University of West Bohemia
Faculty of Applied Sciences

Department of Cybernetics

Heterogeneous Face Recognition from Facial Sketches

Ing. Ivan Gruber

Advisor: Doc. Ing. Miloš Železný, Ph.D.

Pilsen, 2018

Západočeská univerzita
Fakulta aplikovaných věd

Katedra Kybernetiky

Heterogenní Rozpoznávání Lidské Tváře ze Skic Obličeje

Ing. Ivan Gruber

Školitel: Doc. Ing. Miloš Železný, Ph.D.

Plzeň, 2019

Западночешский Университет

Пльзень, Чехия

Гетерогенное распознавание личности по эскизам лица

Ing. Ivan Gruber

Научные руководители: Doc. Ing. Miloš Železný, Ph.D.

Пльзень, 2019

Acknowledgments

I would like to thank everyone who supported me in my studies and also outside of it. Mainly, I would like to thank my advisor Doc. Ing. Miloš Železný Ph.D. and all my Czech and Russian colleagues for their advises and encouragement. Moreover, I would never write this thesis without my great family, my friends, and my beloved girlfriend, thank you all.

Abstract

This dissertation thesis presents a novel system for automatic heterogeneous face recognition from facial sketches based on a novel method named X-Bridge. Such a task is primarily relevant in the security and surveillance domains. In this work are made following contributions: (1) Analysis of the available neural network architectures used for image classification and their testing for face recognition task; (2) Analysis of the state-of-the-art loss functions used in face recognition task and their testing in combination with different neural network architectures; (3) Analysis of methods potentially usable as a cross-modal bridge; (4) Proposing a novel GAN based method named X-Bridge used as a cross-modal bridge; (5) Introducing a novel metric for measuring the performance of cross-modal bridges in the heterogeneous face recognition task; (6) Proposing a complex automatic heterogeneous face recognition system. The system improves state-of-the-art results on an appropriate benchmark face recognition dataset.

Keywords

Face Recognition, Machine Learning, Neural Network, Classification, Verification, Identification, Heterogeneous Face Recognition, Cross-modal bridge, Image-to-sketch translation.

Abstrakt

Tato dizertační práce představuje nový systém automatického heterogenního rozpoznávání lidské tváře ze skic. Systém je založený na nové metodě pojmenované X-Bridge. Heterogenní rozpoznávaní lidské tváře je primárně relevantní pro úlohy bezpečnosti a sledování. Tato práce má následující přínos: (1) Analýzu dostupných architektur neuronových sítí používaných pro úlohu klasifikace obrázků a jejich testování v rámci úlohy rozpoznávání lidské tváře; (2) Analýzu state-of-the-art ztrátových funkcí užívaných v úloze rozpoznávání lidské tváře a jejich testování v kombinaci s různými neuronovými sítěmi; (3) Analýzu metod potenciálně použitelných jako intermodální most; (4) Představení nové metody intermodálního mostu pojmenované X-Bridge založené na generativní adversiální síti; (5) Představení nové metriky určené k měření výkonu intermodálních mostů v úloze heterogenního rozpoznávání lidské tváře; (6) Představení komplexního systému automatického heterogenního rozpoznávání lidské tváře. Představený systém zlepšuje state-of-the-art výsledky na testovaném benchmarkovém datasetu.

Klíčová slova

Rozpoznávání lidské tváře, Strojové učení, Neuronová síť, Klasifikace, Verifikace, Identifikace, Heterogenní rozpoznávání lidské tváře, Intermodální most, Translace obrázek-skica.

Абстракт

В диссертации представлена новая система автоматического гетерогенного распознавания личности человека по эскизам лица, основанная на новом методе распознавания X-Bridge. Такая задача в первую очередь актуальна в сферах безопасности и наблюдения. В этой работе получены следующие основные результаты: (1) Анализ доступных архитектур нейронных сетей, используемых для классификации изображений, и их исследование для задачи распознавания лиц; (2) Анализ современных функций потерь, используемых в задаче распознавания лиц, и их тестирование в сочетании с различными архитектурами нейронных сетей; (3) Анализ методов, потенциально используемых в качестве кросс-модального моста; (4) Предложен новый метод под названием X-Bridge, основанный на GAN моделях (Generative Adversarial Nets - Генеративные состязательные нейросети) и используемый в качестве кросс-модального моста; (5) Предложена новая метрика для измерения производительности кросс-модальных мостов в задаче гетерогенного распознавания лиц; (6) Разработана комплексная автоматическая гетерогенная система распознавания лиц. Система превосходит современные результаты на общепринятом наборе данных, предназначенном для распознавания лиц.

Ключевые слова

Распознавание лиц, машинное обучение, нейронная сеть, классификация, верификация, идентификация, гетерогенное распознавание лиц, кросс-модальный мост, преобразование изображений в эскизы.

Declaration

Hereby I declare that I compiled this rigorous thesis independently, using only the listed literature and resources.

In Pilsen, 31.5.2019

Handwritten signature

Contents

List of Tables

List of Figures

List of Abbreviations

3DMM	3D Morphable Model
CNN	Convolutional Neural Network
CPU	Central processing unit
FR	Face Recognition
GPU	Graphics processing unit
kNN	k-Nearest Neighbors
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
NN	Neural Network
PCA	Principal Component Analysis
ROI	Region of Interest
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine

Chapter 1

Introduction

Face recognition (FR) is defined as a person verification or identification according to a person's face from an image or a video source. FR has been one of the most intensively studied topics in computer vision for the last few decades and received significant attention because of its applications in various tasks. The most notable usage of face recognition is in biometrics. Compared with other biometrics techniques (for example, fingerprints or iris), FR has the potential to recognize the subject without any further cooperation of the subject non-intrusively. Therefore, it can be used for security systems, forensic, or searching for wanted persons in crowds. Moreover, it can be used as another layer of security in login systems. From other domains, we can mention, for example, gender classification, emotion recognition, person database searching, witness face reconstruction, etc.

Despite such great attention, FR is still a very complex and challenging task due to various external conditions, for example, illumination, pose or occlusion, and internal conditions, for example, face expression or aging.

FR tasks can be divided into two main categories: face verification and face identification. More information about this division is in Chapter 2.

1.1 Problem definition

Face Recognition is a process of verification or identification of a person from a digital image or a video source. Prior to the FR, it is usually necessary to perform some preliminary processes. These processes can be generally divided into four following parts (see Figure 1.1):

1. Image preprocessing - Process of suppressing noise (unwilling distortions) in an image while simultaneously maintaining important information in this image. Image preprocessing methods can be divided according to the size around the preprocessed pixel into four following categories: brightness and color corrections and transformations, geometric transformations, local operations of preprocessing (filtration, gradient operators, morphology), and frequency analysis. The most information is always in the original image, and every preprocessing decreases this information. Prior knowledge

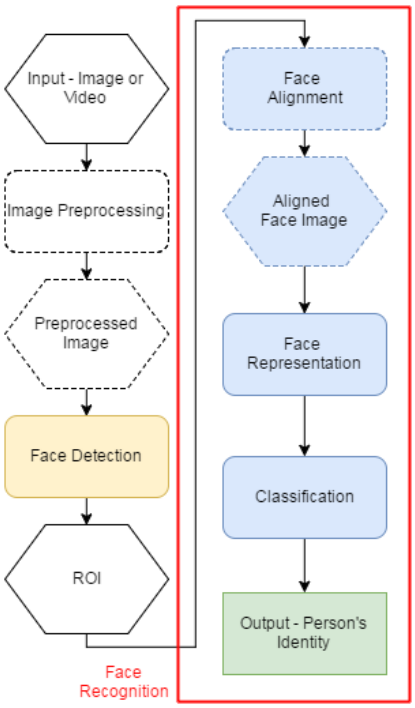


Figure 1.1: Whole Face Recognition process with all prerequisites.

about the image can make preprocessing much easier. Image preprocessing step is not essential, however, the majority of FR systems include this step.

2. Face Detection - Process of a system to detect a human face in an image. There are a lot of different algorithms and approaches, however, the most famous one is probably Haar cascade detection [1].

3. Face Alignment - Process of further processing of the ROI. In this step, there can be employed some image preprocessing techniques from the first step, but more advanced ones can be performed too thanks to priory information about the ROI - it is known there is (or at least should be, if the face detector did not fail) human face. These advanced techniques try to overcome some FR challenges such as pose or non-rigid expression. This step is not obligatory.

4. Face Representation - Process of obtaining a representation of the face image (feature extraction). Among popular description methods can be included, for example, EBGM, PCA, FLDA, Neural Networks, 2D Face Synthesis, and 3D Face Synthesis. More information about this step can be found in Chapters 7 and 8.

5. Classification - A general process of determining to which of a set of categories a new observation belongs. There are many different algorithms, which are used in FR, few examples: Bayesian classifiers, Support Vector Machines (SVM), and Neural Networks (NEUs). More information about classifiers can be found in Chapter 2. The output of the classification step is, according to the type of task (identification vs. verification), a person's identity or answer, if the person truly is, who it claims it is.

1.2 Brief history of Face Recognition

The FR task is as old as computer vision, both because of practical importance and high attractiveness. Why is this task so attractive and demanded? It's not only because of its non-intrusive and uncooperative manner, but it is also because we usually recognize other people according to their face, so it is the most natural way of people recognition for human beings.

The first experiments with semi-automated computer-based facial recognition were done during the sixties by Woodrow Wilson Bledsoe, who used his system for facial feature point detection. A most famous early example of a face recognition system is from 1989 from Kohonen [2], who used a simple neural network to face recognition of aligned and normalized face images. This neural network computes a face description, also known as eigenfaces.

Kirby and Sirovich [3] introduced an algebraic manipulation in 1990, thanks to which could be eigenfaces directly calculated. They also showed that fewer than 100 eigenfaces were required to describe aligned and normalized face images accurately. Turk and Pentland [4] then demonstrated that eigenfaces, coupled with their method for detection and localization of faces in various external conditions, could achieve solid real-time face recognition. This demonstration sparked an explosion of interest in the topic of face recognition.

1.3 Motivation and Application

Why use Face Recognition? There is a growing need for more sophisticated security systems around the world in recent years. A very popular option for these systems is, rather then check, what a person has, whom a person really is. Systems based on body or behavior characteristics are often called biometric systems. More traditional methods rely on possession of some plastic cards, tokens, keys, chips, etc., or knowledge of a password or a PIN code, and are relatively easy to overcome because cards and PINs can be stolen, passwords can be guessed or forgotten. This is the main advantage of biometrics, it cannot be stolen, forgotten, or misplaced.

Probably the most popular biometrics are fingerprints and iris, but there are many others, for example, voice and signature. Some of these techniques are intrusive, some not, however all of these techniques have one crucial drawback - all, in contrast to FR, require the cooperation of the recognized subject. The fact that FR can be done passively is essential for surveillance purposes too. The price of the needed equipment is another advantage of FR in comparison to other biometric techniques - facial images can be easily obtained with a couple of cameras. Perhaps the most crucial thing about FR is that humans identify other people according to their face too, therefore they are likely to be comfortable with systems that use this approach.

There are numerous real-world applications of FR. The most important ones are mentioned in the following list:

- Security - access to buildings, ATM machines, bank account logins, etc.
- Surveillance - searching for wanted or missing persons, airports or other public places security.

- General identity identification - national IDs, passports, driving license.
- Person database investigations - searching for suspected persons in police databases according to witness description, etc.
- Face reconstruction
- Monitoring at childcare or old people's centers
- Labeling faces in video
- Emotion recognition - Customer's reactions observation.

A special case of the FR is heterogeneous FR, which is FR across different visual domains. Such approaches also have many critical real-world applications, especially in the security and surveillance domains. For example, heterogeneous FR is very relevant in assisting law enforcement in identifying subjects, when only a sketch based on eyewitnesses description is available. Another interesting example can be FR from infrared light, whose main advantage is its ability to "see" in the dark. This can be utilized in buildings or places security systems when the lighting of the surrounding is inappropriate for the usage of standard RGB cameras.

It can also be said, FR is a specific case of object recognition, which is very hard due to its nonlinearity. The main problem stems from the fact that different human faces are, in general, still very similar. Moreover, the human face is not a rigid object. The sources of variation of the facial appearance can be divided into two following groups: internal sources and external conditions.

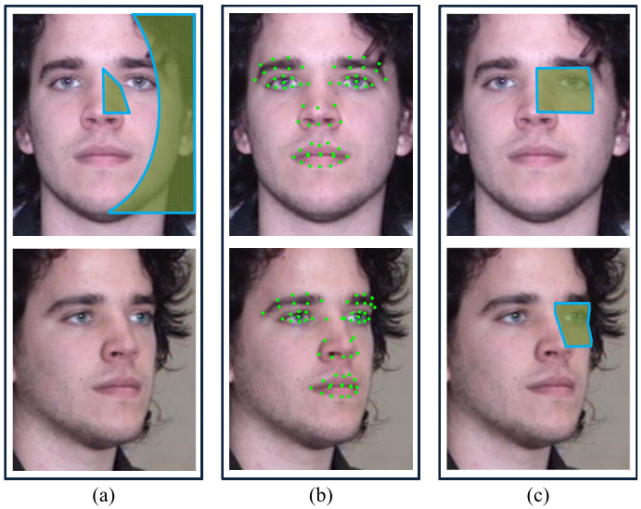


Figure 1.2: Challenges for FR caused by pose variation: (a) self-occlusion; (b) loss of semantic correspondence; (c) nonlinear warping of facial textures; Image taken from [5].

Internal sources result from physical attributes of the face and can't be affected by an observer. Further, internal sources can be categorized into two classes: intrapersonal and interpersonal attributes. Intrapersonal attributes are attributes responsible for differences in the appearance of one person, for example, face expression, aging, different haircuts, glasses, etc. Interpersonal attributes are responsible for variations in appearance between two persons, for example, gender, ethnicity, age, etc.

External conditions cause changes in facial appearance because of the interaction of light with the face or because of the relative position of the face and the observer. Among external conditions, it can be incorporated lighting conditions (illumination), pose, scale, occlusion, and imaging parameters (e.g., resolution, imaging noise, focus, image domain, etc.). Moreover, challenges for FR caused by pose variations can be divided into the following three groups: self-occlusion - loss of information, loss of semantic correspondence - position of facial texture varies nonlinearly following the pose change and nonlinear warping of facial textures (Figure 1.2).

Whereas the interpersonal attributes are desirable, intrapersonal attributes and external conditions cause problems during the FR task. Since differences created by intrapersonal differences and external conditions can be in standard subspaces more significant than interpersonal differences, it makes FR so hard and complex.

At the end of this section, it should be mentioned that there is some controversy about using surveillance systems due to the privacy of citizens. Utilization of systems with face detection and recognition can be abused for the monitoring of citizens' movements and actions.

1.4 Goals of Dissertation

This dissertation thesis's primary goal is to develop a system for automatic heterogeneous FR from facial sketches. Such a task can be divided into three sub-goals. Nowadays, there exist plenty of different classification methods, most of them based on neural networks. Therefore, the first step is to analyze available state-of-the-art FR methods. Furthermore, each heterogeneous FR algorithm needs a cross-modal bridge module to overcome differences in two different modalities. Consequently, the second step is to analyze existing methods potentially usable as the cross-modal bridge in heterogeneous FR tasks. Third, apply these methods while addressing some of their flaws in a novel heterogeneous FR system.

1.5 Outline

The outline of this work is as follows. The description of verification and identification problem can be found in Chapter 2. Modern FR approaches have three primary attributes: (1) Training data; (2) Network architecture; (3) Design of loss function. A quick review of the popular and benchmark datasets used in FR can be found in Chapter 3. In Chapter 4 are described the most important neural network architectures used in recent years. In Chapter 5 is survey of the popular designs of loss functions. Moreover, Chapter 6 describes very important generative models. There is a comprehensive survey of specific FR approaches in Chapters 7 and 8. In Chapter 9, a heterogeneous face recognition algorithm is presented as the main output of this work, whereas in the next chapter experiment and results are presented. Finally, in Chapter 11 conclusions are made, and plans for future work are outlined.

Chapter 2

Classification

Face Recognition task can be considered as a classification task - you have test sample (image or video with someone's face), and you classify it into a specific class (classes are usually identities of the persons, but it is possible to classify people according to their sex, age, ethnicity, etc.). The purpose of a classification algorithm is, therefore, to assign the testing sample to the correct class. Generally speaking, classification can be divided according to an absence or a presence of training data into two main categories: Unsupervised learning (Clustering), and Supervised learning.

Furthermore, FR problem can be categorized into two following problems: Verification, and Identification (Figure 2.1). For more information see Sections 2.1 and 2.2.

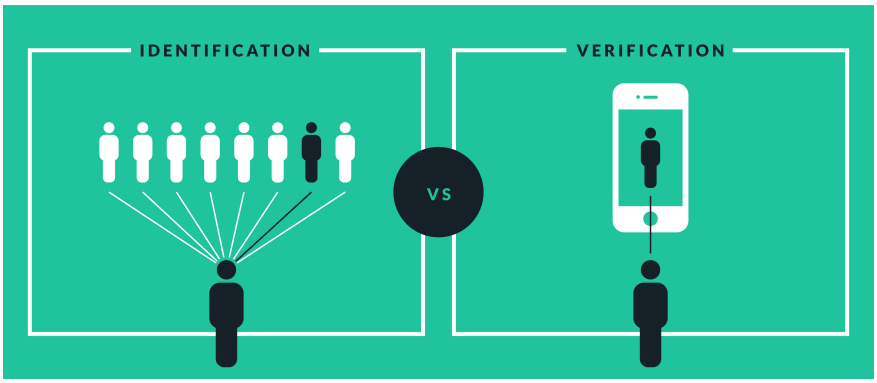


Figure 2.1: Comparison of identification and verification [6].

Moreover, in terms of testing protocol, FR can be evaluated under closed-set or open-set settings, as illustrated in Figure 2.2. You can find more information in Section 2.3.

2.1 Verification

Verification systems are trying to answer the question: "Are you who you claim you are?". In the verification task, an individual presents himself or herself as a specific person. The

system checks his/her biometrics and compares it with biometrics of claimed person (this biometrics has to be already saved in the system's database). Then the system decides if the individual and claimed person are the same person.

In other words, we can say that the verification task is a 1-to-1 matching task. Verification is generally faster than identification because the system compares only two biometrics - the one presented by the individual and the specific one, which is already stored in the system's database.

2.2 Identification

Identification systems are trying to answer the question: "Who are you?". These systems are trying to identify an unknown person's biometrics. It has to compare these biometrics with all other biometrics that are already saved in the system's database.

We can say that identification is 1-to-n matching, where n is the total number of biometrics stored in the system's database. The problem arises when the unknown person is not in the database at all. These cases are usually solved by the implementation of a thrash class (a class with persons outside the database) or by checking some threshold (after crossing this threshold is a person claimed as someone unknown).

