UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Face-reenactment: generation flexibility and identity preservation

Author:
Marian PETRUK

Supervisor:
Markian KOSTIV

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, Marian PETRUK, declare that this thesis titled, "Face-reenactment: generation flexibility and identity preservation " 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:

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Face-reenactment: generation flexibility and identity preservation

by Marian PETRUK

Abstract

Face-reenactment, also knows as puppetry, has become very popular in recent years. The proposed task requires generating new face expression while preserving person identity and scene features. In this work, we propose advancements for recent novel methods of accurate face-reenactment synthesis. We present results using a flexible generation module, and compare different families of encoding backbones, introduce identity loss to preserve a person's identity in image generation with state-of-the-art models in the deep face-recognition domain. We also provide improvements in the training procedure and test on approach weaknesses.

Acknowledgements

I wish to thank my supervisor, Markian Kostiv, who has tactically guided and encouraged me during the work on thesis. Thank to Applied Sciences Faculty at the Ukrainian Catholic Univeristy and specially to Oles Dobosevych who strategically ensured the work is well planned and continues smoothly. I want to express my gratitude also to my collegue Ivan Kosarevych with whom we worked on the related research. Thank to SoftServe and in particularly Mykola Maksymenko for the research idea. Finally, I would like to mention the invaluable support and feedback of my friends and family.

Contents

List of Figures

List of Tables

List of Abbreviations

GAN	**G**enerative **A**dversarial **N**etwork
cGAN	conditional **G**enerative **A**dversarial **N**etwork
ANN	**A**rtificial **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
FPN	**F**eature **P**yramid **N**etwork

List of Symbols

p_{data}	The data generating distribution
$\hat{p}_{\text{data}}$	The empirical distribution defined by the training set
x	The image sample of a source
x'	The image sample of a target
$\hat{x}$	The generated image, *i.e.* reenacted x'

Chapter 1

Introduction

FIGURE 1.1: Visualized aim of face-reenactment

The goal of face-reenactment is to transfer facial expression from the source image to the target face (Fig. 1.1). Re-enacted image of a face should have a realistic expression, same identity, lightning and background. These factors influence the problem complexity. One has to optimise multiple tradeoffs (realism-identity/background preservation) in order to solve this problem.

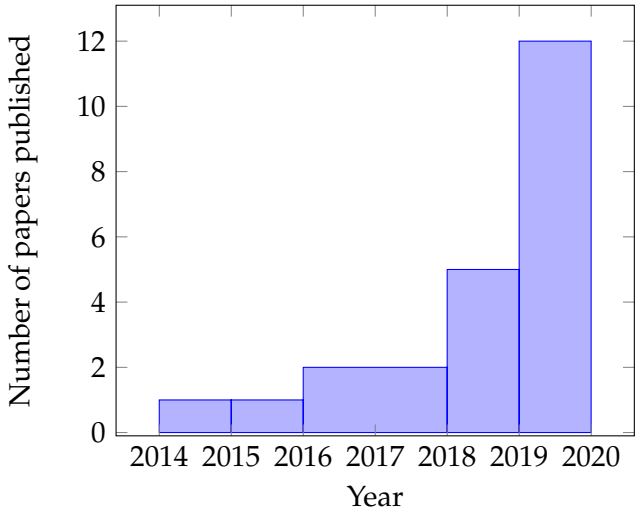


FIGURE 1.2: Histogram of the number of found research papers on face reenactment and synthesised images detection as of April 2020

Face reenactment, also known as puppetry, has myriads of use cases: Multimedia, Augmented Reality, Virtual Reality, Cinematography, Animation of computer graphics (CG) and Face forensics, to name a few. For instance, the film industry uses visual manipulation tools in post-production. It also applies cutting-edge systems

to bring deceased actors "back to life", known as "digital resurrection", for creating film sequels.

Like any other technology, face-reenactment could be used either for good (bona fide) purposes or could be misused. Face reenactment as technology is mainly a computer vision problem. However, it also touches forensics and ethical aspects. In the range between 2014 – 2020 these algorithms gain popularity (Fig. 1.2), in particular among researchers. One may see this fact by looking at the number of research papers published, mainly doing a reenactment of faces. [48, 68, 86, 65, 53] or related, such as detection of generated images [47, 45, 89].

Chapter 2

Related Works

2.1 Image-to-Image Translation

Computer vision algorithms for image processing, similarly to a language translation, can be viewed as a "translation" of an input image into a processed output image. Translation tasks may vary, for instance, translation of RGB image into *e.g.* a grayscale, edge map, semantic mask, *etc*. Automatic translation from one image state into another, given sufficient training data, is defined as image-to-image translation problem. [22]

Face-reenactment could be interpreted as an image-to-image translation problem. Recent studies [22] propose methods for translating/mapping images from one domain to another. However, one should have pairs of images from these domains.

Isola *et al.* [22] condition translation process on some data (*e.g.* edge map images), condition on to both the generator and discriminator as they observe an input image; in terms of GANs it is called conditional GANs (cGANs) [44].

Datasets with paired images are rare due to the high cost and time investments needed to create them. Zhu *et al.* [91] proposed solution to this limitation in Unpaired Image-to-Image translation. Moreover, with this work, one can translate images between domains even if there cannot be a pair-sample in real-life (*e.g.* pair image of oranges on the apple tree). Other works [64, 35, 36, 77] also address this issue.

Pumarola *et al.* [51] synthesise images of higher resolution than previous works, albeit for the price of being less robust.

Images can contain very different head poses, face expressions and lighting conditions. Existing image-to-image translation models could generate unnatural images in extreme conditions such as large poses (Fig. 21 in [22]), or fail on unseen images [51], changing identity of a face.

Inspired by recent studies [48, 86], we incorporate adversarial training with modified discriminator and additional losses, which is successfully adopted in these works, to synthesise realistic face images.

2.2 Face Reenactment

Main objectives of face-reenactment: 1) facial expression transfer; 2) identity preservation; 3) background and illumination retention.

Recent works in face-reenactment commonly use the facial expression of a source face to condition the expression of the synthesised face image. Following ReenactGAN [78] approach, we define the expression of a face with interpolated landmarks. Wu *et al*. use these solid-line landmarks as a medium to map from the face image to the generated face by adapting target and source landmarks.

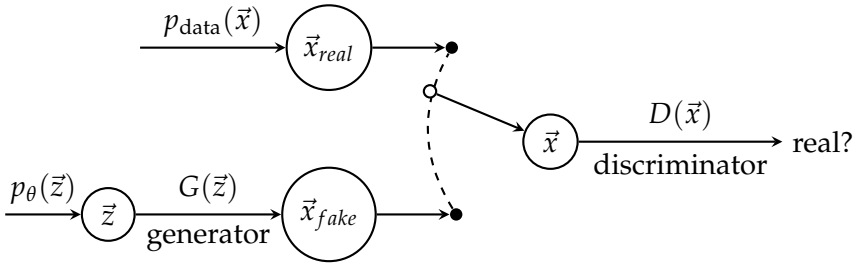


FIGURE 2.1: Diagram of a standard GAN

Another studies *e.g.* [88, 78] apply ANN-based adaptation modules in their pipeline. Delaunay triangulation [33] is also utilised in face-reenactment [48], and in the related problem of face-swap [79].

The other objective is to preserve the identity of the target person in the generated face image. Some works [78] relate to identity in terms of person face texture and apply CycleGAN [91] to maintain the identity of the synthesised face image. We advance the identity constraint via applying models from face recognition domain.

To prevent blurry background in synthesized images [73] researchers apply contextual [41] or perceptual loss [24] between the generated and target images.

Face2Face [66] produce photo-realistic face reenactment but require large expression variability in the input target-video sequence.

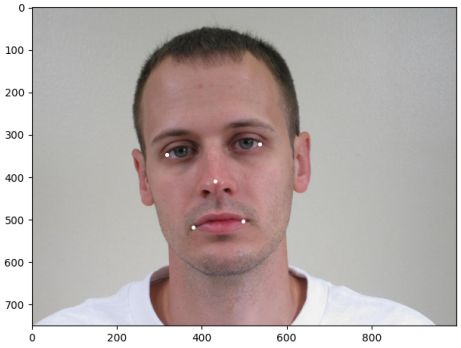
For one-to-one face-reenactment Zakharov *et al.* [85] propose virtual talking heads generation approach with high realism. However, the limitation of this method is in the mimics representation (*e.g.* gaze-reenactment is not possible) and many-to-many reenactment scenario produces noticeable identity mismatch. The approach also requires retraining the model for each new person.

Recent work [58] achieves higher generalisation in the generation of reenacted video. This model feeds one image for a target and a video sequence as a source driving video. The approach generalises well to the range of objects *i.e.* when trained on a set of videos with objects of the same category, their method can be applied to any source video or target image with the object of this class.

