source image. Observe that target and source persons can have either different or the same identities. Consider a sequence of n pairs of source and target images. For every pair of source and target expressions are different (in practice, they are always slightly different as it is difficult to replicate the same expression every time). Denote source identity as Id_s, target identity as Id_t. In this sequence exists several possible scenarios regarding face identity.

1. *"Many-to-many"* - source and target identities are different, arbitrary. More formally, $\forall k : Id_{s_k} \neq Id_{t_k}$ and $Id_{s_1} \neq Id_{s_2} \neq ... \neq Id_{s_n}$ and $Id_{t_1} \neq Id_{t_2} \neq ... \neq Id_{t_n}$;
2. *"Many-to-one"* - target identities are arbitrary, but source identities are the same. Formally, $\forall k : Id_{s_k} \neq Id_{t_k}$ and $Id_{s_1} = Id_{s_2} = ... = Id_{s_n}$ and $Id_{t_1} \neq Id_{t_2} \neq ... \neq Id_{t_n}$;
3. *"One-to-many"* - in opposite to previous, source identities are arbitrary, but target identities are the same. Formally, $\forall k : Id_{s_k} \neq Id_{t_k}$ and $Id_{s_1} \neq Id_{s_2} \neq ... \neq Id_{s_n}$ and $Id_{t_1} = Id_{t_2} = ... = Id_{t_n}$;
4. *"One-to-one"* - source and target identities are the same. Formally, $\forall k : Id_{s_k} = Id_{t_k}$

Scenario 1 is the most practical, yet the most challenging. The ability to reenact arbitrary person into arbitrary person saves time and resources needed to define M for every identity, which is required in Scenarios 2-4. To find M, that can be applied to arbitrary identities, is highly non-trivial problem. In scope of this study we want to define such M that is able to perform many-to-many face reenactment.

From this definition of FR, we see that FR raises several other intriguing problems. The first of them is expression representation. There are several ways to model facial expression. They are described in more detail in Sec. 2.3. In the scope of this work, we study the landmark-based representation of the face. They allow concentrating on specific parts of the face. They are memory-friendly and require a relatively small amount of computational power. Overall, they are a simple yet powerful way to model facial expression.

In order to achieve accurate expression transfer, we have to penalize our neural network with some error between generated and ground truth faces in terms of facial landmarks. It could be the exact landmarks or some embedding, which incorporates information about them in some form. We examine both ways.

Naturally, there arises a problem of keypoint detection that is comprehensively studied in this work. Accurate keypoint locations are essential for facial reenactment. A good keypoint detector should be steady to occlusions (e.g. beards, masks, makeup), large poses, intense illumination, blur. Moreover, such a detector should be lightweight in terms of memory consumption and number of operations, and fast, because FR is highly potential to be used on edge devices, such as cellphones. In this study, we compare different approaches to keypoint detection in terms of accuracy, robustness, speed, and productivity. More importantly, we investigate their influence on the quality of face reenactment.

Reenactment can be applied to various parts of the human body as well as to the whole body. Expression, head pose, and eye gaze transfer are preferable for reenactment in the head region. Expression transfer is the primary step in the comprehensive study of FR. For the sake of simplicity, we decided to study the reenactment of frontal faces only. Meaning, in the scope of this work, we research FR for expression transfer. Pose transfer or even full-body transfer we leave for the future work.

Summarizing, in this work, we focus on the following directions:

1. Impact of GAN loss on face reenactment

2. Stimulation of accurate expression transfer

3. Face reenactment sensitivity to landmark detection technique

Chapter 4

Method

The crucial aspects of our approach include Generator architecture, Discriminator's behavior, expression generation, face normalization, and identity mismatch calculation, which are described in more detail in the corresponding sections below. We more seriously focus on the approaches that influence the correct expression synthesis of a fake person. We study two methods, namely landmark loss and adversarial loss (*i.e.* provided by Discriminator). For landmark estimation, we show the need for a custom landmark detector and propose the possible solution described in Sec. 4.2.2.

In order to provide stable working of the proposed solution for face reenactment, face normalization is needed , *i.e.* source and target faces should be similarly aligned on the image. The normalization process requires a source and a target to have similar head poses. Therefore we consider only frontal faces.

4.1 Expression penalization

We aim to generate a synthetic person with an expression that is the same as in the source person. Person expression is modeled with landmarks. Therefore we want to minimize the distance between generated landmarks K_g and source landmarks K_s. To this end, we study two approaches, namely, landmark loss and landmark discriminator. The first one directly penalizes distance between K_g and K_s. The idea of landmark discriminator is inspired by Siarohin et al., 2018. It may bring more freedom to Generator in terms of landmark locations without a high loss in accuracy of generated expression. However, this approach is more difficult in training compared to landmark loss.

4.1.1 Landmark loss

Landmark loss is the explicit measure of how landmarks of the generated face differ from landmarks of the source. It is a distance between K_g and K_s in the latent space of landmarks. In order to compute K_g and K_s, we need a differentiable function, because we should be able to compute gradients of this function. With these gradients we penalize Generator. Consecutively, it should become a part of our network graph. Therefore, we design our custom landmark detector described in Section 4.2.2, which is a neural network. The detector predicts 56 heatmaps that correspond to 56 facial landmarks. So K_{gu} is a set of these maps predicted for generated face and K_{su} – for source face. To measure distance between them we apply the function used for training our detector, namely IOU loss. Overall, our landmark loss is as follows:

$$L_{landmark} = \frac{K_{gu} \cap K_{su} + \epsilon}{|K_{gu}| + |S_{su}| - K_{gu} \cup K_{su} + \epsilon} \tag{4.1}$$

4.1.2 Landmark Discriminator

We employ the concept of Discriminator introduced by Goodfellow et al., 2014 to penalize Generator for producing faces with inaccurate facial expression. More precisely, Discriminator is a Critic, proposed by Arjovsky, Chintala, and Bottou, 2017, as it outputs a vector, rather than probability value.

As we are interested in generated face expression to match the expression of source person, we provide source landmarks to the Discriminator in the following way. Input to the Discriminator is either a generated image or source image concatenated with source landmarks. Concatenation is done along the channel axis (*i.e.* we add the fourth channel to RGB image). It allows using these landmarks as key additional information for Discriminator. They force Discriminator to focus on the moving parts of a face (*i.e.* eyes, mouth, eyebrows). Finally, the Discriminator provides a score of similarity between image and landmarks. This score is later used to make Generator provide more accurate expression.

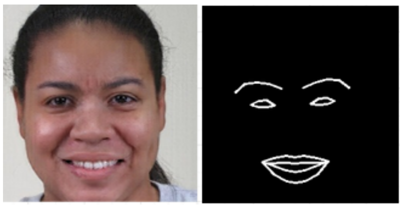


FIGURE 4.1: Schematic representation of Discriminator's input.

4.2 Landmark detection

We model a facial expression with landmarks. Therefore there appears a need to detect them on the image. There already exist several solutions. One of the most common is an image processing library named DLib King, 2009 (details in the corresponding section). We show that DLib has several issues that are critical for successful many-to-many face reenactment. In order to solve them, we propose a more robust landmark detector.

4.2.1 DLib detector

Dlib King, 2009 is a well-known multipurpose library. Among all, it has a valuable set of tools for object detection, including face and landmark detection, face recognition.

Dlib algorithm computes facial landmarks in two steps. Firstly, it finds a bounding box of a face. Then it localizes facial landmarks in the detected bounding box.

Face detection is performed in the following way. The first step is feature extraction from an image using a classic Histogram of Oriented Gradients (HOG). Then based on these features, the linear Support Vector Machine (SVM) predicts whether it is a face or not. The actual detection in the image is performed with a sliding window approach and image pyramids.

The method described above has its benefits. HOG descriptors can be obtained in a reasonable amount of time. Besides, SVM is compact with a sufficient number of parameters to learn HOG features. However, it requires multiple predictions on a single image, which is not efficient. Despite that, HOG detectors provide less rich features compared CNNs, which leads to less accurate localization of small objects, such as facial landmarks.

The second step, namely landmark detection, is done with the approach proposed by Kazemi and Sullivan, 2014. It is based on the cascade of regressors. Regression trees are used here for regressing to the point.

4.2.2 Custom detector

Our landmark detector is a neural network with hourglass architecture. It takes an image of a face as input and generates images that describe landmarks. Out of the generated images, we obtain the exact locations of the landmarks.

Network output

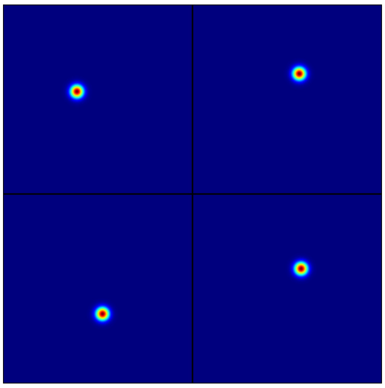


FIGURE 4.2: Examples of a target heatmaps (gaussians).