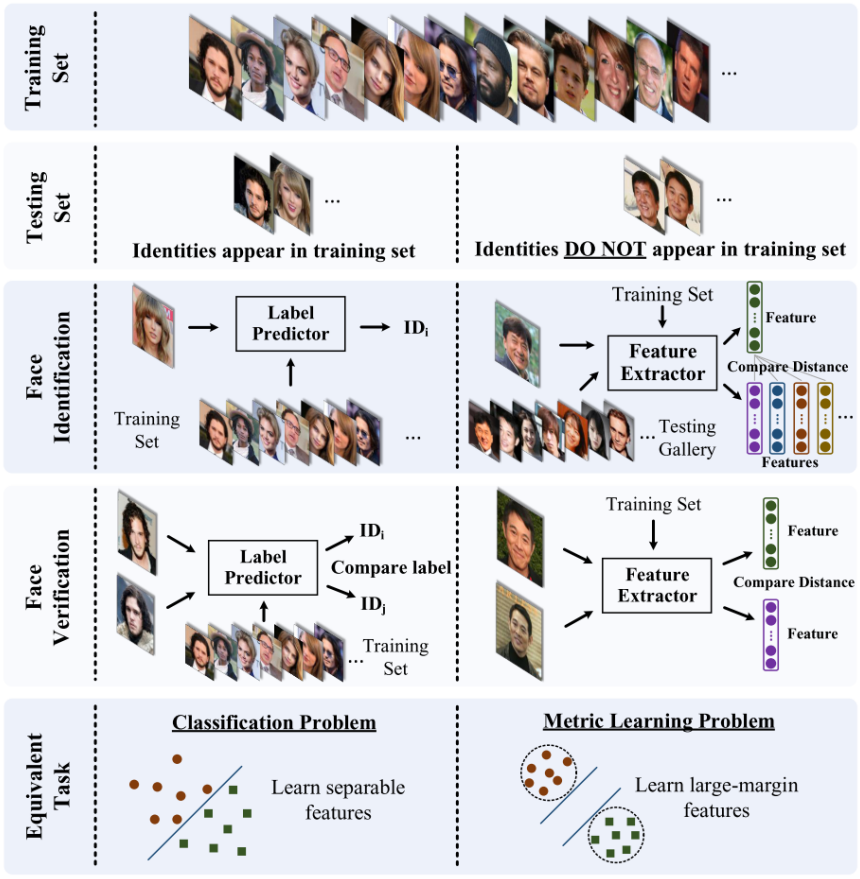


Figure 2.2: Comparison of closed-set and open-set face recognition [7].

2.3 Testing protocols

There are two basic testing protocols in FR: closed-set, and open-set settings [7], see Figure 2.2. In closed-set settings, all testing identities are also presented in the training set. In this scenario, the algorithm performs the standard classification task of each testing image to one of the given identities. Also, in this scenario is a verification task equivalent to performing identification for a pair of faces and comparing their labels. Therefore, closed-set FR can be addressed as a classification problem, where features are expected to be separable.

In the open-set protocol, not all the testing identities have to be presented in the training set, which makes it a much more challenging task. Because it is impossible to classify these cases to known identities in the training set, it is necessary to map the faces to a discriminative feature space. In this scenario, the face identification task can be viewed as performing face verification between the probe image and every identity in the gallery (training set). Open-set FR is, therefore, a metric learning problem, where the key is to learn discriminative large-margin features. In the ideal case, in the certain metric space of the desired features, the maximal intra-class distance is smaller than the minimal inter-class distance. This criterion is necessary to achieve perfect accuracy using the nearest neighbor classifier.

Chapter 3

Face Recognition Datasets

In the world of machine learning, training data are an essential part of modern classification approaches. There is also an utter need for benchmark data on which classification methods can be fairly compared. In this chapter is presented a quick analysis of the most important datasets used for face recognition and also datasets for sketch-based face recognition. At the end of the first part is provided a comparison between these datasets in Table 3.1. It is necessary to notice that in recent years, there was a big gap between the performance of methods thanks to the private (Google, Facebook, Microsoft) datasets. Also, the usage of private datasets causes a problem with the reproducibility of research and, therefore, with an objective evaluation of results. However, this gap is diminished by the newly available datasets containing millions of images. Publicly available datasets and challenges also contribute to the reproducibility of research to a great extent.

3.1 FERET

The Facial Recognition Technology (FERET) [8] program ran from 1993 through 1997, and its primary mission was to develop automatic face recognition capabilities that could be employed to assist security, intelligence, and law enforcement personnel in the performance of their duties. The final corpus consists of 14051 eight-bit grayscale images of human faces with big pose variations. This database was used primarily at the end of the last century and at the beginning of this one.

3.2 XM2VTS

XM2VTS database [9] is a database that consists of four high-quality recordings of 295 subjects taken over a period of four months. Each recording contains a speaking head shot and a rotating head shot. Overall is database outdated for current standards, however, it can be used for commercial purposes, which is untypical.

3.3 LFW

Labeled Faces in the Wild (LFW) [10] is a database of face photographs designed for unconstrained face verification (illumination, pose, and expression variation). It contains 13233 images of 5749 people, whereas 1680 people have in the dataset two or more images. Each face has been labeled with the name of the person pictured. Faces were detected by the Viola-Jones detector. Currently, there are four different sets of LFW images, including the original and three different types of aligned images. LFW was considered the benchmark dataset for FR methods, however in [5] authors remark, that most of the images can be classified as near-frontal, therefore the obtained results were unrealistically optimistic from the position of pose-invariant face recognition. Because of that, the dataset fell out of favor in recent years.

3.4 YouTube Faces

Youtube Faces Database [11] is a database of face videos designed for unconstrained face recognition. The dataset contains 3425 videos of 1595 different people, i.e., an average of 2.15 videos are available for each subject. The average length of the video is 181.3 frames. All videos were downloaded from YouTube. The database was designed with LFW images in mind and was considered to be a benchmark database for video face recognition.

3.5 CMU Multi-Pie

Multi-PIE [12] is a large dataset created at Carnegie Mellon University in 2010. It contains 337 different subjects, captured from 15 viewpoints with 19 different illuminations. The total number of images is more than 750.000, moreover, 6152 of them are annotated with AAM-based style labels. The labels have between 39 and 68 keypoints depending on the pose. All points were annotated manually.

3.6 SFC

The Social Face Classification (SFC) dataset [13] contains 4.4 million labeled faces from 4030 people, each with 800 to 1200 images. Images were collected from Facebook pictures, therefore they are captured in unconstrained conditions, i.e., with variable illumination, pose and expression. The dataset was used for training of breakthrough FR method. Unfortunately, this dataset is not publicly available.

3.7 CAS-PEAL

The CAS-PEAL Face Database [14] has been constructed under the sponsors of the National Hi-Tech Program and ISVISION. The goals to create the PEAL face database were to provide the worldwide researchers of FR community a large-scale Chinese face database for training and evaluating their algorithms, to facilitate the development of FR by providing large-scale face images with different sources of variations, especially Pose, Expression, Accessories, and Lighting (PEAL) and to advance the state-of-the-art face recognition technologies aiming at practical applications especially for the oriental.

3.8 COX Face

COX Face Database [15] consists of images and videos designed for studying three typical scenarios of video-face recognition: Video-to-Image, Image-to-Video, and Video-to-Video FR. The images are taken under a controlled environment, with high quality and resolution, in frontal view, and with neutral person expression. On the contrary, the video frames are often of low resolution and low quality, with blur, and captured by three different camcorders under poor lighting, in non-frontal view. These settings simulate the real-world matching conditions for providing researchers a solid and challenging experimental data.

3.9 PaSC

A creation of Point and Shoot Face Recognition Challenge dataset [16] was motivated by the need of social media users to recognize persons in uploaded pictures or videos automatically. The images and videos in the dataset are balanced with respect to distance to the camera, alternative sensors, frontal vs. non-frontal views, and different locations.

3.10 CelebFaces+

CelebFaces Attributes Dataset [17] is a large-scale face attributes dataset. It contains 202599 face images obtained from the Internet of 10177 different identities. Five landmark locations are annotated on each image. Moreover, each image has 40 binary attribute annotations. The images in the dataset cover large pose variations, background clutter, and have different qualities. In 2019 was this dataset extended by high-quality segmentation masks [18].

3.11 CASIA WebFace

CASIA WebFace database [19] contains 494414 images of 10575 subjects semi-automatically collected from the Internet, i.e., persons are captured in variable conditions. Most faces are centered on the images. The database is publicly available, and it is very popular as a training

dataset among modern FR algorithms, especially for small training set protocol (training set should have under 0.5M images).

3.12 IJB

IARPA Janus Benchmark datasets [20] contains both images and videos of 500 subjects. It includes both, images and video, both in the variable external conditions, whereas all faces were manually localized. The creation of IJB is motivated by a need to push the state-of-the-art in unconstrained face recognition, primarily in pose variations manner. It became a new benchmark standard during 2017 after the LFW dataset (see Sec. 3.3) fell out of the favor. There are three versions of the dataset in total, and it can be expected that authors will produce the fourth version in the foreseeable future.

3.13 MegaFace

The MegaFace dataset [21] is currently the second biggest dataset for FR, moreover, it is one of the most challenging face identification benchmarks. It currently (the number is increasing) contains 4,753,520 images of 672057 people in unconstrained conditions collected from Yahoo. The average number of images per person is 7, while three is minimum, and 2469 is maximum. The faces are detected by a commercial algorithm. The goal of this dataset is to benchmark FR algorithms on a large scale. Both companies Google and Facebook, have available an enormous amount of data, which puts the smaller research groups at a disadvantage. However, the existence of this dataset should help smaller research groups overcome this disadvantage. Unfortunately, nowadays, authors for unknown reasons do not provide access to their database anymore.

3.14 MS-Celeb-1M

MS-1M dataset [22] was designed for purposes of FR benchmark task to recognize one million celebrities from the web images. Moreover, the dataset provides rich knowledge-base information about each of the celebrities. The dataset is even larger than Megaface, which makes it the biggest publicly available dataset right now. The list of the celebrities include persons with more than 2000 different profession and come from more than 200 distinct countries/regions.

3.15 VGGFace2

VGGFace2 [23] is the newest from the large scale datasets. It was collected with three goals in mind: (1) to have both a large number of identities and also a large number of images for each identity; (2) to cover a large range of pose, age, and ethnicity; (3) to minimize the label noise. Experiments of the state-of-the-art algorithms, in which was VGGFace2 used

as a training set, led to improved recognition performance over pose and age. Finally, using the models trained on the dataset, it demonstrated state-of-the-art performance on the face recognition of IJB datasets (see Sec. 3.12), exceeding the previous state-of-the-art by a large margin.

3.16 PIPA

People In Photo Albums (PIPA) dataset [24] is a large-scale recognition dataset collected from Flickr photos. It consists of 63188 images of 2356 identities. The dataset is primary challenging due to occlusions and large pose variations (about only half of the person images containing a frontal face). In comparison to the datasets mentioned above, this dataset contains images of the whole person, therefore is also used for person recognition task (recognition based on the entire body, not just from the face).

Table 3.1: Comparison of datasets for face recognition

Dataset	**Number of Imgs/Vids**	**Number of Ids**	**Conditions**	**Resolution**
FERET [8]	14,051	Unknown	Laboratory	512×768
XM2VTSDB [9]	2,360	295	Laboratory	720×576
LFW [10]	13,233	5,749	Variable	250×250
YouTube [11]	3,425vids	1,595	Variable	Variable
CMU Multi-PIE [12]	750,000	337	Laboratory	High-Res
SFC [13]	4.4M	4030	Variable	Images
CAS-PEAL [14]	99,594	1,040	Laboratory	640×480
COX Face [15]	1,000+1,000vids	1,000	Laboratory	Unknown
PaSC [16]	9,376+2802vids	293	Variable	Unknown
CelebFaces [17]	202,599	10,177	Variable	178×218
CASIA WebFace [19]	494,414	10,575	Variable	250×250
IJB-A [20]	5,712+2,085vids	500	Variable	Variable
MegaFace [21]	4.8M	672,057	Variable	Variable
MS-Celeb-1M [22]	8,456,240	99,892	Variable	300×300
VGGFace2 [23]	3.3M	9000+	Variable	Variable
PIPA [24]	63,188	2,356	Variable	Variable
CFP [25]	7000	500	Variable	Variable

3.17 CFP

The authors have collected a new face data set that will facilitate research in the problem of frontal to profile face verification in the wild [25]. This data set aims to isolate the factor of pose variation in terms of extreme poses like profile, where many features are occluded, along with others in the wild variations. Moreover, they find that human performance on Frontal-Profile verification in this data set is only slightly worse (94.57% accuracy) than that on Frontal-Frontal verification (96.24% accuracy). However, the evaluation of many state-of-the-art algorithms, including Fisher Vector, Sub-SML, and a Deep learning algorithm, shows all of them degrade more than 10% from Frontal-Frontal to Frontal-Profile verification. The Deep learning implementation, which performs comparably to humans on Frontal-Frontal, performs significantly worse (84.91% accuracy) on Frontal-Profile. This suggests that there is a gap between human performance and automatic face recognition methods for large pose variations in unconstrained images. The dataset contains ten frontal and four profile images of 500 individuals.

3.18 CUFS

CUHK Face Sketch database (CUFS) is a database for research on face sketch synthesis and face sketch recognition. It includes 188 faces from the Chinese University of Hong Kong (CUHK) student database, 123 faces from the AR database [26], and 295 faces from the XM2VTS database [9], 606 faces in total. For each face, there is a sketch drawn by an artist based on a photo taken in a frontal pose, under normal lighting condition, and with a neutral expression.

3.19 CUFSF

CUHK Face Sketch FERET Database (CUFSF) [27] is for research on face sketch synthesis and face sketch recognition. It includes 1,194 persons from the FERET database [8]. For each person, there is a face photo in a frontal pose, under the controlled lighting condition, and with a neutral expression. Sketches were drawn by an artist when viewing these photos.

3.20 IIIT-D

IIIT-D Database [28] is a sketch database used in this research comprises of three types of sketch database: (1) Viewed sketch database; (2) Semi-forensic sketch database; (3) Forensic sketch database. The viewed sketch database comprises a total of 238 sketch-digital image pairs. The sketches are drawn by a professional sketch artist for digital images collected from different sources. The semi-forensic sketch database consists of images drawn based on the memory of a sketch artist rather than the description of an eye-witness. To prepare this database, the sketch artist is allowed to view the digital image once and is asked to draw the sketch based on his memory. The database consists of 140 digital images in total.

Forensic sketches are drawn by a sketch artist from the description of an eye-witness based on his/her recollection of the crime scene. The database includes 190 forensic sketches with corresponding digital face images in total. Unfortunately, all three parts of the database consist of images collected from the internet, therefore, authors are sharing direct links to the face images. That means that some of these links are already dead after the years.

3.21 Memory Gap Database

Memory Gap Database (MGDB) [29] is a sketch database addressing a memory problem of a description of the suspect from eye-witnesses. 100 subjects were chosen from a page with mugshots of real criminals, and four types of sketches were drawn: (1) Viewed sketches were drawn while artist looks directly at the mugshot; (2) Sketches drawn one hour after viewing the photo; (3) Sketches drawn 24 hours after viewing the photo; (4) Sketches drawn based on the description of an eye-witness, who has seen the photo immediately before. The sketches were drawn by 20 different artists, however, all four kinds of sketches for each subject was always drawn by the same one, so sketches do not have inter-artist variability.

3.22 ILSVRC

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [30] is dataset used to evaluate algorithms for object detection and image classification at large scale. The training subset contains 1.3 million images, validation set 50 thousand images and testing subset 100 thousand images of objects from 1000 categories. This dataset is currently a top dataset for image classification, and many large-scale image classification algorithms are tested during yearly ILSVRC challenges.

Chapter 4

Network Architectures

In this chapter, there will be first described the idea behind artificial neural networks (ANNs), followed by a brief description of the most common neural network features. After that, there is a comprehensive survey of the most important neural network architectures used in recent years.

ANNs are models inspired by biological neural systems (for example, the human brain) [31][32]. ANNs are, same as a human brain, composed of neurons. In the human nervous system, there can be found approximately 86 billion neurons, that are connected with approximately 10^{14} synapses.

The biological neuron is composed of the following parts:

- Soma - body of the neuron.
- Axon - output, each neuron has only one axon.
- Dendrites - input, each neuron can have up to several thousands dendrites.
- Synapses - links between Axons and Dendrites, one-way gates, which allows the transfer of the signal only in the Dendrite $\rightarrow$ Axon direction.

Transmitted signals between neurons are electrical impulses, these signals are carried to the neuron's body, where they get summed. If the final sum is above a certain threshold, the neuron send (fires) signal into its axon. The main ingenious idea of this system is that synapses can have different synaptic strengths, which is learnable, and it controls the strength of the influence of the neuron to the next one. The artificial neuron (see Figure 4.1) is arranged very similarly. The strength of the axioms is modeled by weights $\boldsymbol{W}$, and the threshold is ensured by activation function $\mathbf{f}$ (see more about activation functions in the next Section).

There are two basic types of artificial neural networks: feed-forward networks and recurrent networks. Feed-forward networks allow the signal to travel from input to output only. They are mostly used in pattern recognition. Recurrent networks can have signal traveling in both directions because of loops in the network. They are usually used for sequential tasks: time series prediction or sequence classification. It is worth mention that ANN can also be 'trained'

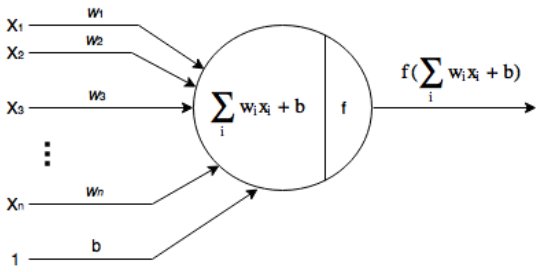


Figure 4.1: Scheme of the artificial neuron.

using unsupervised learning. This type of ANN is then called a Self-organizing map (SOM) [33] or Generative Adversarial Networks (GANs) [34].

4.1 Activation functions

Each activation function defines the output of the neuron based on the input(s) and some fixed mathematical operations. Artificial neurons necessarily don't have to have activation function. There are many different activation functions, but let's mention the most used ones in practice.

4.1.1 Sigmoid

Sigmoid function is defined as follows:

$$f(\xi) = \frac{1}{1 + e^{\xi}}, \tag{4.1}$$

where ξ is activation, see Equation 4.2.

$$\xi = \sum_{i=1}^{n} (\boldsymbol{w}_i^T \boldsymbol{x}_i + \boldsymbol{b}), \tag{4.2}$$

where $\boldsymbol{W}$ is weight matrix, $\boldsymbol{X}$ is matrix of inputs, and $\boldsymbol{b}$ is bias. The range of the values of the sigmoid function is the open interval from 0 to 1. The sigmoid function has been frequently used historically, however, it fell out of favor because it has two major drawbacks. Firstly, the sigmoid function saturates and kills gradient. This very undesirable fact is based on the saturation of the neuron when the output approaches 0 or 1, it means that the gradient is almost zero. This causes problems during back-propagation (see Section 4.4), where this very small number during multiplication "kills" the gradient, and no significant signal will flow through the neuron to its weights and recursively to its data. The second problem of the sigmoid function is that the output is not zero-centered. This is undesirable because this non-zero-centered output will come to the inputs of neurons in the next layer, and it will cause, that gradient during back-propagation will always be either positive or negative.

