

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Face recognition for forensic applications - Methods for matching facial sketches to mugshot pictures

Leonardo Gomes Capozzi



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ana Rebelo, PhD

Co-supervisor: João Ribeiro Pinto, Msc

July 25, 2020

Face recognition for forensic applications - Methods for matching facial sketches to mugshot pictures

Leonardo Gomes Capozzi

Mestrado Integrado em Engenharia Informática e Computação

July 25, 2020

Abstract

The traditional task of locating suspects using forensic sketches posted on public spaces, news, and social media can be a difficult task. Recent methods that use computer vision to improve this process present limitations, as they either do not use end-to-end networks for sketch recognition in police databases (which generally improve performance) or/and do not offer a photo-realistic representation of the sketch that could be used as an alternative should the automatic matching process fail. This dissertation proposes a method that combines these two properties, using a conditional generative adversarial network (cGAN) and a pre-trained face recognition network that are jointly optimised as an end-to-end model. While the model can identify a short list of potential suspects in a given database, the cGAN offers an intermediate realistic face representation to support an alternative manual matching process. Evaluation on sketch-photo pairs from the CUFS and CUFSF databases reveal the proposed method outperforms the state-of-the-art in most tasks, and that forcing an intermediate photo-realistic representation only results in a small performance decrease.

Agradecimentos

Aos meus orientadores, a Professora Ana Rebelo e João Ribeiro Pinto, pela sua atenção, pelas suas críticas construtivas, por estarem sempre disponíveis para me ajudar, pelo seu esforço, honestidade, inteligência e profissionalismo. Foram uns orientadores e pessoas exemplares e estou muito feliz por ter tido a oportunidade de trabalhar com eles.

Quero agradecer também ao Professor Jaime Cardoso, pelo seu tempo, pela atenção e pelas suas sugestões, que permitiram melhorar a qualidade do trabalho.

Aos membros dos grupos de Facial Analytics e VCMI, pelo bom ambiente de trabalho e pela sua ajuda nas reuniões.

Aos meus amigos, por estarem sempre presentes, por me apoiarem e pelos bons momentos ao longo destes anos.

Finalmente, e acima de tudo, à minha família. Aos meus pais, por estarem sempre lá para mim e por serem um exemplo a seguir, pelo seu apoio incondicional, por me motivarem a trabalhar pelos meus objetivos e por me ensinarem que com esforço e dedicação conseguirei alcançar os meus sonhos. À minha irmã, pelo seu apoio, pelos bons momentos e por, para além de irmã, ser uma amiga. Aos meus avós, bisavós, tios e primos por me apoiarem e acreditarem sempre em mim.

Leonardo Gomes Capozzi

“The difference between ordinary and extraordinary is that little extra.”

Jimmy Johnson

Contents

1	Introduction	1
1.1	Objectives	2
1.2	Contribution	2
1.3	Document Structure	2
2	Related Work	5
2.1	Background Knowledge	5
2.1.1	Convolutional Neural Networks	5
2.1.2	Generative Adversarial Networks	6
2.1.3	Conditional Generative Adversarial Network	7
2.2	State of the Art	8
2.2.1	Image Generation	8
2.2.2	Matching	11
2.2.3	Approaches to the sketch-to-face matching problem	13
2.3	Conclusions	17
3	Final Methodology	19
3.1	Network Architecture	19
3.1.1	Sketch-to-render generator	19
3.1.2	Matching network	22
3.2	Loss	23
4	Experimental Settings	25
4.1	Data	25
4.2	Pre-processing	26
4.3	Experiments	27
4.3.1	Experiment 1	28
4.3.2	Experiment 2	29
4.3.3	Experiment 3	29
4.3.4	Experiment 4	30
4.3.5	Experiment 5	31
4.3.6	Conclusions	32
4.4	Training	33
5	Results and Discussion	35
5.1	Realistic Generation Performance	36
6	Conclusions and future work	41

References**43**

List of Figures

2.1	Convolutional Neural Network used to detect handwritten digits.	6
2.2	Generative Adversarial Network diagram.	7
2.3	Conditional Generative Adversarial Network diagram.	8
2.4	The process of adding layers to the network to increase the resolution of the generated images.	9
2.5	Results obtained by training the network using the CELEBA-HQ dataset.	9
2.6	Pix2pix model.	10
2.7	Results obtained by training a model to transform edges to shoes.	10
2.8	Results obtained by training a model to transform Google Map images to satellite images and vice-versa.	11
2.9	Results obtained by using different loss functions.	11
2.10	Different methods developed over the years to improve face recognition.	12
2.11	Different loss functions used over the years for face recognition.	13
2.12	Decision margins of different loss functions.	13
2.13	Architecture from Iranmanesh et al. (2018)	14
2.14	Architecture from Kazemi et al. (2018)	14
2.15	Generated images using the method in Kazemi et al. (2018)	15
2.16	Architecture from Chao et al. (2019)	15
2.17	Generated images using the method in Chao et al. (2019) with different loss functions.	16
2.18	Generated images using the method in Osahor et al. (2020)	16
2.19	Architecture from Osahor et al. (2020)	17
3.1	Overview schema of the proposed method.	19
3.2	Network architecture of the generator.	20
3.3	Network architecture of the discriminator.	21
3.4	Comparison of the pix2pix discriminator (old) and the modified pix2pix discriminator (new).	22
3.5	Loss component for the matching part of the model.	23
3.6	Loss component for the generator of the sketch-to-render part of the model.	24
3.7	Loss component for the discriminator of the sketch-to-render part of the model.	24
4.1	Example images from CUHK Face Sketch database.	25
4.2	Example images from CUHK Face Sketch FERET database.	26
4.3	CUFSS dataset images before and after colorization using the DeOldify API.	26
4.4	CUFSS dataset images before and after cropping.	27
4.5	Images generated using the method described in experiment 1.	28
4.6	Images generated using the method described in experiment 2.	29

4.7	Images generated using the method described in experiment 3.	30
4.8	Images generated using the method described in experiment 4.	31
4.9	Images generated using the method described in experiment 5.	32
5.1	Images generated by our method using the CUFSF dataset.	38
5.2	Intermediate representation of the sketches using the CUFSF dataset.	38
5.3	Images generated by our method using the CUFS dataset.	39
5.4	Intermediate representation of the sketches using the CUFS dataset.	39

List of Tables

2.1	Different methods developed over the years for face recognition/matching.	12
5.1	Matching accuracy on the CUFSF dataset, using different methods to enhance the sketch (r.r.g.: realistic rendering generation).	35
5.2	Matching accuracy on the CUFS dataset, using different methods to enhance the sketch (r.r.g.: realistic rendering generation).	36
5.3	Rank-1 accuracy of different methods for sketch-face matching.	36
5.4	Comparison between different methods to enhance the sketch, using the CUFSF dataset (r.r.g.: realistic rendering generation).	37
5.5	Comparison between different methods to enhance the sketch, using the CUFS dataset (r.r.g.: realistic rendering generation).	37

Abreviaturas e Símbolos

NN	Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
CGAN	Conditional Generative Adversarial Network
CUHK	Chinese University of Hong Kong
CUFS	CUHK Face Sketch database
CUFSF	CUHK Face Sketch FERET dataset

Chapter 1

Introduction

When a crime is committed there are occasions when a picture of the suspect is not available. In these cases, an artist draws a forensic sketch of the suspect based on descriptions from eye witnesses. These sketches are typically posted in public places and in the media with the hope that someone in the public recognizes the suspect and helps to identify and locate them. This process is typically slow and inefficient to provide a service that needs to be quick and accurate. Therefore, there is a need to make this process faster and more efficient, so that the suspect can be located as quickly as possible.

Over the years, convolutional neural networks (CNN) have been very successful for several pattern recognition and computer vision tasks, including that of face recognition. Recent publications in this topic often report significantly improved accuracy in the matching process when using deep learning methodologies (Taigman et al., 2014; Sun et al., 2014; Parkhi et al., 2015b; Ranjan et al., 2017; Deng et al., 2018; Wang and Deng, 2018). However, face recognition based on photos or video is obviously an easier task than face recognition based forensic sketches, since a sketch might not be the most accurate representation of an individual, as it was drawn based on descriptions from eye witnesses (Pramanik and Bhattacharjee, 2013).

Sketch-to-face matching consists of matching a sketch to a real picture of a person. Typically the n most similar individuals to the sketch are chosen from a database of face images, to narrow down the list of suspects (n is an arbitrary number).

On the problem of sketch-to-face matching, several recent state-of-the-art methods have also used CNNs to match a sketch to the corresponding identity (Iranmanesh et al., 2018; Osahor et al., 2020; Kazemi et al., 2018; Chao et al., 2019). However, several of these do not take full advantage of the potential of deep learning as they are not end-to-end deep approaches. Jointly optimised end-to-end deep models are able to use all information carried by the input data as effectively as possible, as shown in the literature (*e.g.*, by the state-of-the-art in end-to-end face recognition). However, including processes that are manually tuned or separately optimised can cause dissonance between different processes and limit achievable performance.

One of the examples of separate processes in sketch-to-face matching is the prior transformation of a sketch to resemble a real face photo. Several literature approaches apply such transforma-

tions including, more recently, sophisticated adversarial methods based on CycleGANs (Kazemi et al., 2018) and cGANs (Osahor et al., 2020; Chao et al., 2019). Having a photo-realistic representation of the suspect’s face is a great advantage for manual identification when the automatic matching process fails to deliver useful results. Nevertheless, the separate optimisation of these processes will, as aforementioned, induce performance limitations.

1.1 Objectives

This dissertation tackles two important aspects of forensic sketch-to-face matching, that have not yet been addressed in the literature. The first aspect is avoiding the combination of separate processes that are individually optimised, which often limits achievable performance, instead proposing an end-to-end jointly optimised model. The second aspect is enforcing this end-to-end model to offer an intermediate representation that is photo-realistic, so the authorities have access to a realistic face rendering that will help manual identification of the suspect.

To achieve these, we propose an end-to-end model composed of a conditional generative adversarial network (cGAN) and a matching CNN that are jointly optimised. When trained, the model receives a sketch and returns a template that can be used for matching using simple distance metrics. Although the approach is end-to-end, the training strategy induces the cGAN to generate intermediate latent representations that look realistic and are similar to the corresponding real photographs. Hence, we avoid the performance limits often linked to non-end-to-end approaches, while retaining the intermediate realistic images that could help an alternative manual identification process.

1.2 Contribution

The method proposed in this dissertation improves current methodologies by developing an end-to-end model which optimizes the model as a whole. This improves the accuracy of the matching process, while also forcing an intermediate realistic representation of the sketch that can also be used with a manual matching method.

In case the realistic representation of the sketch is not necessary and the only concern is the matching accuracy we can remove the loss component related to the realistic representation and optimize the model for matching, which significantly improves the accuracy.

The work in this dissertation was included in an article that was submitted for the 2020 International Joint Conference on Biometrics (IJCB 2020).

1.3 Document Structure

In addition to the introduction, this dissertation contains five more chapters. In Chapter 2 the basic version of the algorithms referenced in this dissertation are explained, and the state of the art is reviewed. In Chapter 3 a solution to this problem is proposed. In Chapter 4 the

experimental settings are described and some of the experiments that were performed to reach our final methodology are presented. In Chapter 5 we show the results and the performance of the proposed methodology. In Chapter 6 the final conclusions and future work are presented.