It is well-known that Convolutional neural networks suit well for image processing. CNNs are able to extract meaningful features from an image and manipulate with them. Therefore, we train our network to predict 56 heatmaps instead of landmark coordinates, which are scalar values. A single heatmap is simply a square gaussian centered in the location of a particular landmark with a standard deviation S. Such a model approximates the gradient class activation map Selvaraju et al., 2019. Grad-CAMs show where neural network thinks a particular object is located on the image.

We empirically found that $S = 6$. If $S > 6$ our confidence gets smaller. If $S < 6$ model gets less amount of information and converges slower. Therefore one heatmap is mostly a black image with the relatively small circle-like object in the location of a particular landmark, as shown in Figure 4.2. We can use such an approach because the landmark location that we predict is a single point. Therefore this point naturally becomes the center of gaussian.

Network architecture

Network finds landmarks in the RGB image. It is designed to predict 56 heatmaps, that correspond to each of 56 landmarks. Network is organized in encoder-decoder way. It has three parts. The first part, namely encoder extracts features from the input image and compresses them into a vector. The second piece is a bottleneck, which is a medium between encoding and decoding. The third part, namely decoder reconstructs 56 heatmaps out of the bottleneck vector.

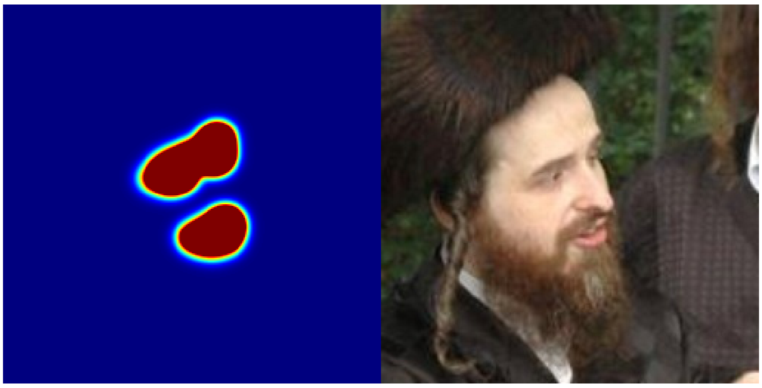


FIGURE 4.3: All 56 heatmaps merged together (left). Note, that model did not observe this heatmap. From it we see, that heatmaps correspond to the image (right).

Encoding part is usually a fully-convolutional neural network. We design it in the way, that different CNN architectures can be used. It is known as a "backbone" of a network. By default we use ResNet-18. ResNet is one of classic CNN architectures. Its skip connections allow creating deeper networks with more capacity. This leads to better feature extraction performance. There different variations of ResNet. We observed that our baseline has sufficient accuracy and is faster than its counterparts (*e.g.* ResNet-152, ResNext), as it has less parameters.

In ResNet-18 we use all five convolutional blocks for encoding. In each convolution block we learn features with different semantic information. In deeper blocks we learn more high-level features (e.g. nose, eye, mouth).

While at the encoding part we downscale our input image, on reconstruction stage we upscale obtained feature maps in order to reconstruct initial resolution of input image. In the encoding part we learn the mapping from image to vector. In the decoding part we use that knowledge to more easily learn the backward mapping (*i.e.* from vector to image). Feature maps from encoder enrich reconstruction abilities. It is provided as additional information for decoder. Decoding part has the same number of levels as encoding part (five in our case). Resolutions of feature maps on the i^{th} level of encoder and decoder are the same. It is needed for concatenation of feature maps from encoder and decoder.

A basic block of decoder on i^{th} level has upscaling operation followed by feature extraction part. Upscaling is performed in the following way. Firstly, pixels of every feature map expand uniformly. Then, gaps, that occur are filled with bilinear interpolation. Such approach prevents high information loss. Upscaling operation does not involve any learning. That is done by two consecutive convolutional layers. These convolutional layers learn the mapping from $(i-1)^{th}$ level to i^{th} one. Features that have useful information needed for particular landmark localization are located in specific local region. Therefore we use convolutions with kernel size of 3x3 and padding equal to 1, which have small receptive field. Zero-padding increases amount of features that correspond to object location and improves learning of features near the border.

In the end we have five sets of feature maps from decoder. We upscale them to the same size and concatenate altogether. Then we extract meaningful features from each of the five sets by convolving to the smaller dimension. We obtain less number of feature maps, but with more rich information. Finally we apply the last convolution to produce 56 heatmaps. We also place sigmoid function on top, so the model converges faster.

Network training

We employ supervised learning. For every image in the dataset, we generate 56 heatmaps, that correspond to a particular landmark. Those heatmaps are used as ground truth for the model.

We make our model pursue a single objective. In our model, we operate with heatmaps, which are spatial objects. Given a ground truth heatmap and the inferred heatmap, we want them to overlap as much as possible. More precisely, a single map contains gaussian, which can be seen as a set of (x, y) coordinates of pixels that belong to this gaussian. Therefore in order to measure the overlap, we do the following. Given ground truth set S_y and synthesized set S_θ we calculate ratio between intersection and union of those two sets in the following way:

$$IOU = \frac{S_y \cap S_\theta + \epsilon}{|S_y| + |S_\theta| - S_y \cup S_\theta + \epsilon} \tag{4.2}$$

We aim to maximize this value, so our cost function, which should be minimized, looks as follows: $C = 1 - IOU$.

We use augmentations of training subset images to avoid overfitting and make our model more robust. We use both pixel-level and spatial-level transforms. The transforms include Gaussian noise, motion and median blur, optical and grid distortions, horizontal flip, shift, scale, and rotate. Every augmentation is applied with some probability. Pixel-level augmentations are more likely to happen, as they are less harmful. Some kind of blur, distortion, or noise is added to every sample.

Before passing the image to the network, we normalize it with normalization used in ImageNet Deng et al., 2009. That is needed for our feature extractor to work correctly, as it is pre-trained on ImageNet dataset.

4.3 GAN Network Architecture

The proposed pipeline (Fig. 4.4) uses a standard adversarial setup with a single Generator (Fig. 4.5) and a Discriminator (Fig. 4.6).

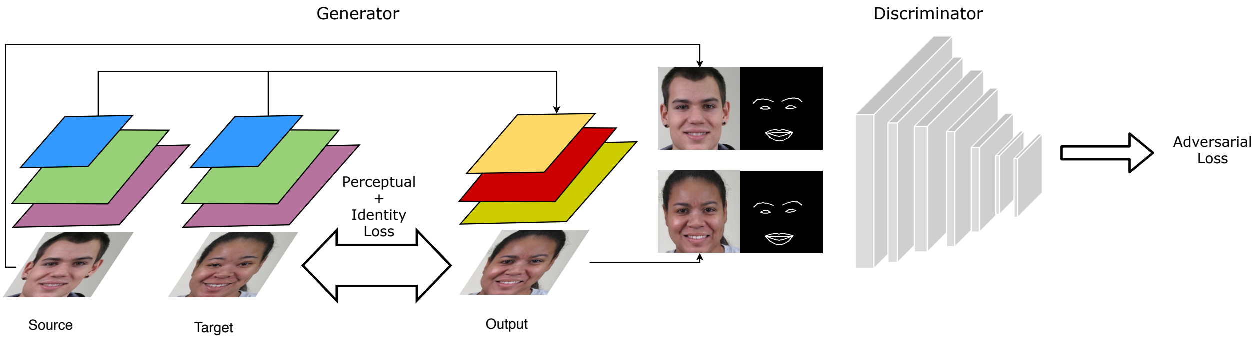


FIGURE 4.4: High-level architecture diagram.

4.3.1 Generator

Our generator is designed in an end-to-end fashion. It infers how to reconstruct the desired image $\hat{x}$ straight from the source x and target $x\prime$ pictures.

It is organized in an encoder-decoder way. Generator has two identical encoders meant for either source or target and a single decoder. Encoders are based upon Feature Pyramid Network introduced by Lin et al., 2016

In recent studies, Feature Pyramid Network framework Lin et al., 2016 shows impressive results, *e.g.* in object detection and segmentation Kirillov et al., 2019. Rich feature extraction and reconstruction abilities of FPN allow us to blend images of source and target effectively.

FPN has two paths with equal number of semantic levels, that correspond to convolutional blocks. The bottom-up path is a FCNN, that is responsible for feature extraction from input image. By default we took InceptionResNetV2 (Szegedy et al., 2016) with weights pre-trained on ImageNet. In the top-down path reconstruction of semantic maps is performed. Extracting such maps for both source and target allows to reconstruct fake image efficiently. This path is enriched with skip connections from the bottom-up path passed through pointwise convolutions. These feature maps are added to maps of top-down path.