4.1.2 Tanh

Tanh function has following form:

$$f(\xi) = \frac{2e^{\xi}}{1+e^{\xi}}(2\xi) - 1, \tag{4.3}$$

where ξ is the activation value (see Equation 4.2). The range of the values of the tanh function is the open interval from -1 to 1. This function has an advantage over the sigmoid function that it is zero-centered, but it still can saturate. Overall, the tanh function is usually used over the sigmoid function.

4.1.3 ReLU

The Rectified Linear Unit (ReLU) is define as follows:

$$f(\xi) = max(0, \xi), \tag{4.4}$$

where ξ is the activation value (see Eq. 4.2) once again. It can be noticed activation is thresholded at zero. ReLU is probably the most popular activation function of recent years. The main advantages of ReLU are its computational simplicity and faster convergence of stochastic gradient descent (see Sec. 4.5) compared to the previous activation functions. The main disadvantage of ReLU is that there is a danger of the creation of dead neurons. This can be caused by large gradients flowing through the ReLU neuron. According to this gradient, weights can be updated in such a way that the neuron will never activate on the data point again.

4.1.4 Leaky ReLU

The Leaky ReLU is defined as follows:

$$f(\xi) = \begin{cases} \xi & \text{if } \xi > 0 \\ \alpha\xi & \text{otherwise} \end{cases} \tag{4.5}$$

where α is a small number (usually 0.01). Leaky ReLU tries to fix the "dead neuron" ReLU problem, however, the consistency of the benefit over "ordinary" ReLU is still unclear.

4.1.5 Parametric ReLU

The Parametric ReLU is a type of Leaky ReLU that, instead of having a fixed predetermined α, makes it a parameter for the neural network to train and find it itself.

4.1.6 Maxout

Maxout doesn't use the standard functional form $f(\boldsymbol{W}^T\boldsymbol{X} + \boldsymbol{b})$, where function is applied on activation value. Instead maxout neuron computes the function $\max(\boldsymbol{w}_i^T\boldsymbol{x}_i + \boldsymbol{b}_i)$. The main

advantage of maxout is that it doesn't saturate and it doesn't suffer from dead neurons, but for the price of higher computational complexity.

4.1.7 Softmax

Softmax activation function is for j-th neuron is defined as follows:

$$f(\xi)_j = \frac{e^{\xi_j}}{\sum_N e^{\xi_N}}, \tag{4.6}$$

where N represents the number of different possible outcomes (i.e., the number of neurons in the layer). The Softmax function is usually used only in the final layer of NN trained for classification tasks. Softmax converts a raw value into a posterior probability.

4.2 Layers

ANN is formed by connecting (acyclic) of the artificial neurons together. The final purpose and function of the ANN are to determine by these connections (architecture of the network), by weights, and by types of neurons (activation functions). ANN are usually organized into distinct layers of neurons. The most common ones are described in this subsection.

4.2.1 Fully-connected

The fully-connected layer is the most common type of layer. Each neuron has trainable weights, and each neuron in one layer is connected to all neurons in the previous one, however, neurons in a single layer don't share any connections, see example in Figure 4.2.

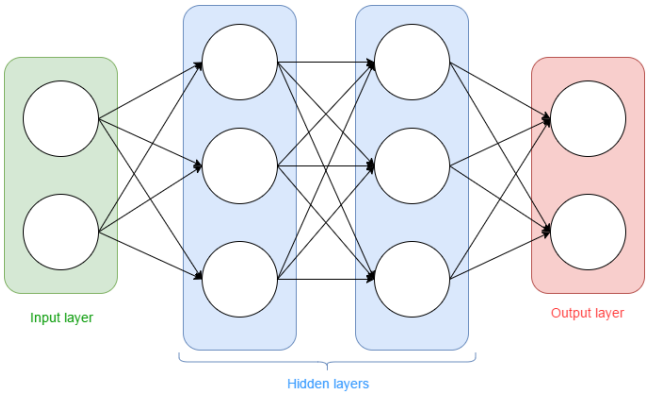


Figure 4.2: A 3-layer (input layer is not counted) neural network with two fully-connected hidden layers.

4.2.2 Convolutional

Convolutional layers are basic building stones of convolutional neural networks (CNNs). Unlike fully-connected layers, neurons in convolutional layers are connected only to a local region of the previous layer - the size of the region (height and width) is hyperparameter called the receptive field of the neuron. In signal processing, they can be imagined as a set of filters that are applied to a specific part of the signal. The number of "filters" is called depth and depends on the concrete task. The number of neurons is dependent on the length of the processed signal (size of an image) - It is necessary to cover the whole signal with each filter and sometimes it is good to use overlapping regions, and on the receptive field of the neurons. Neurons related to the one concrete filter have shared weights.

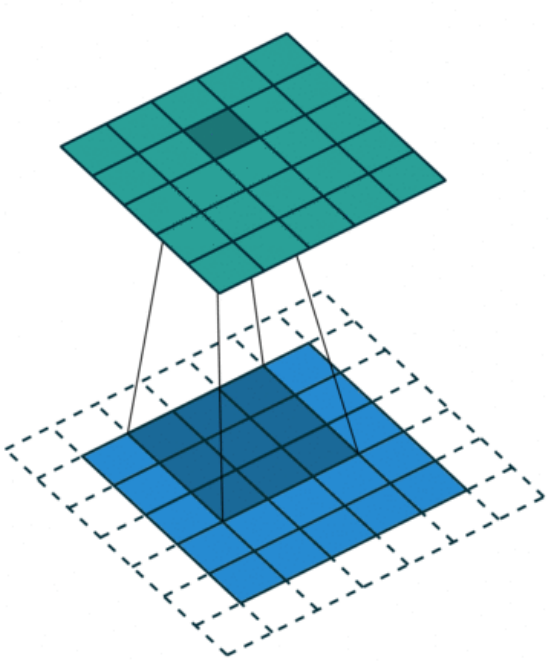


Figure 4.3: Example of the convolution with zero padding.

There exist many different types of convolutions, for example: traditional (see Figure 4.3), dilated, transposed, and depthwise (atrous) separable. Generally, their usage is dependent on the concrete task again, for example, in paper [35] is revealed that in the encoder-decoder type of NN structure, depthwise separable convolution provides better results than traditional convolution, moreover, with parameter savings.

4.2.3 Pooling

The pooling layer is another important layer, which is commonly used in CNN. Its function is to progressively reduce the spacial size of the representation to reduce the number of parameters and, therefore, a number of computations in the network. This all together decreases the chance of overfitting. Neurons in pooling layers are spatially connected to the previous layer once again, however, they don't have any trainable weights. The neurons only

make some specific mathematical operations over the related region. Average-pooling was historically very popular, but then it has fallen out of favor, and it was replaced by max-pooling, see Figure 4.4. Due to the aggressive reduction in the size of the representation (which is helpful only for smaller datasets to control overfitting), the current trend in the literature is towards using smaller filters or discarding the max-pooling layer altogether.

In [36], the authors proposed a novel network structure called Network in Network. With this enhanced approach, authors were able to utilize the global average pooling layer over feature maps in the classification layer instead of a more traditional fully-connected layer, which leads to huge parameter saving. In traditional CNN, it is difficult to interpret how the category level information from the objective cost layer is passed back to the previous convolution layer due to the fully connected layers which act as a black box in between. In contrast, global average pooling is more meaningful and interpretable as it enforces correspondence between feature maps and categories, which is made possible by stronger local modeling using the micro-network. Furthermore, the fully connected layers are prone to overfitting and heavily depend on dropout regularization, while global average pooling is itself a structural regularizer, which natively prevents overfitting for the overall structure. Experiments proved the effectiveness of this method. With the same approach in [37] they showed, that the global average pooling layer enables the convolutional neural network to have localization ability despite being trained only on the image classification task.

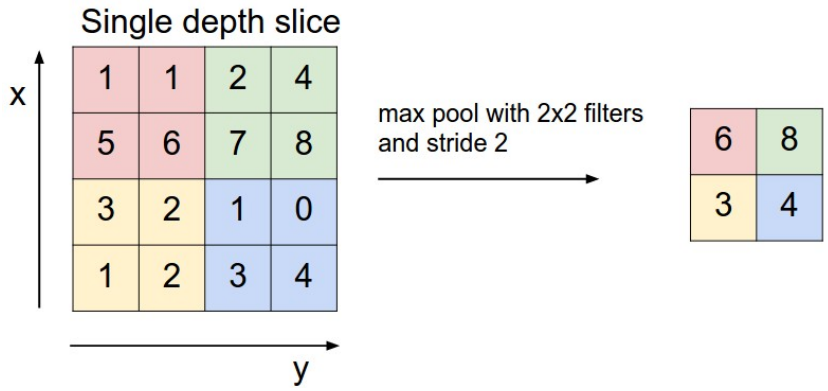


Figure 4.4: Example of the max-pooling operation, taken from [31].

4.2.4 Normalization

The normalization layer performs mathematical normalization over local input regions. A very popular type of normalization nowadays is batch normalization [38], which task is to fight with covariance shift in hidden layers by normalizing the inputs to the layers. Batch normalization (BN) also effectively increases the stability and the speed of NN training. During the training a BN layer firstly calculates the batch mean μ_B and variance σ_B^2 of the layer's input:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{4.7}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2, \tag{4.8}$$

where m is the number of samples in a mini-batch, and x_i is i-th sample and the input into the layer. In the second step is the input normalized using these calculated batch statistics:

$$\bar{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{4.9}$$

where ϵ is a small number. Lastly, is the normalized input scaled and shifted:

$$y_i = \gamma \bar{x_i} + \beta, \tag{4.10}$$

where γ and β are trainable parameters of the batch normalization layer.

From other normalization techniques lets mention weight normalization [39], layer normalization [40], and instance normalization [41].

4.2.5 Loss

The loss layer is used as the last layer during the training of ANN, and it specifies the enumeration of the difference between predicted and ground-truth values (loss, error). The choice of the loss function depends on the concrete type of problem. Common problems can be divided into two following categories: classification, and regression. Most common functions used during the classification are hinge loss L_h defined as follows:

$$L_h = \sum_{j \neq y} \max(0, f_j - f_y + 1), \tag{4.11}$$

where f_y is functional value of correct class, and f_j is functional value of predicted class while $f = f(\boldsymbol{x}_i, \boldsymbol{W})$ is output of the penultimate layer. Another popular loss function for classification problems is cross entropy loss L_{ce}:

$$L_{ce} = -\sum_{i}^{N} p_i \log \hat{p}_i, \tag{4.12}$$

where p_i is the target probability distribution, and $\hat{p}_i$ is predicted probability distribution, and N is a total number of classes. Regression is the task of predicting real-valued quantities. The most popular loss for regression is L_2 squared norm defined in Equation 4.13.

$$L_2 = ||f - y||_2^2, \tag{4.13}$$

where f is predicted value and y is ground-truth value.

The design of the loss function is one of the most attributes of modern FR approaches. Nowadays, there exist plenty of different loss functions. The most important ones can be found in Chapter 5.

4.3 Regularization techniques

There are several other ways how to prevent overfitting of the network, the most popular is probably the Dropout method [42] (but there are others, for example, L1 regularization, and L2 regularization). Dropout is a really simple but very effective approach - at each stage of training, each neuron has some probability **p** (a hyperparameter) to stay active. It is "dropped out" otherwise.

4.4 Gradient and back-propagation

Back-propagation is the most common training method of ANN used in conjunction with an optimization method. It is used for gradient computing through recursive application of chain rule. The whole algorithm can be divided into the following parts:

1. The forward pass - At the beginning, the algorithm lets ANN predicts the output with given weights and biases.
2. Calculating the total error - In the second step, the loss layer calculates the total error L.
3. The backward pass - In this step is computed gradient ∇L for the individual parameters $(\boldsymbol{W}, \boldsymbol{b})$. The gradient is then used to perform a parameter update (more in the next section) - it is found a direction with the biggest descent.

4.5 Parameter update - optimization methods

Before the training process, it is necessary to initialize parameters. Historically was used the setting with all parameters equal zero, but nowadays is initialization with a small random number or pretraining more common. The most popular initializer is Xavier normal initializer [43].

During the training, once the analytic gradient is computed with back-propagation, the gradients are used to perform a parameter update. There are several commonly used methods for performing the update, which are discussed next.

4.5.1 SGD

Stochastic gradient descent (SGD) is a first-order optimization algorithm. SGD has the same mathematical principle as Gradient Descent, but since only limited memory is available, training data are divided into batches, and further SGD works only with them. SGD update in step $t + 1$ for n observation is defined as follows:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^{t} - \gamma_t \sum_{i=1}^{n} \nabla L_i(\boldsymbol{\omega}^{t}), \tag{4.14}$$

where $\boldsymbol{\omega}$ are trainable parameters, γ is learning rate (hyperparameter), and L is loss (error) function. SGD's main advantage is its low computational time, the disadvantage is, that method doesn't know the size of the step, that it should take in the negative gradient direction.

4.5.2 Momentum

The momentum method usually achieves better results than SGD in most cases. Momentum can be imagined as a weighted average between the newly computed gradient and the past

gradients. The update of parameters with momentum $\Delta\omega^t$ has then the following form:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t + \Delta\boldsymbol{\omega}^t = \boldsymbol{\omega}^t - \gamma_t \nabla L(\boldsymbol{\omega}^t) + \alpha\Delta\boldsymbol{\omega}^{t-1}, \tag{4.15}$$

where α is momentum hyperparameter, usually chosen between 0.9 and 1.0.

4.5.3 Nestorov Momentum

Nestorov momentum is a different version of the momentum method, which has been gaining popularity in recent years. Unlike in the Momentum method, the gradient is computed AFTER the momentum step. The idea behind this step is that the gradient in the "look-ahead" point should be more accurate, see Figure 4.5. Nestorov Momentum is defined as follows:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t + \Delta\boldsymbol{\omega}^t - \gamma_t \nabla L(\boldsymbol{\omega}^t + \Delta\boldsymbol{\omega}^t), \tag{4.16}$$

where $\Delta\boldsymbol{\omega}^t = -\gamma_t \nabla L(\boldsymbol{\omega}^t) + \alpha\Delta\boldsymbol{\omega}^{t-1}$ is same momentum as before.

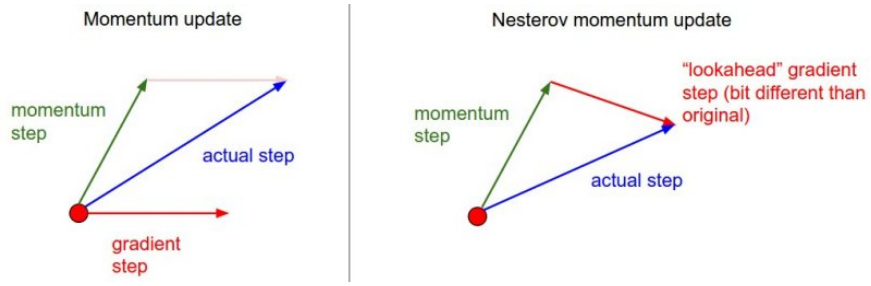


Figure 4.5: Comparison of Momentum and Nestorov Momentum methods.

4.5.4 Adagrad

All previous methods use the learning rate globally and equally for all parameters, however, the adaptive approach shows great promise. Adagrad is an adaptive learning rate method proposed in [44], see Equations 4.17 and 4.18. Its main disadvantage is that Adagrad is sometimes too aggressive, and it stops (or slows) learning too early.

$$\sigma_i^{t+1} = \sigma_i^t + ||\nabla L(\boldsymbol{\omega}_i^t)||_2^2, \tag{4.17}$$

$$\boldsymbol{\omega}_i^{t+1} = \boldsymbol{\omega}_i^t - \frac{\gamma_t \nabla L(\boldsymbol{\omega}_i^t)}{\sqrt{\sigma_i^{t+1}} + \epsilon}, \tag{4.18}$$

where ϵ is a small number used to avoiding division by zero.

4.5.5 RMSprop

RMSprop is very effective, but currently, unpublished [45] adaptive learning rate method. RMSprop attempts to reduce the Adagrad's aggressiveness and therefore adjusts the Adagrad method in a very simple way:

$$\sigma_i^{t+1} = \alpha\sigma_i^t + (1-\alpha)||\nabla L(\boldsymbol{\omega}_i^t)||_2^2, \tag{4.19}$$

$$\boldsymbol{\omega}_i^{t+1} = \boldsymbol{\omega}_i^t - \frac{\gamma_t \nabla L(\boldsymbol{\omega}_i^t)}{\sqrt{\sigma_i^{t+1}} + \epsilon}, \tag{4.20}$$

where α is hyperparameter (decay), thanks to which parameter's updates do not become monotonically smaller.

4.5.6 Adam

Adam is recently proposed [46] adaptive learning rate update method, which, in contrast with the RMSprop method, uses the "smooth" version of gradient m (see Equation 4.21) instead of the raw gradient vector. It can be said Adam is RMSprop with momentum.

$$m^{t+1} = \beta_1 m^t + (1 - \beta_1)\nabla L(\boldsymbol{\omega}_i^t), \tag{4.21}$$

$$v^{t+1} = \beta_2 v^t + (1 - \beta_2)||\nabla L(\boldsymbol{\omega}_i^t)||_2^2, \tag{4.22}$$

$$\boldsymbol{\omega}_i^{t+1} = \boldsymbol{\omega}_i^t - \frac{\gamma_t m^{t+1}}{\sqrt{v^{t+1}} + \epsilon}, \tag{4.23}$$

where β_1 and β_2 are hyperparameters, usually chosen between 0.900 and 0.999.

4.5.7 Nadam

Much like Adam is essentially RMSprop with momentum, Nadam is RMSprop with Nestorov Momentum [47].

4.6 AlexNet

In 2012, Krizhevsky et al. published an article [48] about a novel neural network, named AlexNet. AlexNet is composed of eight layers - five convolutional and three fully connected layers, see Fig. 4.6, and was trained on ImageNet dataset. This dataset contains over 15 million labeled high-resolution images belonging to 1000 categories. In that time, the network of such size was too large to be trained on a single GPU, so it was necessary to perform training of multiple GPUs. The novelty of this work stems from using few, until that moment very unusual, features. The first most significant upgrade was the usage of ReLU nonlinearity (Sec. 4.1.3). Until that, it was Tanh nonlinearity much more usual. Another important upgrade was the usage of overlapping in pooling layers. They reported that a model with overlapping pooling is less prone to overfitting. However, for CNN with so many parameters (60 million) was overfitting still a significant problem, therefore they used the Dropout method and data augmentation. Achieved results (on ILSVRC-2010) were stunning - top1 and top5 test set error rates of 37.5% and 17.0%, whereas state-of-the-art performance till that moment was 45.7% and 25.7%. Moreover, they tested their CNN in other competitions, and in all of them, they improved state-of-the-art results significantly.