Chapter 2

Related Work

In this chapter, the fundamental concepts behind the algorithms used and referenced throughout this dissertation are introduced. We also review some of the methods used for image generation and for face matching that have been developed over the years using machine learning. With this survey of related work, we aim to discuss and understand the flaws of current algorithms. This helped shape the proposed algorithm to address problems and limitations with current methodologies.

2.1 Background Knowledge

2.1.1 Convolutional Neural Networks

A CNN is a type of Deep Neural Network that is commonly used for image analysis ([O'Shea and Nash, 2015](#); [Gu et al., 2015](#)). The main difference between a Convolutional Neural Network and a Fully Connected Neural Network is the use of convolutional layers.

Each convolutional layer is comprised of filters. Filters can have an arbitrary height and width, but will have a depth equal to the depth of the current input. These filters are adjusted during training using backpropagation, which allows a trained network to extract the relevant features in an image and pass them to the next layer.

Pooling layers are used to reduce the dimension of the data outputted by the previous layer of the network. Max pooling extracts the maximum value within the kernel window. Average pooling calculates the average of the values within the kernel window.

CNNs can use a Fully Connected Neural Network, with an activation function such as softmax at the end, if it is a classification problem. Figure [2.1](#) shows an example of a Convolutional Neural Network used to classify handwritten digits.

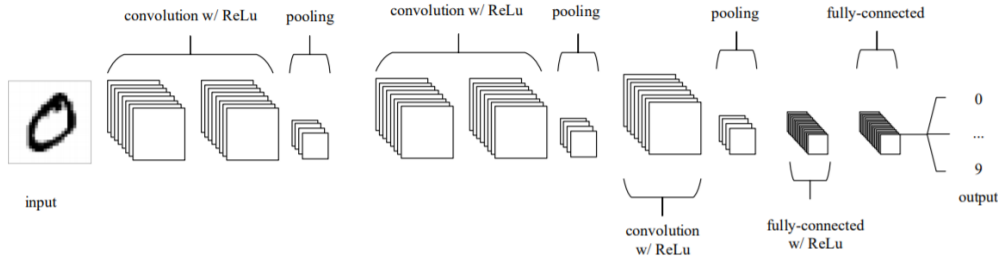


Figure 2.1: Convolutional Neural Network used to detect handwritten digits (Image extracted from the paper [O'Shea and Nash \(2015\)](#)).

2.1.2 Generative Adversarial Networks

GANs have been around since 2014, when [Goodfellow et al. \(2014\)](#) first proposed the idea. They have become very popular nowadays when it comes to image generation, and many variations to the original paper have been published. An example of a GAN can be seen in [Figure 2.2](#).

A Generative Adversarial Network consists of two networks: a generator and a discriminator. The objective of a GAN is to generate images that are indistinguishable from the training dataset. When training, the two networks are adversaries, in the sense that each network is constantly trying to become better than the other.

The generator network takes as input a random vector of numbers and outputs an image. The discriminator takes as input an image and outputs the probability of it being real. In other words, the generator creates images that look real and the discriminator tries to distinguish generated images from real images.

The loss function of a GAN can be written as:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_x[\log D(x)] + \mathbb{E}_y[\log(1 - D(G(y)))]. \quad (2.1)$$

where x is a "real" image and y is a random noise vector. The generator (G) tries to minimize this loss and the discriminator (D) competes to maximize it. With an adequately balanced training the generator will output increasingly realistic images, until they are practically indistinguishable from the "real" images.

After some experimentation researchers found that using GANs was very challenging, especially when using high-resolution pictures. They noticed that if the discriminator got much better than the generator, then the gradients that were calculated for the generator during training became less useful. There is also the problem of mode collapse, which is when the generator fails to produce a wide variety of outputs, and generates the same output every time ([Arjovsky and Bottou, 2017](#)).

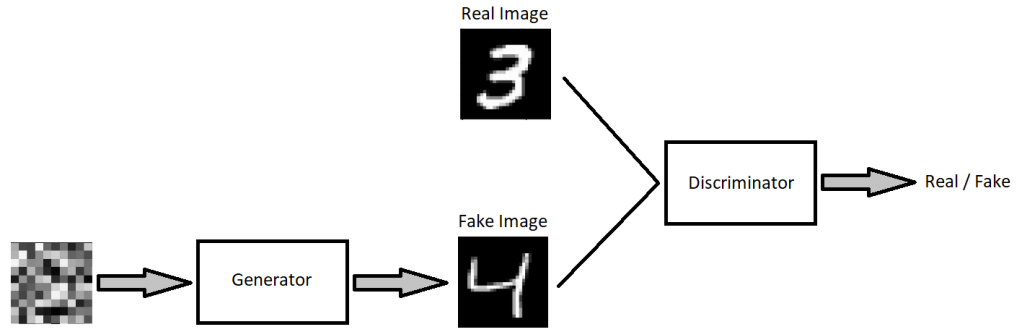


Figure 2.2: Generative Adversarial Network diagram.

2.1.3 Conditional Generative Adversarial Network

As seen in section 2.1.2 generative models are very useful for generating images from a random vector of numbers. Since a random vector of numbers is used as input we have no control over the output of the generator, which would be very useful for some problems. This is where cGANs come into play (Mirza and Osindero, 2014). They aim to condition the output of the generator. To do so, an additional input is added to the generator and the discriminator. An example of a CGAN can be seen in Figure 2.3. After the model is trained we can control the output of the generator by specifying the class/label of what we want the generator to produce. The random vector of numbers is still used, to allow some variation in the generated images belonging to the same class.

The loss function of a cGAN can be written as:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]. \quad (2.2)$$

where x is a class label, y is a "real" image from class x and z is a random noise vector. The generator (G) tries to minimize this loss and the discriminator (D) competes to maximize it. While training the generator will output increasingly realistic images, until they are practically indistinguishable from the "real" images.

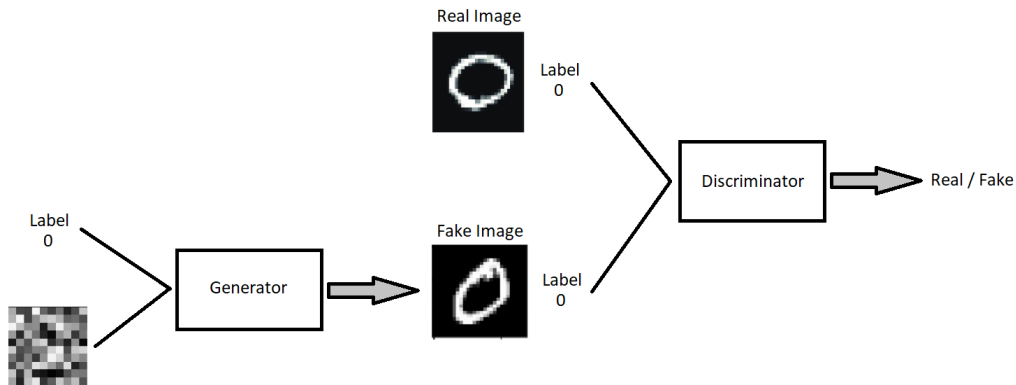


Figure 2.3: Conditional Generative Adversarial Network diagram.

2.2 State of the Art

2.2.1 Image Generation

Although there are various methods for image generation, such as GANs, cGANs and CycleGANs, those that specialize in face generation are not so common. In the next subsections the most recent or relevant algorithms for image generation and face generation are presented.

ProGAN

In order to create high-resolution pictures, to solve the issue of mode collapse and to improve training efficiency, the team at NVIDIA presented a solution based on Generative Adversarial Networks (Karras et al., 2017).

Their method still used the adversarial approach of using a generator and a discriminator, but instead of training the full networks at once, they took a more incremental approach. They trained the networks one layer at a time, starting with a very low resolution and incrementally increasing the size of the network and the resolution of the pictures generated. To do this they decreased the resolution of the training images to match the output of the generator and the input of the discriminator, then they trained the network to generate very low-resolution images (4x4 pixels), which was relatively fast.

After the training was completed another layer was added to the generator and to the discriminator (the generator and the discriminator have the same layer architecture, but mirrored) which doubled the resolution of the generated images to 8x8 pixels. The weights of the previously trained layers were kept, but were not final, meaning they were still trainable. The newly added layer was faded in with a parameter alpha, which was increased linearly during training, to allow for a more stable transition. This process was repeated until the desired resolution was reached.

The growing process of the network can be visualized in Figure 2.4 (extracted from the paper). The fact that the model was trained with an incrementally bigger resolution allowed the first layers

to learn a more high-level representation and the following layers to "fill in the details", which resulted in more realistic pictures. They found that it was also much more computationally efficient than training a model with all the layers at once. The results obtained by the authors are shown in Figure 2.5.

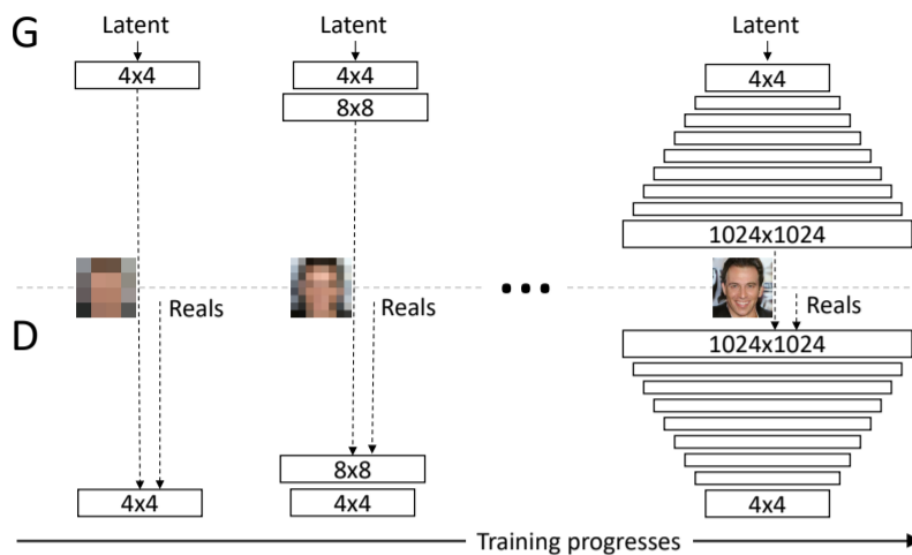


Figure 2.4: The process of adding layers to the network to increase the resolution of the generated images (Image extracted from the paper [Karras et al. \(2017\)](#)).



Figure 2.5: Results obtained by training the network using the CELEBA-HQ dataset (Images extracted from the paper [Karras et al. \(2017\)](#)).

Pix2pix

The Pix2pix algorithm consists of a cGAN (Figure 2.6), and it was designed as a general solution to image-to-image translation problems (Isola et al., 2016). Image-to-image translation refers to the process of applying transformations to an image to change its appearance. This is done to change certain features in the original image, such as the color and textures of certain elements. The network architecture of the generator consists of a U-Net, which is an encoder-decoder with skip connections. The network architecture of the discriminator is a PatchGAN, which means that instead of classifying an image as real or fake, as it is normally done, the discriminator classifies patches of the image as real or fake.