We use separate encoders, which means they do not share weights as for example in siamese network. Both encoders extract five feature maps that correspond to semantic level in the pyramid.

Within the decoding part, we employ a sequence of convolutions and upsamplings on those sets of the maps concatenated altogether along the channel dimension. It allows the model to learn information on different semantic levels from both of encoded images simultaneously.

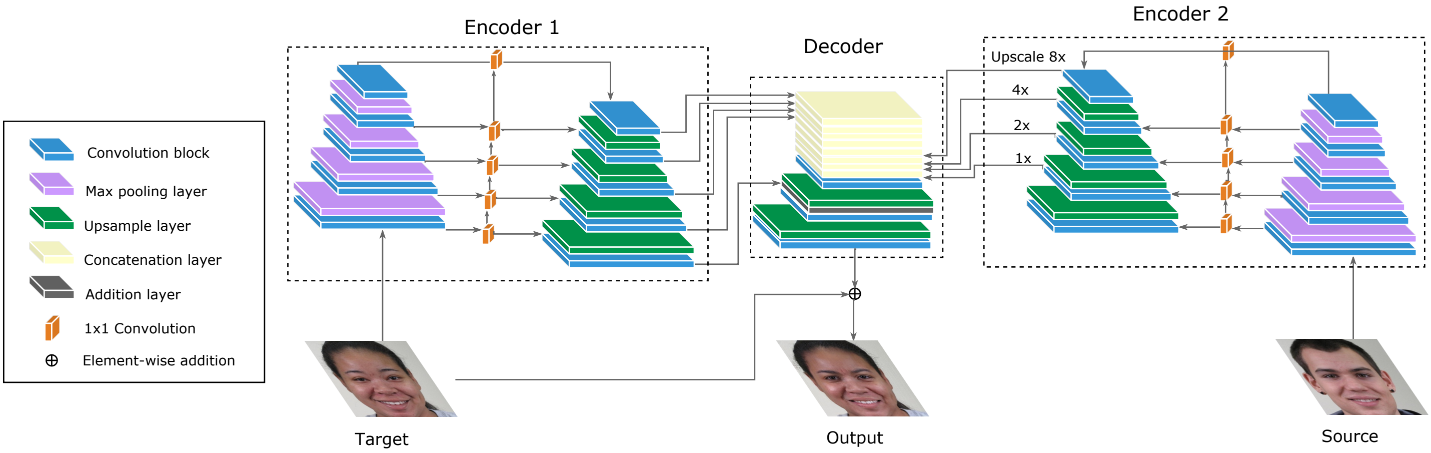


FIGURE 4.5: High-level FPN-based Generator architecture diagram.

4.3.2 Discriminator

The Discriminator is a five-layer fully-convolutional network similar to those in Isola et al., 2016. The network has $4x4$ convolutions in every layer except for the last, where we apply $1x1$ convolution (*i.e.* pointwise convolution). Pointwise convolution is a common practice to flatten feature maps in latest works, as it utilizes less parameters compared to fully connected layer, while losing insignificant amount of information. In every convolutional layer we use padding of size $2x2$ to not decrease size of feature map rapidly and in the last 3 layers maintain it on the nearly the same level. Alongside with padding we use stride of size 2 for the first three layers to have bigger receptive field in the first layers. Meanwhile in the last 2 layers we use stride equal to 1 to preserve local information.

Hidden layers use InstanceNorm (Ulyanov, Vedaldi, and Lempitsky, 2016), which is preferred over BatchNorm (Ioffe and Szegedy, 2015) for GANs. Neurons in all the layers, but output get activated with Leaky ReLU (with slope 0.2), that prevents the appearance of "dying neurons" (Lu et al., 2019).

Discriminator outputs the matrix, that represents the correspondence of each face part to the provided landmarks.

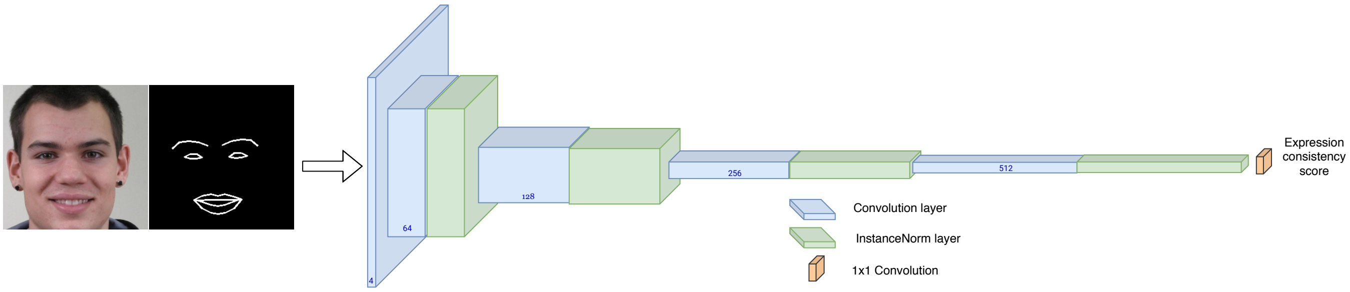


FIGURE 4.6: Discriminator architecture diagram.

4.4 Face normalization

We do normalization (alignment) to make landmarks of different images appear similar to a predefined configuration. This operation is performed on the original image, meaning we do not crop face region beforehand. The process works as follows; first, we select five facial points, namely eyes, nose, and two mouth corners. Then, these points are adopted to perform similarity transformation. Finally, we obtain a cropped face, which is then resized to $N \times N$ region.

4.5 Face Identity Loss

To represent a person's identity, we encode it in a vector of features. In order to extract such identity embedding from the image, we adopt Additive Angular Margin Loss (ArcFace) proposed by Deng et al., 2018, which is a state-of-the-art model in the face-recognition domain. In our experiments, we use the pre-trained ArcFace model with Squeeze-and-Excitation ResNet-50 (Hu et al., 2017) backbone. The authors of the aforementioned paper show that by adding SE-blocks to ResNet-50, one can expect almost the same accuracy as ResNet-101 delivers. This way, we need less computational resources to obtain higher accuracy.

Finally, to find how much identity of $\hat{x}$ varies from $x\prime$ we evaluate the distance between corresponding calculated embeddings $E_{\hat{x}}$ and $E_{x\prime}$.

$$L_{\text{identity}} = \sum (E_{\hat{x}} - E_{x\prime})^2 \tag{4.3}$$

4.6 Network Training

We train our GAN following the common practices. Generator and Discriminator are trained for an equal number of steps, which is less resource hungry. Both these players have the same speed of training (*i.e.* learning rate), which prevents one player from being more powerful than another. For the baseline, we penalize our GAN by a combination of Relativistic average GANs (Jolicoeur-Martineau, 2018) and Least-squares GANs (Mao et al., 2016), that shows the most pleasant results. In the experiments section, we present an empirical study of different adversarial losses. Discriminator loss is as follows:

$$\mathrm{L}_D^{\mathrm{RaLSGAN}} = E_{x_\gamma}[(D(x_\gamma) - E_{(x,x')}D(G_\gamma(x,x')) - 1)^2] + E_{(x,x')}[(D(G_\gamma(x,x')) - E_{x_\gamma}D(x_\gamma) + 1)^2] \quad (4.4)$$

where x_γ - source image x concatenated with its face landmarks γ; $G_\gamma(x,x')$ - generated image concatenated with source's face landmarks γ.

We train our Generator to pursue three objectives that enforce proper face reenactment.

The first objective, namely content (*i.e.* background, illumination, image structure) preservation of target image is achieved with perceptual loss (Johnson, Alahi, and Fei-Fei, 2016). Consecutively it is calculated between $x\prime$ and $\hat{x}$. Our L_{content} consists of two parts. The first one is MSE between ReLU activations of the third convolutional layer of pre-trained VGG19 model. We also add l2 regularization. The loss with coefficients is as follows:

$$L_{content} = 0.06 \cdot MSE(l_{\hat{x}}^{\phi,relu_{3,3}}, l_{x'}^{\phi,relu_{3,3}}) + 0.5 \cdot MSE(\hat{x}, x') \quad (4.5)$$

Identity preservation is covered with proposed identity loss described in Section 4.5. It enforces Discriminator to produce faces, whose identity embedding matches the embedding of a target face precisely.