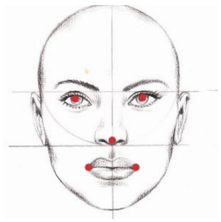
2.3 Generative Adversarial Network

Standard/"Vanilla" GAN [17] (represented in Fig. 2.1) consists of two networks: Generator and Discriminator. The Generator feeds latent random variable (noise vector) $p_\theta(\vec{z})$ and ouputs sample $\vec{x}_{fake}$ which should resemble $\vec{x}_{real}$ from the real data distribution $p_{\text{data}}(\vec{x})$. The Discriminator is a binary classification model that consequently feeds $\vec{x}_{real}$ and $\vec{x}_{fake}$, and ouputes likelihood value whether the input was real (close to 1.0) or not (close to 0). Whole GAN framework is a minimax (Eq. 2.1) two-player game [46, 23]. Each subnetwork *i.e.* generator $G(\vec{z})$ and discriminator $D(\vec{x})$ tries to beat each other.

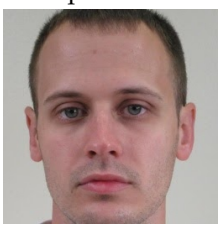
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.1}$$



(A) Five face landmarks estimation



(B) Matching with predefined pattern



(C) Applying similarity transformation to get the normalized face

FIGURE 2.2: Face normalization process using similarity transformation

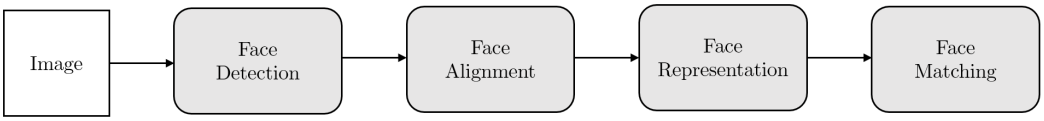


FIGURE 2.3: Face recognition pipeline [69]

2.4 Face Recognition

Face identity preservation is an important part of a face-reenactment pipeline. Face recognition as research blossomed in the seventies and today methods range from classical to more novel approaches using deep learning trained on massive datasets [69].

General face-recognition pipeline consists of four steps (Fig. 2.3). Face detection [26] is the first step - finding the coordinates of a bounding box representing the location of the face on an image. Many algorithms exist for face detection [1, 2, 84, 49]. Methods range from more classical [8, 32], *e.g.* Viola-Jones method [71], to more modern ANN-based [87, 83, 13, 34, 7].

Second step is face landmarks estimation and face normalization (alignment) procedure (Fig. 2.2). After this step, we obtain aligned face image.

During the third step, we map face image to some face representation, *i.e.* embedding vector in most cases. Finally, this latent vector we can compare to some other embedding vector. If the vectors are similar, we can recognise with some probability the face identity of the person on the image.

To conclude, Face identity is represented as an embedding vector, typically of size 128, 256 or 512.

Chapter 3

Methods

3.1 Network Architecture

Our pipeline structure [30] is similar to the standard GAN (2.3). It consists of a Generator (described in 3.1.1) and a Discriminator (3.1.4).

Images of a source (x) and a target (x') are propagated through the Generator (Fig. 3.2) to produce the reenacted image ($\hat{x}$) of the target. First, the Discriminator computes adversarial loss, which we use to improve the generator in producing more accurate expression. Second, Identity loss (3.2.5) ensures same identity for the synthesised face image. Finally, perceptual content loss (3.4.1) is responsible for overall image content preservation.

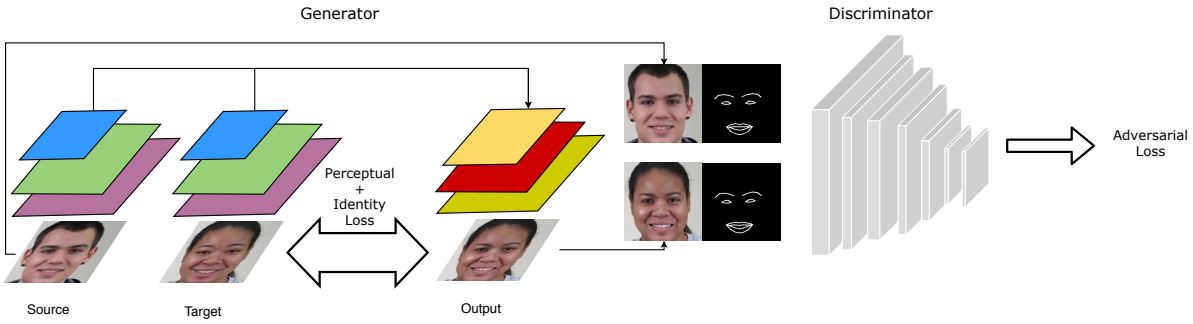


FIGURE 3.1: High-level full system architecture diagram

3.1.1 Generator

In our work we consider FPN [37]-based (Fig. 3.2) generator. Both source and target images are fed into separate FPN-like feature extraction modules (encoders) to generate multi-scale feature maps from source and target images. During our experiments with siamese [4] encoders we observed (Tab. 5.2) that separate encoders for source and target extract more unique features which yield better results.

3.1.2 FPN

Inspired by recent advancements in semantic segmentation [43] we chosen Feature Pyramids [37] as the feature extraction module.

In order to utilise better both low and high feature resolutions the FPN (Fig. 3.3) consists of: a bottom-up (in the left part of the Fig. 3.3), top-down (in the center) and lateral connections. The bottom-up pathway is a CNN for extracting features from the image. It produces multi-scale feature maps with a different semantic value which are then redirected to the top-down path.

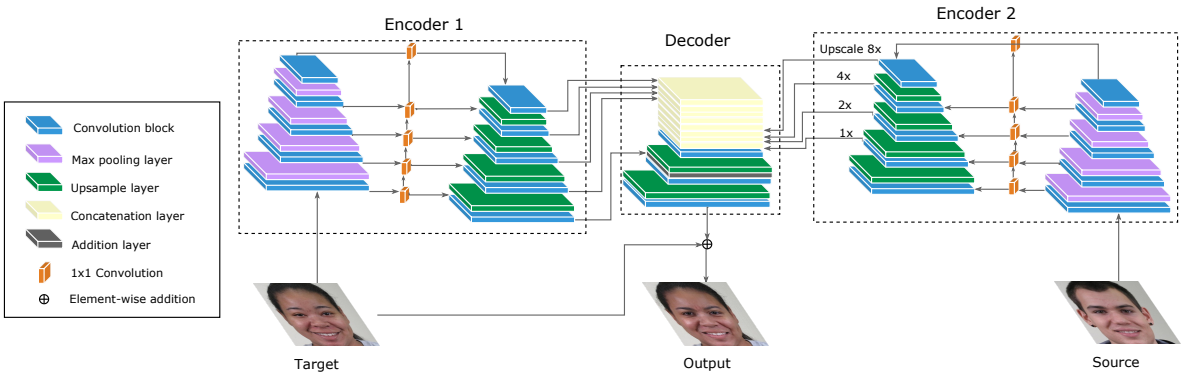


FIGURE 3.2: High-level FPN-based Generator architecture diagram

In the top-down pathway semantically "rich" feature maps increase in resolution. The top-down path could be viewed as a hierarchical "super-resolution" network. Some location information is lost due to downsampling operations.

To recover it we add 1×1 lateral connections after each scale in the top-down path from a corresponding bottom-up feature map.

Further, those multi-scale features from the top-down path could be upscaled, concatenated and processed by convolution to produce "enriched" with local and global context information prediction.

In our pipeline, we receive five separate feature maps from target and source FPN-Encoders. Then, in the Decoder, top four maps after upscaling and concatenation in FPN are propagated through a series of convolutional and upsampling layers. After first upsampling, we add the last (fifth) bottom-most feature map from the target Encoder. In the end, the target image, as a skip connection, is added to the result of the Decoder to get the final generated the face.

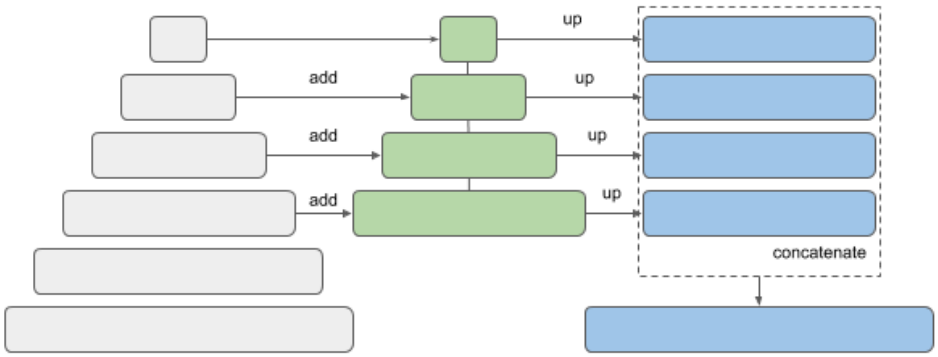


FIGURE 3.3: Skeleton of FPN. Figure from [82]

We can easily swap FPN-"backbones", *i.e.* classification models for feature extraction (bottom-up pathway). To conclude, the FPN module is an efficient feature

extractor and flexible in a sense depending on "backbone" it can train faster or produce results of higher quality.

3.1.3 Feature Extractor "Backbones"

InceptionResNetV2

Inception-ResNet-v2 [62] is a modification of InceptionV3 [63]. This model achieved top-1 accuracy of 80.4% on ILSVRC2012 [11].