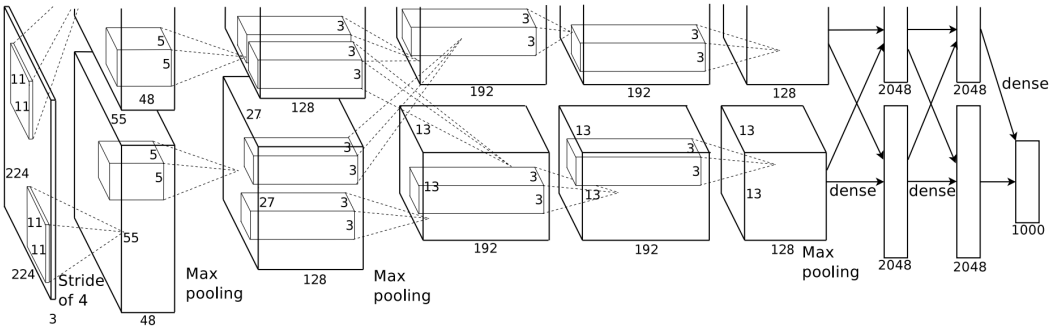


Figure 4.6: Architecture of AlexNet [48]. Figure taken from the original paper.

4.7 VGG

In 2014, Simonyan et al. [49] presented novel CNN architecture, and it fast became a gold standard among the neural networks designed for image recognition. Their paper has three main contributions: (1) it examines and evaluates the influence of the depth of NN; (2) it utilizes very small convolutional filters (3x3) with great success; (3) it improves state-of-the-art results significantly.

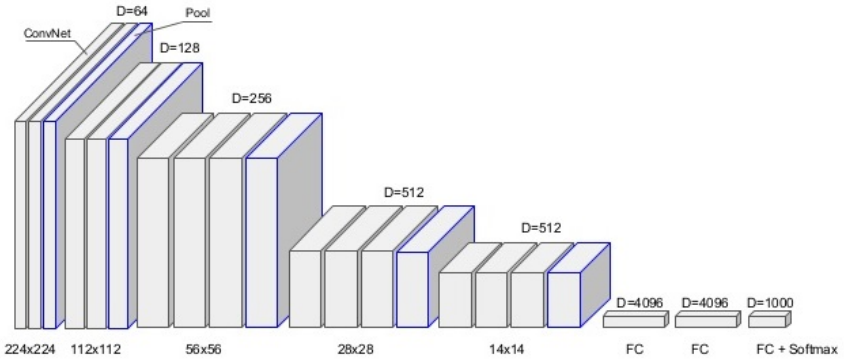


Figure 4.7: Topology of CNN VGG16 (CNN with 16 weight layers).

Authors tested overall five topologies (11-19 weight layers) of CNN, all of them contained convolutional layers, where were used filters with very small receptive field: 3x3. The convolution stride was always fixed to 1 pixel. After some number of convolutional layers, the authors utilized the max-pooling layer (which is performed over a 2x2 pixel window with stride 2). Moreover, are ALL hidden layers equipped with the ReLU non-linearity. Each architecture is ended with three fully-connected layers, while the first two have 4096 neurons each and since the third one contains 1000 neurons as it performs classification into 1000 classes. The final layer utilizes the Softmax activation function. See Figure 4.7 for the most popular topology, which is used in many applications today.

However, in 2014, thus topology was quite different from the ones used in the top-performing algorithms. The main difference is in the size of convolutional filters. The main idea stems from the fact that two stacked 3x3 convolutional layers without spatial pooling between them have an effective receptive field 5x5. Three such layers have a 7x7 effective receptive field. There are two main differences between this approach and using a single 7x7 convolutional

layer. First, this approach can incorporate up to three non-linearities instead of a single one, which makes decision function more discriminative, Second, this approach decreases the number of parameters, which can lead to the faster convergence during training. The last peculiarity of the VGG topology is that after the max-pooling layer is the number of filters always doubled - this leads to constant computational complexity during the training of each convolutional layer.

The NN was optimized according to the multinomial logistic regression objective using SGD with momentum. The size of the mini-batch was 256, and the size of input images is fixed to 224x224 pixels. VGG was trained and tested on the ILSCVRC-2012 dataset [30]. The VGG architecture won the whole challenge and secured second place in the localization part.

4.8 InceptionNet + NiN

In 2014, Szegedy et al. [50] came with an important milestone in the development of CNN classifiers. Previous most popular CNNs stacked convolution layers going deeper or add more filters going wider, hoping to get better performance. This is a very easy and safe way of training higher quality models, however, this solution intuitively comes with two major drawbacks. Bigger size means a larger number of parameters, which firstly makes the enlarged network more prone to overfitting, and secondly, it dramatically increases computational complexity. Moreover, during the process of neural network designing, it can be tough to choose the right size of the kernel. This problem stems from the fact that important parts in the image can be an extremely large variation in size. Generally, a larger kernel is preferred for information that is distributed more globally, while a smaller kernel is preferred for information that is distributed more locally.

To address these problems, the authors decided to have multiple filters of different sizes operating on the same level, which is technically just going wider, but more clever. The designed module was named inception module, and its standard version can be viewed in Fig 4.8.

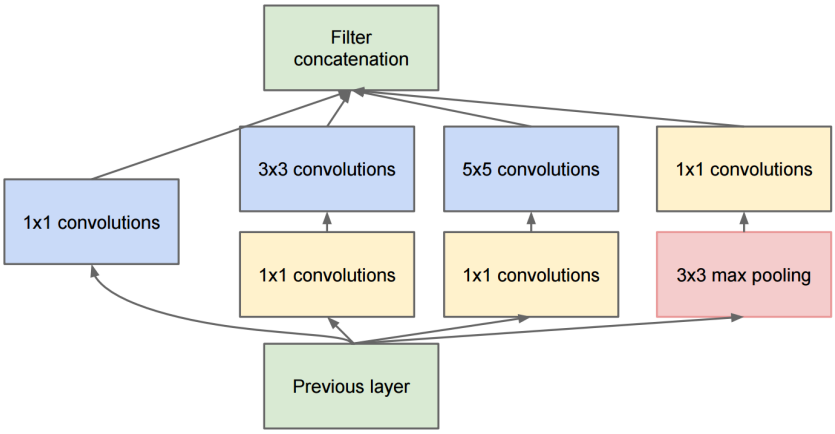


Figure 4.8: The standard version of the inception module.

The standard inception module performs convolution with three different sizes of filters, and additionally, max-pooling (with stride 1) is also performed. All the outputs are concatenated

and sent to the next neural network block. It should be pointed on the usage extra 1×1 convolutions before 3×3 and 5×5 convolutions and after the max pooling. This approach is called Network in Network (NiN), and it was first introduced in [51], and though adding an extra operation seems counter-intuitive, employing additional convolution to decrease the number of channels of an input spares plenty of computational resources. The usage of such convolution can be viewed as a very similar operation as the usage of max pooling but for the channel dimension instead of width and height dimension, nevertheless, with a possible advantage to learn complex cross-channel information.

Lin et al. [51] also compared the usage of fully-connected layers with the usage of global average pooling at the end of the network. The main disadvantages of fully-connected layers are their propensity to overfitting and a huge number of parameters. Therefore, they replaced these last layers with the global average pooling operation. The idea behind this is to generate one feature map for each corresponding category of classification tasks in the last convolutional layer. After that, instead of adding fully connected layers on top of the feature maps, the average of each feature map is taken, and the resulting vector is fed directly into the softmax layer. This approach has the following advantages: (1) the convolutional structure is enforced to correspondence between feature maps and categories; (2) overfitting avoidance; (3) global average pooling is very robust to spatial transformations. Furthermore, Zhou et al. [52] showed that fully-convolutional networks have a great ability to encode localization information despite being trained only for the classification task.

Up to date, there were proposed five different versions of inception net, where with its last version Inception-Resnet [53] in 2016, the authors reached state-of-the-art results in ILSVRC challenge.

4.9 Highway Networks

Srivastava et al. [54] presented a method for training of very deep networks utilizing attention mechanism called Highway networks. The output y of typical plain network is defined as follows: $y = H(\boldsymbol{x}, \boldsymbol{W_H})$, where H is the transform function followed by an activation function, $\boldsymbol{x}$ is input and $\boldsymbol{W_H}$ are weights. In the highway networks, however, two non-linear transform T, and C were introduced:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C), \tag{4.24}$$

where T is the Transform gate and C is the Carry Gate. In the final implementation $C = 1 - T$, that means:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_C)). \tag{4.25}$$

With this more simple implementation, there are interesting conditions for particular T values:

$$y = \begin{cases} x & \text{if } T(x, W_T) = 0 \\ H(x, W_H) & \text{if } T(x, W_T) = 1 \end{cases} \tag{4.26}$$

To further elaborate, when $T = 0$, the layer passes the input as output directly, whereas, when $T = 1$, the highway network performs the same operation as a plain network. That means that block can smoothly vary its behavior between standard nonlinear transform and simple passing its input through. The authors tested the method on CIFAR-10 and CIFAR-100 dataset, whereas reached state-of-the-art results on the latter one.

4.10 ResNet

In 2016, He et al. [55] proposed novel DNN architecture to address a problem of degradation during training very deep networks. During testing, the authors observed the counter-intuitive phenomenon - adding more layers to the architecture cause higher training error. Historically this problem occurred because vanishing/exploding of gradients, however, this problem has been largely addressed by normalized initialization and intermediate normalization layers. This degradation of training accuracies indicates that not all systems are similarly easy to optimize. Authors argue that after adding layers to shallower architecture, these new layers should be trained to the identity mapping, and the other layers should remain unchanged. But experiments showed that current solvers probably have problems with the identity mapping of multiple nonlinear layers and therefore are unable to find such a solution or comparably good one.

The authors address the degradation problem by introducing a deep residual learning framework. Instead of learning every few stacked layers directly fit a desired underlying mapping, they let these layers fit a residual mapping. Let $H(\boldsymbol{x})$ be the desired underlying mapping of a few stacked layers with $\boldsymbol{x}$ denoting the input to the first of these layers. Based on the hypothesis that multiple nonlinear layers can asymptotically approximate complicated functions, multiple nonlinear layers should be able to approximate the residual function asymptotically, i.e., $F(\boldsymbol{x}) = H(\boldsymbol{x}) - \boldsymbol{x}$. The original function thus becomes $F(\boldsymbol{x}) + \boldsymbol{x}$. Although both forms should be able to approximate the desired functions asymptotically, the ease of learning is different. The formulation of $F(\boldsymbol{x})+\boldsymbol{x}$ authors realized by the shortcut connections as element-wise addition, see Figure 4.9.

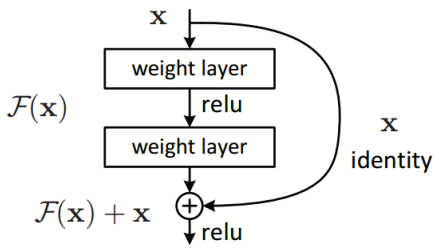


Figure 4.9: Residual learning: a building block [55].

Formally is building block defined as:

$$\boldsymbol{y} = F(\boldsymbol{x}, \{\boldsymbol{W}_i\}) + \boldsymbol{x}, \tag{4.27}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{y}$ is the output vector of the layers and $F(\boldsymbol{x}, \{\boldsymbol{W}_i\})$ represents the residual mapping to be learned. The main advantage of this configuration is that the computational complexity of the element-wise addition is negligible. The dimensions of $\boldsymbol{x}$ and F must be equal, however, if this is not the case, then it can be performed a linear projection of $\boldsymbol{x}$. The function F can represent multiple fully connected or convolution layers, in the latter case, the element-wise addition is performed channel by channel. Authors experimented with F that contains one, two, or three layers, however, if F has only a single layer, no advantages were observed.

The authors proposed residual nets with a depth of up to 152 layers and evaluated them on on the ImageNet2012 classification dataset [30]. The models are trained on the 1.28

million training images. They 152-layer ResNet outperformed state-of-the-art with a top-5 validation error of 4.49%. After that, they combined six ResNets of different depth. This leads to 3.57% top-5 error rate on the test set. This entry won the 1st place in ILSVRC-2015. Overall, ResNet has a very high potential for face recognition, and it is probably the most significant upgrade of neural networks since AlexNet from 2012 [48].

It should be noted, that there exist plenty of modification of original ResNet, for example, pre-activation ResNet [56] or ResNeXt [57], whereas the former deals with problems of gradient explosion/vanishing in the shortcut connection, whereas, the latter improves classification accuracy by increasing the size of set of transformation while maintaining same computational complexity.

4.11 DenseNet

Densely connected convolutional network (DenseNet) was proposed by Huang et al. [58] and according to testing this approach has three main advantages: (1) it alleviates the vanishing-gradient problem; (2) it strengthen feature propagation and encourages feature reuse; (3) it substantially reduce the number of parameters, because of their efficient usage. The paper is also Best CVPR2017 article award winner.

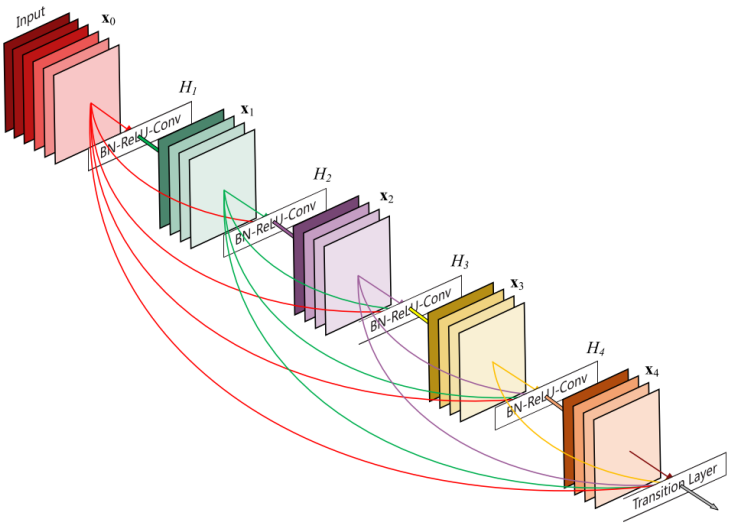


Figure 4.10: A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input [58].

The main idea of this architecture is the following: Each layer is connected to every other layer in feed-forward fashion, i.e., each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. In contrast to ResNet, these additional features are not summed together, instead, they are combined by concatenation (Figure 4.10). That means, each layer adds a small set of feature-maps to the collective knowledge of network, therefore final classifier can make a decision based on all feature-maps in the network.

Formally the output of l-th layer is defined as follows:

$$\boldsymbol{y}_l = F([x_0, x_1, ..., x_{l-1}]), \tag{4.28}$$

where $[x_0, x_1, ..., x_{l-1}]$ refers to concatenation of the feature-maps produces in layers 0,1,..,l-1. Authors also introduced new hyperparameter - growth rate k. If each function H_l produces k feature maps, it follows that l^{th} layer has $k_0 + k \times (l-1)$ input feature maps, where k_0 is the number of channels in the input layer. It should be noted, that even with a relative small growth rate the architecture obtains state-of-the art results.

Experiments also show the superiority of this architecture over the standard ResNet, which is in fact quite similar, however, the small modifications in the architecture lead to substantially different behavior of the two network architectures: (1) feature reuse - all subsequent layers can access the feature maps learned by any of DenseNet layers; (2) shortcuts provide additional supervision from loss function; (3) lower computational complexity - DenseNet reaches better results than ResNet with about 30% fewer parameters.

4.12 PyramidalNet

PyramidNet [59] method enhances ResNet, to by more concrete its pre-activation version [56]. In comparison with original ResNet, pre-activation ResNet moves ReLU activation from shortcut-connection path to convolution-layer path, see Fig. 4.11.

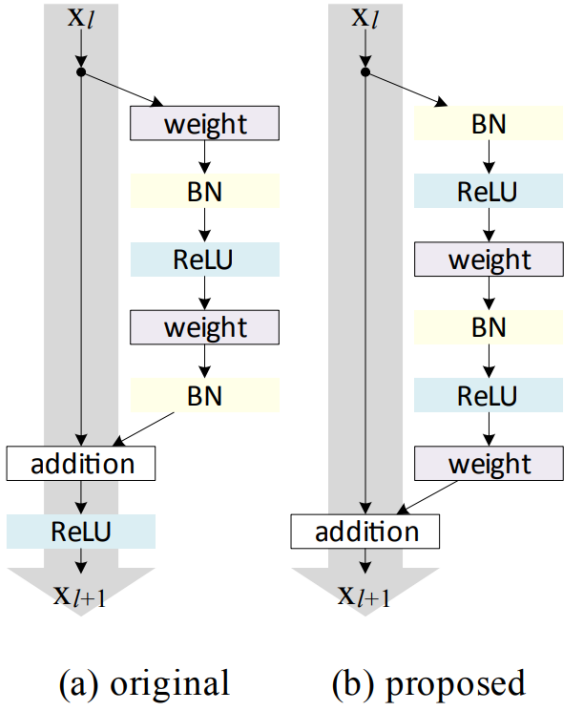


Figure 4.11: Comparison of original and pre-activation ResNet. Taken from [56].

With this simple modification, Pre-activation Resnet provides better results. In their ablation study, authors tried plenty of modifications of the shortcut connection, but the network with

"clean" shortcuts always provided the best results. This is caused by a potential gradient explosion/vanishing phenomenon, which can be caused by an identity mapping violation by the presence of math operation in shortcut connections.

Authors of PyramidNet found out that by gradually increasing the feature map dimension, instead of increasing the feature map dimensions sharply, the classification accuracy is improved. They also continued experiments with the positions of ReLU and Batch Normalization operation within a residual building block.

4.13 Squeeze-and-Excitation Networks

Hu et al. [60] proposed Squeeze-and-Excitation (SE) block, which can be used directly in existing architectures at minimal additional computational cost (approximately 4% relative parameter increase), while improves results significantly. The main idea behind the SE block stems from the statement that during the training, convolutions have to model not only spatial dependencies but also channel relationships. After adding some tools to model these channel relationships instead of them, the training of convolutions should be easier, because they can focus on the spatial dependencies only. Therefore, the main goal of the SE block is to improve the representation power of a network (increase its sensitivity to informative features) by explicitly modeling of inter-dependencies between the channels of its convolutional features.

To further elaborate, let's have following the transformation of an input $\boldsymbol{X} \in \Re^{H' \times W' \times C'}$ to feature maps $\boldsymbol{U} \in \Re^{H \times W \times C}$:

$$\boldsymbol{u}_c = \boldsymbol{v}_c * \boldsymbol{X} = \sum_{s=1}^{C'} \boldsymbol{v}_c^s * \boldsymbol{x}^s, \tag{4.29}$$

where $*$ denotes convolution, $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_C]$ is the learned set of filter kernels, where $\boldsymbol{v}_c = [\boldsymbol{v}_c^1, \boldsymbol{v}_c^2, ..., \boldsymbol{v}_c^{C'}]$, $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^{C'}]$ and $\boldsymbol{u}_c \in \Re^{H \times W}$. $\boldsymbol{v}_c^s$ is a 2D spatial kernel representing a single channel of $\boldsymbol{v}_c$, that interacts with corresponding channel of $\boldsymbol{X}$. Bias term is omitted to simplify the notation. Since the output is by default produces by an unweighted summation through all channels, channel dependencies are implicitly embedded in $\boldsymbol{v}_c$. That means the dependencies are entangled with the local spatial correlation captured by the filters.