In this study we compare two approaches, namely landmark loss and landmark discriminator, to expression transfer penalization described in Section 4.1. The first one implies both adversarial and landmark loss, while the later has only adversarial. Applying RaLSGANs,

$$L_{\mathrm{adversarial}} = E_{x_\gamma}[(D(x_\gamma) - E_{(x,x')}D(G_\gamma(x,x')) + 1)^2] + E_{(x,x')}[(D(G_\gamma(x,x')) - E_{x_\gamma}D(x_\gamma) - 1)^2] \quad (4.6)$$

The full objective for Generator in the baseline (*i.e.* expression penalization using discriminator) combines three losses with appropriate scales and is as follows:

$$L_{\mathrm{G}} = \lambda_{\mathrm{content}} \times L_{\mathrm{content}} + \lambda_{\mathrm{adv.}} \times L_{\mathrm{adv.}} + \lambda_{\mathrm{identity}} \times L_{\mathrm{identity}} \quad (4.7)$$

If we train with landmark loss we simply add it to the above equation with appropriate scale $\lambda_{lamdmark}$. Note that in these two cases adversarial loss is computed in different way.

The input for the model is target-source pairs of images selected at random. We feed a pair per propagation because we adopt Instance normalization in both Generator and Discriminator.

Chapter 5

Experiments

5.1 Implementation details

In our work, we use a standard toolkit for Machine Learning. We implement our solution alongside with the experiments on Python 3.7 (Van Rossum and Drake, 2009), which is a dynamic language for quick prototyping. For building and training neural networks, we prefer Pytorch 1.0 framework (Paszke et al., 2019 over Tensorflow (Abadi et al., 2015, as it implements dynamic graph (*i.e.* nodes may be added on runtime) and, in our opinion, is more intuitive. In order to visualize the learning process, we employ TensorBoard, which has a simple API for plotting losses and images. For image processing, matrix manipulations and some visualizations we use OpenCV (Bradski, 2000, NumPy (Oliphant, 2006 and Matplotlib (Hunter, 2007 correspondingly.

5.2 Datasets

5.2.1 GAN

For training and evaluating our model we selected Compound facial expressions of emotion (CFEE) dataset by Du, Tao, and Martinez, 2014. It contains images of 26 emotions (*i.e.* facial expressions) presented by 244 different persons (*i.e.* identities). The white background of the images makes it simpler for the model to learn transitions from one expression to another. Despite simple background used for training we show in Table 6.5, that our model is able to retain background on FaceForensics++ dataset (Rössler et al., 2019).

We randomly select 100 images of each emotion, which results in 2600 training samples. The rest we leave for evaluation. Those images are then normalized concerning the procedure described in Sec. 4.4, cropped around the face and resized to 256x256. All our models are trained on this image resolution.

We evaluate on 500 randomly selected from test subset image pairs.

5.2.2 Custom landmark detector

Out detector is trained on WFLW dataset Wu et al., 2018, which contains 10000 faces with 98 fully manual annotated landmarks. Each annotation is a (X, Y) coordinate of a landmark on the image of size (H x W). Out of these landmarks, we selected 56 landmarks that correspond to mouth, eyes, and eyebrows. Based on visual inspection, we see that these parts of a face change the most drastically from one expression to another.

As mentioned in Sec. 4.2.2, our network predicts heatmaps rather than scalar values. Therefore we additionally generate 56 heatmaps for every image in the dataset right before training.

We split our dataset into train and test parts. For the training part, we select 5000 images and leave 3000 images for evaluation. We additionally split the training subset randomly into training and validation parts with a ratio of 9:1.

5.3 Training details

Both our models were trained on NVIDIA GTX 1080 GPU using the Adam solver (Kingma and Ba, 2014), which is recommended as a default algorithm to use. We have not found meaningful reasons to use another algorithm. We use a typical initial learning rate equal to 0.0001 and Xavier algorithm (Glorot and Bengio, 2010) for weights initialization.

5.3.1 GAN

We train our baseline model for 250 epochs. As we observed, training it for a longer time leads to divergence and poor results. We use a linear decay scheduler, which decreases the learning rate down to 1e−7 starting from 40 epoch.

Before evaluation we experimented with different coefficients for our full objective (Equation **??**). Experimentally we obtained the following: $\lambda_{\text{content}} = 0.01$, $\lambda_{\text{adv.}} = 0.1$, $\lambda_{\text{identity}} = 0.001$. While training with landmark loss we get slightly different value of $\lambda_{\text{adv.}} = 0.001$. We scale the landmark penalty $L_{landmark}$ with the value of $\lambda_{lamdmark} = 2$.

5.3.2 Custom landmark detector

Our custom detector is trained for 200 epochs. We use manual scheduling with a decreasing learning rate during the training on the 60th and 80th epochs by 10. That allows the optimizer to reach minima with higher certainty.

5.3.3 DLib models

We use pre-trained models for face detection and landmark estimation provided with the library to compare GAN losses. They were trained on iBUG 300-W face landmark dataset Sagonas et al., 2016. It has 300 indoor and 300 outdoor faces under different poses, illumination, scale, annotated with 68 landmarks. This database lacks in the amount as well as the diversity of pictures. It is limited in images of faces with occlusions, makeup, blur.

We additionally train DLib detector on WFLW dataset. We use the training script provided with DLib with standard options.

5.4 Metrics

For evaluating results of face reenactment, we use Fréchet inception distance (FID Heusel et al., 2017), Normalized mean square error (NMSE), and Cosine similarity (CSIM) described in the corresponding sections. Landmark detection results are evaluated with NMSE.

5.4.1 NMSE

MNSE shows a similarity between the two sets of landmarks. It is a an Euclidian distance between two sets of facial landmarks, which is normalized by inter-ocular distance and number of landmark points in one set. That is an Euclidian distance between centroids (*i.e.* pupils) of two eyes.

$$\text{NMSE} = \frac{\sum\limits_{i=1}^{L} \sqrt{(x_i^{\theta} - x_i^{gt})^2 + (y_i^{\theta} - y_i^{gt})^2}}{L \cdot \sqrt{(x^{gt} - x_r^{gt})^2 + (y_l^{gt} - y_r^{gt})^2}} \cdot 100 \tag{5.1}$$

where L - number of landmarks, x'_l - x-coordinate of left pupil of the source (ground truth), y'_l - y-coordinate of left pupil of the source, similarly x'_r and y'_r - coordinates of the right pupil.

We use NMSE for two purposes. Firstly we measure expression transfer accuracy produced by our GAN for FR. We calculate it in a pair of landmarks of generated and source. Secondly, we evaluate the accuracy of landmark detection, which is a standard metric, widely used in works on landmark detection (Kazemi and Sullivan, 2014; Cao et al., 2014; Sun, Wang, and Tang, 2013; Ranjan, Patel, and Chellappa, 2016).

5.4.2 FID

Fréchet inception distance is used for measuring image consistency and realism. It runs on several images rather than single ones and calculates statistics of two sets, which is its main advantage over well-known Inception score Salimans et al., 2016. We apply it for measuring realism of generated faces and content-identity preservation on average.

5.4.3 CSIM

Cosine similarity is an explicit cosine of an angle between two vectors. We use it to explicitly measure the distance between identity embedding of generated and target persons. For a single pair of images, CSIM is given as follows:

$$CSIM = \frac{\sum\limits_{i=1}^{S} E_i^{\hat{x}} \cdot \sum\limits_{i=1}^{S} E_i^{x'}}{\sqrt{\sum\limits_{i=1}^{S} (E_i^{\hat{x}})^2} \cdot \sqrt{\sum\limits_{i=1}^{S} (E_i^{x'})^2}} \tag{5.2}$$

5.5 Empirical studies

All the models for comparison are trained for 100 epochs in setup described in the beginning of Sec. 4.6 on the dataset described in Sec. 5.2.1. We evaluate results both quantitatively (*i.e.* with FID, NMSE, CSIM) and qualitatively with User study (5.6).

5.5.1 Expression penalization

We study the abilities of landmark loss and landmark discriminator to provide effective FR. We train two models with slightly different architectures and training scenarios described in Sec. 4.1. We assume that landmark loss should provide more accurate expression and stable training.

5.5.2 Landmark detectors

We evaluate our custom landmark detector against DLib landmark detector in two scenarios. To this end, we additionally train DLib model (described in 4.2.1) on WFLW dataset. Firstly, we compare the accuracy of predicted landmarks using NMSE metric and performance (*i.e.* average prediction time). Secondly, we run the investigation to find out how the landmark detection technique influences face reenactment quality. To this end, we train our GAN using landmark discriminator firstly with DLib detector trained by us, and then with our custom detector. Recall, that landmark discriminator is responsible for the penalization of expression transfer. Therefore we are mostly interested to see the difference in the accuracy of reenacted mimics. We also take into account other FR properties.

5.5.3 GAN losses

Recent studies show that adversarial loss influences the stability of the training process, perceptual image quality, provide a more or less meaningful representation of produced results (Lucic et al., 2018; Dong and Yang, 2019; Kurach et al., 2019). Therefore we examine the impact of this loss function onto FR results. We selected four losses, namely Wasserstein GAN (WGAN Arjovsky, Chintala, and Bottou, 2017), Relativistic average GAN (RaGAN Jolicoeur-Martineau, 2018), Least-Squares GAN (LSGAN Mao et al., 2016) and combination of RaGAN and LSGAN – RaLSGAN. These losses are among those, which may be applied to a wide range of problems and are not designed for a specific task.