ResNext WSL

ResNext WSL [40] is a ResNext101 [80] model pre-trained with billion scale weakly-supervised data and fine-tuned on ImageNet. Nearly 1 billion (940 million) image-data this model was trained on were collected from social media Instagram. Users of this platform "voluntarily" put hashtags on the images they upload to their profiles. Most of these profiles are public. Those image-hashtags Mahajan *et al.* [40] used as labels for training ResNext model on the image classification task. Different capacity variations of the model scored from 82.2% to 85.4 top-1 accuracy on ImageNet [11]. In our experiments, we use WSL-ResNext-101-32x8d (32 groups of convolutions whose input and output channels are 8-dimensional) which has 88 million of parameters.

MobileNetV2

MobileNetV2 [56] is a lightweight mobile ANN architecture that reached top-1 ImageNet accuracy of 72.0% with only 3.4 million of parameters.

3.1.4 Discriminator

Markovian discriminator called PatchGAN proposed by Isola *et al.* [22] works on patches of the image, producing sharper results.

Our discriminator is similar to PatchGAN architecture, but with additional information on landmarks. Discriminator feeds the full input image with concatenated dense interpolated landmarks, so the input tensor shape is 4D (RGB and a "landmarks" channel).

3.2 Identity

3.2.1 Face Identity Estimation

To achieve a high-quality reenactment effect, we need to preserve the original face geometry and illumination. Primarily, it is crucial when we conduct a reenactment of a person different from a source one.

To represent a person's identity, we encode it in a vector of features, *i.e.* embedding vector. In order to extract such identity embedding vector from the image, we use a face-recognition model. We want the face-recognition model to have a well structured latent space. The distance between the two latent vectors must be small if the embeddings are of the same person, while the distance between the vectors of distinct individuals to be large. The concept illustrated in Fig. 3.4.

We conduct experiments with different identity loss models to choose the one with the most sattisfactory results: CosFace [72], SphereFace [39] and ArcFace [12].

3.2.2 SphereFace

Many loss functions for face recognition use a Euclidean-based margin to make a better separation between feature embeddings. Softmax applied to face recognition problem have an intrinsic angular distribution of learned features.

SphereFace [39] uses CNN and adopts euclidean margin idea but in angular terms. It proposes angular margin to softmax, named angular softmax (A-Softmax) loss for face feature discrimination. Method map face image embedding onto the hypersphere with good enough latent space properties to recognise identity.

This face recognition model achieves 99.42% accuracy on Labeled Faces in the Wild (LFW) [20] and 95% on YouTube Faces (YTF) [76] database.

3.2.3 CosFace

Wang *et al.* [72] introduce in CosFace large margin cosine loss (LMCL). Similarly to SphereFace (3.2.2) this loss is another angular margin technique, which improves on the weaknesses of the SphereFace. CosFace defines the decision margin in cosine space instead of angular space. It adds L2 normalisation for features and weights to addresses the problem of different margin for different classes which exists in SphereFace.

CosFace shows accuracy of 99.73% on LFW and 97.6% on YTF.

3.2.4 ArcFace

We adopt Additive Angular Margin Loss (ArcFace) [12] as our baseline identity model. It has exact correspondence to the geodesic distance.

ArcFace has better separable decision boundary than SphereFace (3.2.2) and CosFace (3.2.3). It has constant linear angular margin, while other works have only a nonlinear angular margin. One may see the result of MSE of Arcface embeddings illustrated in Fig. 3.4.

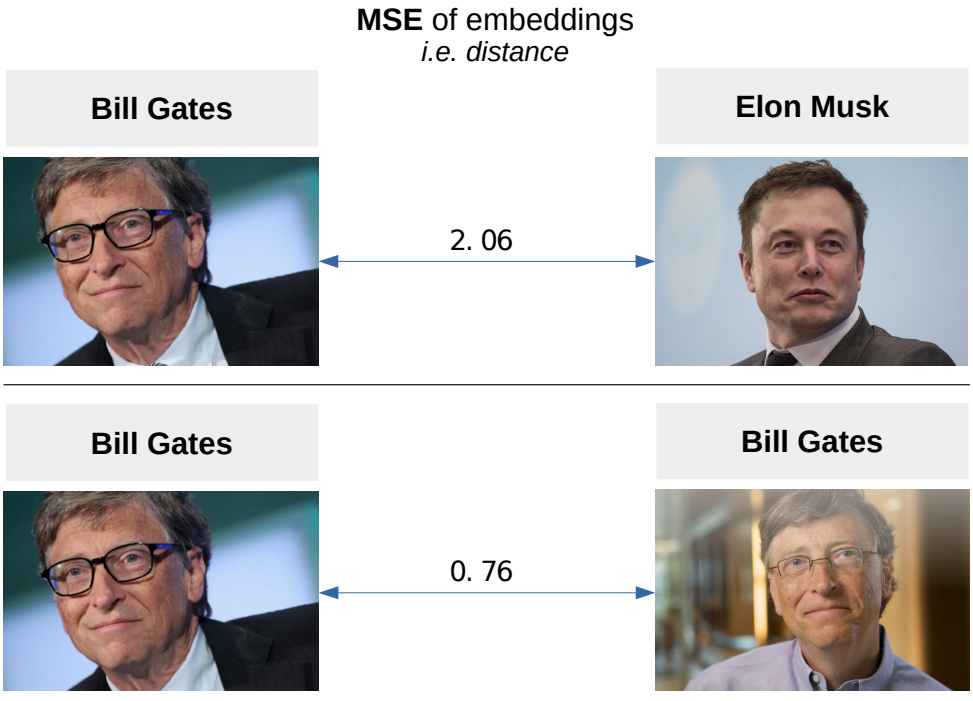


FIGURE 3.4: MSE of "ArcFace embeddings"

ArcFace scores 99.83% LFW and 98.02% on YTF dataset.

3.2.5 Identity loss

To make synthesised identity as same as possible to the target face identity we employ identity loss. It is a distance between the target and generated face feature embeddings learned by separate Deep CNN with the use of recently proposed state-of-the-art loss in the face-recognition domain (described in 3.2.1).

In our study, we adapted and compared ArcFace, CosFace and SphereFace models to generate face embedding. We use open-source pre-trained models.

The identity loss L_{identity} prevents the generator from modifying face characteristics of a target person. Here we use a concept of a "similarity" function.

$$d(\text{img}_1, \text{img}_2) = \text{degree of difference between images} \tag{3.1}$$

Having image of a face (img_1 we want to compute a similarity (distance) to some other face (img_2).

To measure the identity dissimilarity between target x' and synthesised $\hat{x}$ image we compute the following distance between the embedding vectors $e_{x'}$ and $e_{\hat{x}}$ as the identity loss:

$$L_{\text{identity}} = \sum (e_{\hat{x}} - e_{x'})^2 \tag{3.2}$$

This loss we use when training with ArcFace model (3.2.4) for face identity estimation.

When we train with CosFace (3.2.3) or SphereFace (3.2.2) we employ cosine distance as the models were trained using such distance. The L_{identity} in this case is as follows:

$$L_{\text{identity}} = 1 - \cos(e_{x'}, e_{\hat{x}}) \tag{3.3}$$

3.3 Face Normalisation

Face normalisation (alignment) is important in our pipeline since it is sensitive (5.1.4) to head scale and its position on an image. We conduct face alignment (Fig. 2.2) as a preprocessing step during dataset generation procedure. Face normalisation procedure makes a face on different images appear similar to a predefined template (reference).

We apply normalisation on the raw image in order to obtain aligned face. The five facial landmark points (eyes, nose and two mouth corners; Fig. 2.2a) for each image in the dataset. Then we choose a type of transformation that makes landmarks from the template and the image the most similar. There are euclidean, similarity, affine and projective transformation (Fig. 3.5). We choose similarity transformation, which in contrast to rigid (euclidean) transformation also includes scaling, therefore does not preserve the distance between points. Similarity transformation includes rotation, translation and scaling. It produces more natural-looking faces than other types of transformations. Finally, we add some padding to obtain a canonical (aligned) cropped face of size 256×256 pixels.

We use Dlib [27] for face landmarks estimation. Landmarks are estimated twice in our pipeline: 1) during dataset generation; 2) before feeding the discriminator (mentioned in 3.1.4, 3.4).