SE block has to main parts - squeeze part, which embeds global information, and excitation part, which performs adaptive recalibration, see Fig. 4.12. That means first part squeeze global information into channel descriptor by performing global average pooling operation. Formally, a statistic $z \in \Re^C$ is generated by shrinking $\boldsymbol{U}$ through its spatial dimensions $H \times W$:

$$z_c = \boldsymbol{F}_{sq}(\boldsymbol{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j). \tag{4.30}$$

Such output can be interpreted as a collection of the local descriptors whose statistics are expressive for the whole image. To utilize such information, squeeze operation is followed by a second one (excitation operation), which aims to capture channel-wise dependencies fully. Such a function should be primary flexible, and it must learn a non-mutually exclusive relationship. Simple gating mechanism with sigmoid activation meets such criteria:

$$\boldsymbol{s} = \boldsymbol{F}_{ex}(\boldsymbol{z}, \boldsymbol{W}) = \sigma(g(\boldsymbol{z}, \boldsymbol{W})) = \sigma(\boldsymbol{W}_2 \delta(\boldsymbol{W}_1 \boldsymbol{z})), \tag{4.31}$$

where δ is the ReLU activation function, $\boldsymbol{W}_1 \in \Re^{\frac{C}{r} \times C}$, and $\boldsymbol{W}_2 \in \Re^{C \times \frac{C}{r}}$. The gating mechanism is parameterize by forming a bottleneck with two fully-connected layers, first with reduction ratio r, and then second a dimensionality-increasing layer. The final output of the block is obtained by rescaling $\boldsymbol{U}$ with the activations $\boldsymbol{s}$:

$$\tilde{\boldsymbol{x}}_c = \boldsymbol{F}_{scale}(\boldsymbol{u}_c, s_c) = s_c \boldsymbol{u}_c, \tag{4.32}$$

where $\boldsymbol{F}_{scale}(\boldsymbol{u}_c, s_c)$ refers to a channel-wise multiplication between scalar s_c and the feature map $\boldsymbol{u}_c$.

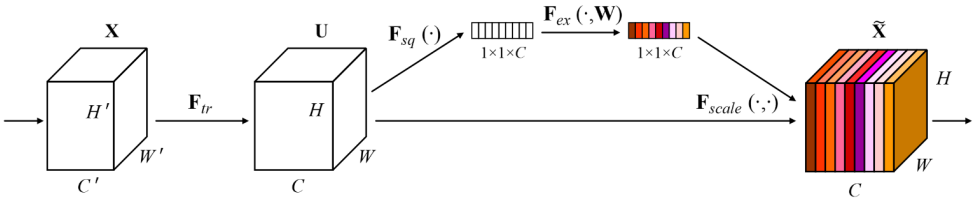


Figure 4.12: Squeeze-and-Excitation block. Taken from [60].

Best results were reached with $r = 16$. Also, because lower layer features are typically more general, while higher layers produce features with greater specificity, removing the SE block for the last few layers can improve computation cost dramatically for only a marginal cost of performance.

4.14 Autoencoders

Autoencoders generally are unsupervised learning techniques used for representation learning. More specifically, a feed-forward neural network is using the "bottleneck" structure (also called bow-tie structure), which enforces to learn compressed knowledge representation of the original input. The main idea of this structure is, firstly, in an Encoder part compressing the data from input raw pixels into a feature vector representation (i.e., latent space representation). Secondly, the Decoder takes these features, and via upsampling produces an output map (or outputs maps) with the same size, see Figure 4.13.

The network is then trained by minimizing the reconstruction error $L(x, \hat{x})$, where x is the original input to the network, and $\hat{x}$ is its output. As a distance metric can be used l_2 distance, for example.

The size of the latent space constrains the amount of information that can be encoded. During designing the network is important to create the latent space to be big enough to be able to encode all important information. Nevertheless, it should also be small enough to filter out all unnecessary information. When the latent space is too big, there is also a much more significant danger of overfitting.

In the usual implementation of a plain autoencoder, there are no restrictions applied to the latent space. However, such an approach has its limits, so in 2013, Kingma and Welling [61] presented Variational Autoencoder (VAE) network. VAE incorporates regularization by explicitly learning a joint distribution over data via forcing the latent space to follow

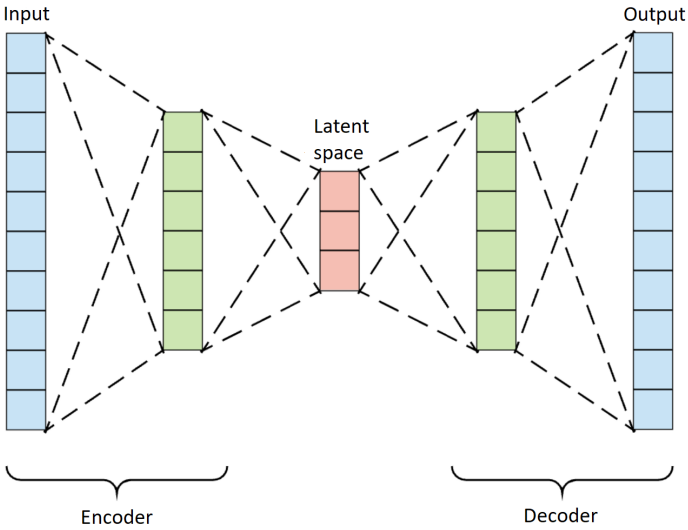


Figure 4.13: Standard autoencoder structure.

a Gaussian distribution. The loss function of the variational autoencoder is the sum of reconstruction loss and a regularizer, i.e., the negative log-likelihood of a single datapoint and Kullback-Leibler divergence between Encoder's distribution and the forced Gaussian distribution. The first term encourages the decoder to learn to reconstruct data, while the second one measures how close the two distributions are.

Chapter 5

Loss Functions

The design of loss function is one of the three primary attributes of the modern FR approach. Such design primary depends on testing protocol, which should be used in, for more information, see Section 2.3. In the scenario of closed-set settings, the used algorithm presumes that the class features are separable, therefore, it is not necessary to learn any margin between them.

However, things get much more complicated in the case of the open-set classification. Since it is impossible to classify faces to known identities in the training set, it is necessary to map faces to a discriminative feature space. Therefore, this scenario can be viewed as a metric learning problem.

There are two main approaches to design margin between classes according to the used metric: (1) losses based on Euclidean margin; (2) losses based on Angular and cosine margin.

5.1 Euclidean margin based losses

5.1.1 Softmax loss

Technically, there is no term as such Softmax loss, however, in literature is this term commonly used referring cross-entropy loss over the output with a Softmax activation function. Softmax loss for a mini-batch of size m is defined as follows:

$$L = \frac{1}{m}\sum_{i=1}^{m} -\log\frac{e^{fy_i}}{\sum_{j=1}^{n} e^{f_j}}, \tag{5.1}$$

where n is number of classes, $\boldsymbol{x}_i$ is the input feature with the ground-truth label y_i, f denotes class score vector. In neural networks, f is usually the output of fully-connected layer, so for i-th training sample can be loss L_i reformulate as follows:

$$L = -\log\frac{e^{\boldsymbol{W}_{y_i}^T\boldsymbol{x}_i+b_{y_i}}}{\sum_{j=1}^{n} e^{\boldsymbol{W}_j^T\boldsymbol{x}_i+b_j}}, \tag{5.2}$$

where W_i and b_i are weights and bias of last fully-connected layer corresponding to class i, respectively.

The decision criteria of Softmax is based on the posterior probability, whereas, the tested will be assigned to the class with the highest one. The main advantages of Softmax loss are: (1) the score values are easily interpretable because they represent log probabilities for each class; (2) it does not require any changes in standard ground-truth labels. Its main disadvantage is the lack of explicit optimization of the features to have a higher similarity score for positive pairs and lower negative ones. This fact makes usage of Softmax loss in open-set classification problematic.

5.1.2 Contrastive loss

Contrastive loss [62][63] is a distance-based loss, which means it tries to ensure that samples from the same class are embedded close together. In other words, the Contrastive loss function is employed to learn the parameters W in such a way that samples from the same class are pulled together, and examples from different ones are pushed apart. Unlike Softmax loss, where the loss is calculated as a sum over samples, the loss function is calculated over pairs of samples.

Formally, let $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \Re$ be a pair of input vectors and let Y be a binary label assigned to this pair ($Y = 0$, if $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are from the same class, $Y = 1$ otherwise). Then Contrastive loss for one pair is defined as follows:

$$L = yd + (1 - y)\max(m - d, 0)^2, \tag{5.3}$$

where m is a margin used to "tighten" the constraint, d is Euclidean distance $d = ||f_1 - f_2||_2$ between the two features f_1 and f_2 derived from input vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ respectively.

Contrastive loss is mostly used with a Siamese network structure, which is a feed-forward network with two identical branches with shared parameters. Each pair sample is used as input to one of these branches to obtain the embedding. The main advantage of the Contrastive loss is its low computational complexity, easy implementation, and ability to produce discriminative embedding even for open-set classification protocol. The main disadvantage is the necessity of data pairs, which usually means additional preprocessing of the training data.

5.1.3 Triplet loss

In 2015, Schroff et al. [64] presented a novel loss function, which can be viewed as an extension of Contrastive loss. Instead of sample pairs, Triplet loss is using triplets, which always contains one anchor sample x_a, one positive sample x_p from the same class as the anchor sample, and one negative sample x_n from a different class. Given these three samples, we want to be valid the following equation:

$$||x^a - x^p||_2^2 + \alpha < ||x^a - x^n||_2^2, \forall (x^a, x^p, x^n) \in T, \tag{5.4}$$

where α is margin enforced between positive and negative pairs, and T is the set of all possible triplets in the training set. The triplet loss for one triplet is then defined as follows:

$$L_{tr} = \left[||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + \alpha\right]_+, \tag{5.5}$$

where $f(*)$ are feature vectors (embeddings) derived from their respective inputs.

Therefore, during training, a network is trying to produce such embedding, that positive sample is closer to anchor than a negative one, see Fig. 5.1.

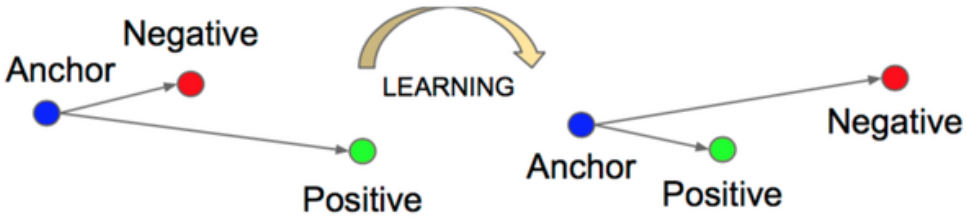


Figure 5.1: Triplet loss training. Taken from [64].

The main advantage of Triplet loss is its very effective mapping to compact Euclidean space, huge representation efficiency, and very good results for open-set classification tasks. However, there exist a huge number of possible triplets, whereas a big portion of them already fulfill Eq. 5.4. It means that during the training of the network is necessary to select only the triplets, which violates this constraint. Training is very slow and ineffective otherwise. Unfortunately, it is impossible to compute all these triplets in advance, therefore, triplets are usually generated online during the training. This brings a big computational complexity of the Triplet loss. Moreover, experiments showed the loss is very greedy for a large amount of the training data and that it generally reaches much better results with big mini-batch size.

5.1.4 Center loss

In order to enhance the discriminative power of deeply learned features, Wen et al. [65] proposed Center loss. The Center loss simultaneously learns a center for features of each class and penalizes the distance between the features and their corresponding class centers.

Center loss is defined as follows:

$$L_C = \frac{1}{2}\sum_{i=1}^{m} ||\boldsymbol{x}_i - \boldsymbol{c}_{y_i}||_2^2, \tag{5.6}$$

where $\boldsymbol{x}_i$ is an input feature with the ground-truth label y_i, c_{y_i} denotes the y_ith class feature center, and m is mini-batch size. In an ideal case, the center for each class will be calculated from the whole training set. However, this approach is very impractical, so the centers are updated in each training iteration based on mini-batch. To avoid large changes caused by a few mislabeled samples, α parameter is used to control the learning rate of centers:

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i, y_j)(\boldsymbol{c}_j - \boldsymbol{x}_i)}{1 + \sum_{i=1}^{m} \delta(y_i, y_j)}, \tag{5.7}$$

where $\delta(\cdot,\cdot)$ is the indicator function. To further improve the loss performance, Softmax loss supervision is utilized:

$$L = L_S + \lambda L_C, \tag{5.8}$$

where λ is balancing parameter usually fixed to $\lambda = 0.003$. In comparison with Contrastive and Triplet loss, Center loss reaches very similar results with more efficiency and without any further training data preprocessing (i.e., pairs or triplets creation). Moreover, Center loss is much easier to implement and incorporate into the standard neural network architectures.

5.2 Angular and cosine margin based losses

5.2.1 Angular Softmax loss

In 2017, Liu et al. [7] presented novel loss function - Angular Softmax loss (A-Softmax). A-Softmax addresses the problem of the original Softmax, which does not explicitly optimize the features to have a higher similarity score for positive pairs and lower for negative ones.

Presuming standard CNN, first step towards the A-loss is a very simple reformulation of the original Softmax:

$$L_i = -\log \frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j}} = -\log \frac{e^{||\boldsymbol{W}_{y_i}||\,||\boldsymbol{x}_i||\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_{j=1}^{n} e^{||\boldsymbol{W}_j||\,||\boldsymbol{x}_i||\cos(\theta_{j,i}) + b_j}}, \tag{5.9}$$

where $\theta_{j,i} (0 \leq \theta_{j,i} \leq \pi)$ is the angle between vector $\boldsymbol{W}_j$ and $\boldsymbol{x}_i$. Experimental results showed, that weight normalization, i.e. $||\boldsymbol{W}_i|| = 1$ and $b_i = 0$ slightly improves results, because the prediction is then dependent only on the angle between the feature vector and weight vector, which helps the network with its training. Therefore, modified Softmax loss gets following form:

$$L_{modified} = -\log \frac{e^{||\boldsymbol{x}_i||\cos(\theta_{y_i,i})}}{\sum_{j=1}^{n} e^{||\boldsymbol{x}_i||\cos(\theta_{j,i})}}. \tag{5.10}$$

Unfortunately, features learned by such loss function are still not necessarily discriminative. Since angles are used as the distance metric, it is natural to incorporate angular margin to learned features to enhance the discrimination power.

Assume a learned feature $\boldsymbol{x}$ from class 1, modified Softmax requires $\cos(\theta_1) > \cos(\theta_{others})$ to correctly classify $\boldsymbol{x}$. Using this condition margin m is incorporated, i.e. $\cos(m\theta_1) > \cos(\theta_{others})$. This is making the decision more stringent than previous one:

$$L_{ang} = -\log \frac{e^{||\boldsymbol{x}_i||\cos(m\theta_{y_i,i})}}{e^{||\boldsymbol{x}_i||\cos(m\theta_{y_i,i})} + \sum_{j \neq y_i} e^{||\boldsymbol{x}_i||\cos(\theta_{j,i})}}, \tag{5.11}$$

where $\theta_{y_i,i}$ has to be in the range of $[0, \frac{\pi}{m}]$. To get rid of this restriction, the authors designed a piece-wise monotonic function.

The main advantage of A-Softmax is that it can be trained the same way as classic Softmax. Unfortunately, A-Softmax has problems with convergence leading to training instabilities. To overcome this problem is recommended to incorporate standard Softmax supervision or pretraining. Another disadvantage is the difficulty of A-Softmax implementation.

5.2.2 COCO loss

In 2017, Liu et al. [66][67] address problem of open-set large-scale face recognition by introducing Congenerous cosine (COCO) loss. Same as other losses in this section, COCO loss is optimizing the cosine similarity among data. It inherits the Softmax property to make inter-class features discriminative. Moreover, it shares the idea of the class centroid in metric learning. However, in previous works (see Subsection 5.1.4), the center is a temporal, statistical variable withing one mini-batch during training without any consulting to the network parameter update. This can lead to unstable training or even stopping the training.

On the other hand, COCO loss updates centroids simultaneously during the network training. To be more concrete, let x_i be the feature vector of i-th sample with label y_i. Cosine similarity of two features is then defined as follows:

$$C(x_i, x_j) = \frac{\boldsymbol{x}_i^T \boldsymbol{x}_j}{||\boldsymbol{x}_i||\,||\boldsymbol{x}_j||}. \tag{5.12}$$

Let denote K the total number of classes naive algorithm wants to maximize the following loss:

$$L_{naive} = \sum_{i,j\in\beta} \frac{\delta(y_i, y_j) C(\boldsymbol{x}_i, \boldsymbol{x}_j)}{[1 - \delta(y_i, y_j)] C(\boldsymbol{x}_i, \boldsymbol{x}_j) + \epsilon}, \tag{5.13}$$

where β denotes the mini-batch, $\delta(\cdot, \cdot)$ is the indicator function and ϵ is a trivial number for computational stability. Such design is reasonable in theory but suffers from computational inefficiency and unstable parameter update, therefore unstable convergence.

To address these flaws, COCO loss uses the centroid for each class and thus enforcing features to be learned around these points. The centroid for class k is defined as follows:

$$c_k = \frac{1}{M_k} \sum_{i\in\beta} \delta(y_i, y_j) \boldsymbol{x}_i, \tag{5.14}$$

where M_k is the number of samples that belong to class k within the mini-batch. Incorporating class centroids into traditional Softmax loss:

$$L_{upgraded} = \sum_{i\in\beta} \frac{e^{C(x_i, c_{y_i})}}{\sum_{n\neq y_i} e^{C(x_i, c_n)}}, \tag{5.15}$$

where n is a class index. Such approach measures the distance of one sample against other samples by way of the class centroid instead of direct pairwise comparison. The numerator ensures sample i is close enough to its class center c_{y_i}, and denominator enforces a minimal distance against samples in other classes.

To further improve loss performance, the features and centroids are normalized by l_2 norm and then scale the features by scale factor α before feeding them into the loss layer:

$$\hat{c}_k = \frac{c_k}{||c_k||}, \;\; \hat{x}_i = \frac{\alpha x_i}{||x_i||}. \tag{5.16}$$

So the final loss to minimize will have the following form:

$$L_{COCO} = -\log \frac{e^{\hat{c}_k^T \hat{x}_i}}{\sum_n e^{\hat{c}_n^T \hat{x}_i}}. \tag{5.17}$$

Note that both the features and cluster centroids are trained end-to-end during network parameters update.

The main advantages of COCO loss are its easy implementation, training stability, and very good results for large-scale face recognition. The main disadvantage is its very long training and the necessity to "calculate" new centroids in every training iteration.