WGANs are well-known for their training stability (they have no sign of mode collapse) and the correlation between error value and image quality. Usually, GAN loss is selected empirically. WGAN is a solid starting point in search of the proper loss function. Note that we use WGAN with a gradient penalty for weight clipping, which is an improvement of the original Wasserstein GAN.

We assume that a single LSGAN may perform the most poorly in terms of content and identity preservation, as it has a tendency to smooth pixels on the image. Its main advantage is the stabilization of the training procedure. The main advantage of RaGANs is that when evaluating a sample from the positive set, they take into account statistics of a negative one and vice versa. So we expect, that combination of LSGAN and RaGAN should provide higher perceptual quality and stable training.

5.6 User study

Quantitative evaluation of face reenactment is quite bounded. In recent days there appears no proper metric. Even a combination of different metrics provides an incomplete representation of the actual situation. Therefore it is common for works in this field additionally to quantitative results provide human evaluation, which is usually conducted on Amazon mechanical turk (Crowston, 2012). We do not have the possibility to use AMTurk. However, we conduct a similar study.

We incorporated our user study in a Google Form. The protocol is as follows. We selected ten pairs of images (source and target). With each of our six approaches (described in Sec. 5.5) we produced ten fake persons out of these pairs. For every person, we ask the participants to say which fake looks more real to them. In the case where we compare more than two approaches, we ask people to select two, in their opinion, the most realistic fakes.

We collected 32 responses and present results in Figures 6.2, 6.3, 6.4. It is worth to mention that most of the participants are familiar with face manipulation technologies.

Chapter 6

Results

6.1 Expression penalization

From metrics in Table 6.2 it is visible, that landmark loss significantly outperforms landmark discriminator. It was expected in terms of NMSE, as the loss directly penalized distance between landmarks of source and generated. What is interesting, that landmark loss shows higher results in perceptual quality and identity preservation. Observe, that user study correlates with quantitative results. A higher number of participants in total selected landmark loss over the discriminator.

6.2 Landmark detectors

Observe from Table 6.3 that our custom landmark detector outperforms DLib detector in terms of NMSE, while loses in performance (*i.e.* prediction time). However, our detector is more robust to corner cases, such as severe poses, occlusions, mustache, beard (Table 6.1). The use of augmentations highly influences it.

When we apply both detectors to face reenactment, they show close results. From Table 6.2, one may see, that our detector improved expression accuracy and identity preservation, while slightly lost in perceptual quality. The reason may be a small inconsistency in predictions on the same images observed in our detector. By inconsistency, we mean a slight difference in landmark locations for the same image predicted on different runs. (see in Fig. 6.1).

6.3 GAN losses

From Table 6.2 we can see, that WGAN clearly shows the highest performance in terms of the metrics. RaGAN, LSGAN and RaLSGAN show quite similar results. However, while LSGAN has the highest perceptual quality (*i.e.* FID) among those three, it produces the least accurate expression and identity. RaGAN shows 'mirror' results to LSGAN. RaGAN has the poorest perceptual quality (which also correlates with user study), but the highest expression transfer and identity preservation. RaLSGAN is in-between those two quantitatively. At the same time, based on conducted user study, it produces the most realistic faces and exceeds WGAN significantly. Observe, that visual quality degrades with nearly the same step from RaLSGAN to RaGAN.

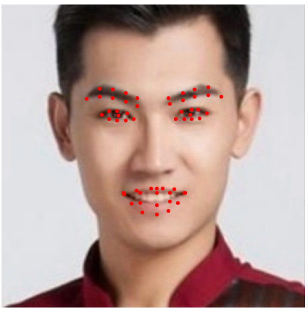
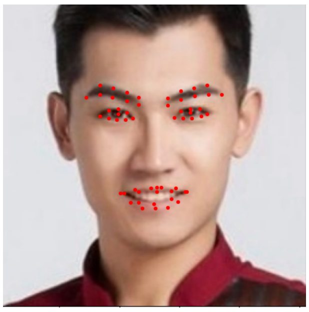
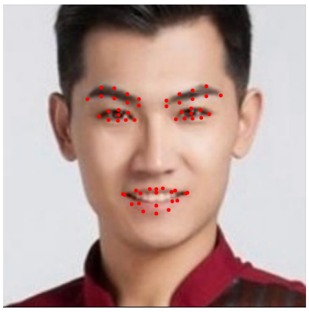
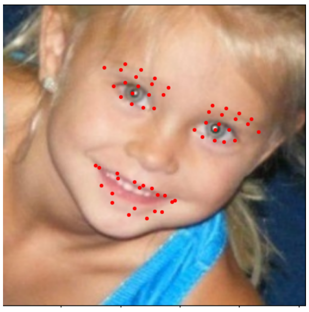
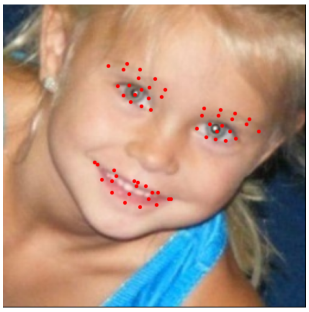
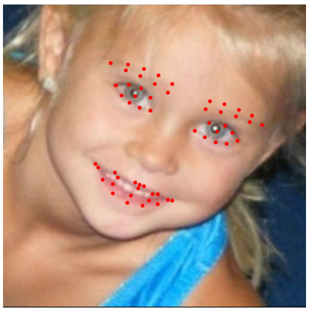
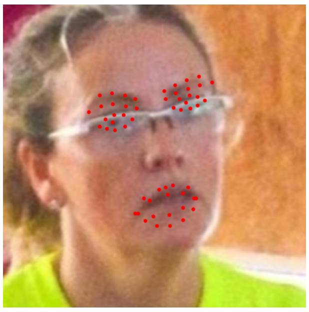
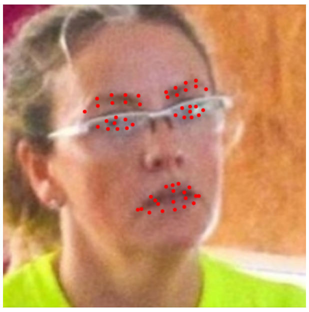
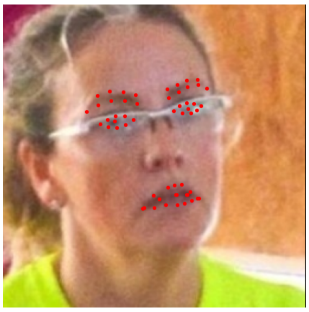
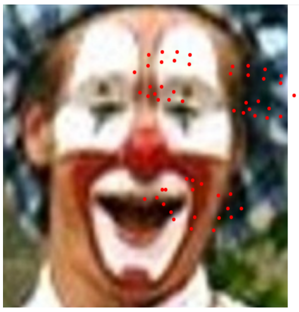
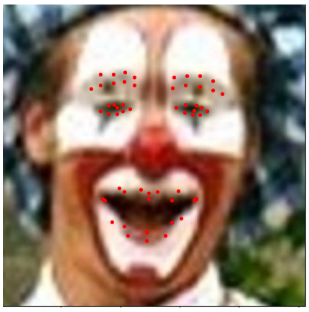
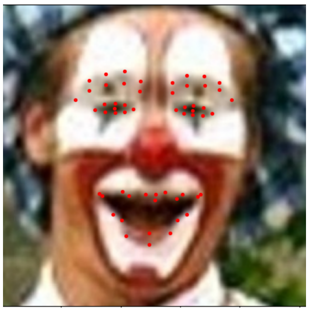
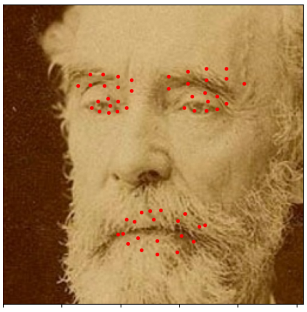
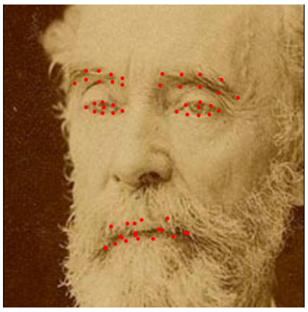
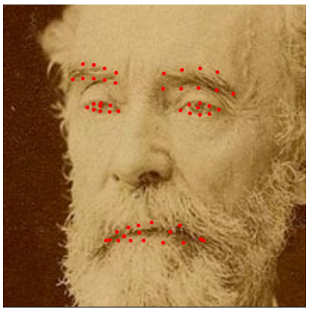
Case	DLib	U-Net	Ground truth
Average			
Pose			
Occlusion			
Mask			
Beard			

TABLE 6.1: Qualitative comparison of landmark detectors.