The loss function used for the discriminator is the same as the loss function of a cGAN. On the other hand, the loss function used for the generator is the loss function of a cGAN, with the addition of the L1 distance between the generated image and the real image. The authors chose L1 instead of L2 since it encourages less blurring. Some tests were performed to compare loss functions, such as using only L1 loss, using only CGAN loss, and combining both. Comparisons shown in Figure 2.9, extracted from the paper. They found that combining both was the best option. They trained the model for different image translation problems, as shown in Figures 2.7 and 2.8.

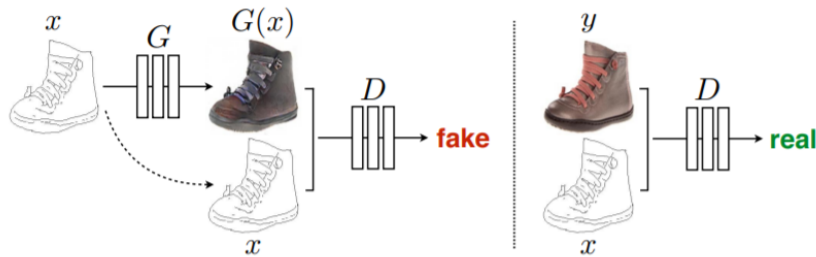


Figure 2.6: Pix2pix model. G is the generator and D is the discriminator (Images extracted from the paper Isola et al. (2016)).



Figure 2.7: Results obtained by training a model to transform edges to shoes (Images extracted from the paper Isola et al. (2016)).



Figure 2.8: Results obtained by training a model to transform Google Map images to satellite images and vice-versa (Images extracted from the paper [Isola et al. \(2016\)](#)).



Figure 2.9: Results obtained by using different loss functions (Images extracted from the paper [Isola et al. \(2016\)](#)).

2.2.2 Matching

Over the years many algorithms were implemented with the goal of face recognition and face matching. The best ones have an accuracy of more than 99% (see Table 2.1), which surpass algorithms that did not use machine learning, as can be seen in Figure 2.10 ([Wang and Deng, 2018](#)). State of the art methods use CNNs to transform an image into a feature vector. This feature vector can be compared to the feature vectors of other images to check if both images are a representation of the same person.

Network architectures consist of CNNs as they are capable of extracting relevant features in an image, allowing the network to more easily recognize a person (Wang and Deng, 2018). They produce better results because the networks learn the filters that need to be applied in each layer to extract the relevant features.

Method	Public. Date	Loss	Architecture	Accuracy (%)
FaceNet (Schroff et al., 2015)	2015	triplet loss	GoogleNet-24	99.63
VGGface (Parkhi et al., 2015a)	2015	triplet loss	VGGNet-16	98.95
Range Loss (Zhang et al., 2016)	2016	range loss	VGGNet-16	99.52
L2-softmax (Ranjan et al., 2017)	2017	L2-softmax	ResNet-101	99.78
CoCo loss (Liu et al., 2017)	2017	CoCo loss	-	99.86
Cosface (Wang et al., 2018)	2018	cosface	ResNet-64	99.33
Arcface (Deng et al., 2018)	2018	arcface	ResNet-100	99.83

Table 2.1: Different methods developed over the years for face recognition/matching (table from Wang and Deng (2018)).

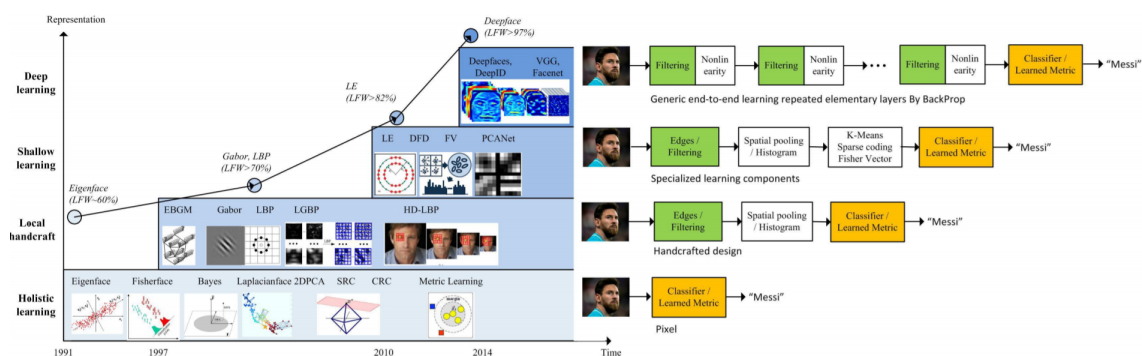


Figure 2.10: Different methods developed over the years to improve face recognition (Image extracted from the paper Wang and Deng (2018)).

Loss functions

Over the years many loss functions have been applied. These include variations of the following losses: Euclidean-distance-based loss, angular/cosine-margin-based loss, softmax loss. See Figure 2.11 for more details on the loss functions that have been used over the years.

With CosFace (Wang et al., 2018) and ArcFace (Deng et al., 2018) the authors proposed a modification to the softmax loss. The softmax loss gives separate feature embeddings but is ambiguous in the decision boundaries. With CosFace and ArcFace the authors solve this problem by increasing the margin between different classes. Figure 2.12 compares the decision boundaries in a binary classification problem between ArcFace and CosFace. ArcFace has a constant linear

angular margin, while CosFace does not. This difference increases the accuracy of the ArcFace method slightly, as can be seen in Table 2.1.

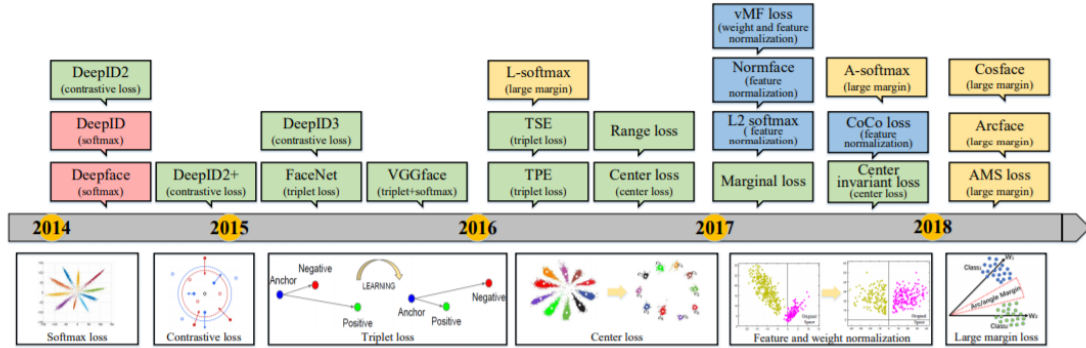


Figure 2.11: Different loss functions used over the years for face recognition (Image extracted from the paper [Wang and Deng \(2018\)](#)).

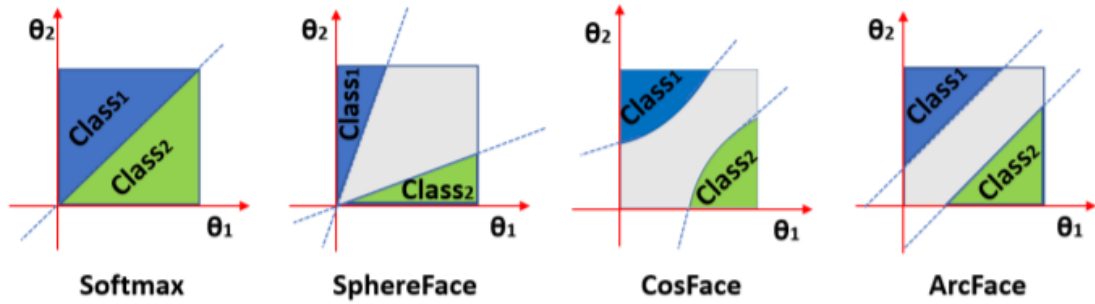


Figure 2.12: Decision margins of different loss functions (Image extracted from the paper [Deng et al. \(2018\)](#)).

2.2.3 Approaches to the sketch-to-face matching problem

Recent approaches to the sketch-to-face matching problem include methods based on CNNs ([Iranmanesh et al., 2018](#)), CycleGANs ([Kazemi et al., 2018](#)) and cGANs ([Osahor et al., 2020](#); [Chao et al., 2019](#)).

In [Iranmanesh et al. \(2018\)](#) the authors propose a method that uses two VGG-16 like networks ([Simonyan and Zisserman, 2014](#)) to match a sketch to a real picture of a person. One of the networks receives as input the sketch and a set of attributes, while the other receives a real picture. The architecture is shown in Figure 2.13. The loss function they use pulls the real pairs towards each other into a similar common latent feature subspace and pushes the false pairs apart from each other. When the networks are trained we can find the correct person by computing simple distance measures between the vector returned for a sketch and the vector returned for real pictures. One of the downsides of this method is that it only matches a sketch to a real picture. Other methods also

generate a realistic representation of the sketch, in case there is the need to use a manual method for the matching process.

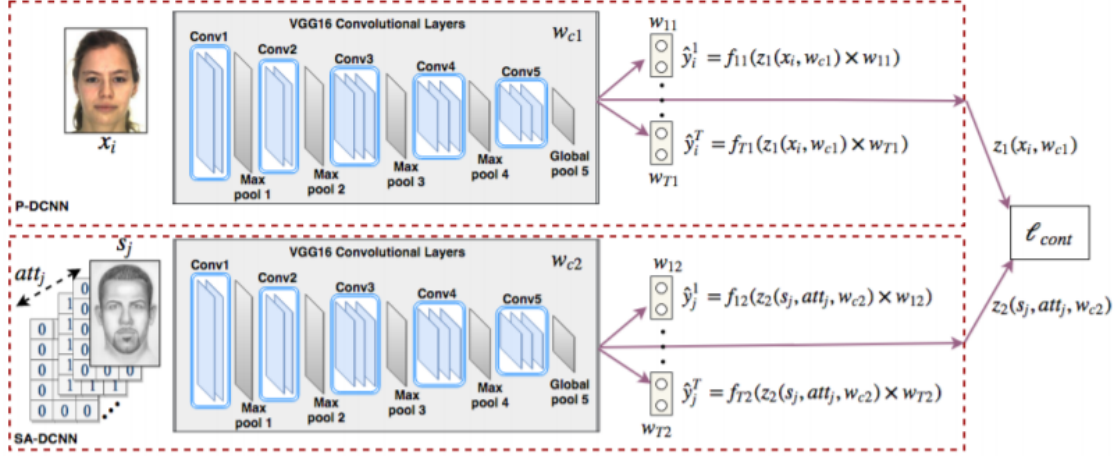


Figure 2.13: Architecture from Iranmanesh et al. (2018).

In Kazemi et al. (2018) the authors propose a method for generating sketches from real pictures and realistic pictures from sketches using a CycleGAN. One of the benefits of using a CycleGAN is that there is no need for paired data, a dataset of sketches and a dataset of real pictures could be used and the network would learn the transformations that needed to be applied to go from one domain to the other. They also conditioned the sketch-photo generator with facial attributes to force a more accurate representation in the generated photos, and to be able to change certain attributes in the generated photo. Figure 2.14 shows the architecture used. Figure 2.15 shows some examples of generated images with different attributes.