Dlib returns 68 face landmarks as keypoints in cartesian-coordinates, (x, y) position on an image. During the first landmarks estimation, we pick only five of them to

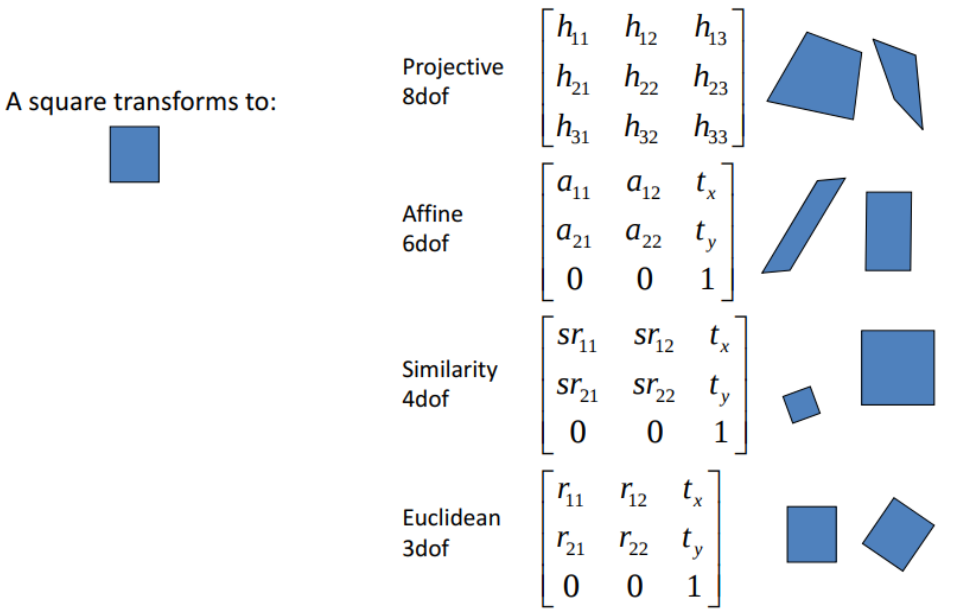


FIGURE 3.5: Types of transformations

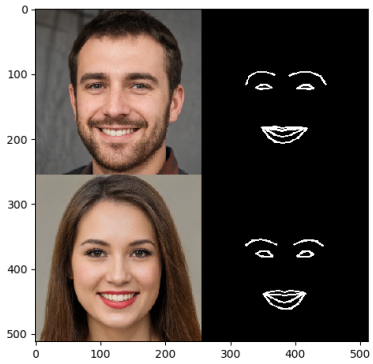


FIGURE 3.6: Examples of interpolated face-landmarks

normalise/align the face. Before feeding the discriminator we interpolate/rasterize landmarks (Fig 3.6) following the idea of Wu *et al.* [78].

3.4 Network Training

GANs are notorious for being hard to train [42]. Per one feedforward training path, we pull two images from the training dataset at random: a source (x) and a target (x').

Our discriminator feeds face image concatenated with rasterised landmarks, similarly to [57]. First, the discriminator calculates the value for the source image and its landmarks. Second, it processes the synthesised image ($\hat{x}$) concatenated with landmarks of the source (x), to calculate the "accuracy" of expression transfer.

In this way, our discriminator learns to tell the difference between real and synthesised facial expression. We then use discriminator skill to penalise the generator in order to produce face with the correct expression.

Additionally, identity loss and content loss penalise the generator on identity and image content dissimilarity.

3.4.1 Content loss

To constrain the proposed model in retaining the background, we adopt some representation learning approach for computing content loss. We use Johnson *et al.* [24] perceptual loss. Using pre-trained on ImageNet [11] VGG-19 network [59] we calculate MSE (squared ℓ^2) loss not only between raw pixels to preserve low-level details, but also between features maps to preserve general content, to preserve semantic information.

The combination of two MSE parts produce more "eye-pleasing" results:

$$L_{\text{content}} = 0.06 \cdot l_{\text{feat}}^{\phi,\text{relu3\_3}} + 0.5 \cdot \text{MSE} \tag{3.4}$$

where ϕ is the VGG-19, $l_{\text{feat}}^{\phi,\text{relu3\_3}}$ is the feature reconstruction loss *i.e.* squared ℓ^2 norm between feature representations of VGG-19 relu3_3 layer; MSE is the squared ℓ_2 norm between raw images (per-pixel difference).

3.4.2 Adversarial loss

For adversarial loss, we have chosen RaGAN-LS loss function from the family of Relativistic average GANs (RaGANs) proposed by Jolicoeur-Martineau [25], which generate higher quality data than non-reltivistic ones.

Therefore adversarial loss L_{adv} is as follows:

$$\begin{aligned} L_{\text{adv}} = \mathrm{L}_G^{\text{RaLSGAN}} = \mathbb{E}_{x_\gamma}[(D(x_\gamma) - \mathbb{E}_{(x,x')} D(G_\gamma(x,x')) + 1)^2] \\ + \mathbb{E}_{(x,x')}[(D(G_\gamma(x,x')) - \mathbb{E}_{x_\gamma} D(x_\gamma) - 1)^2] \end{aligned} \tag{3.5}$$

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.

3.4.3 Full Objective

Generator's objective

Full objective combines three losses scaled by correpsonding λ (4.1): content (3.4.1), adversarial (3.4.2) and identity (3.2.5).

The final objective formula is as follows:

$$L_{\text{total}} = \lambda_{\text{content}} \times L_{\text{content}} + \lambda_{\text{adv}} \times L_{\text{adv}} + \lambda_{\text{identity}} \times L_{\text{identity}} \tag{3.6}$$

Discrminator's objective

Finally, discriminator objective function is as follows:

$$\begin{aligned} \mathrm{L}_D^{\text{RaLSGAN}} = \mathbb{E}_{x_\gamma}[(D(x_\gamma) - \mathbb{E}_{(x,x')} D(G_\gamma(x,x')) - 1)^2] \\ + \mathbb{E}_{(x,x')}[D(G_\gamma(x,x')) - \mathbb{E}_{x_\gamma} D(x_\gamma) + 1)^2] \end{aligned} \tag{3.7}$$

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.

Chapter 4

Experiments

We conduct experiments with different generator encoder types, generator feature extractor backbones, identity loss models, alignment methods. We evaluate three components, which are significant for face reenactment: 1) image realism, 2) expression accuracy, 3) identity preservation.

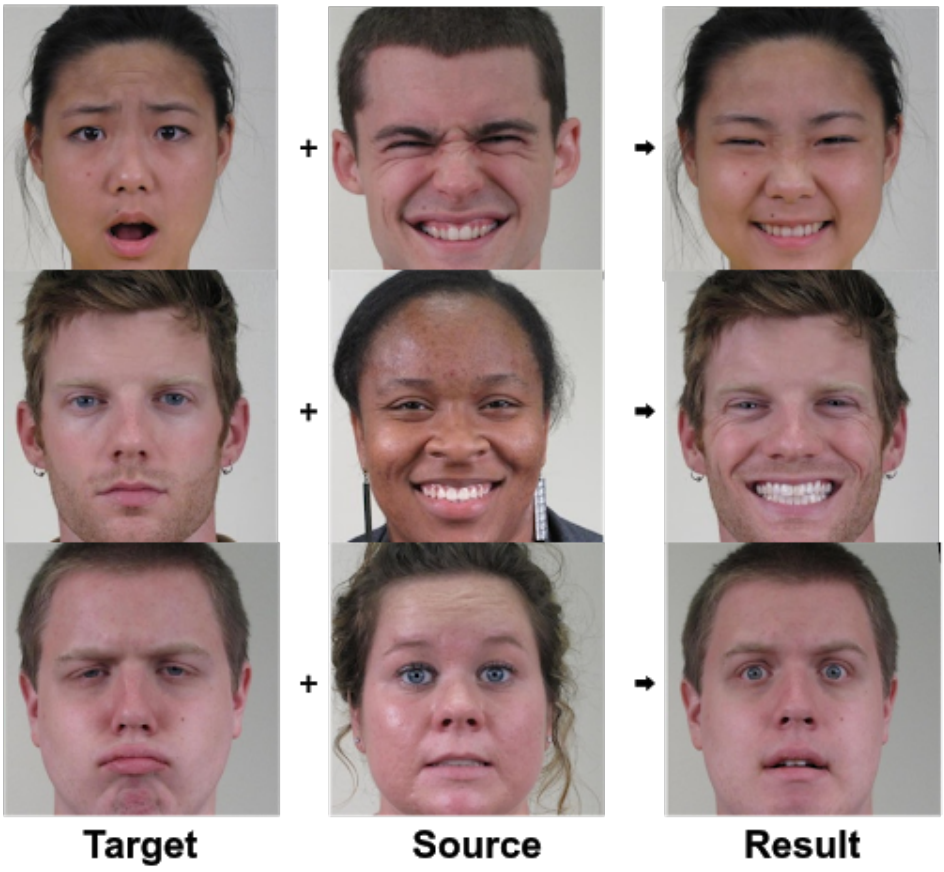


FIGURE 4.1: Visualisation of many-to-many face-reenactment

Results in these three components may vary depending on different reenactment scenarios. This work explores two possible scenarios. For every scenario, we created the corresponding dataset.

1. *"Many-to-many"* - source and target identities and expressions are different, randomised (*i.e.* many identities) in the dataset (Fig. 4.1);

2. *"One-to-one"* - source and target identities are the same, facial expressions are different (Fig. 4.2); This case may be used when identity in the source image

changed his hair or wearing a hat. In such a case the task is to generate an face image of the same identity with expression from the source but with the look, background of the target.

The first case is the one in which we are most interested. It is the most challenging so far, mainly in identity and background preservation. Within the second scenario, we study whether self-reenactment, *i.e.* of the same person, can improve the overall quality of the image and modify a person's identity less than in a *"many-to-many"* case. However, we note that for the best possible results fine-tuning on particular people is recommended.

FIGURE 4.2: Visualisation of one-to-one face-reenactment. (Source image is different from the result image in the real-world application)

4.1 Implementation Details

Experimentaly the following coefficients produced the most satisfactory results: $\lambda_{\text{content}} = 0.01$, $\lambda_{\text{adv}} = 0.001$, $\lambda_{\text{identity}} = 0.001$ (for Formula 3.6) . The coefficients were chosen to bound the loss values, to insure the gradients from being too stochastic and ensure there are no exploding gradients (a common problem in GANs [18]).