	FID ↓	NMSE ↓	CSIM ↑
Landmark detetectors			
DLib	**17.36**	17.77%	−0.13
U-Net	18.83	**15.38**%	**0.02**
GAN losses			
RaLSGAN	21.97	11.1%	0.38
LSGAN	22.25	13.26%	0.17
RaGAN	23.24	8.96%	0.57
WGAN	**15.22**	**6.49**%	**0.75**
Landmark penalization			
Discriminator	18.83	15.38%	0.02
Loss	**12.15**	**3.37**%	**0.89**

TABLE 6.2: Quantitative face reenactment experiments results.

	Error (NMSE) ↓	Average prediction time on CPU (ms.) ↓
DLib	10.86%	7.83
U-Net	**6.73**%	48.81

TABLE 6.3: Landmark detectors comparison

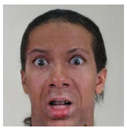
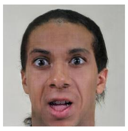
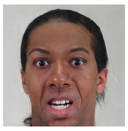
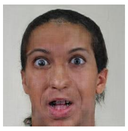
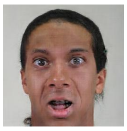
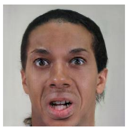
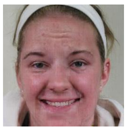
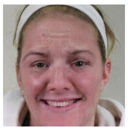
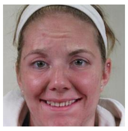
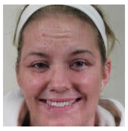
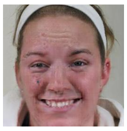
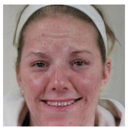
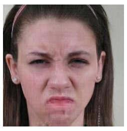
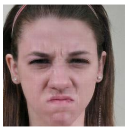
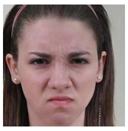
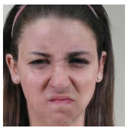
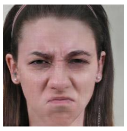
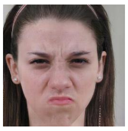
RaLSGAN/DLib	U-Net/Discriminator	Loss	LSGAN	RaGAN	WGAN
					
					
					

TABLE 6.4: Qualitative results of face reenactment

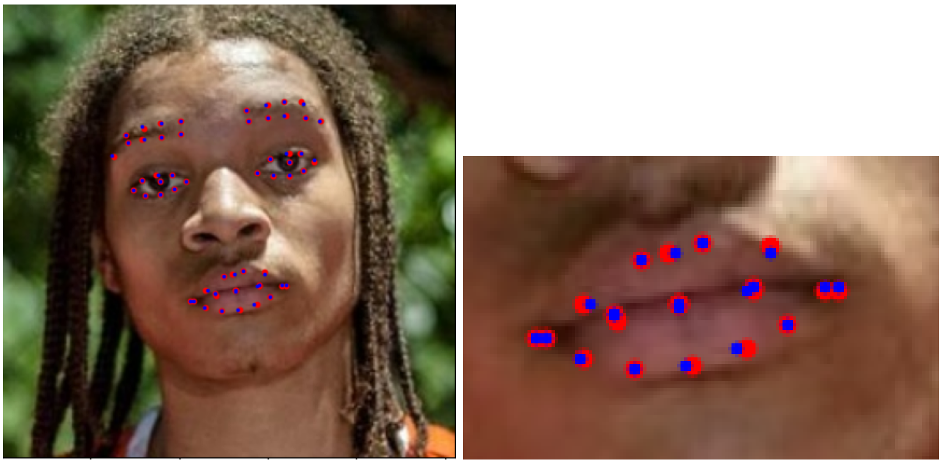


FIGURE 6.1: Example of small inconsistency between two distinct predictions of U-Net detector.