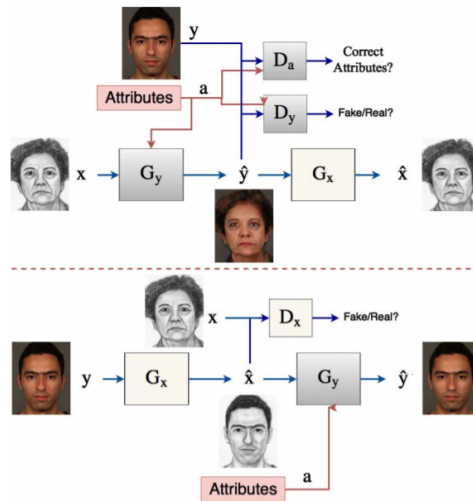


Figure 2.14: Architecture from Kazemi et al. (2018).

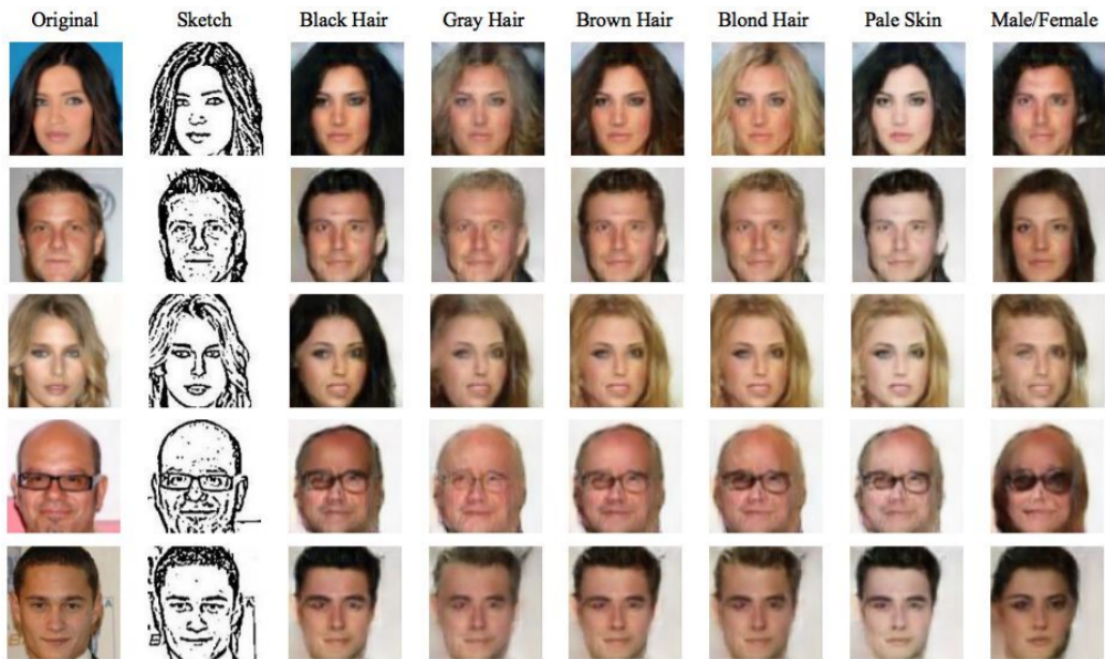


Figure 2.15: Generated images using the method in [Kazemi et al. \(2018\)](#).

In [Chao et al. \(2019\)](#) the authors use a modified cGAN to transform a sketch into a realistic picture. Their loss function included the loss of a cGAN, the difference between the real picture and the generated picture, the difference of the edges between the real picture and the generated picture, the difference of the face descriptors of the real picture and the face descriptors of the generated image. The face descriptors were obtained by training a VGG-19 model to extract high-level features. Figure 2.16 shows the model architecture. This method achieved very realistic results in the generated images as can be seen in Figure 2.17. They used the Dlib module for face recognition ([King, 2009](#)) to match the generated photo to a real photo.

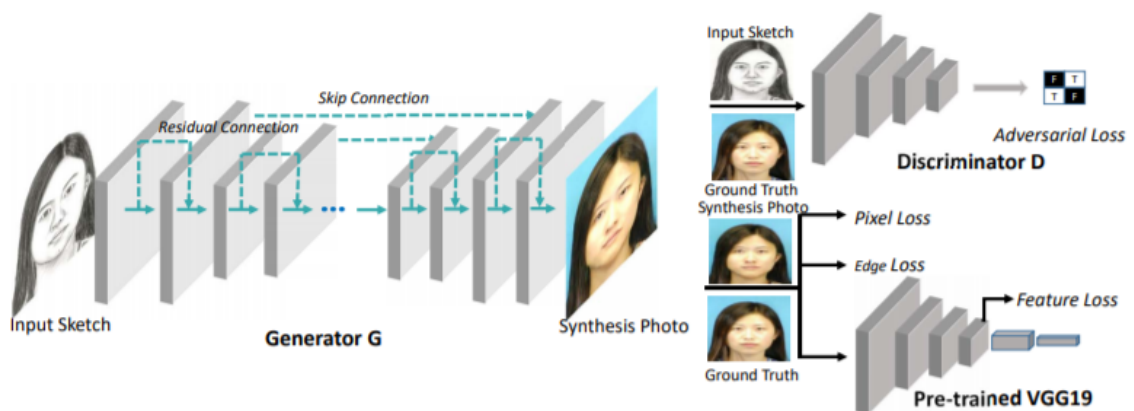


Figure 2.16: Architecture from [Chao et al. \(2019\)](#).



Figure 2.17: Generated images using the method in [Chao et al. \(2019\)](#) with different loss functions.

In [Osahor et al. \(2020\)](#) the authors used a modified cGAN to transform a sketch into a realistic picture. The authors used an XDOG filter ([Winnemoeller et al., 2012](#)), which detects edges in an image, to generate sketches from real pictures. They also trained a generator to transform real pictures into sketches. This is because the databases for sketch-photo pairs are not very large, and some do not have a very good quality. Therefore they took large face datasets, such as the CelebA dataset, which is a large dataset of celebrity face images, and generated their own sketches. Their loss function was designed to preserve the identity of the person in the generated image and to allow the user to choose certain facial attributes in the generated picture, such as hair color and sex. Figure 2.19 shows the model architecture. Examples of generated photos can be seen in Figure 2.18. To match the generated photo to a real photo they implemented a face verifier called DeepFace.

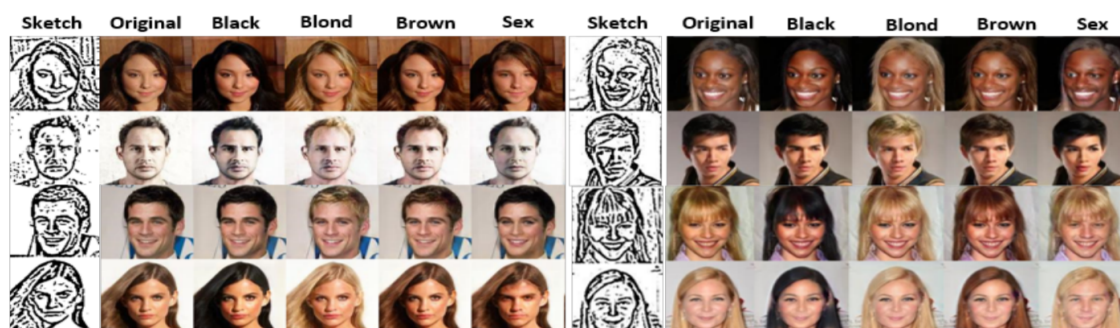


Figure 2.18: Generated images using the method in [Osahor et al. \(2020\)](#).

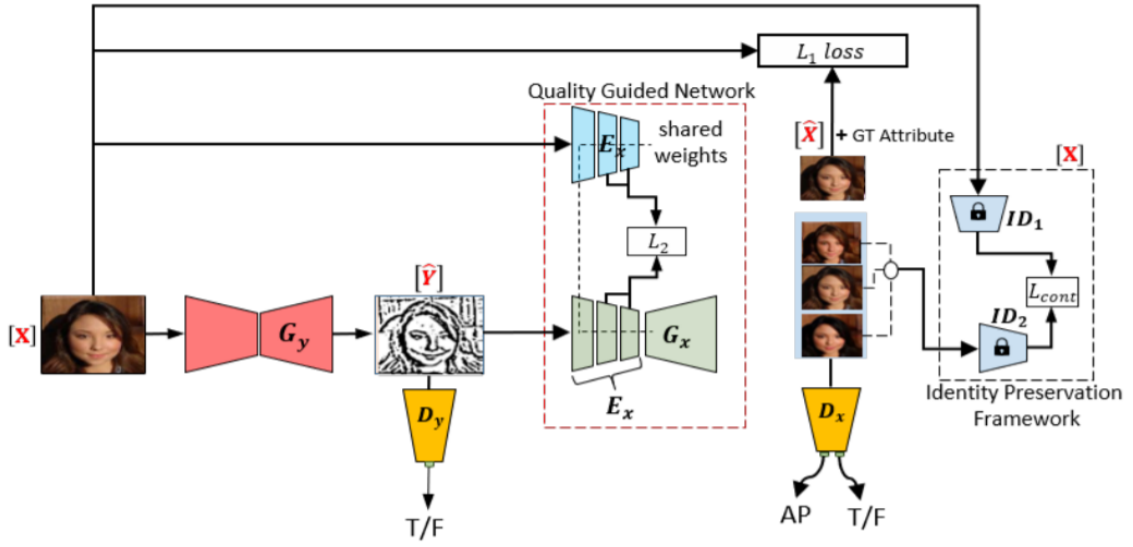


Figure 2.19: Architecture from [Osahor et al. \(2020\)](#).

2.3 Conclusions

Convolutional Neural Networks are mainly used for image analysis, for example, face recognition or object recognition tasks. Generative models such as Generative Adversarial Networks and Conditional Generative Adversarial Networks are used for image generation tasks. In this chapter we showed how important these algorithms are for the current state of the art methods, both in face generation and face matching/recognition.

Chapter 3

Final Methodology¹

The proposed method is an end-to-end model that, although integrated and optimizable as a whole, is composed of two main parts: a sketch-to-render generator and a matching network (see Figure 3.1). The sketch-to-render generator will transform the input sketch into a face rendering that is photo-realistic and similar to the real face that corresponds to the sketch. The matching network will receive the realistic rendering and output a template that can be used for matching through a simple distance measure. In the next sections, the architecture of both parts, the loss function, and the training process of the model are described in higher detail.

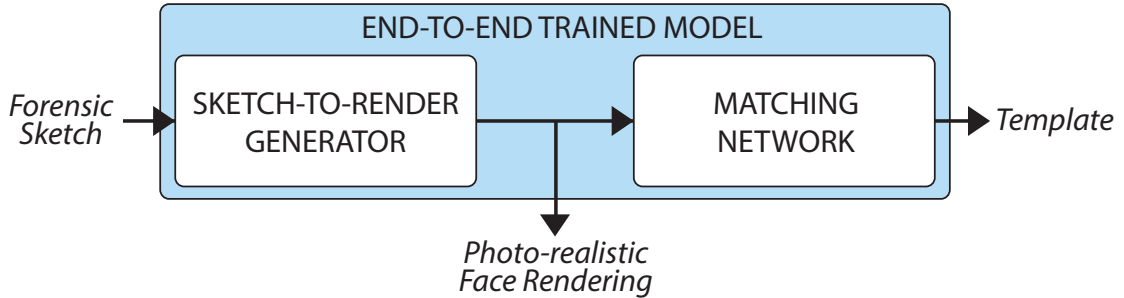


Figure 3.1: Overview schema of the proposed method.