For this work, we are using Pytorch [50] as our main machine learning framework, Pytorch Lightning [15] for better scalability and Hydra [81] for configuration management. The experiments were trained from scratch on NVIDIA RTX 2080 Ti, GTX 1080, Tesla K80, Tesla M60 GPUs using the Adam solver [28] with a range of batch sizes: $1, 2, 4, 8$. The learning rate both for discriminator and generator is initially set to $1e{-}4$ with reducing learning rate down to $1e{-}7$ when a metric has stopped improving.

For monitoring training, experiments tracking and metrics logging we extensively use Weights & Biases [3] (`https://www.wandb.com/`) and CometML (`https://comet.ml`).

4.2 Datasets

For training, we selected Compound facial expressions of emotion (CFEE) dataset [14]. This database consists of 244 human identities; each of them expressed 26 emotions (Fig. 4.3). They are totalling to 6344 raw RGB images.

We split the dataset into three parts: train (85% of the dataset without test part), validation (15% of the dataset without test part), test (20% of the entire dataset).

FIGURE 4.3: 26 expressions of one person from CFEE dataset

4.3 Evaluation Metrics

It is important to note that there is no "silver-bullet" metric exists which could tell definitely that one image will be evaluated better over some other by humans. Therefore better metrics is still needed to be found. Nonetheless, we evaluate our method on standard metrics connected to image quality, realism and expression transfer accuracy, namely: FID, NMSE and CSIM.

Though the metrics mentioned above give approximate quantitative measures, it is still the visuals and human perception that may tell when the results are plausible. The problem this research aims to solve still lacks appropriate metrics.

4.3.1 FID

We apply Fréchet inception distance (FID) [19] to measure the variation and realism of generated images. It compares the statistics of generated samples to real samples. Lower FID is better, corresponding to more similarity between real and generated samples (Fig. 4.4), as measured by the distance between their activation distributions.

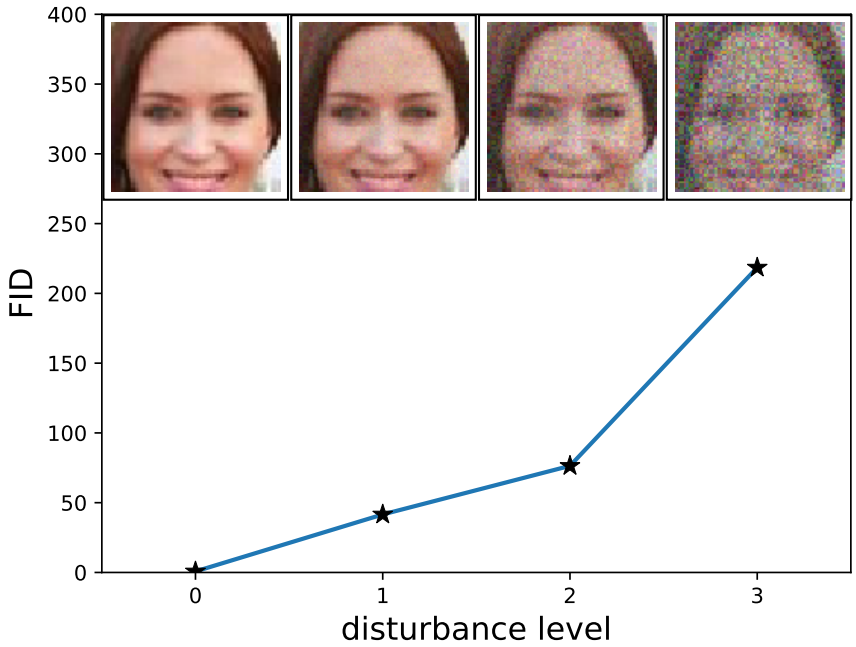


FIGURE 4.4: Gaussian noise effect on FID value. Figure from Heusel *et al.* [19]

This metric is widely used in image generation projects. It could be seen as an "analytical function" that tries to imitate capturing the variance of the generated samples as good as possible.

4.3.2 NMSE

For semantic evaluation, *i.e.* the correspondence between the source landmarks and the landmarks on the synthesized image, we employ NMSE (normalized (by inter-ocular (centroid of an eye) distance) mean squared error (times 100%)) commonly

used in many [38, 5, 61, 90, 52, 6] papers to compare semantic information *i.e.* face expression through landmarks.

The formula to calculate the NMSE between target (source) landmarks and generated (reenacted) landmarks is the following:

$$\text{NMSE} = \frac{\sum_{i=1}^{L} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{L \cdot \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2}} \cdot 100 \tag{4.1}$$

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.

4.3.3 CSIM

To compare identity preservation of the generated image, we use CSIM metric - cosine similarity between embedding vectors of a face-recognition network - CosFace (3.2.3).

$$\text{CSIM} = \frac{e_{x'} \cdot e_{\hat{x}}}{\|e_{x'}\| \cdot \|e_{\hat{x}}\|} \tag{4.2}$$

where $\|\cdot\|$ - Euclidean norm of a vector, $e_{x'}$ - embedding vector of a target face x', $e_{\hat{x}}$ - embedding vector of a generated face $\hat{x}$.

Chapter 5

Results

5.1 Quantitative results

Here we present metrics from our experiments. We want to notice that these metrics are a theoretical approximation to the final goal of a well-reenacted face image. These metrics may show general estimation for comparing failure case (if metrics are orders of magnitude different from some other model) with good enough results. During our experiments, we stumbled upon such cases where metrics were good; however, visuals were not. The idea is similar to notorious "panda-gibbon" case [16] - generated samples were such to satisfy some metrics while being full of visual artefacts. These are the main arguments why having both types of data we prioritised visual to numerical results. Nowadays, it is still mainly the visuals and human judgement that could select a better model. To conclude, a holistic view of both metrics and visual images should be used for judgement.

5.1.1 Comparison of Identity estimation models

We experimented with state-of-the-art models in the deep face-recognition domain for identity preservation of the synthesised face images.

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
ArcFace	7.33	**4.38**	0.71
CosFace	**4.43**	4.5	**0.86**
SphereFace	5.62	4.41	0.79
	"one-to-one"		
ArcFace	7.04	3.56	0.76
CosFace	**4.24**	**3.38**	**0.89**
SphereFace	5.5	3.46	0.84

TABLE 5.1: Quantitative resuls using different identity estimators

One may see from the Table 5.1 that CosFace (3.2.3) showed better identity preservation and higher realism in both reenactment scenarios. It scored best in "one-to-one" in all metrics, whereas in pure "many-to-many" scenario it gave way to ArcFace (3.2.4). However, due to the large difference in FID and CSIM, the difference in NMSE, in this case, may be neglected. The results presented here were measured in evaluation mode (5.5) for all models on the test dataset. All models were trained for

our proposed method with separate encoders (FPNMobileNetV2 (3.1.3) as a backbone)), batch norm [21] and batch size = 8 for 330 epochs for fair comparison.

To conclude, overall, CosFace identity model works best as identity loss for our method, improving realism and reenactment accuracy.

5.1.2 Comparison of siamese vs separate encoders in the Generator

We compared siamese [4] Generator (3.1.1) encoders, *i.e.* same weights for each encoder, with separate weights for source and target encoder.

In this experiment, one may see from the results in Table 5.2 that separate encoders achieved a better result in both scenarios in FID and CSIM, but scored lower in NMSE. Considering a significant gap in FID, *i.e.* image realism, we incorporated separate encoders in our main proposed method.

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
Siamese encoders	9.77	**4.94**	0.8
Separate encoders	**6.0**	5.04	**0.81**
	"one-to-one"		
Siamese encoders	9.73	**3.76**	**0.85**
Separate encoders	**5.82**	3.92	**0.85**

TABLE 5.2: Quantitative resuls using siamese encoders vs separate in the Generator

5.1.3 Comparison of Generator "backbones" for Encoder(s) feature extraction

In this work we have experimented with different feature extractors for our generator (3.1.1) encoder (3.1.2). Here we compare models trained with ArcFace idenity model (3.2.4), batch norm and separate encoders of different heaviness (capacities): lightweight (MobileNetV2 (3.1.3)), middleweight (InceptionResNetV2 (3.1.3)) and heavyweight (WSLResNext101 (3.1.3)).

As can be seen in Table 5.3, heavyweight WSLResNext model scored best in FID and CSIM in both scenarios but scored a little worse in NMSE. It should be noticed that here we present results for models of different capacities trained for the equal amount of epoch (steps), however heavier models take longer to train, so the result may not be final.

InceptionResNetV2 scored better than MobileNetV2, in this case, we neglect small difference in NMSE in "one-to-one" scenario. However, for final judgement, one has to see the actual synthesised images (Fig. 5.3).

5.1.4 Comparison of dataset preprocessing alignment methods

We experimented with different data preprocessing algorithms - here we compare our main face normalisation (3.3) procedure with another normalisation which uses distance between eyes in each face image.