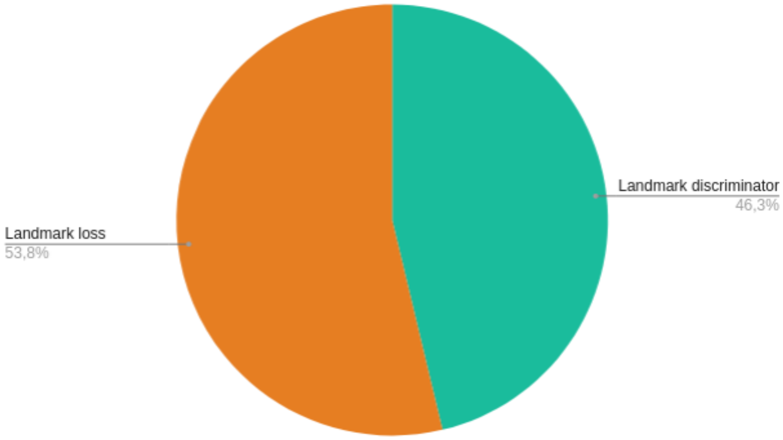


FIGURE 6.2: User study. Expression penalization

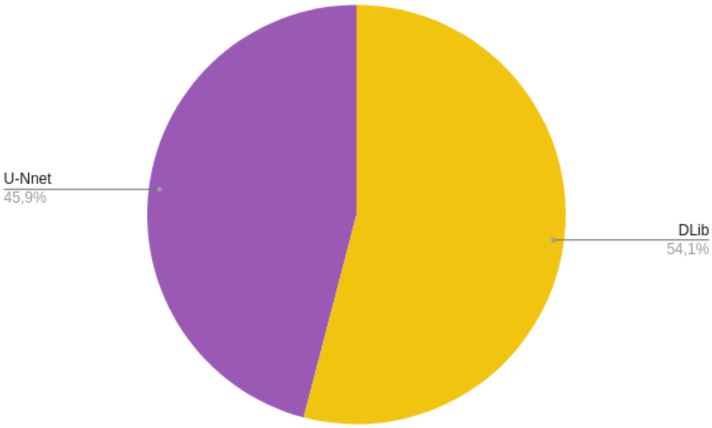


FIGURE 6.3: User study. Landmark detectors

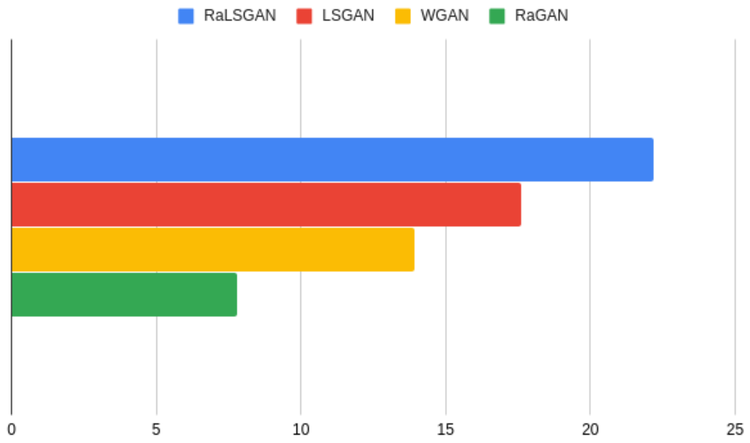


FIGURE 6.4: User study. GAN loses

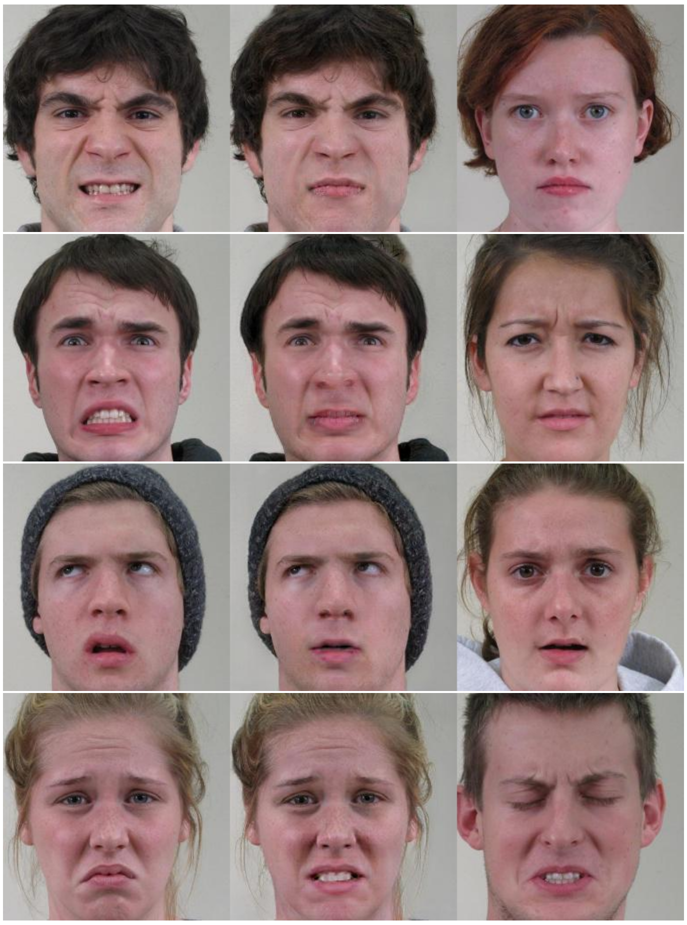


FIGURE 6.5: Mouth expression reenactment

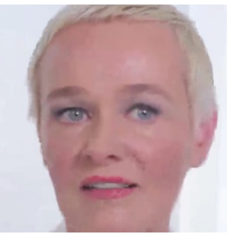
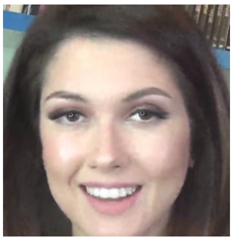
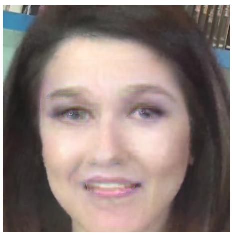
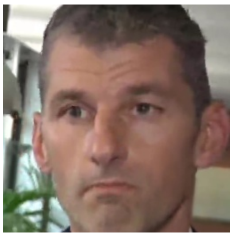
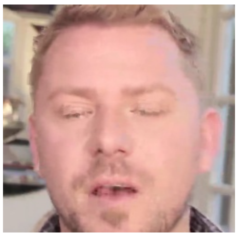
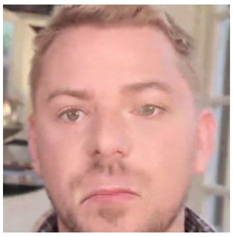
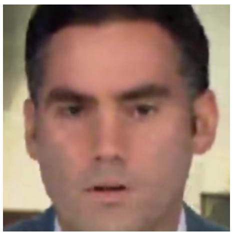
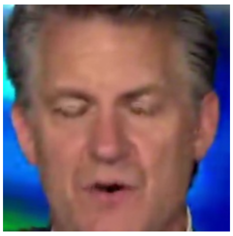
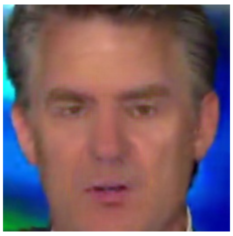
Source	Target	Result
		
		
		

TABLE 6.5: Example of background preservation.

Chapter 7

Conclusion and Future work

7.1 Conclusion

In this work, we provide a flexible and efficient one-shot solution for many-to-many face reenactment problem using GAN architecture.

We studied various adversarial losses and showed, that combination of Relativistic-average GANs and Least-squares GANs is the most visually pleasing and produces accurate expression transfer.

We focused on landmark representation of facial expression and built our custom landmark detector based on U-Net architecture, which shows higher accuracy compared to DLib detector. We showed that DLib detector is sensitive to numerous corner cases (*i.e.* occlusions, masks, poses), while our detector is highly robust.

We examined two alternatives for stimulating accurate expression transfer, namely landmark discriminator and landmark loss, and found that the later shows satisfactory results both quantitatively and qualitatively.

We admit that no proper metric for evaluation of face reenactment is present yet. Therefore in our work, we conducted a user study to measure the visual quality of synthetic images.

We acknowledge that the problem of face reenactment touches ethical questions and can be used for violating purposes. Therefore we do not open-source our code. However, we are open to cooperation in the improvement of existing neural generated content detection systems.

7.2 Future work

For future work, we consider the following directions.

In terms of face reenactment, we plan to enhance the abilities of our GAN to pose and eye gaze transfer. That would provide better higher realism and would be highly useful for videos of synthetic faces.

There is still a gap in the identity preservation of a target person. We observe that sometimes face characteristics of source (*i.e.* eyes shape, mouth forms) may be copied. Therefore we consider an additional module that would emphasize the identity features of the target. As a starting point, we can use style discriminator similar to. An important step towards better identity preservation may be adaptation (*i.e.* warping) of source landmarks to target in calculating landmark loss. Consecutively it would make generated landmarks more natural for a target face.

For our landmark detector, we plan to work towards higher consistency of predictions. To enhance our study, we want to provide a broad comparison of existing landmark detectors with ours.

Finally, we want to expand our user study. For example, we may add real images to the form and ask participants to say whether the person is real or fake. Another option would be to use AMTurk.

Appendix A

GitHub repository link: https://gitlab.com/vosar/thesis

User study Google Form link: https://forms.gle/7Yho4KqFVbjixu8d7

Bibliography

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: http://tensorflow.org/.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). *Wasserstein GAN*. arXiv: 1701.07875 [stat.ML].

Bitouk, Dmitri et al. (Aug. 2008). "Face swapping: Automatically replacing faces in photographs". In: *ACM Trans. Graph.* 27. DOI: 10.1145/1399504.1360638.

Blanz, Volker and Thomas Vetter (Sept. 2002). "A Morphable Model for the Synthesis of 3D Faces". In: *SIGGRAPH'99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. DOI: 10.1145/311535.311556.

Blanz, Volker et al. (2004). "Exchanging Faces in Images". In: *Computer Graphics Forum*. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2004.00799.x.

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Cao, Xudong et al. (Apr. 2014). "Face Alignment by Explicit Shape Regression". In: *Int. J. Comput. Vision* 107.2, pp. 177–190. ISSN: 0920-5691. DOI: 10.1007/s11263-013-0667-3. URL: http://dx.doi.org/10.1007/s11263-013-0667-3.

Choi, Yunjey et al. (2018). "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Crowston, Kevin (2012). "Amazon Mechanical Turk: A Research Tool for Organizations and Information Systems Scholars". In: *Shaping the Future of ICT Research. Methods and Approaches*. Ed. by Anol Bhattacherjee and Brian Fitzgerald. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 210–221. ISBN: 978-3-642-35142-6.

Deng, J. et al. (2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*.

Deng, Jiankang et al. (2018). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *arXiv e-prints*, arXiv:1801.07698, arXiv:1801.07698. arXiv: 1801.07698 [cs.CV].

Dong, Hao-Wen and Yi-Hsuan Yang (2019). *Towards a Deeper Understanding of Adversarial Losses*. arXiv: 1901.08753 [cs.LG].

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. ISSN: 0027-8424. DOI: 10.1073/pnas.1322355111. eprint: https://www.pnas.org/content/111/15/E1454.full.pdf. URL: https://www.pnas.org/content/111/15/E1454.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

Goodfellow, Ian (2016). *NIPS 2016 Tutorial: Generative Adversarial Networks*. arXiv: 1701.00160 [cs.LG].

Goodfellow, Ian J. et al. (2014). “Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1406.2661, arXiv:1406.2661. arXiv: 1406.2661 [stat.ML].

Ha, Sungjoo et al. (2019). *MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets*. arXiv: 1911.08139 [cs.CV].

Heusel, Martin et al. (2017). “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *arXiv e-prints*, arXiv:1706.08500, arXiv:1706.08500. arXiv: 1706.08500 [cs.LG].

Hu, Jie et al. (2017). “Squeeze-and-Excitation Networks”. In: *arXiv e-prints*, arXiv:1709.01507, arXiv:1709.01507. arXiv: 1709.01507 [cs.CV].

Huang, Xun et al. (2018). *Multimodal Unsupervised Image-to-Image Translation*. arXiv: 1804.04732 [cs.CV].

Hunter, J. D. (2007). “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

Ioffe, Sergey and Christian Szegedy (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv: 1502.03167 [cs.LG].

Isola, Phillip et al. (2016). “Image-to-Image Translation with Conditional Adversarial Networks”. In: *arXiv e-prints*, arXiv:1611.07004, arXiv:1611.07004. arXiv: 1611.07004 [cs.CV].

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *arXiv e-prints*, arXiv:1603.08155, arXiv:1603.08155. arXiv: 1603.08155 [cs.CV].

Jolicoeur-Martineau, Alexia (2018). “The relativistic discriminator: a key element missing from standard GAN”. In: *arXiv e-prints*, arXiv:1807.00734, arXiv:1807.00734. arXiv: 1807.00734 [cs.LG].

Karras, Tero et al. (2017). “Progressive Growing of GANs for Improved Quality, Stability, and Variation”. In: *arXiv e-prints*, arXiv:1710.10196, arXiv:1710.10196. arXiv: 1710.10196 [cs.NE].

Kazemi, Vahid and Josephine Sullivan (June 2014). “One Millisecond Face Alignment with an Ensemble of Regression Trees”. In: DOI: 10.13140/2.1.1212.2243.