3.1 Network Architecture

3.1.1 Sketch-to-render generator

The sketch-to-render process is performed by an image-to-image model that receives a sketch and transforms it into a realistic face rendering. For this model, we use the generator of a conditional generative adversarial network (cGAN) (Mirza and Osindero, 2014), that follows the typical structure of a U-Net (Ronneberger et al., 2015), with an encoder, that reduces data resolution at

¹The method described in this chapter was included in an article that was submitted for the 2020 International Joint Conference on Biometrics (IJCB 2020).

each level, followed by a decoder, that processes data up to the original resolution (see Figure 3.2). Skip connections enable the transmission of information between corresponding levels from the encoder to the decoder.

The U-Net encoder mimicked the architecture of VGG-16 (Simonyan and Zisserman, 2014) and it is composed of thirteen convolutional layers and four max-pooling layers distributed over five resolution levels. Convolutional layers have 64 – 512 filters, with size of 3×3 and stride of 1×1 , each followed by rectified linear unit (ReLU) activations. Max-pooling layers, which reduce data resolution for the next level, use window size and stride of 2×2 . The decoder mirrors the structure of the encoder with convolutional and deconvolution layers. Convolutional layers, which extract features from the data, have 64 – 512 filters with size and stride similar to their respective encoder counterparts. Deconvolution layers, that increase data resolution for the next level, have 64 – 512 filters with size and stride of 2×2 .

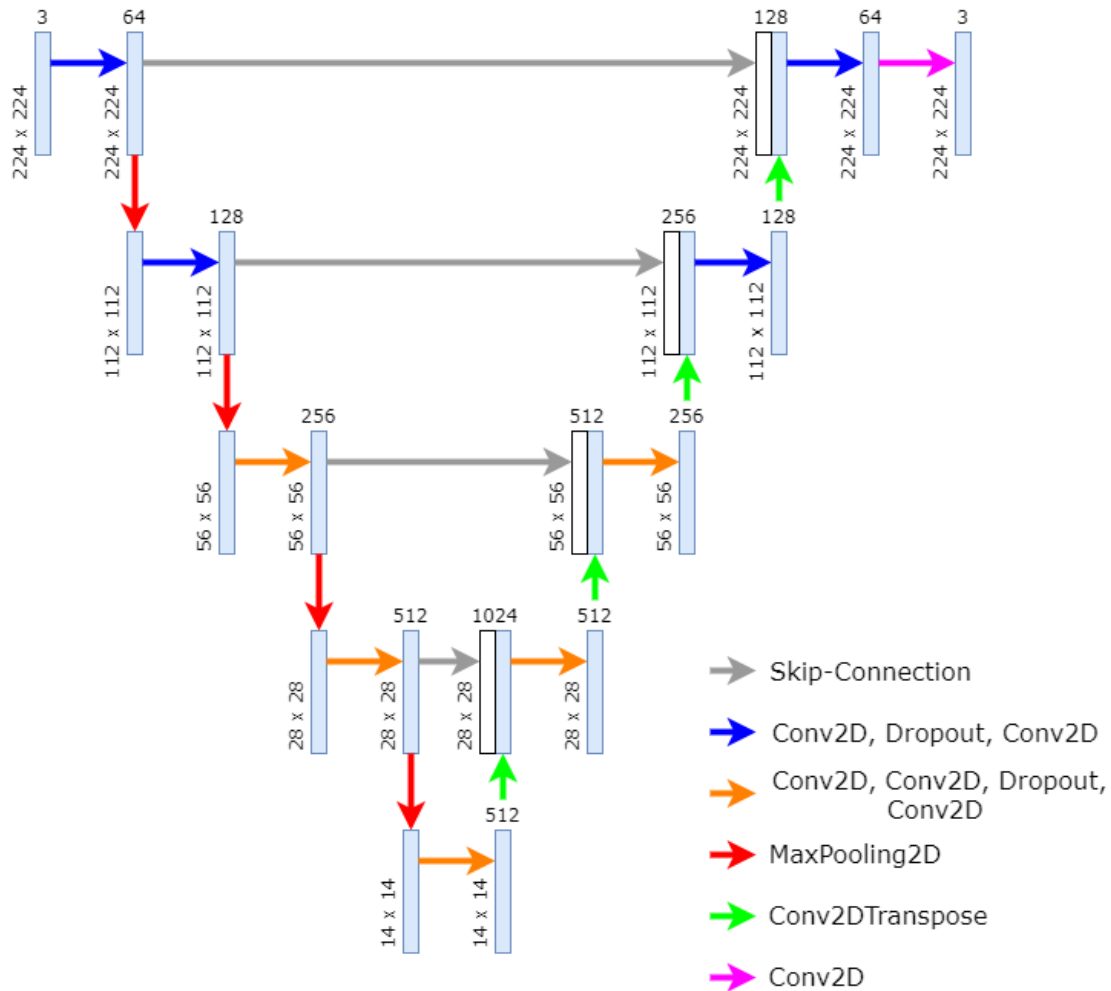


Figure 3.2: Network architecture of the generator.

The discriminator used during training is a CNN adapted from the cGAN of pix2pix (Isola et al., 2016) (see Figure 3.3), which receives as input the photo-realistic representation outputted by the generator and the corresponding sketch, and outputs a prediction on whether it is real or generated. This discriminator is composed of three convolutional layers, one fully-connected layer, and batch normalization. The convolutional layers have 64 – 256 filters with a size of 4×4 and stride of 2×2 , and are followed by Leaky ReLU activations.

The most important modification made to the discriminator was the output shape. The pix2pix discriminator returns a matrix indicating if patches of the input images are real or generated. We noticed that if instead of returning a matrix it returned a single number which classified the whole image as real or generated then it would reduce distortions in the generated image. An example can be seen in Figure 3.4. The generated images in this figure have a worse quality than our final method because this test was done in earlier stages where we had not cropped the dataset in the best way and the network architectures and hyperparameters were not fully optimized yet. Nevertheless, it still shows that the modified version of the pix2pix discriminator improves the quality of the generated images.

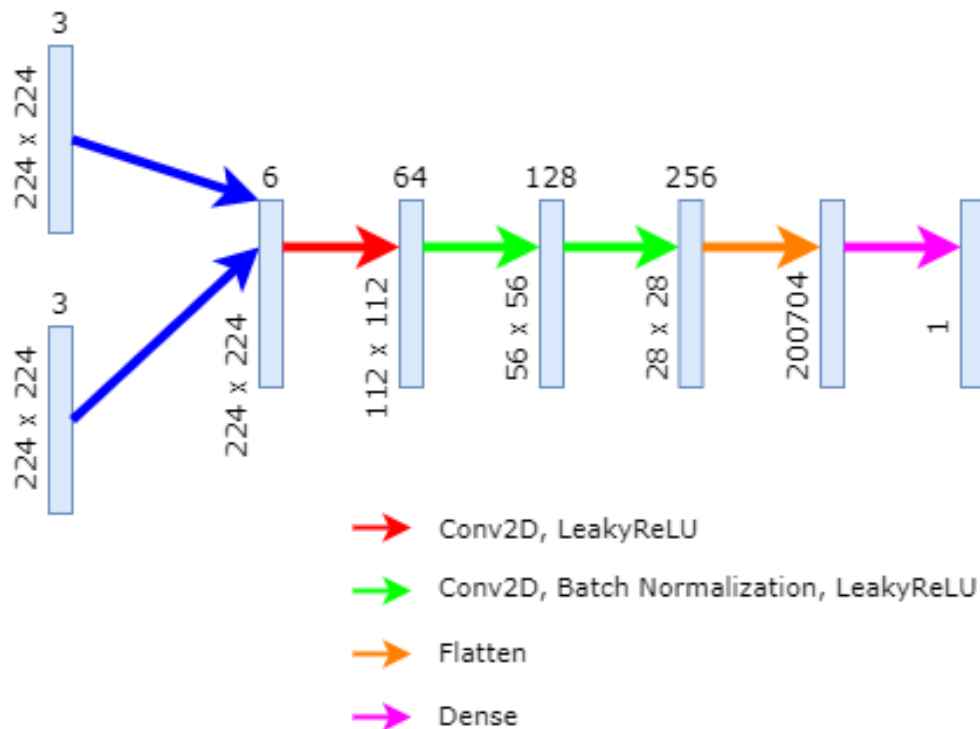


Figure 3.3: Network architecture of the discriminator.



Figure 3.4: Comparison of the pix2pix discriminator (old) and the modified pix2pix discriminator (new).

3.1.2 Matching network

After the sketch is transformed into a photo-realistic rendering of the face, the matching part of the model transforms this rendering into a template that can easily be used for matching. In this work, the matching network follows the structure of the VGG-16 ([Simonyan and Zisserman, 2014](#)) and uses pre-trained VGG-Face weights ([Parkhi et al., 2015b](#)).

Given a sketch, the output of the matching network (and of the method as a whole) is a template that can be used for matching. This template (or face descriptor) is a numerical uni-dimensional vector of 2622 features that describe the face represented on the input sketch. The sketch is thus matched with face photos from a database by computing the cosine distance between the respective templates.

3.2 Loss

The loss function used for training is a composition of several loss components relative to each part of the proposed model. The first component of the loss corresponds to the sketch-to-render, and comes from the typical training methodology of a cGAN. This part can be written as following:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))], \quad (3.1)$$

where x is a sketch and y is the corresponding ground-truth photo. The generator (G) tries to minimize this loss and the discriminator (D) competes to maximize it. This competition, if adequately balanced, will lead the generator to output increasingly realistic face images. Figures 3.6 and 3.7 show a graphical representation of the generator and discriminator loss functions, respectively.

Nevertheless, the generator should not only transform the sketch into a photo-realistic image, but also should closely mimic the respective ground-truth face photograph. In order to achieve this, a second loss term is defined as:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[||y - G(x)||_1], \quad (3.2)$$

which minimizes the $L1$ norm of the difference between the real image (y) and the generated image ($G(x)$). Moreover, we need to ensure the identity information in the sketch is preserved and refined throughout the network, and matches that of the respective ground-truth image.

Hence, a third component is added to the loss function (see Figure 3.5), corresponding to the matching part, such as:

$$\mathcal{L}_{match}(G) = \mathbb{E}_{x,y}[||V(y) - V(G(x))||_2]. \quad (3.3)$$

Combining the loss components mentioned above, the final loss function becomes:

$$\mathcal{L}(G, D) = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda_1 \mathcal{L}_{L1}(G) + \lambda_2 \mathcal{L}_{match}(G). \quad (3.4)$$

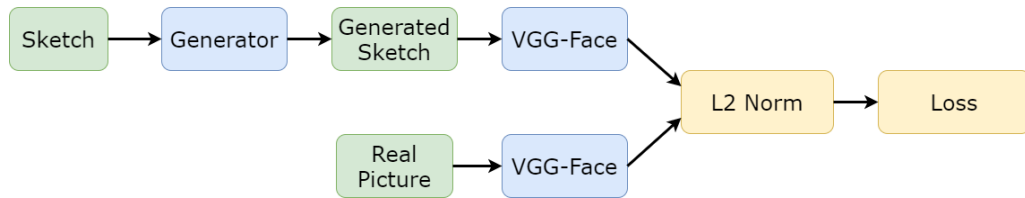


Figure 3.5: Loss component for the matching part of the model.