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
FPNInceptionResNetV2	6.22	**4.27**	0.68
FPNWSLResNext	**5.64**	4.61	**0.77**
FPNMobileNetV2	6.43	4.3	0.72
	"one-to-one"		
FPNInceptionResNetV2	6.23	3.48	0.73
FPNWSLResNext	**5.24**	3.66	**0.82**
FPNMobileNetV2	6.22	**3.45**	0.77

TABLE 5.3: Quantitative resuls using different backbones for the generator

The latter method for normalisation works as follows: first we locate 68 facial landmarks with dlib [27]; second we center (rotate) and scale found landmarks around mid-point between eyes via calculating the rotation matrix and apply isotropic scaling; third we shift the landmarks to make all faces have constant position in the 256×256 image; finally we apply the computed transformation matrix to the image and obtain the normalised by interocular distance face image.

We observed that the second method we use for comparison is not optimal for our project since people in our dataset have different face compositions, *i.e.* different forms of eyes, lips, nose; different distance between eyes and lips, which as we saw from the result produce unnatural face images.

In Table 5.4 one may see that our main normalisation procedure (3.3) produces better results in all metrics.

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
Similarity transformation (3.3)	**3.44**	**6.02**	**0.86**
Interocular distance normalisation	4.11	7.49	0.85
	"one-to-one"		
Similarity transformation (3.3)	**3.3**	**4.47**	**0.89**
Interocular distance normalisation	3.75	6.89	0.87

TABLE 5.4: Quantitative resuls using different normalisation methods for dataset preprocessing

5.1.5 Comparison of batch sizes and normalisation layers

Here we compare training with batch normalistaion [21] and instance normalisation [70] layer, with respective batch sizes. From the Table 5.5 one may see that instance normalisation with a batch size of 1, scores better in FID and CSIM, however, batch norm outperformed instance norm in NMSE. We notice that training the

model with batch normalisation layer and bigger batch size is more efficient and faster.

Despite that model trained with batch norm performed worse in FID and CSIM, one may use it for training much larger dataset in the shorter amount of time than training with instance norm would take. We observed that setting the model in train mode (5.5) during inference could improve the results more in batch size than in instance norm.

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
Batch norm (batch = 8)	6.22	**4.27**	0.68
Instance norm (batch = 1)	**3.44**	6.02	**0.86**
	"one-to-one"		
Batch norm (batch = 8)	6.23	**3.48**	0.73
Instance norm (batch = 1)	**3.3**	4.47	**0.89**

TABLE 5.5: Quantitative resuls using different normalisation layers and batch size in network architecture

5.2 Qualitative Results

In Figure 5.1 one can see results of the proposed method trained with separate generator encoders, FPNWSLResNext101 (3.1.3) as a backbone, batch norm (batch size = 2), ArcFace Identity model (3.2.4). Additionaly, we show the result of this model on unseen targets in Figure 5.4.

5.3 Comparison with other works

In the domain of face-reenactment, it is hard to compare with others due to the lack of published datasets. Here we compare with Pix2PixHD (5.3.1) method. The results show advantages of our model both in metrics and visually. Our proposed method has better mimics reconstruction, plausible identity preservation and fewer artefacts.

5.3.1 Pix2PixHD

We compare with pix2pixHD [74] trained on CFEE dataset (4.2). As one may see from the Table 5.6, the model we propose has much better identity preservation, higher realism and plausible expression transfer. Even though Pix2PixHD has better NMSE, it has drastically different FID and lower CSIM.

5.4 Forensics

To evaluate our results we use close to state-of-the-art model from FaceForensics Benchmark [55] – Xception c40. The results are presented in Table 5.7. One may see that Xception c40 fails to discriminate on our data. We acknowledge that for the more fair evaluation fine-tuning of the aforementioned model on our data is required; however, the code for the model training was not released yet.

FIGURE 5.1: Qualitative results in "many-to-many" scenario

FIGURE 5.2: Qualitative results in “one-to-one” scenario

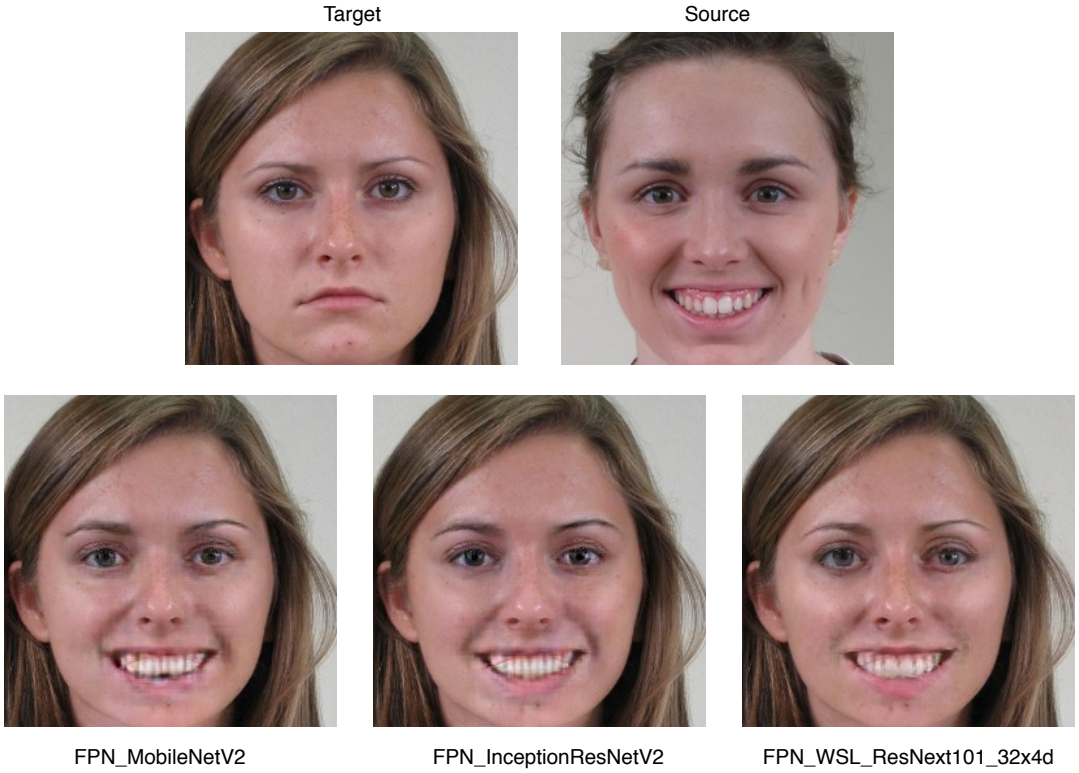


FIGURE 5.3: Comparison of generator backbones for encoders feature extractor

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many"		
Pix2PixHD	26.16	**4.17**	0.49
Our method	**6.66**	4.56	**0.77**

TABLE 5.6: Quantitative comparison of Pix2PixHD with Our proposed method

5.5 Interesting "eval" case

Experimenting we observed (Tab. 5.8) that sometimes during inference our generator being in `.train()` mode produce better results than in `.eval()` mode. We do not yet have the exact answer to why this happens.

When the model is set to `.eval()` mode all layers will behave in accordance, it is especially important when using some particular type of layers which have different behaviour during training and during the evaluation, *e.g.* Batch normalisation [21], Dropout [60]. We suppose that in our case this may be due to the use of Batch norm layer in the Generator.

During production use one has to ensure a robust and not stochastic behaviour, therefore setting the model into `.eval()` mode is a must. That is why for measuring our models with metrics, we set our generator in evaluation mode, although `.train()` could produce better outcomes.