Kim, Hyeongwoo et al. (July 2018). “Deep Video Portraits”. In: *ACM Trans. Graph.* 37.4, 163:1–163:14. ISSN: 0730-0301. DOI: 10.1145/3197517.3201283. URL: http://doi.acm.org/10.1145/3197517.3201283.

Kim, Taeksoo et al. (2017). “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1703.05192, arXiv:1703.05192. arXiv: 1703.05192 [cs.CV].

King, Davis E. (2009). “Dlib-ml: A Machine Learning Toolkit”. In: *Journal of Machine Learning Research* 10, pp. 1755–1758.

Kingma, Diederik P. and Jimmy Ba (2014). “Adam: A Method for Stochastic Optimization”. In: *arXiv e-prints*, arXiv:1412.6980, arXiv:1412.6980. arXiv: 1412.6980 [cs.LG].

Kirillov, Alexander et al. (2019). “Panoptic Feature Pyramid Networks”. In: *arXiv e-prints*, arXiv:1901.02446, arXiv:1901.02446. arXiv: 1901.02446 [cs.CV].

Kosarevych, Ivan et al. (2020). *ActGAN: Flexible and Efficient One-shot Face Reenactment*. arXiv: 2003.13840 [cs.CV].

Kurach, Karol et al. (2019). “A Large-Scale Study on Regularization and Normalization in GANs”. In: *International Conference on Machine Learning*. URL: https://arxiv.org/abs/1807.04720.

Lassner, Christoph, Gerard Pons-Moll, and Peter V. Gehler (2017). “A Generative Model of People in Clothing”. In: *arXiv e-prints*, arXiv:1705.04098, arXiv:1705.04098. arXiv: 1705.04098 [cs.CV].

LeCun, Yann and Corinna Cortes (2010). “MNIST handwritten digit database”. In: URL: http://yann.lecun.com/exdb/mnist/.

Lee, Hsin-Ying et al. (2018). *Diverse Image-to-Image Translation via Disentangled Representations*. arXiv: 1808.00948 [cs.CV].

Lin, Tsung-Yi et al. (2016). “Feature Pyramid Networks for Object Detection”. In: *arXiv e-prints*, arXiv:1612.03144, arXiv:1612.03144. arXiv: 1612.03144 [cs.CV].

Liu, Ming-Yu, Thomas Breuel, and Jan Kautz (2017). “Unsupervised Image-to-Image Translation Networks”. In: *arXiv e-prints*, arXiv:1703.00848, arXiv:1703.00848. arXiv: 1703.00848 [cs.CV].

Lu, Lu et al. (2019). *Dying ReLU and Initialization: Theory and Numerical Examples*. arXiv: 1903.06733 [stat.ML].

Lucic, Mario et al. (2018). “Are GANs Created Equal? A Large-Scale Study”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., pp. 700–709. URL: http://papers.nips.cc/paper/7350-are-gans-created-equal-a-large-scale-study.pdf.

Mao, Xudong et al. (2016). “Least Squares Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1611.04076, arXiv:1611.04076. arXiv: 1611.04076 [cs.CV].

Mirza, Mehdi and Simon Osindero (2014). “Conditional Generative Adversarial Nets”. In: *arXiv e-prints*, arXiv:1411.1784, arXiv:1411.1784. arXiv: 1411.1784 [cs.LG].

Nash, John (1951). “Non-Cooperative Games”. In: *Annals of Mathematics* 54.2, pp. 286–295. ISSN: 0003486X. URL: http://www.jstor.org/stable/1969529.

Nirkin, Yuval, Yosi Keller, and Tal Hassner (2019). “FSGAN: Subject Agnostic Face Swapping and Reenactment”. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Oliphant, Travis E (2006). *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.

Park, Taesung et al. (2019). “Semantic Image Synthesis with Spatially-Adaptive Normalization”. In: *arXiv e-prints*, arXiv:1903.07291, arXiv:1903.07291. arXiv: 1903.07291 [cs.CV].

Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *arXiv e-prints*, arXiv:1912.01703, arXiv:1912.01703. arXiv: 1912.01703 [cs.LG].

Pumarola, Albert et al. (2018). “GANimation: Anatomically-aware Facial Animation from a Single Image”. In: *arXiv e-prints*, arXiv:1807.09251, arXiv:1807.09251. arXiv: 1807.09251 [cs.CV].

Ranjan, Rajeev, Vishal M. Patel, and Rama Chellappa (2016). “HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition”. In: *arXiv e-prints*, arXiv:1603.01249, arXiv:1603.01249. arXiv: 1603.01249 [cs.CV].

Robbins, H. and S. Monro (1951). “A stochastic approximation method”. In: *Annals of Mathematical Statistics* 22, pp. 400–407.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs.CV].

Rössler, Andreas et al. (2019). “FaceForensics++: Learning to Detect Manipulated Facial Images”. In: *International Conference on Computer Vision (ICCV)*.

Sagonas, Christos et al. (Jan. 2016). “300 Faces In-The-Wild Challenge: database and results”. In: *Image and Vision Computing* 47. DOI: 10.1016/j.imavis.2016.01.002.

Salimans, Tim et al. (2016). “Improved Techniques for Training GANs”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 2234–2242. URL: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

Selvaraju, Ramprasaath R. et al. (2019). “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *International Journal of Computer Vision* 128.2, 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

Siarohin, Aliaksandr et al. (2018). *Animating Arbitrary Objects via Deep Motion Transfer*. arXiv: 1812.08861 [cs.GR].

Siarohin, Aliaksandr et al. (2019). “First Order Motion Model for Image Animation”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 7137–7147. URL: http://papers.nips.cc/paper/8935-first-order-motion-model-for-image-animation.pdf.

Sun, Y., X. Wang, and X. Tang (2013). “Deep Convolutional Network Cascade for Facial Point Detection”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476–3483. DOI: 10.1109/CVPR.2013.446.

Szegedy, Christian et al. (2016). “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *arXiv e-prints*, arXiv:1602.07261, arXiv:1602.07261. arXiv: 1602.07261 [cs.CV].

Thies, J. et al. (2016). “Face2Face: Real-Time Face Capture and Reenactment of RGB Videos”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2387–2395. DOI: 10.1109/CVPR.2016.262.

Thies, Justus et al. (Oct. 2015). “Real-time Expression Transfer for Facial Reenactment”. In: *ACM Trans. Graph.* 34.6, 183:1–183:14. ISSN: 0730-0301. DOI: 10.1145/2816795.2818056. URL: http://doi.acm.org/10.1145/2816795.2818056.

Tripathy, Soumya, Juho Kannala, and Esa Rahtu (2019). *ICface: Interpretable and Controllable Face Reenactment Using GANs*. arXiv: 1904.01909 [cs.CV].

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). *Instance Normalization: The Missing Ingredient for Fast Stylization*. arXiv: 1607.08022 [cs.CV].

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Vlasic, Daniel et al. (2005). “Face transfer with multilinear models.” In: *ACM Trans. Graph.* 24.3, pp. 426–433. URL: http://dblp.uni-trier.de/db/journals/tog/tog24.html#VlasicBPP05.

Wang, Miao et al. (2019). “Example-Guided Style Consistent Image Synthesis from Semantic Labeling”. In: *arXiv e-prints*, arXiv:1906.01314, arXiv:1906.01314. arXiv: 1906.01314 [cs.CV].

Wang, Ting-Chun et al. (2017). “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs”. In: *arXiv e-prints*, arXiv:1711.11585, arXiv:1711.11585. arXiv: 1711.11585 [cs.CV].

Wu, Wayne et al. (2018). “Look at Boundary: A Boundary-Aware Face Alignment Algorithm”. In: *CVPR*.

Wu, Wayne et al. (2018). “ReenactGAN: Learning to Reenact Faces via Boundary Transfer”. In: *arXiv e-prints*, arXiv:1807.11079, arXiv:1807.11079. arXiv: 1807.11079 [cs.CV].

Yi, Zili et al. (2017). “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *CoRR* abs/1704.02510. arXiv: 1704.02510. URL: http://arxiv.org/abs/1704.02510.

Zakharov, Egor et al. (2019). “Few-Shot Adversarial Learning of Realistic Neural Talking Head Models”. In: *arXiv e-prints*, arXiv:1905.08233, arXiv:1905.08233. arXiv: 1905.08233 [cs.CV].

Zhang, Jiangning et al. (2019). “FaceSwapNet: Landmark Guided Many-to-Many Face Reenactment”. In: *arXiv e-prints*, arXiv:1905.11805, arXiv:1905.11805. arXiv: 1905.11805 [cs.CV].

Zhu, Jun-Yan et al. (2017). *Toward Multimodal Image-to-Image Translation*. arXiv: 1711.11586 [cs.CV].

Zhu, Jun-Yan et al. (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *arXiv e-prints*, arXiv:1703.10593, arXiv:1703.10593. arXiv: 1703.10593 [cs.CV].