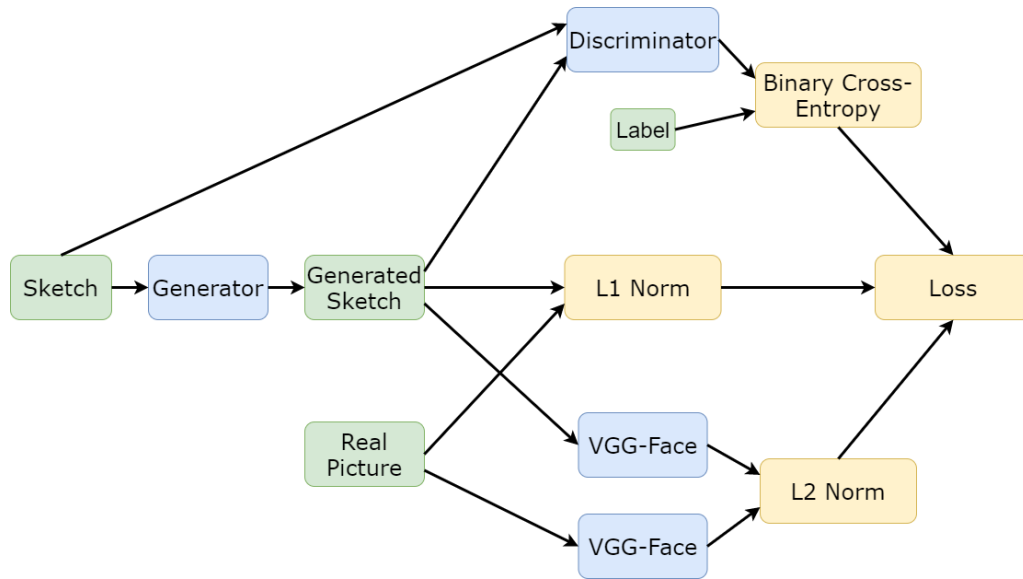


Figure 3.6: Loss component for the generator of the sketch-to-render part of the model.

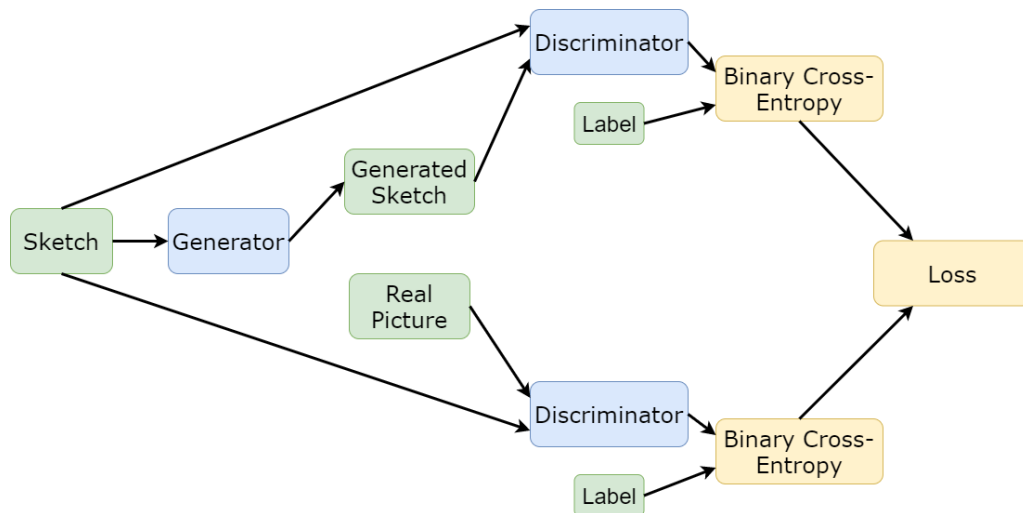


Figure 3.7: Loss component for the discriminator of the sketch-to-render part of the model.

Chapter 4

Experimental Settings

4.1 Data

The proposed model has been trained using pairs of sketches and corresponding face images from the CUHK Face Sketch database (CUFS) ([Wang and Tang, 2009](#)) and the CUHK Face Sketch FERET dataset (CUFSF) ([Wang and Tang, 2009](#); [Zhang et al., 2011](#)).

The CUFS dataset contains 606 sketch-photo pairs. The sketches correspond to face images acquired from students of the Chinese University of Hong Kong (CUHK) and from the AR and XM2GTS databases. Due to the current unavailability of the latter two databases, this work only used the 188 sketch-photo pairs relative to CUHK students. Of those, 88 pairs were used for model training and 100 pairs were reserved for testing.



Figure 4.1: Example images from CUHK Face Sketch database ([Wang and Tang, 2009](#)).

The CUFSF contains 1189 sketches that correspond to photos in the FERET database ([Phillips et al., 1998, 2000](#)). Of those, 1089 pairs were used for training and 100 pairs were used for testing.



Figure 4.2: Example images from CUHK Face Sketch FERET Database (Wang and Tang, 2009; Zhang et al., 2011).

4.2 Pre-processing

To uniformize the CUFS and CUFSF databases, the DeOldify API¹ was used to colorize the photos of CUFSF from the FERET database (see Figure 4.3) that, unlike CUFS images, were originally in grayscale. Additionally, this allows the proposed model to learn to generate color images.

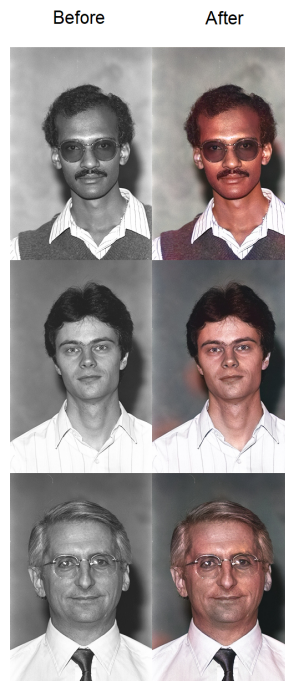


Figure 4.3: CUFSF dataset images before and after colorization using the DeOldify API.

¹DeOldify API. Available on: <https://github.com/jantic/DeOldify>.

In style transfer problems using conditional GANs, image spatial consistency is paramount. When the locations of image landmarks are consistent between the input and the expected output, the network is able to offer realistic results. This is related to the loss component that compares the output of the generator and the corresponding ground-truth. However, in the case of sketch-to-photo transformation, the shape, size, and location of facial landmarks in the sketch and the ground truth can vary considerably. For instance, the eyes in a sketch can be drawn farther apart than they are in the ground-truth image. These situations reflect on the generator loss values and generally cause distortions that damage the realism of the rendering. To solve this problem, the sketches and photos are transformed so that the faces are aligned and the position of the eyes are consistent in each sketch-photo pair (see Figure 4.4 for an example of images before and after they were aligned and cropped). This makes the data more consistent and enables higher photo-realism in the generated renderings. Additionally, the accuracy of the matching process is also improved (Schroff et al., 2015).

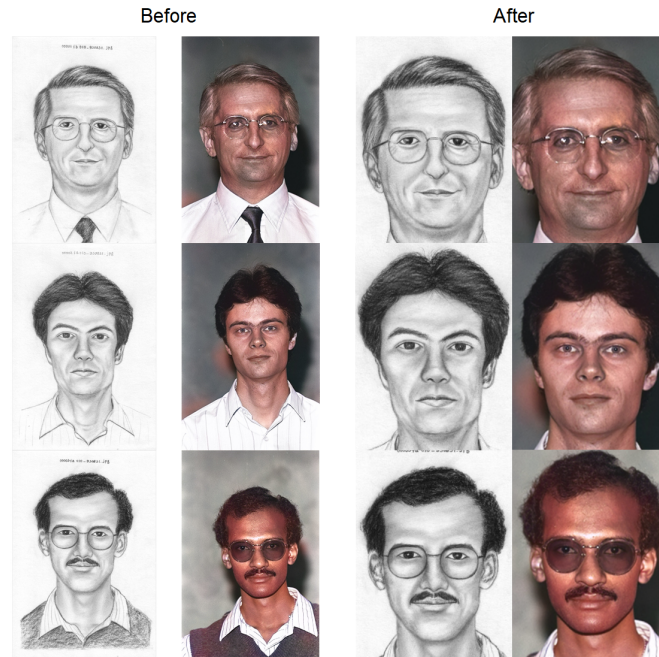


Figure 4.4: CUFSF dataset images before and after cropping.

4.3 Experiments

In this section, we present some of the experiments that took place in the development process of the methodology explained in chapter 3.

The generator and discriminator used in the following experiments have the same architecture as the ones shown in chapter 3.

4.3.1 Experiment 1

For this experiment we used the CUFSF dataset, after applying pre-processing.

The used loss function is the following:

$$\mathcal{L}(G) = \mathcal{L}_{L1}(G) + \mathcal{L}_{match}(G). \quad (4.1)$$

The $\mathcal{L}_{L1}$ loss makes the generated image similar to the real image and the $\mathcal{L}_{match}$ loss makes the identity of the person in the generated image similar to the identity of the corresponding real image. The objective was to have a good matching accuracy while having a realistic intermediate representation of the sketch.

The resulting images looked blurry because the $\mathcal{L}_{cGAN}$ loss was not included. Nevertheless, the model was somewhat able to transform the sketches into a realistic representation, at least when it comes to color. Generated images can be seen in Figure 4.5.



Figure 4.5: Images generated using the method described in experiment 1.

4.3.2 Experiment 2

For this experiment we used the CUFSF dataset, after applying pre-processing.

The loss function used is the following:

$$\mathcal{L}(G) = \mathcal{L}_{match}(G). \quad (4.2)$$

The $\mathcal{L}_{match}$ loss makes the identity of the person in the generated image similar to the identity of the corresponding real image, which improves the matching accuracy since the matching network recognizes the transformed sketch and the real image as the same person.

The resulting images were an intermediate representation that increased the matching accuracy without concern for realism. In Chapter 5 we discuss the increase in matching accuracy using this method when compared to the method that enforced a realistic representation. Generated intermediate images can be seen in Figure 4.6.