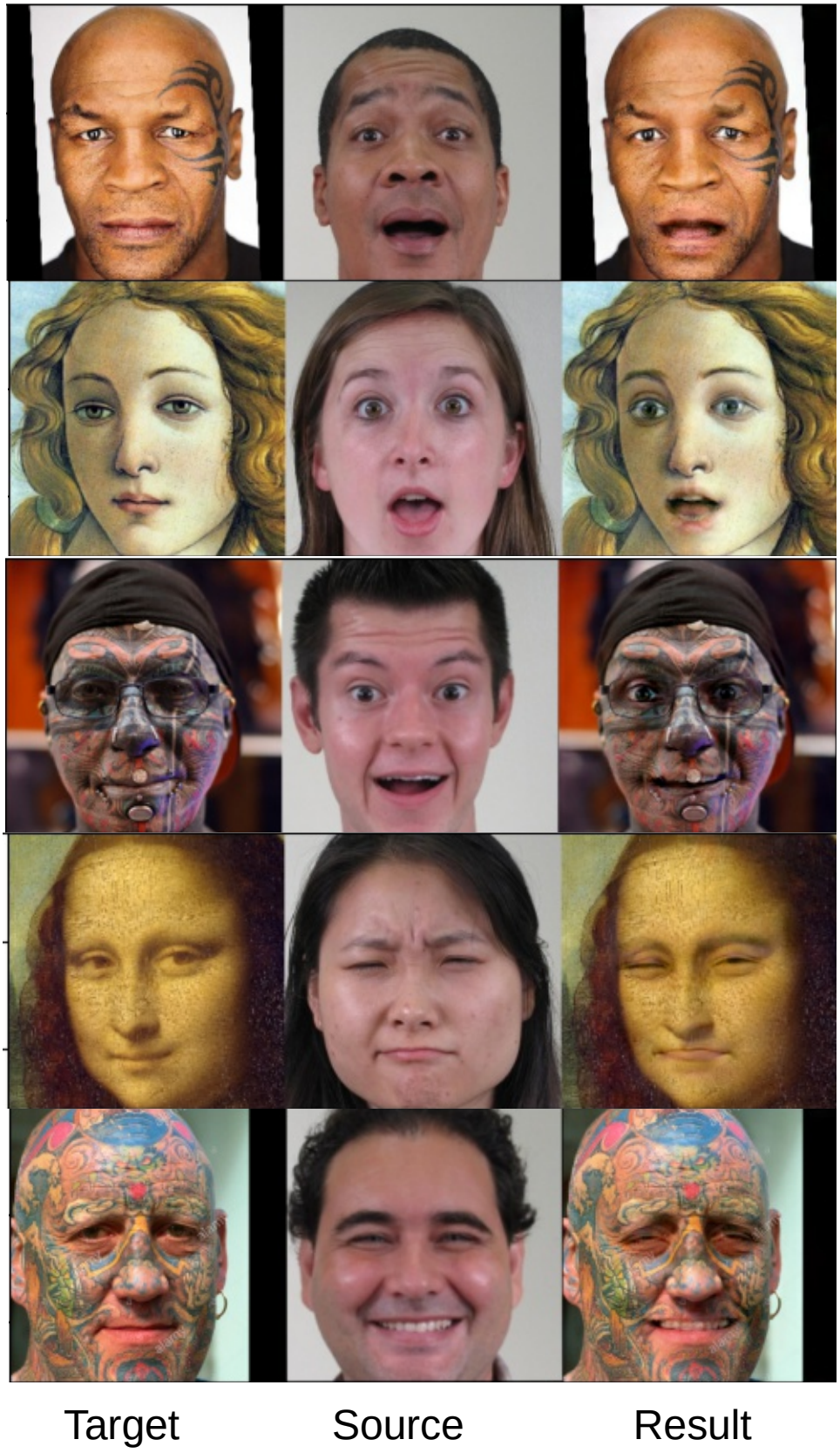


FIGURE 5.4: Results of Our proposed method on unseen targets

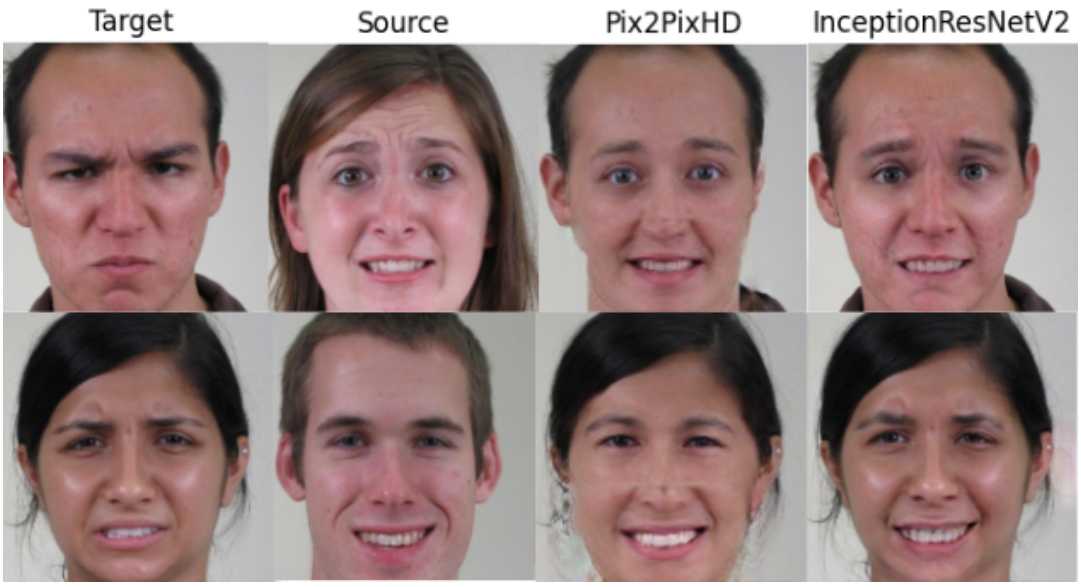


FIGURE 5.5: Qualitative comparison of our proposed model vs Pix2PixHD.

	Accuracy of the Xception c40
	"many-to-many"
Pix2PixHD	70%
Our proposed method	14%

TABLE 5.7: Classification accuracy of Xception c40

	FID ↓	NMSE ↓	CSIM ↑
	"many-to-many" w/o *"one-to-one"*		
`.eval()`	7.43	**4.31**	0.71
`.train()`	**5.13**	4.71	**0.78**
	"one-to-one"		
`.eval()`	7.27	3.42	0.76
`.train()`	4.93	3.66	0.83

TABLE 5.8: Quantitative resuls comparing FPNMobileNetV2 (3.1.3) generator trained with arcface (3.2.4) and batch size = 8 being in evaluation and training mode

Chapter 6

Conclusions

6.1 Ethical questions

Face-reenactment, like any other technology, could be used either for good or harmful purposes. There are examples of videos of persons appear to do or say things that did not happen. These generated videos have been named as DeepFakes. Deepfake title is a combination of words *i.e.* "deep learning", and "fake" [75].

Nowadays, an average human cannot discern bona fide image of a face from the generated fake face. Even some AI-based models fail (5.4) to discriminate.

Defense Advanced Research Projects Agency (DARPA) has started developing technology that can detect deepfakes [29]. Big tech giants also understand the emerging threat and encourage skilled professionals to participate in the challenge to build innovative new counter technologies for detecting deepfakes and manipulated media (`https://deepfakedetectionchallenge.ai/`) with a bounty of $1 M.

We plan to publish our synthesised dataset in order to help researchers train more accurate, more robust classification models for deepfake detection.

6.2 Our contribution

In this work, we proposed:

- an efficient and flexible FPN-based generator architecture for face-reenactment;
- separate encoders for more accurate feature extraction of the source and target image in the generator;
- a range of feature extraction backbones following the recent advancement in semantic segmentation and image classification domain;
- an identity loss for generated image identity features preservation based on the state-of-the-art model in deep face recognition domain;

We conducted experiments and provided quantitative and qualitative comparisons for architectural decisions and training procedures. Finally, we explored our approach in terms of forensics and showed the results using close to a state-of-the-art model for AI-generated content detection on our data.

Chapter 7

Future Advancements

- Big structural gap exists between the source face and the target (5.1.4). Therefore better ways of landmarks adaptation may be utilized, such as separate NN module (like in [88]) for mapping from actor landmark space onto avatar landmark space.
- Conduct Human validation of the results, using *e.g.* Amazon Mechanical Turks (AMT) [9].
- Use depth (3D) images of faces [54, 92] for more features.
- Train on larger dataset. Need more computing acceleration.
- Conduct hyperparameter tuning, to find best parameters.
- Research train-eval case (5.5) in more details. For instance, one may try training our pipeline with freezed batch norm layers.
- Compare our approach on Face forensics++ dataset [55] which consists of videos generated using four face manipulation methods, namely Deepfakes [10], Face2Face [66], FaceSwap [31] and NeuralTextures [67].

Bibliography

[1] Takeshi Agui et al. “Extraction of FaceRecognition from Monochromatic Photographs Using Neural Networks”. In: 1992.

[2] O. Bernier et al. “MULTRAK: a system for automatic multiperson localization and tracking in real-time”. In: *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*. Vol. 1. 1998, 136–140 vol.1. DOI: 10.1109/ICIP.1998.723444.

[3] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/.

[4] Jane Bromley et al. “Signature Verification Using a “Siamese” Time Delay Neural Network”. In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS’93. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1993, 737–744.

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

[6] Yu Chen et al. “Adversarial Learning of Structure-Aware Fully Convolutional Networks for Landmark Localization”. In: *arXiv e-prints*, arXiv:1711.00253 (2017), arXiv:1711.00253. arXiv: 1711.00253 [cs.CV].

[7] Cheng Chi et al. *Selective Refinement Network for High Performance Face Detection*. 2018. arXiv: 1809.02693 [cs.CV].

[8] Ian Craw, David Tock, and Alan Bennett. “Finding Face Features”. In: *Proceedings of the Second European Conference on Computer Vision*. ECCV ’92. Berlin, Heidelberg: Springer-Verlag, 1992, 92–96. ISBN: 3540554262.

[9] Kevin Crowston. “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, 2012, pp. 210–221. ISBN: 978-3-642-35142-6.

[10] deepfakes. *Deepfakes github*. Github. 2018. URL: https://github.com/deepfakes/faceswap.

[11] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009.

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

[13] Jiankang Deng et al. *RetinaFace: Single-stage Dense Face Localisation in the Wild*. 2019. arXiv: 1905.00641 [cs.CV].

[14] Shichuan Du, Yong Tao, and Aleix M. Martinez. "Compound facial expressions of emotion". In: *Proceedings of the National Academy of Sciences* 111.15 (2014), 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.

[15] W.A. et al. Falcon. *PyTorch Lightning*. https://github.com/PytorchLightning/pytorch-lightning. 2019.