5.2.3 Arc loss

Inspired by A-Softmax, Deng et al. [68] presented their loss function - Arc loss. During experiments was observed that the networks learn to respond to the quality of the image by the $L2$-norm of its feature descriptor. To further elaborate, for good quality frontal face images have a high $L2$-norm while blurry faces with extreme pose have low $L2$-norm, see Fig. 5.2.

Figure 5.2: Comparison of $L2$-norms for different facial images. Taken from [69].

This is caused by the fact Softmax loss is weak in modeling difficult or extreme samples, therefore, the loss gets minimized by increasing $L2$-norm of features for easy samples and ignoring hard ones. That means, the trained network learns to respond to the quality of the image by the $L2$-norm of its feature descriptor, which is a very unwanted element. Also, the gradient norm may be extremely large when the feature from the low-quality image is very small, which potentially increases the risk of gradient explosion. So it is logical to add $L2$-constraint to the feature descriptor and restrict features to lie on a hypersphere of a fixed radius s. Such $L2$-normalization can be easily implemented and significantly boost the performance of the face verification.

Wang et al. [70][71][72] modified A-Softmax by usage additive cosine margin m instead of original multiplicative one. Cosine loss is then defined as follows:

$$L_{cos} = -\log \frac{e^{s(\cos(\theta_{y_i})-m)}}{e^{s(\cos(\theta_{y_i})-m)} + \sum_{j \neq y_i} e^{s \cos \theta_j}}. \tag{5.18}$$

Compare to A-Softmax, Cosine loss has three advantages: (1) It's much easier to implement; (2) It provides much better convergence without the necessity of using additional Softmax supervision; (3) It reaches better results overall.

Although an improvement, which Cosine loss provides, the angular margin has a more clear geometric interpretation compared to cosine margin, and the margin in angular space corresponds to the arc distance on hypersphere manifold, see Fig. 5.3.

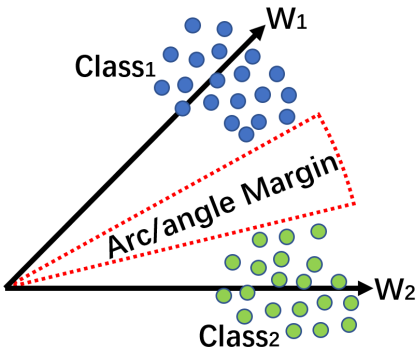


Figure 5.3: Angular margin. Taken from [68].

Arc loss is defined as follows:

$$L_{arc} = -\log \frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j \neq y_i} e^{s\cos\theta_j}}. \tag{5.19}$$

Apart from the best geometric interpretation, Arc loss also reaches the best results, and its implementation is straightforward and intuitive. For better intuition of decision boundaries difference, see Table 5.1

Table 5.1: Decision boundaries of different loss functions for two classes.

Loss Functions	**Decision Boundaries**
Softmax	$(\boldsymbol{W}_1 - \boldsymbol{W}_2)\boldsymbol{x} + b_1 - b_2 = 0$
W-Norm Softmax	$\|\|\boldsymbol{x}\|\|(\cos\theta_1 - \theta_2) = 0$
Angular Softmax	$\|\|\boldsymbol{x}\|\|(\cos m\theta_1 - \cos\theta_2) = 0$
F-Norm A-Softmax	$s(\cos m\theta_1 - \cos\theta_2) = 0$
Cosine loss	$s(\cos\theta_1 - m - \cos\theta_2) = 0$
Arc loss	$s(\cos(\theta_1 + m) + \cos\theta_2) = 0$

Chapter 6

Generative Adversarial Networks

Generative Adversarial Networks (GANs) were introduced by Goodfellow et al. [34] in 2014 and have had a huge success since then. Their popularity stems from the fact, they are first effective generative models based on neural networks (all standard implementation before were discriminative models). The main advantage of generative models is its ability to generate new unseen content. To be more specific, discriminative models care about the relation between sample x and its label y, i.e., during the training process, they are learned to predict correct label y based on the input x (=supervised learning). Meanwhile, generative models care about how to model x from data distribution (=unsupervised learning).

Standard GAN is composed of two feed-forward neural networks - generator G, and discriminator D. The generator's task is creating a new, synthetic sample x_g based on its input, which is n-dimensional noise vector z (sometimes also called latent space). On the input of discriminator are fed real images x_r from the ground-truth dataset or fake images generated by the generator x_g. The discriminator's task is to reveal the fake images, whereas the generator is trying to generate images as real as possible to fool the discriminator, see Figure 6.1.

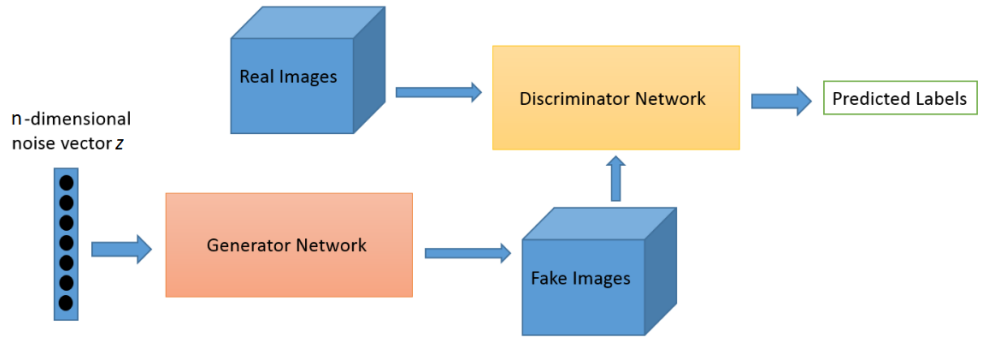


Figure 6.1: Scheme of standard Generative Adversarial Network. Taken from [73].

With such setup, the two networks are trying to beat each other, and, doing so, they are both becoming better and better via the training process. From a game theory point of view, this setting is a minimax two-player game where the equilibrium state corresponds to the situation

where the generator produces data from the exact target distribution as ground-truth data and where the discriminator predicts "real" or "fake" with probability $\frac{1}{2}$ for any sample it receives. Such a state is called the Nash equilibrium.

The expected value function (also often called as Adversarial loss) of the discriminator can be expressed as:

$$V(G, D) = \frac{1}{2}E_{x\sim p_r}[\log D(x)] + \frac{1}{2}E_{z\sim p_z}[\log(1 - D(G(z)))], \tag{6.1}$$

where p_r is real data distribution and p_z is fake data distribution. The goal of the generator is to fool the discriminator, so the generator is naturally trying to minimize this value function. On the other hand, the discriminator is trying to distinguish between real and fake data, therefore, is maximizing this value:

$$\min_G(\max_D E(G, D)). \tag{6.2}$$

Therefore, the best possible discriminator is the one which maximizes:

$$E_{x\sim p_r}[\log D(x)] + E_{z\sim p_z}[\log(1 - D(G(z)))]. \tag{6.3}$$

And the best possible generator is the one which minimizes:

$$E_{z\sim p_z}[\log(1 - D(G(z)))]. \tag{6.4}$$

However, the training of GANs is a challenging problem. There are two main possible unwanted results: (1) lack of convergence; (2) mode collapse.

The problem with convergence mostly stems from an unbalance speed of the training of generator and discriminator. Basically, there are two possible reasons. Firstly, the generator is trained faster and become superior to the discriminator. In this situation, the generator produces perfect images (from the discriminator point of view), and the discriminator is, therefore, unable to distinguish between fake and real ones. This leads to the stopping of the training. Secondly, the discriminator is trained much faster and become flawless in revealing fakes. In such a situation, the generator does not know how to improve itself, because, despite all of its efforts, it can not fool the discriminator. This again leads to the stopping of the training. There exist different strategies to prevent these problems. The discriminator is usually trained faster than the generator, therefore, a very common simple heuristic strategy is to perform two weight updates of the generator for every discriminator update. Another very popular strategy includes tracking of loss of both GAN parts and stops training of some of them when its loss becomes much lower than the loss of the second part.

During the training, there does not exist any lever to force the generator to generate different outputs. This can lead to mode collapse when the generator learns to generate only a few different outputs, but entirely real, and completely ignore the others. The discriminator is then unable to distinguish between real and fake ones. Let's mention an example of a numeral generation. The generator may decide during the training, that it is much easier for it to generate only numerals *1*, and *9* and totally ignore the other possibilities. Such a system can produce very realistic images of numerals *1* and *9*, however, it is unable to produce anything else.

There are two basic approaches to prevent the mode collapse. Firstly, it is provided some information about a wanted sample to the generator and also to the discriminator (for example, information for the generator which numeral should be generated and information for the

discriminator which numeral should be on the image). This approach is called Conditional GAN, for more information, see Section 6.2.

Secondly, it is possible to employ Wasserstein distance instead of standard Adversarial loss (Eq. 6.1). The Wasserstein distance (or the EM distance) is the minimum cost of transporting mass, which can also be used for the distance calculation between two different distributions. In our case, between real data distribution p_r and the distribution of generated data p_g:

$$W(p_r, p_g) = \inf_{\gamma \in \Pi(p_r, p_g)} E_{(x,y)\sim\gamma}[||x - y||], \tag{6.5}$$

where $\Pi(p_r, p_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively p_r and p_g. In other words, Π is the set of all possible transport plans γ from point x to point y as to make x follows the same probability distribution of y. Using Kantorovich-Rubenstein duality is Wasserstein distance reformulated as follows:

$$W(p_r, p_g) = \frac{1}{K} \sup_{||f||_L \leq K} E_{x\sim p_r}[f(x)] - E_{x\sim p_g}[f(x)], \tag{6.6}$$

where f is a 1-Lipschitz function following this constraint:

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|. \tag{6.7}$$

Such a function can be parameterized by a set of weights w. In the modified GAN utilizing Wasserstein distance (called Wasserstein GAN [74]) is the discriminator model (called critic) used to learn w to find optimal f_w. The loss function is then defined as follows:

$$L(p_r, p_g) = W(p_r, p_g) = \max_{w \in \boldsymbol{W}} E_{x\sim p_r}[f_w(x)] - E_{z\sim p_z}[f_w(G(z))]. \tag{6.8}$$

The critic is not directly distinguishing fake and real samples, it is trained to compute Wasserstein distance instead. However, the smaller the distance gets, the generator's output distribution is closer to the real data distribution.

The major problem is to maintain the 1-Lipschitz continuity of f_w during the training. To enforce the constraint, Wasserstein GAN (WGAN) is using clipping of the weights w to restrict the maximum weight value, resulting in a compactness parameter space $\boldsymbol{W}$ and thus f_w obtains its lower and upper bounds to preserve Lipschitz continuity. However, even authors admitted that weight clipping is a terrible way to enforce Lipschitz constraint because it leads to slow convergence and sometimes to vanishing gradients. Gulrajani et al. [75] replaced weight clipping with a gradient penalty, which leads to significant improvements.

6.1 VAEGAN

VAEGAN [76] [77] is an approach combining standard Generative Adversarial network and Variational Autoencoder. The main disadvantage of Autoencoders is the necessity of the design of the appropriate distance metric (used for the calculation of the input-output difference), which can be really dependent on the task. The most popular metric is l_2 distance, which provides reasonable results for a big variety of tasks. However, Autoencoders using l_2 produce blurry images. This directly stems from the nature of l_2 distance when it is much easier for the network to produce only "average features" (and more or less meet required

criteria) and therefore stuck in some local minima during the training process. The content of such generated images is usually very real, nevertheless, for a human, it is easily distinguishable from the real images.

To overcome this flaw, the VAEGAN approach is utilizing another neural network (discriminator) instead of engineered designed metric distance, see Fig. 6.2. Same as in a plain GAN, the discriminator in VAEGAN can reveal fake images, and therefore, it forces generator to improve its output.

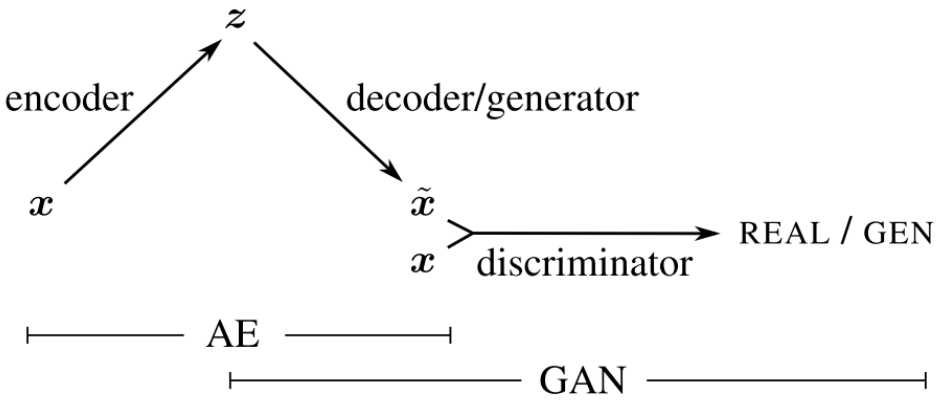


Figure 6.2: Overview of VAEGAN approach - combining a VAE with a GAN. Taken from [76].

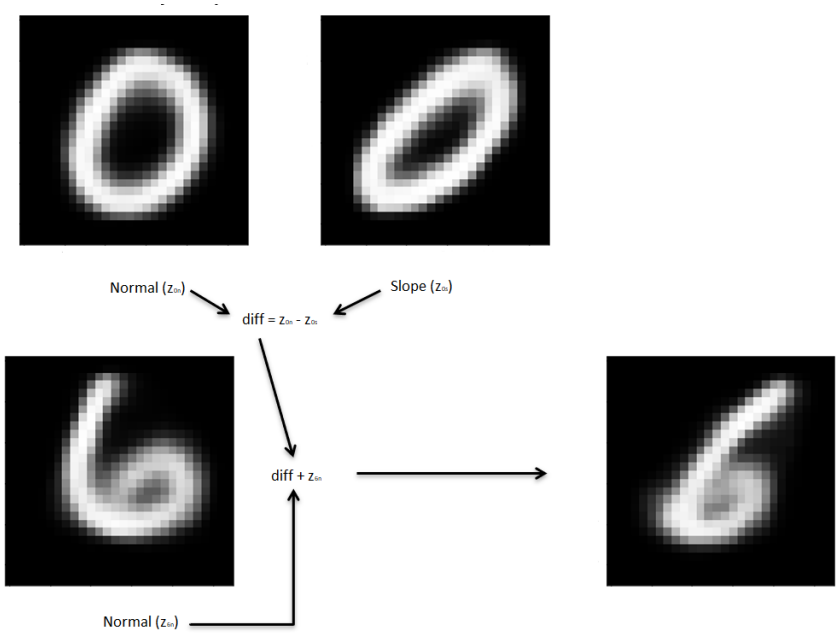


Figure 6.3: An example of the latent space arithmetic.

From the GAN point of view, the encoder addition allows the system to encode real images to the latent space. The latent space codes z can be then used as discriminative features for the encoded real images. Moreover, despite the unsupervised manner of the training and

without any additional constraints on the latent space, in the latent space works latent vector arithmetic phenomenon. Encoding and decoding is a highly non-linear process, however, some linearity is preserved. Experiments show, it is easier to learn to map similar inputs close to each other, whereas different ones far from each other. Thanks to this phenomenon, a good-trained encoder naturally holds big clustering ability across image attributes despite the lack of any additional information about them.

For example, let's assume, VAEGAN is trained to generate images of numerals. By subtracting z_{0s}, which is latent vector belonging to an image of "slope" zero, from z_{0n}, which is latent vector belonging to an image of "normal" zero, and by adding this subtract to z_{6n}, which is latent vector belonging to an image of "normal" six, it is get z_{6s}, which is latent vector belonging to "slope" six, see Figure 6.3.

6.2 Conditional GAN

Mode collapse is a real problem for traditional GAN approaches. Condi-GAN [78] addresses this problem by providing additional information about the class label y to both generator, and discriminator, see Fig. 6.4. From the generator's perspective, these labels act as an extension to the latent space z, whereas it is usual to concatenate the label vector with z vector directly.

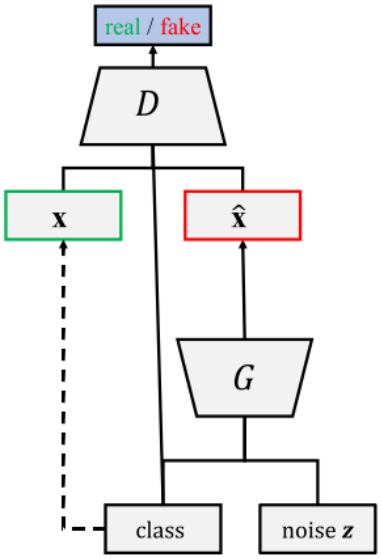


Figure 6.4: Conditional GAN. Taken from [79].

Results of the two-player minimax game are now not only dependent on the degree of the realness of the generated images, but also on the fact if the images look like other images from the corresponding class. During the training process, for the discriminator, it should be very easy to reveal the fake image when it looks like perfectly real numeral 0, however, the provided label says it should be numeral 6. The value function of the Conditional GAN is

defined as follows:

$$V(G,D) = \frac{1}{2}E_{x\sim p_r}[\log D(x|y)] + \frac{1}{2}E_{z\sim p_z}[\log(1 - D(G(z|y)))]. \tag{6.9}$$

Conditional GAN successfully fights against mode collapse, but for the cost of the necessity of data labels.

6.3 DR-GAN

Pose-invariant FR is still a real challenge even for modern FR approaches. Obtaining face identity representation completely disentangled from any external conditions is the holy grail of FR algorithms. Disentangled Representation GAN (DR-GAN) [79] is an approach effectively providing representation disentangled from pose variations. Moreover, thanks to the encoder-decoder structure, the generator can frontalize or rotate a face to an almost arbitrary pose.

Beside standard GAN structure, DR-GAN is utilizing encoder, which gives it an option to obtain representation $f(x)$ of a real image during the testing phase. Moreover, latent space z is enriched by pose information c. With the addition of the two independent parameters, the generator is forced to generate $\hat{x}$ independently on the original pose of x. Also, the encoder is trying to ease the generator's job by encoding x independently on its pose because they are both penalized when discriminator reveals a fake image, therefore, it is important for them to work together against the discriminator.

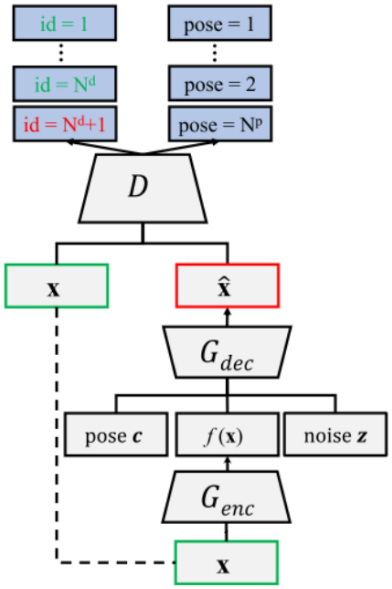


Figure 6.5: DR-GAN. Taken from [79].