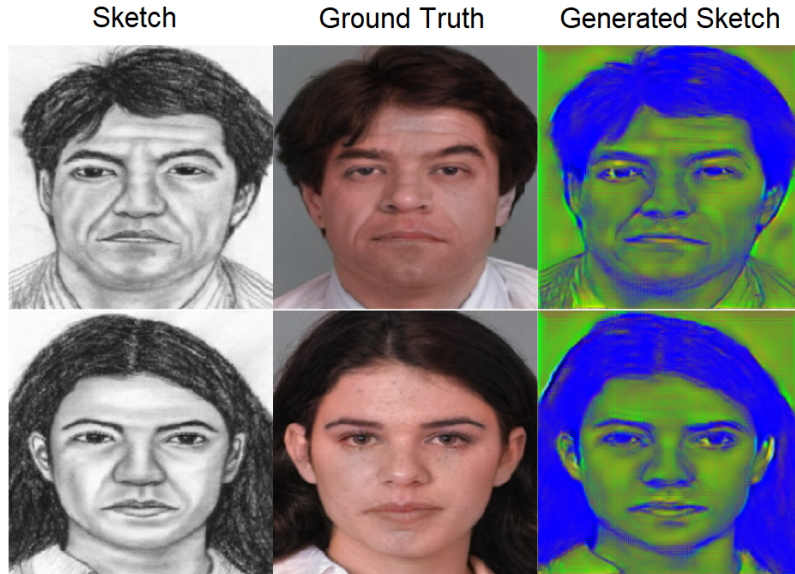


Figure 4.6: Images generated using the method described in experiment 2.

4.3.3 Experiment 3

For this experiment we used the CUFS dataset, after applying the processing.

The loss function used is the following:

$$\mathcal{L}(G) = \mathcal{L}_{match}(G). \quad (4.3)$$

The resulting images were an intermediate representation that increased the matching accuracy without concern for realism. In Chapter 5 we discuss the increase in matching accuracy using

this method when compared to the method that enforced a realistic representation. Generated intermediate images can be seen in Figure 4.7.

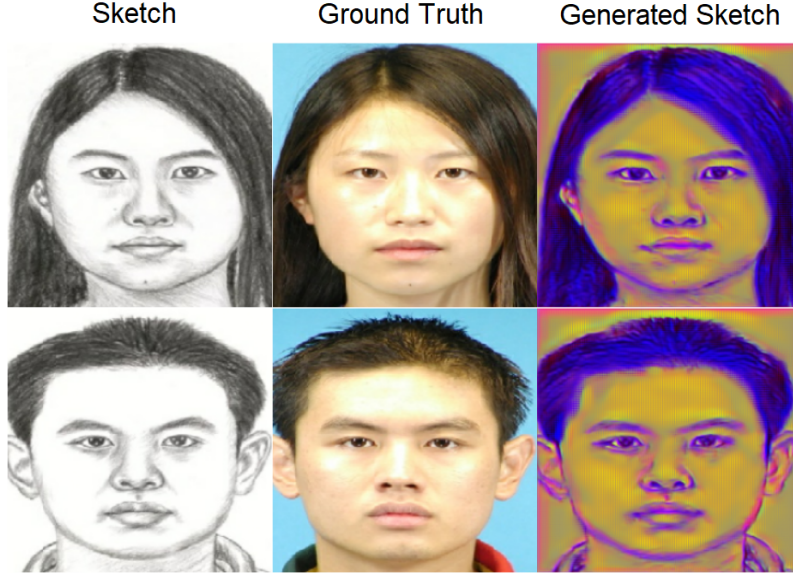


Figure 4.7: Images generated using the method described in experiment 3.

4.3.4 Experiment 4

For this experiment we used the CUFSF dataset, after applying pre-processing.

The loss function used is the following:

$$\mathcal{L}(G, D) = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda_1 \mathcal{L}_{L1}(G). \quad (4.4)$$

with $\lambda_1 = 100$ as recommended in the pix2pix paper (Isola et al., 2016).

The $\mathcal{L}_{cGAN}$ loss gives the resulting images a sharper look and the $\mathcal{L}_{L1}$ loss makes the generated image similar to the real image. The $\mathcal{L}_{L1}$ loss also helps avoid mode collapse, which is when the generator fails to produce a wide variety of outputs, and generates the same (or very similar) outputs every time.

This generator architecture gave better results than the generator of pix2pix. The generated images had fewer distortions and were smoother. Figure 4.8 shows the generated images using this model.



Figure 4.8: Images generated using the method described in experiment 4.

4.3.5 Experiment 5

For this experiment we used the CUFS dataset, after applying pre-processing.

The loss function used is the following:

$$\mathcal{L}(G, D) = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda_1 \mathcal{L}_{L1}(G). \quad (4.5)$$

with $\lambda_1 = 100$ as recommended in the pix2pix paper (Isola et al., 2016).

Figure 4.9 shows the generated images using this model.



Figure 4.9: Images generated using the method described in experiment 5.

4.3.6 Conclusions

The experiments allowed us to reach the final model shown in chapter 3. Experiments 2 and 3 showed that in case there is no need for a realistic representation of the sketch this method is a good alternative, as it increases the matching accuracy significantly (as will be explained in further detail in chapter 5). Experiments 4 and 5 showed acceptable image quality, but it could be improved by adding the $\mathcal{L}_{match}$ loss, as it would prevent the model from changing the identity of the person in the generated image when compared to the sketch, and it would also improve the matching accuracy. Therefore the final methodology is a combination of experiments 2 and 4 since it allowed for a good matching accuracy and the best looking images without changing the identity of the person in the original sketch.

4.4 Training

The model presented in chapter 3 was trained on two experimental settings. The first one followed the afore-described loss function to promote both an intermediate realistic face rendering and high matching accuracy. Here, the model was trained during 780 and 1500 epochs, for CUFSF and CUFS respectively, with batch size 8, using the Adam optimizer with a learning rate of 2×10^{-4} and parameter $\beta_1 = 0.5$. The loss parameters λ_1 and λ_2 were experimentally set to 100 and 1, respectively.

The second setting studied the effect on performance if the realistic face rendering generation was disposable. To achieve this, the loss terms $\mathcal{L}_{cGAN}$ and $\mathcal{L}_{L1}$, which promotes the realistic intermediate representation, were removed. In this setting, the model was trained during 1320 and 2800 epochs, for CUFSF and CUFS respectively, with batch size 12, using the Adam optimizer with a learning rate of 1×10^{-3} and parameter $\beta_1 = 0.9$.

To improve the robustness of the model and to avoid overfitting, data augmentation was applied to each pair of images. These were randomly cropped and horizontally mirrored before each epoch.

Chapter 5

Results and Discussion

To evaluate the matching performance of the proposed model, its rank- N accuracy was computed on the test sets of CUFS and CUFSF. Rank- N accuracy measures the fraction of test instances where the true correspondence is successfully found among the N strongest predictions offered by the model. In this case, we consider $N \in \{1, 5, 10\}$.

After training the model with the CUFSF dataset (1089 train pairs), 54% rank-1 accuracy (100 test pairs) was attained. Over all considered ranks, the matching accuracy of the proposed method is superior to the alternative methods, as presented in Table 5.1. However, when dismissing photo-realistic rendering generation, the matching accuracy increases significantly, showing a trade-off between intermediate representation realism and matching performance that could be tuned to fit the objectives of specific application scenarios.

Method	Accuracy (%)		
	Rank-1	Rank-5	Rank-10
Sketch	49	77	88
pix2pix	53	83	92
Proposed (with r.r.g.)	54	87	96
Proposed (without r.r.g.)	73	98	99

Table 5.1: Matching accuracy on the CUFSF dataset, using different methods to enhance the sketch (r.r.g.: realistic rendering generation).

On the CUFS dataset, using 88 sketch-photo pairs for training and 100 pairs for testing, the proposed method attained 44% rank-1 matching accuracy (see Table 5.2). These performance results are slightly below, but nevertheless aligned with the alternative methods evaluated on the same settings. When removing the rendering realism constraints in the loss, the accuracy at all ranks increases considerably, achieving much better performance than the alternative methods.

Considering the significantly smaller size of the CUFS training set vs. the size of the training set of the CUFSF dataset, these results may denote that the proposed method is more sensitive to scarce data than the alternatives, failing to offer more general solutions. Furthermore, knowing the images in the CUFS dataset are much less diverse than those of CUFSF (regarding subject

Method	Accuracy (%)		
	Rank-1	Rank-5	Rank-10
Sketch	47	80	90
pix2pix	52	79	86
Proposed (with r.r.g.)	44	77	86
Proposed (without r.r.g)	59	85	91

Table 5.2: Matching accuracy on the CUFS dataset, using different methods to enhance the sketch (r.r.g.: realistic rendering generation).

ethnicity, skin color, hair color, background), one can argue that the proposed method offers the greatest advantages over the alternatives in more challenging scenarios.

On recent literature, Table 5.3 shows the rank-1 accuracy of different methods for matching a sketch to a real picture.

Paper	Data	Accuracy (%)
Chao et al. (2019)	CUFS	36.00
Pramanik and Bhattacharjee (2012)	CUFS	80.00
Bhatt et al. (2010)	CUFS	87.89
Wang and Tang (2009)	CUFS	96.30
Qingshan Liu et al. (2005)	CUFS	87.67
Xiaoou Tang and Xiaogang Wang (2004)	CUFS	71.00
Xiaoou Tang and Xiaogang Wang (2003)	CUFS	81.30
Bhatt et al. (2012)	CUFS + IIIT-Delhi	93.16
Kazemi et al. (2018)	CelebA	65.53

Table 5.3: Rank-1 accuracy of different methods for sketch-face matching.

These values are not directly comparable to the results reported in this paper, as the data used is not the same. Even on the CUFS dataset, some prior works used 606 pairs of images, while currently only 188 are available, and this puts the proposed method at a disadvantage. Nevertheless, the results of the proposed method are aligned and sometimes superior to these prior art results.

5.1 Realistic Generation Performance

After the evaluation of matching performance, the realism of the intermediate representations generated by the proposed method were evaluated. Examples of these photo-realistic images and the corresponding sketches and ground-truth photos from the CUFSF and CUFS datasets can be seen in Figures 5.1 and 5.3, respectively.

Measuring realism is a difficult task, since the concept is highly subjective. However, we can assume that, if a generated image is sufficiently similar to the corresponding ground-truth, it is realistic. Hence, we use similarity/dissimilarity metrics that are common in related literature

works for an objective evaluation, along with a visual subjective inspection of the test results. These metrics were the Fréchet Inception Distance (FID) (Heusel et al., 2017), the Inception Score (IS) (Salimans et al., 2016), and the Structural Similarity Index (SSIM) (Wang et al., 2004).

As expected, the results through the similarity/dissimilarity metrics are better when the proposed method includes realistic rendering generation (see Tables 5.4 and 5.5), as the method has learned to generate images that are similar to the respective ground-truth photos. With CUFSF data, the results with the proposed method are slightly inferior but comparable to the alternatives, once again showing that the proposed method finds advantages in more challenging data.

Method	FID ↓	IS ↑	SSIM ↑
Ground Truth	0.0	1.71	1.0
pix2pix	70.54	1.69	0.59
Proposed (with r.r.g.)	83.51	1.47	0.60
Proposed (without r.r.g.)	330.41	1.48	0.32

Table 5.4: Comparison between different methods to enhance the sketch, using the CUFSF dataset (r.r.g.: realistic rendering generation).

Method	FID ↓	IS ↑	SSIM ↑
Ground Truth	0.0	1.39	1.0
pix2pix	41.46	1.26	0.62
HFFS2PS Chao et al. (2019)	58.50	1.43	0.70
Proposed (with r.r.g.)	74.74	1.54	0.61
Proposed (without r.r.g.)	321.96	1.35	0.45

Table 5.5: Comparison between different methods to enhance the sketch, using the CUFS dataset (r.r.g.: realistic rendering generation).