[16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *arXiv e-prints*, arXiv:1412.6572 (2014), arXiv:1412.6572. arXiv: 1412.6572 [stat.ML].

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

[18] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 2017. arXiv: 1704.00028 [cs.LG].

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

[20] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, 2007.

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

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

[23] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. *What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?* 2019. arXiv: 1902.00618 [cs.LG].

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

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

[26] Inseong Kim, Joon Hyung Shim, and Jinkyu Yang. *Introduction Face detection*.

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

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

[29] Will Knight. *The US military is funding an effort to catch deepfakes and other AI trickery*. 2020. URL: https://www.technologyreview.com/2018/05/23/142770/the-us-military-is-funding-an-effort-to-catch-deepfakes-and-other-ai-trickery/.

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

[31] Marek Kowalski. *FaceSwap github*. Github. 2018. URL: https://github.com/MarekKowalski/FaceSwap/.

[32] A. Lanitis, CJ J. Taylor, and TF F. Cootes. "Automatic face identification system using flexible appearance models". English. In: *Image and Vision Computing* 13.5 (June 1995), pp. 393–401. ISSN: 0262-8856.

[33] D. T. Lee and B. J. Schachter. "Two algorithms for constructing a Delaunay triangulation". In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242. DOI: 10.1007/bf00977785. URL: https://doi.org/10.1007%2Fbf00977785.

[34] Jian Li et al. *DSFD: Dual Shot Face Detector*. 2018. arXiv: 1810.10220 [cs.CV].

[35] Yu Li et al. "Asymmetric GAN for Unpaired Image-to-Image Translation". In: *IEEE Transactions on Image Processing* 28.12 (2019), 5881–5896. ISSN: 1941-0042. DOI: 10.1109/tip.2019.2922854. URL: http://dx.doi.org/10.1109/TIP.2019.2922854.

[36] Jianxin Lin et al. "TuiGAN: Learning Versatile Image-to-Image Translation with Two Unpaired Images". In: *arXiv:2004.04634 [cs, eess]* (Apr. 2020). arXiv: 2004.04634. URL: http://arxiv.org/abs/2004.04634 (visited on 04/14/2020).

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

[38] Lingbo Liu et al. "Facial Landmark Machines: A Backbone-Branches Architecture with Progressive Representation Learning". In: *CoRR* abs/1812.03887 (2018). arXiv: 1812.03887. URL: http://arxiv.org/abs/1812.03887.

[39] Weiyang Liu et al. "SphereFace: Deep Hypersphere Embedding for Face Recognition". In: *arXiv e-prints*, arXiv:1704.08063 (2017), arXiv:1704.08063. arXiv: 1704.08063 [cs.CV].

[40] Dhruv Mahajan et al. *Exploring the Limits of Weakly Supervised Pretraining*. 2018. arXiv: 1805.00932 [cs.CV].

[41] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. *The Contextual Loss for Image Transformation with Non-Aligned Data*. 2018. arXiv: 1803.02077 [cs.CV].

[42] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. *Which Training Methods for GANs do actually Converge?* 2018. arXiv: 1801.04406 [cs.LG].

[43] Shervin Minaee et al. *Image Segmentation Using Deep Learning: A Survey*. 2020. arXiv: 2001.05566 [cs.CV].

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

[45] Paarth Neekhara et al. *Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples*. 2020. arXiv: 2002.12749 [cs.CV].

[46] John von Neumann. "Communication on the Borel notes". In: *Econometrica: journal of the Econometric Society* (1953).

[47] Huy H. Nguyen et al. *Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos*. 2019. arXiv: 1906.06876 [cs.CV].

[48] Yuval Nirkin, Yosi Keller, and Tal Hassner. *FSGAN: Subject Agnostic Face Swapping and Reenactment*. 2019. arXiv: 1908.05932 [cs.CV].

[49] Sakrapee Paisitkriangkrai, Chunhua Shen, and Jian Zhang. *Face Detection with Effective Feature Extraction*. 2010. arXiv: 1009.5758 [cs.CV].

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

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

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

[53] Yurui Ren et al. *Deep Image Spatial Transformation for Person Image Generation*. 2020. arXiv: 2003.00696 [cs.CV].

[54] Elad Richardson et al. *Learning Detailed Face Reconstruction from a Single Image*. 2016. arXiv: 1611.05053 [cs.CV].

[55] Andreas Rössler et al. "FaceForensics++: Learning to Detect Manipulated Facial Images". In: *International Conference on Computer Vision (ICCV)*. 2019.

[56] Mark Sandler et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2018. arXiv: 1801.04381 [cs.CV].

[57] Aliaksandr Siarohin et al. "Animating Arbitrary Objects via Deep Motion Transfer". In: *CoRR* abs/1812.08861 (2018). arXiv: 1812.08861. URL: http://arxiv.org/abs/1812.08861.

[58] Aliaksandr Siarohin et al. *First Order Motion Model for Image Animation*. 2020. arXiv: 2003.00196 [cs.CV].

[59] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 [cs.CV].

[60] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), 1929–1958. ISSN: 1532-4435.

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

[62] Christian Szegedy et al. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016. arXiv: 1602.07261 [cs.CV].

[63] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[64] Hao Tang et al. *AttentionGAN: Unpaired Image-to-Image Translation using Attention-Guided Generative Adversarial Networks*. 2019. arXiv: 1911.11897 [cs.CV].

[65] Ayush Tewari et al. *State of the Art on Neural Rendering*. 2020. arXiv: 2004.03805 [cs.CV].

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

[67] Justus Thies, Michael Zollhöfer, and Matthias Nießner. *Deferred Neural Rendering: Image Synthesis using Neural Textures*. 2019. arXiv: 1904.12356 [cs.CV].

[68] Justus Thies et al. "Headon". In: *ACM Transactions on Graphics* 37.4 (2018), 1–13. ISSN: 1557-7368. DOI: 10.1145/3197517.3201350. URL: http://dx.doi.org/10.1145/3197517.3201350.

[69] Daniel Sáez Trigueros, Li Meng, and Margaret Hartnett. *Face Recognition: From Traditional to Deep Learning Methods*. 2018. arXiv: 1811.00116 [cs.CV].

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

[71] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.

[72] Hao Wang et al. "CosFace: Large Margin Cosine Loss for Deep Face Recognition". In: *arXiv e-prints*, arXiv:1801.09414 (2018), arXiv:1801.09414. arXiv: 1801.09414 [cs.CV].

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

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

[75] Mika Westerlund. "The Emergence of Deepfake Technology: A Review". In: *Technology Innovation Management Review* 9 (Nov. 2019), pp. 39–52. DOI: 10.22215/timreview/1282.

[76] L. Wolf, T. Hassner, and I. Maoz. "Face recognition in unconstrained videos with matched background similarity". In: *CVPR 2011*. 2011, pp. 529–534.

[77] Ruizheng Wu et al. "Attribute-Driven Spontaneous Motion in Unpaired Image Translation". In: *arXiv:1907.01452 [cs]* (Oct. 2019). arXiv: 1907.01452. URL: http://arxiv.org/abs/1907.01452 (visited on 04/02/2020).

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

[79] Shengtao Xiao et al. "A Live Face Swapper". In: *Proceedings of the 24th ACM International Conference on Multimedia*. MM '16. Amsterdam, The Netherlands: Association for Computing Machinery, 2016, 691–692. ISBN: 9781450336031. DOI: 10.1145/2964284.2973808. URL: https://doi.org/10.1145/2964284.2973808.

[80] Saining Xie et al. "Aggregated Residual Transformations for Deep Neural Networks". In: *arXiv e-prints*, arXiv:1611.05431 (2016), arXiv:1611.05431. arXiv: 1611.05431 [cs.CV].

[81] Omry Yadan. *A framework for elegantly configuring complex applications*. Github. 2019. URL: https://github.com/facebookresearch/hydra.

[82] Pavel Yakubovskiy. *Segmentation Models*. https://github.com/qubvel/segmentation_models. 2019.

[83] Shuo Yang et al. *Faceness-Net: Face Detection through Deep Facial Part Responses*. 2017. arXiv: 1701.08393 [cs.CV].

[84] Baosheng Yu and Dacheng Tao. "Anchor Cascade for Efficient Face Detection". In: *IEEE Transactions on Image Processing* 28.5 (2019), 2490–2501. ISSN: 1941-0042. DOI: 10.1109/tip.2018.2886790. URL: http://dx.doi.org/10.1109/TIP.2018.2886790.

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

[86] Xianfang Zeng et al. *Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose*. 2020. arXiv: 2003.12957 [cs.CV].

[87] Faen Zhang et al. *Accurate Face Detection for High Performance*. 2019. arXiv: 1905.01585 [cs.CV].

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

[89] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. *Detecting and Simulating Artifacts in GAN Fake Images*. 2019. arXiv: 1907.06515 [cs.CV].

[90] Zhanpeng Zhang et al. "Learning and Transferring Multi-task Deep Representation for Face Alignment". In: *CoRR* abs/1408.3967 (2014). arXiv: 1408.3967. URL: http://arxiv.org/abs/1408.3967.

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

[92] M. Zollhöfer et al. "State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications". In: *Computer Graphics Forum* 37.2 (2018), pp. 523–550. DOI: 10.1111/cgf.13382. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13382.