DR-GAN's discriminator, instead only revealing fake images, has two main tasks. Firstly, it is classifying its input to $d + 1$ classes, where d is the number of identities in the training set, and the last class is assigned to fake images. Secondly, it classifies input's face pose into p pose classes. The generator is then penalized not only for generating an unrealistic facial

image, but also for generating face from incorrect class or in the wrong pose, see Fig. 6.5. DR-GAN's error is then defined as follows:

$$\max_D V_D(D,G) = E_{(x,y)\sim p_d}[\log D^d_{y^d}(x) + \log D^p_{y^p}(x)] + E_{(x,y)\sim p_d, z\sim p_z, c\sim p_c}[\log(D^d_{N^d+1}(G(x,c,z)))], \tag{6.10}$$

$$\max_D V_G(D,G) = E_{(x,y)\sim p_d, z\sim p_z, c\sim p_c}[\log(D^d_{y^d}(G(x,c,z))) + \log(D^p_{y^t}(G(x,c,z)))]. \tag{6.11}$$

The authors also expand plain DR-GAN with the possibility of extracting feature representation from an arbitrary number of images of a single person simultaneously.

6.4 FaceID-GAN

Typical GAN approaches are formulated as a two-player game, where a discriminator distinguishes real face images from synthesized ones, while a generator reduces its discriminativeness by synthesizing more and more realistic faces. Unlike these typical approaches, FaceID-GAN [80] threats a face classifier as the third player, competing with the generator by distinguishing the identities of real and synthesized faces. A stationary training point is then reached when the generator produces high-quality images with a well-preserved identity. Moreover, the face classifier is used to extract identity features from both input (real) and output (synthesized) images of the generator, which ensures that the real and synthesized images are projected into the same feature space and alleviate the training difficulty of typical GAN.

FaceID-GAN is utilizing both the discriminator D and the additional identity classifier C. Given d different identities, the classifier C classifies the input into $2d$ classes, whereas there exist two classes for each identity, one for real images and one for the fakes. This pushes real and synthesized domains as close to each other as possible. For even better results, there is incorporated shape estimator P to project facial images into a shape feature space, representing pose and expression. The main advantage of the Face-ID GAN is the possibility to employ other additional estimators easily. The final pipeline of the Face-ID GAN approach can be seen in Fig. 6.6.

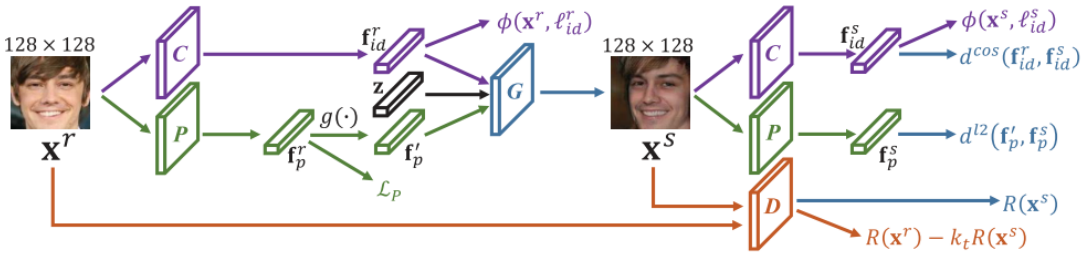


Figure 6.6: FaceID-GAN pipeline. Taken from [80].

FaceID-GAN outperforms the other state-of-the-art approaches in identity preservation, and also the capability to control pose and expression. It also provides images of very good quality, but PG-GAN outperforms Face-ID GAN in this domain.

6.5 PG-GAN

The key idea of Progressive Growing GAN (PG-GAN) [81] is to grow both the generator and the discriminator progressively. Training starts from a low resolution, and by adding new layers, model increasingly fine details of its output (in the paper authors started with the resolution of 4×4 pixels and ended with the resolution of 1024×1024 pixels). This both speeds the training up and greatly stabilizes it, allowing PG-GAN to produce images of state-of-the-art quality.

Later in 2018, Shaobo Guan proposed Transparent Latent-space GAN (TL-GAN) [82] TL-GAN is based on original PG-GAN, which is modified with the ability to control the generation process. It offers users the ability to tune one or multiple features using a single network gradually. To achieve this goal, it is necessary to understand latent space representation and find features axes in it. Separate feature extractor is therefore trained using which is established link between latent space z and features $\boldsymbol{y}$, see Fig. 6.7. The main advantage of this approach is that to add a new feature is not necessary to re-train the GAN model, but only to establish the link between latent space and the new feature.

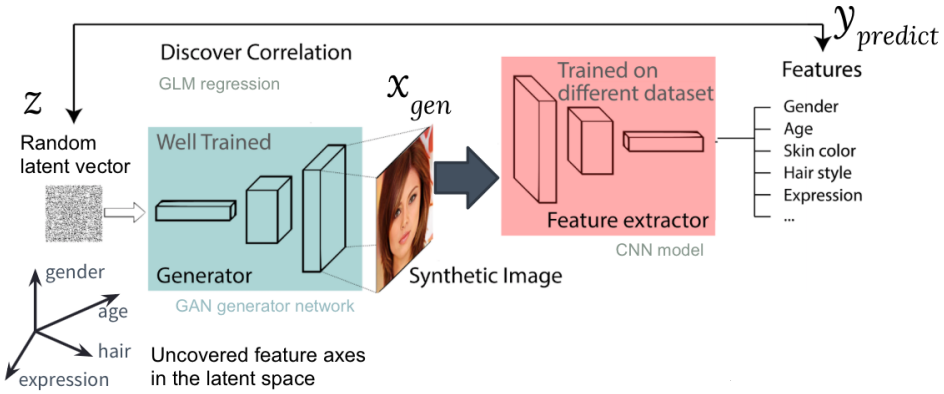


Figure 6.7: TL-GAN structure. Taken from [82].

6.6 Pix2pix

Image-to-image translation (image modality change) is a very popular task nowadays, because of the variety of its real-world application. Probably the best existing supervised method is Pix2pix [83], which based conditional adversarial network. Its main advantage is efficiency across many different generic tasks.

Pix2pix has a traditional pipeline - generator (with two parts - encoder and decoder), and discriminator. The loss function of GAN can be expressed as:

$$L_{GAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z))]. \tag{6.12}$$

Moreover, the authors found out it is beneficial to mix the GAN objective with some traditional loss. This motivates the generator to produce images corresponding with original inputs. Testing also showed that L_1 distance is more suitable than L_2 distance because it encourages less blurring:

$$L_1(G) = E_{x,y,z}[||y - G(x, z)||_1]. \tag{6.13}$$

Final loss function is then defined as follows:

$$L = \min_G \max_D L_{GAN}(G, D) + \lambda L_1(G), \tag{6.14}$$

where λ is proportionality constant.

Inspired by U-Net [84], the generator has encoder-decoder structure with additional skip connections between each layer i in the encoder and layer n-i in the decoder, where n is the total number of layers. The skip connections are implemented as channel concatenation of all channels at layer i with those at layer n-i. Such skip connections allow better propagation of information through the network, see Fig. 6.8.

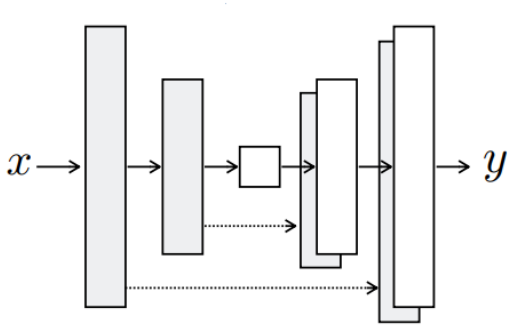


Figure 6.8: U-Net. Taken from [83].

To further overcome blurry results on image generation, instead of the standard discriminator, the authors utilized Markovian discriminator. Markovian discriminator penalized structure at the scale of patches rather than penalizing an image as a whole. In other words, Markovian discriminator tries to classify if each $N \times N$ patch in an image is real or fake. The discriminator is run convolutionally across the image, averaging all responses to provide the ultimate output of D. Markovian discriminator not only fights against blurriness but also has fewer parameters and can be applied to arbitrary large images.

6.7 UNIT/MUNIT

The main disadvantage of the Pix2pix method is the necessity of training data in the form of image pairs. To overcome this problem, UNIT [85] aims at unsupervised learning a joint distribution of images in different domains using images from the marginal distributions in individual domains assuming shared-latent space. By shared-latent space assumption is meant that a pair of corresponding images (x_1, x_2) from two different domains X_1, X_2 can be mapped to the same latent code z in a shared-latent space Z. Such an assumption also implies cycle-consistency.

UNIT framework is based on one VAEGAN for each domain, i.e. it consist of two domain encoders E_1 and E_2, two domain generators G_1 and G_2, and two domain discriminators D_1 and D_2. To enforce the shared-latent space, the last few layers (high-level layers) of encoders, and the first few layers (high-level layers again) of generators are sharing their weights, see Fig. 6.9. During the training, for image x_1 there are performed two main operation - image reconstruction $\hat{x_1}^{1\to1} = G_1(E_1(x_1))$, and image translation $\hat{x_1}^{1\to2} = G_2(E_1(x_1))$.

Assuming image reconstruction, image translation, and cycle-consistency, the final loss is defined as follows:

$$L = \min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} L_{VAE_1}(E_1,G_1) + L_{GAN_1}(E_1,G_1,D_1) + L_{CC_1}(E_1,G_1,E_2,G_2) + L_{VAE_2}(E_2,G_2) + L_{GAN_2}(E_2,G_2,D_2) + L_{CC_2}(E_2,G_2,E_1,G_1). \tag{6.15}$$

L_{VAE_1} is defined as follows:

$$L_{VAE_1} = \lambda_1 \mathrm{KL}(q_1(z_1|x_1)||p_\nu(z)) - \lambda_2 E_{z_1\sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)], \tag{6.16}$$

where λ_1 and λ_2 are proportional constants and $p_\nu(z) = \mathcal{N}(z|0,I)$ is a zero mean Gaussian.

L_{GAN_1} is given:

$$L_{GAN_1}(E_1,G_1,D_1) = \lambda_0 E_{x_1\sim P_{x_1}}[\log D_1(x_1)] + \lambda_0 E_{z_2\sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))], \tag{6.17}$$

where λ_0 is proportional constant again.

Finally, L_{CC_1} is defined as follows:

$$L_{CC_1}(E_1,G_1,E_2,G_2) = \lambda_3 \mathrm{KL}(q_1(z_1|x_1)||p_\nu(z)) + \lambda_3 \mathrm{KL}(q_2(z_2|x_1^{1\to 2})||p_\nu(z)) - \lambda_4 E_{z_2\sim q_2(z_2|x_1^{1\to 2})}[\log p_{G_1}(x_1|z_2)], \tag{6.18}$$

where λ_3 and λ_4 are proportional constants once more. The loss functions $L_{VAE_2}, L_{GAN_2},$ andL_{CC_2} has the same form with appropriate indexes.

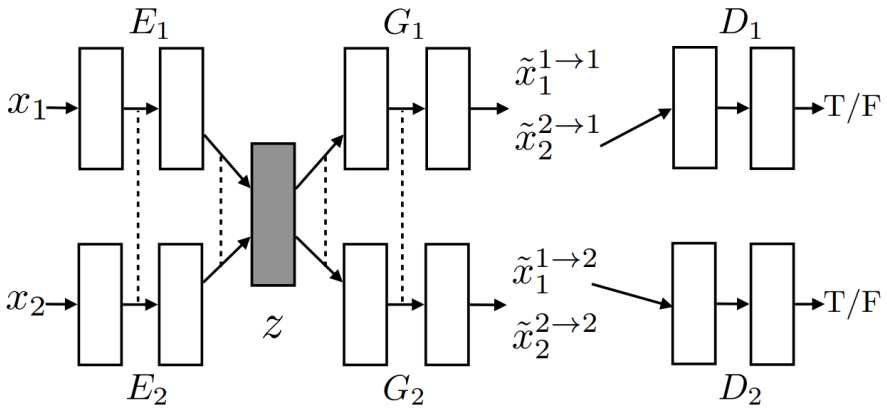


Figure 6.9: UNIT framework. Taken from [85].

UNIT effectively translate images between two domains, however, it has its limitations. First, the translation model is unimodal due to the Gaussian latent space assumption, and second, training is sometimes unstable.

MUNIT [86] framework scraps deterministic assumption about one-to-one mapping, and instead, it assumes that the image representation can be decomposed into a content code that is domain-invariant, and a style code that captures domain-specific properties. To translate an image to another domain, it recombines its content code with a random style code sampled from the style space of the target domain. In other words, instead of a fully-shared latent space, MUNIT assumes that only part of the latent space (domain-invariant) is shared across domains, whereas the other part is domain-specific.

The MUNIT framework consists of an encoder, a decoder, and a discriminator for each domain. The model is trained with adversarial losses that ensure the translated images to be indistinguishable from real images in the target domain, as well as bidirectional reconstruction losses that reconstruct both images and latent codes. Again, MUNIT enables two modes - image reconstruction and cross-domain translation. Overall, MUNIT approach surpasses the quality and diversity of other unsupervised methods and is comparable to the state-of-the-art supervised approaches.

Chapter 7

Single Image-Based Recognition

7.1 Engineered-based methods

Engineered-based methods use engineered-based features for FR and can be classified into two categories: the local feature-based methods and the local appearance-based methods. Initial FR methods mostly used this approach. Local feature-based methods detect the position of local features first and then extract features on these positions, while local appearance-based methods partitions the face image into sub-regions, and based on these, they extract features directly.

7.1.1 Local feature-based methods

In 1993, Brunelli and Poggio [87] developed a FR system that extracts a vector of 3 geometrical facial features to form a 35-dimensional vector. They reported 90% recognition ration on a dataset of 47 people (4 images per person) while using a Bayes classifier. Despite the very low storage cost of such a system, this approach has two main disadvantages: geometrical features can be hard to be extracted, and primarily geometrical features alone are not entirely sufficient to represent a face because a lot of useful information is irretrievably lost.

Two research directions try to solve these problems of the geometrical features [88]. The first direction focuses on better facial features detection. There are two main aspects of detection - robustness, and accuracy. Intuitively, the larger number of features points is obtained, the tighter semantic correspondence that can be achieved. This direction is subject to a lot of studies [89][90], but despite all the efforts, the problem of the facial feature points detection can be hardly called solved.

The second direction is focused on finding more powerful local representation methods rather than purely geometrical ones (in practice, approaches from both directions are combined to reach the best possible results). Manjunath et al. [91] presented a method based on facial feature points representation with Gabor wavelet decomposition [92]. There are extracted two kinds of information for each feature point - local information S_i and feature information $\boldsymbol{q}_i$. Feature information $\boldsymbol{q}_i$ is a vector defined as follows: $\boldsymbol{q}_i = [\boldsymbol{Q}_{i1}(x_1, y_1, \theta_1), ..., \boldsymbol{Q}_{iN}(x_N, y_N, \theta_N)]$,

where N is $\boldsymbol{q}$'s predefined number of neighbors $\boldsymbol{Q}$ represented by the spatial and angular distance. According to these vectors is constructed a topological graph (two feature points in some spatial range with minimal distance are connected with an edge) and FR is formulated as a graph matching problem. The graph matching is divided into two parts: the similarity between feature points and global topology similarity. Manjunath et al. reported 86% recognition rate on a dataset of 86 people, containing express and pose variation. The main disadvantage of this approach is the impossibility of any graph modifications after its construction.

Another used approach is the Elastic Bunch Graph Matching (EBGM) - graph matching method proposed by Wiskott et al. [93]. As in [91], a topology graph is constructed first with each node attached to one or several specific Gabor wavelets (see Fig. 7.1). The main advantage of these Gabor features is that they are robust against illumination changes, distortion, and scaling. The upgrade of this method compared to [91] is the second step of graph matching (the first step is very similar) - a deformable matching algorithm is employed, which means each node of the graph is allowed to change its scale and position according to the facial appearance variations. Although this method showed big robustness against appearance changes and it was among the best-performing ones in the FERET competition [8] of 1996, it has three serious drawbacks. The first one is its high computational complexity. The second one is a common problem of all feature-based methods - the algorithm uses the information only from feature points positions and discards the rest of the image. The last one is a requirement of the manual placement of the graph for the first 70 faces before the elastic graph matching becomes adequately dependable, however, this drawback was overcome by Campadelli and Lanzarotti [94] by using parametric models. In 2013, Biswas et al. [95] presented a very similar approach, but instead of Gabor features, they described each landmark with SIFT features [96]. They used then the concatenation of these SIFT features of all landmarks as the face representation for FR task.

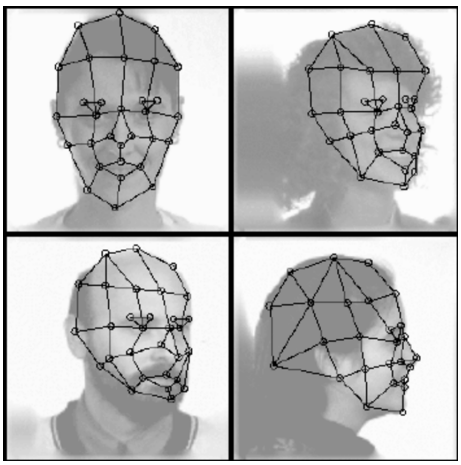


Figure 7.1: Elastic Bunch Graph matched to faces. Image taken from [91].

There exist some other successful variations to this approach that replace Gabor features by a graph matching algorithm [97]. Another variation uses Histograms of Gradients (HoGs) instead [98]. The main advantage of HoGs is their invariance to illumination and shadowing [99]. The last but not the least important variation was made by Kepenekci et al. [100]. The main idea of their approach is to use candidates automatically chosen by Gabor filter for face representation. There is an assumption that since the resulting feature points are

different face to face, the possibility of finding class-specific features is increased. Moreover, Kepenekci's approach also has lower computational complexity than traditional EBGM and is better suitable for handling occlusion. In 2014, Lenc presented SIFT based adapted Kepenekci's method in his doctoral thesis [101].

Gao et al. [102] proposed a geometrical feature descriptor called Directional Corner Point (DPC). A DPC is a feature point with additional information about the connectivity to its neighbors. The experiments show that this configuration is economical and add robust to illumination changes.

The features gaining some popularity in recent years are Local Binary Patterns (LBP) [103][104]. In 2013, Chen et al. [105] extracted multi-scale LBP features from patches around 27 landmarks. After that, these features from all patches were concatenated to become a high-dimensional feature vector. Concatenating the features of all landmarks across the face brings highly nonlinear intrapersonal differences. To overcome this problem, Ding et al. [106] employed Dual-Cross Patterns (DCP) features. DCP of landmarks belonging to the same facial component was concatenated as the component's descriptor.

Li et al. [107] presented feature points detection based on a generic 3D face model. Similar, but more accurate approach choose Yi et al. [108] in 2013. They used a 3D morphable face model with 352 pre-labeled landmarks to which 2D face image is aligned using the weak perspective projection model [109]. After that, there are 352 landmarks projected to the 2D image. Lastly, Gabor features are extracted from the found positions of the landmarks and concatenated into a feature vector.