Visually, we can confirm that the results of the proposed method are, indeed, very similar to the alternative method (see Figures 5.1 and 5.3). Considering the improvement in performance, maintaining the degree of photo-realism is a positive aspect of the proposed method.

Furthermore, the proposed method delivers, in most cases, images that retain most information needed to identify the person represented in the sketch. Nevertheless, the method presents a worrying lack of diversity in specific facial features (such as hair, skin, or eye color) that should be addressed. In fact, when trained without photo-realistic generation, the method avoids these details by using unrealistic color schemes (see Figures 5.2 and 5.4). To improve this shortcoming, an approach similar to the one proposed in Iranmanesh et al. (2018), which consists in giving facial attribute information along with the sketch, could be an important addition to the proposed model.



Figure 5.1: Images generated by our method using the CUFSF dataset.

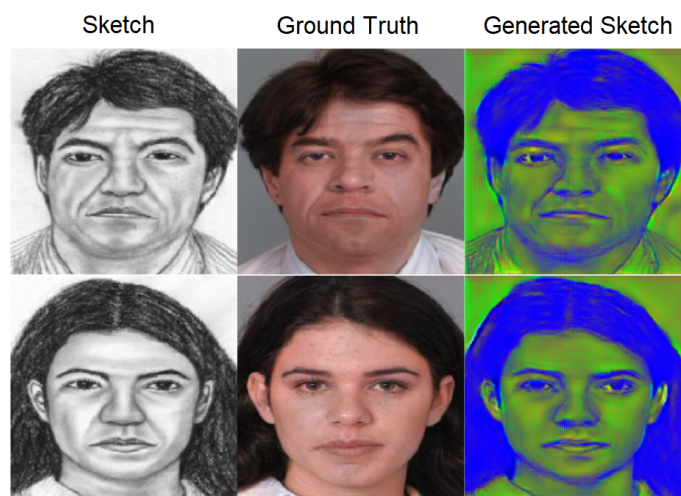


Figure 5.2: Intermediate representation of the sketches using the CUFSF dataset.

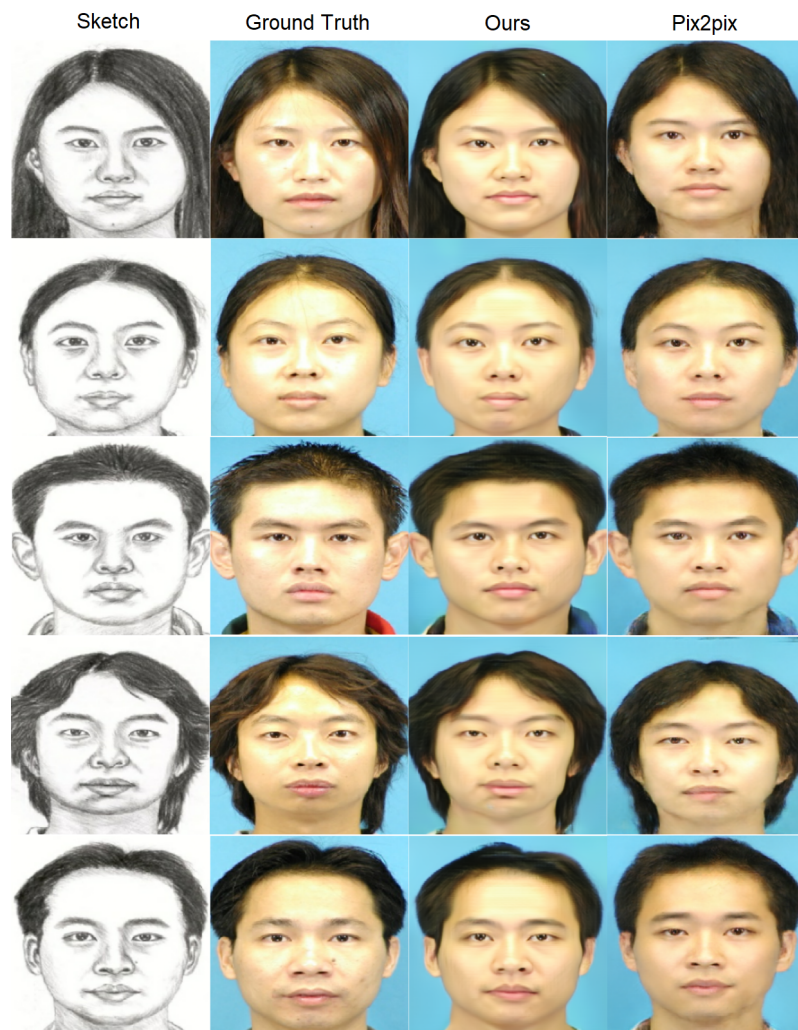


Figure 5.3: Images generated by our method using the CUFS dataset.

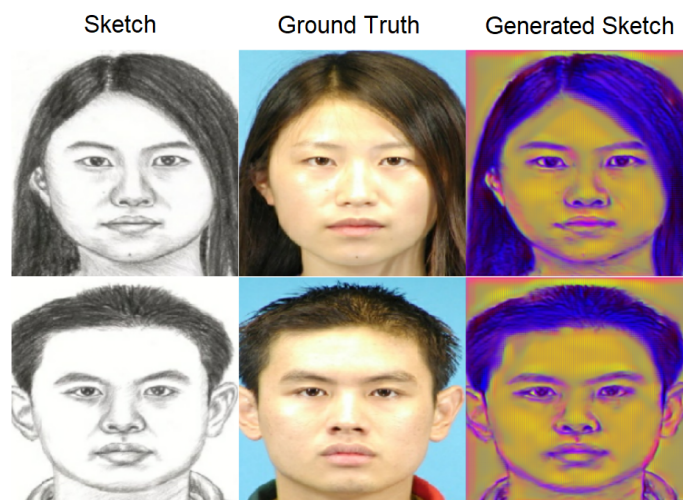


Figure 5.4: Intermediate representation of the sketches using the CUFS dataset.

Chapter 6

Conclusions and future work

This dissertation proposes an end-to-end deep method for sketch-to-photo matching that promotes the generation of a photo-realistic intermediate representation of the face depicted on the input sketch. Upon receiving a forensic sketch as input, the model will first convert the sketch into a photo-realistic rendering and then process it to return a face descriptor template. This template can be compared to templates obtained from photos in a database, using simple distance measures, to find a matching identity.

As an end-to-end model, jointly trainable, it aims to eliminate performance limitations associated with separately optimized processes. Nevertheless, by including a sketch-to-render component based on a conditional GAN, the method also learns to offer an intermediate realistic face representation that can be used for alternative manual matching, in case the automatic matching fails to offer useful results.

Upon evaluation on the CUFS and CUFSF sketch-photo databases, the matching process showed performance improvements over the state-of-the-art, especially in the more diverse database (CUFSF). The generation of face renderings offered realistic results, comparable to the state-of-the-art, especially on the CUFS dataset. When disregarding realistic rendering generation, the performance results improved.

Despite the promising results, further efforts should be devoted to improve the face rendering generation component. Namely, the limited diversity of face characteristics like hair, eyes, and skin color on generated images should be addressed, in order to improve both the realism of the results and the matching performance. Another interesting research topic would be learning from unpaired sketches and photos, enabling the easy construction of larger databases for more robust training.

References

- Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017.
- H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa. On matching sketches with digital face images. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7, 2010.
- H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa. Memetically optimized mcwld for matching sketches with digital face images. *IEEE Transactions on Information Forensics and Security*, 7(5):1522–1535, 2012.
- W. Chao, L. Chang, X. Wang, J. Cheng, X. Deng, and F. Duan. High-fidelity face sketch-to-photo synthesis using generative adversarial network. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4699–4703, 2019.
- Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2018. URL <http://arxiv.org/abs/1801.07698>.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *CoRR*, abs/1512.07108, 2015. URL <http://arxiv.org/abs/1512.07108>.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. 2017.
- Seyed Mehdi Iranmanesh, Hadi Kazemi, Sobhan Soleymani, Ali Dabouei, and Nasser M. Nasrabadi. Deep sketch-photo face recognition assisted by facial attributes. *CoRR*, abs/1808.00059, 2018. URL <http://arxiv.org/abs/1808.00059>.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. URL <http://arxiv.org/abs/1611.07004>.
- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL <http://arxiv.org/abs/1710.10196>.

- H. Kazemi, M. Iranmanesh, A. Dabouei, S. Soleymani, and N. M. Nasrabadi. Facial attributes guided deep sketch-to-photo synthesis. In *2018 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 1–8, 2018.
- Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(60):1755–1758, 2009. URL <http://jmlr.org/papers/v10/king09a.html>.
- Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *CoRR*, abs/1710.00870, 2017. URL <http://arxiv.org/abs/1710.00870>.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL <http://arxiv.org/abs/1411.1784>.
- Uche Osahor, Hadi Kazemi, Ali Dabouei, and Nasser Nasrabadi. Quality guided sketch-to-photo image synthesis. *arXiv*, 2020. 2005.02133.
- Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015. URL <http://arxiv.org/abs/1511.08458>.
- O.M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, pages 6., 2015a.
- Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015b.
- P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing Journal*, 16(5):295–306, 1998.
- P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET Evaluation Methodology for Face Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.
- S. Pramanik and D. Bhattacharjee. Geometric feature based face-sketch recognition. In *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, pages 409–415, 2012.
- Sourav Pramanik and Debotosh Bhattacharjee. An approach: Modality reduction and face-sketch recognition. *CoRR*, abs/1312.1681, 2013. URL <http://arxiv.org/abs/1312.1681>.
- Qingshan Liu, Xiaoou Tang, Hongliang Jin, Hanqing Lu, and Songde Ma. A nonlinear approach for face sketch synthesis and recognition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 1005–1010 vol. 1, 2005.
- Rajeev Ranjan, Carlos Domingo Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *CoRR*, abs/1703.09507, 2017. URL <http://arxiv.org/abs/1703.09507>.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015. ISBN 978-3-319-24574-4.

- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. 2016.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. URL <http://arxiv.org/abs/1503.03832>.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
- Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1891–1898, Washington, DC, USA, 2014. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.244.
- Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014. doi: 10.1109/CVPR.2014.220.
- Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *CoRR*, abs/1801.09414, 2018. URL <http://arxiv.org/abs/1801.09414>.
- Mei Wang and Weihong Deng. Deep face recognition: A survey. *CoRR*, abs/1804.06655, 2018. URL <http://arxiv.org/abs/1804.06655>.
- X. Wang and X. Tang. Face photo-sketch synthesis and recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 31. 2009.
- Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- Holger Winnemoeller, Jan Kyprianidis, and Sven Olsen. Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36: 740–753, 10 2012. doi: 10.1016/j.cag.2012.03.004.
- Xiaoou Tang and Xiaogang Wang. Face sketch synthesis and recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 687–694 vol.1, 2003.
- Xiaoou Tang and Xiaogang Wang. Face sketch recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):50–57, 2004.
- W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tail. *CoRR*, abs/1611.08976, 2016. URL <http://arxiv.org/abs/1611.08976>.