In summary, local feature-based methods offer relatively high robustness to translation and rotation variations in the input image, low computational complexity, high speed and can deal with one sample problem. Unfortunately, they depend critically upon the accurate feature points detection, which is hardly a solved problem. Another disadvantage is that the implementer of these methods has to make an arbitrary decision about which features should be used. This is not an easy decision because the number of features has to be as low as possible, but simultaneously the features have to be sufficiently informative.

7.1.2 Local appearance-based methods

Local appearance methods divide face images into local regions and extract features from them. Local appearance-based methods generally include four following steps: local region partitions, local feature extraction, feature selection, and classification (see Figure 7.2). It should be noted that not every step is compulsory, some of them can be united into one or omitted (feature selection). More details about these steps are given below.

Local region partition: Firstly, it is necessary to define local regions, which means it should be decided about region shape, region size, and overlapping. The most commonly used region shape is a rectangular window, but elliptical and strip regions are also used. The size can be very diverse, and there is not any general precedent about using overlapping.

Local feature extraction: Once the local regions are defined, local features can be chosen. Each local feature has its advantages and disadvantages (robustness again, different lighting conditions, noise, pose, etc.). Among most widely used features belong gray-value features

[110], HoGs [99], Gabor wavelets [92], and Haar wavelets [111].

Feature selection: Due to the fact that in the previous step is usually extracted the big amount of features, it is essential to select only the most important ones. This is ensured by employing feature selection methods as PCA [112] or LDA [113].

Classification: There are two main approaches - combining classifiers (each classifier is applied to only one feature, and the final decision is made according to major vote, weighted sum, or probabilistic measure) or one classifier trained on all features.

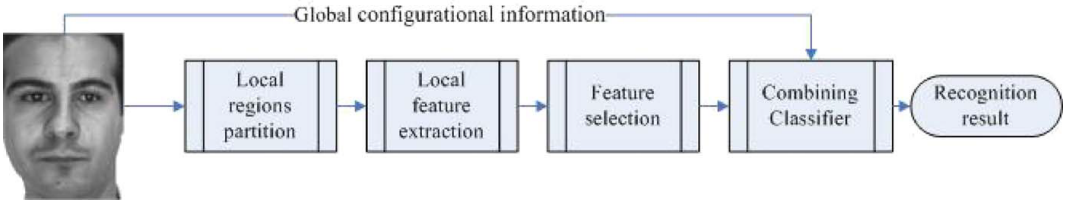


Figure 7.2: General scheme of local appearance based methods. Image taken from [88].

In 2002, Martinez [114] proposed a local probabilistic approach to recognizing imprecisely localized, partially occluded, and expression variant face from a single sample per person. His algorithm divides the face into six overlapping elliptical areas. The same areas of each face generate a face subspace, which is further transformed into eigenspace. In these eigenspaces is during training estimated distribution by means of a Gaussians mixture model (GMM) using the expectation-maximization (EM) algorithm. Then, during identification, the sample image is also divided into six elliptical areas, and probabilistic classification is performed in computed eigenspaces. In 2013, Li et al. [115] proposed Probabilistic Elastic Matching (PEM) model based on a GMM too. Firstly, PEM learns a GMM from the spatial-appearance features [116] of densely sampled image subregions in the training set, i.e., each Gaussian component stands for subregions of the same semantic meaning. A testing image is therefore represented as a bag of spatial-appearance features. In the next step, there is found the subregion whose feature induces the highest probability on each Gaussian component. Concatenating the feature vector of these subregions forms the representation of the face. Since all testing images follow the same procedure, the semantic correspondence is established.

Tan et al. [110] proposed an alternative representation of the face subspace, to the probabilistic approach, with Self-Organizing Maps (SOMs) [33]. After a division of the face into local regions, the SOM is trained for each of this region. As training data for each SOM serves all the same local regions obtained from all training images. After the SOM training is done, each region is mapped to its corresponding best matching unit by the nearest neighbor strategy. The unit's location in the 2D SOM topological space is represented as a location vector. Therefore, all location vectors from the same face can be grouped - this is called SOM-face representation. SOM-face representation has good robustness against noise, and it is compact.

Kanade and Yamada [117], in 2003, proposed a multi-subregion based probabilistic approach for face verification. The face is divided into a set of local regions again and then is constructed a probabilistic model of each subregion of appearance change according to pose change. Experiments were performed on the CMU Multi-PIE dataset [12].

Samaria used hidden Markov models (HMMs) [118] in his FR system [119]. The main feature of the HMM technique is that it characterizes face as a dynamic random process with a set

of parameters. The author divided a face into five overlapping subregions. Each of these subregions is represented as one state of an HMM. A face is then represented by five model's states, which can be modeled by a multivariate Gaussian distribution. Therefore FR is done by output probabilities of observed image calculation.

Arashloo and Kittler [120] presented a method based on Markov Random Field (MRF) used to found a correspondence of subregions between two images. The subregion has in proposed method dynamic size and adaptable shape. In the first step, a face image is densely divided into these subregions, and each of them is represented as a node of the MRF model. Labels are represented by the 2D displacement vectors. The goal of the optimization is to find the assignment of labels with minimum energy (the energy is measured from the gradient), while it is necessary to take both translation and projective distortion into account.

Pentland et al. [121] based their research on relevant studies in psychophysics and neuroscience, which revealed that different parts of the human face are differently important for FR, for example, eyes and mouth are much more important than the nose for remembering faces. Their algorithm uses four masks to obtain regions of eyes, mouth, nose, and whole face. Local features describe these regions, and these features are then projected into eigenspace [4]. The obtained eigenfeatures are used for FR. The results of this method confirmed the fact about the different importance of different face regions. Newer work that adopts similar ideas [122].

Ahonen et al. [123] proposed a method using three different levels of locality: pixel-level, regional level, and holistic level. The first two levels are represented by LBP features extracted from appropriate regions. The holistic level is then represented by concatenating the regional LBP features. The classification algorithm uses the nearest neighbor classifier with Chi-square distance as a dissimilarity measure.

In 2013, Simonyan et al. [124] proposed a method based on Fisher vectors and SIFT features. The work makes two main contributions: (1) It shows that state-of-the-art verification results can be achieved by applying the Fisher vector on densely sampled SIFT features; (2) It shows that compact descriptors can be learned from Fisher vector using discriminative metric learning. Firstly, the method detects a face in the image using the Viola-Jones detector. In the second step, rather than sampling locations and scales sparsely by running a face landmark detector, the method extracts SIFT features densely in scale and space. The process is repeated at five scales, which results in about 26 thousand 128-D descriptors per face. The dimensionality is reduced to 64-D using PCA. The nonlinear Fisher vector encoding is then used to aggregate these descriptors. This is done by fitting a GMM model to the features and then encoding the derivatives of the log-likelihood of the model with respect to its parameters (for example, mean and variances). To preserve spatial information of the features, dense features are augmented with their spatial coordinates. This whole process results in the Fisher vector with dimensionality 67584. To improve the performance of the method in the FR step, the discriminative dimensionality reduction is applied. The step-by-step overview of this method can be seen in Figure 7.3.

The method was tested on Labeled Faces in the Wild dataset (LFW) [10] and achieved the classification accuracy of $93.03\% \pm 1.05$, which was state-of-the-art result in 2013. The authors presume that by using multi-feature image representation, the results can be further improved. Moreover, they show that Fisher vectors, coupled with discriminative dimensionality reduction, can automatically capture the most discriminative parts of the face.

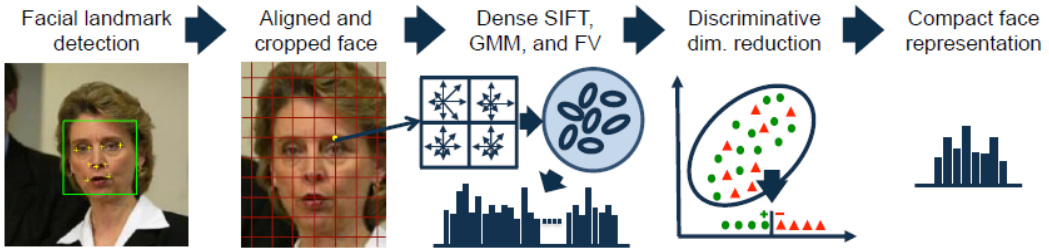


Figure 7.3: Overview of the method from article [124].

In conclusion, the main advantage of the local appearance-based methods is that they usually have high robustness to occlusion. Moreover, with the right selection of extracted features, they can be robust, for example, to different lighting conditions (LBP). However, it is still not clear which features should be used and which are more appropriate for a given task. Another problem can be a selection of suitable regions. Adaptive methods of local region selection can partially address this problem.

7.2 Learn-based methods

Learn-based methods classify a face using global representations (the whole face image) rather than local features. The two main advantages of these methods are: 1) They don't lose any information in the image by extracting features only on limited regions or points of interest; 2) There is no need to manually designed features, which should be extracted, therefore they are automatically extracted by statistical or machine-learning (ML) models. Moreover, this approach can capture more global aspects of the human face than local methods. Learn-based approaches can be divided into two following groups: statistical and AI approaches.

7.2.1 Statistical methods

The most straightforward representation of a face image is a 2D array of intensity values. First, very naive approaches [125] tried directly compare intensity images. On the one hand, this approach was easily implemented, on the other hand, it could work only under limited circumstances (constant illumination, scale, pose, etc.), it was very sensitive to intrapersonal differences, and it was computationally very expensive, because of huge dimensionality. To overcome the curse of dimensionality, it can be employed statistical methods for dimensionality reduction (PCA, LDA, etc.).

Sirovich and Kirby [112] were the first to utilize PCA to compact face representation. They showed that any face image can be effectively represented in an eigenpictures coordinate space and that any face can be approximately reconstructed by using a small set of eigenpictures and the corresponding coefficients (projections).

Based on this approach [112], Turn and Pentland [4] employed eigenfaces projections as a classification features for FR, see Figure 7.4. For an unknown face, their FR system firstly found optimal projection into eigenfaces space and then, based on the obtained coefficients, perform classification. The test of the method on the dataset containing 2500 images of 16

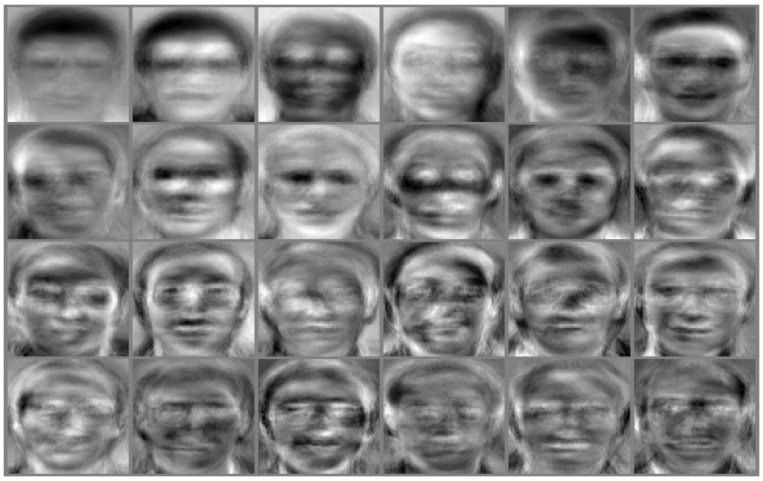


Figure 7.4: Example of the trained eigenfaces corresponding to the first 25 eigenvectors.

different people revealed relatively high robustness to different lighting conditions (96% recognition rate). Unfortunately, the test also shows the high sensitivity of the method to pose and scale changes (85% and 64% recognition rate, respectively). To overcome these difficulties, this system was extended in several ways in the already above-mentioned method [121].

PCA method appears to work quite well when only a single image of each individual is available, but according to Belhumeur et al. [126], when multiple images per person are presented, PCA retains unwanted variations due to lighting conditions and facial expressions. These variations can be in eigenfaces subspace bigger than interpersonal differences. Therefore, they proposed a method, which utilizes Fisher's linear discriminant analysis, and called it Fisherfaces. Opposed to PCA, which is an unsupervised method and maximizes the variance in variables, LDA is a supervised method and maximizes the ratio of between-class scatter to within-class scatter. This method was intensively tested, and in [127] was shown that for the small training dataset, PCA could outperform LDA and also that PCA is less sensitive to different training sets. From that point, numerous variations and extension to the standard eigenfaces and Fisherfaces have been presented.

In 2006, Prince et al. [128] proposed an approach based on Tied Factor Analysis (TFA). TFA assumes that there exists an ideal identity subspace, where each identity is represented by one multidimensional variable $\boldsymbol{h}_i$, considering different poses. Given a face image in jth pose, x_{ij}^k, where k is a number of sample for given ith subject, is generated by the pose-specific linear transformation as follows:

$$x_{ij}^k = \boldsymbol{W}_j \boldsymbol{h}_i + \mu_j + \epsilon_{ij}^k, \tag{7.1}$$

where $\boldsymbol{W}_j$ stands for model parameters, μ_j is offset, and ϵ_{ij}^k is Gaussian noise $\epsilon_{ij}^k \sim G(0, \Sigma_j)$. $\boldsymbol{W}_j$, μ_j, and Σ_j are estimated from the training data by using the EM algorithm. The method was applied to both identification and verification tasks. Cai et al. [129] presented the Regularized Latent Least Square Regression (RLLSR) method, which is based on the same assumption as TFA, however, RLLSR reformulates Eq. 7.1 in the least square regression manner. Moreover, authors incorporated some prior knowledge as two regularization terms into the least square: firstly, the transformation for nearby poses should not differ too much, and secondly, the distribution of the latent ideal objects should preserve the geometric structure of the observed image space.

In 2007, Prince et al. [130] proposed another algorithm designed to handle illumination and pose changes based on Probabilistic LDA (PLDA) this time. This work presented two main findings: (1) inference is more powerful in PLDA than LDA because there can be used more sophisticated noise model; (2) the probabilistic approach allows the development of the non-linear extension. Given a training dataset consists of J images each of I individuals, the model is defined as follows:

$$x_{ij} = \mu + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{ij} + \epsilon_{ij}, \tag{7.2}$$

where the μ represents the overall mean of the training dataset, the matrix $\boldsymbol{F}$ contains a basis for the between-individual subspace, $\boldsymbol{h}_i$ represent position in this subspace, or one can say that represents the identity of individual i, the matrix $\boldsymbol{G}$ contains a basis for the within-individual subspace, $\boldsymbol{w}_{ij}$ represents position in that subspace, and ϵ_{ij} is residual Gaussian noise $\epsilon_{ij} \sim G(0, \Sigma_j)$, see Figure 7.5. This model can be intuitively divided into two parts: the first part which depends only on the identity of the person, but not the particular image ($\mu + \boldsymbol{F}\boldsymbol{h}_i$) and the second part which is different for every image of the individual, i.e. it models pose and illumination changes ($\boldsymbol{G}\boldsymbol{w}_{ij} + \epsilon_{ij}$). During verification, it is considered a likelihood that two faces were generated from the same $\boldsymbol{h}_i$, i.e., they are the same identity. Training of the model is ensured by the EM algorithm. Authors, because of the non-linear nature of the FR problem, also tried to use Mixtures of PLDAs and Tied PLDAs with satisfying results. The whole algorithm was tested on the XM2VTS database.

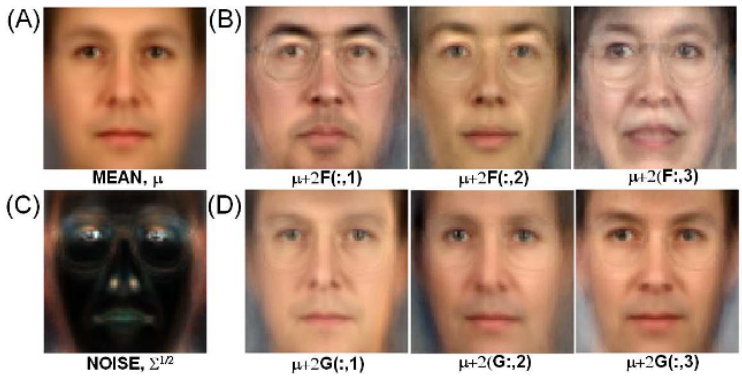


Figure 7.5: Components of PLDA model. (A) Mean face; (B) Three direction in between-individual subspace; (C) noise covariance; (D) Three directions in within-individual subspace. Taken from [130].

Linear approaches (Eigenfaces, Fisherfaces) assume the existence of an optimal projection that projects the face images to distinct non-overlapping regions in the reduced subspace. However, in reality, this assumption may not always be true (images of different persons can be projected into the same region in the projection subspace), because it was proved, that pose-varied face images are distributed on a highly nonlinear manifold. Moghaddam et al. [131] presented an approach based on difference images. The difference image is defined as the signed arithmetic difference of the intensity values of corresponding pixels. Moreover, it is assumed that both intrapersonal and extrapersonal classes are normally distributed within the space of all possible difference images. Then, given the difference image between images I_1 and I_2, the probability that the images are two images of the same person is defined as follows:

$$P(\Omega_I|d(I_1,I_2)) = \frac{P(d(I_1,I_2)|\Omega_I)P(\Omega_I)}{P(d(I_1,I_2)|P(\Omega_I)P(\Omega_I) + P(d(I_1,I_2)|\Omega_E)P(\Omega_E)}, \tag{7.3}$$

where Ω_I is intrapersonal class, and Ω_E is extrapersonal class. It can be seen that the algorithm has to solve the binary classification problem using MAP rule.

Another disadvantage of the approaches based on the linear method of dimensionality reduction as PCA or LDA is that these methods can't discover any connections between images in a nonlinear manifold. Especially PCA extracts only low-dimensional representation, therefore it is used only first and second-tier statistics. This drawback can be removed by using nonlinear techniques (or by extending linear ones via the kernel technique), that can discover these nonlinear structures, for example Isometric Feature Mapping (ISOMAP) [132], Laplacian Eigenmap [133], Local Linear Embedding [134], Locality Preserving Projection [135], Nearest Manifold Approach [136], Discriminant Manifold Learning [137], Laplacianfaces [138] or Embedded Manifold [139].

Probably the most important nonlinear statistic method, that is used in FR, is Independent Component Analysis (ICA) [140]. ICA is a generalization of PCA, but it aims to find an independent, rather than uncorrelated, image decomposition and representation. Furthermore, ICA provides one main advantage over PCA - it is not only exploiting the covariance matrix, but it is also considering the high-order statistics. Bartlett et al. [141] made testing on FERET dataset under two different architectures of the algorithm: the first one treated images as random variables and pixels as an outcome, the second one treated the other way around or more specifically it treated the pixels as random variables and the images as an outcome. Both approaches outperformed PCA representation in their testing. The best results were reached by combining both classification decisions.

In 2015, Lu et al. [142] proposed a face verification method based on Discriminative Gaussian Process Latent Variable Model, named GaussianFace and for the first time in history their algorithm surpassed the human-level performance on LFW dataset. The authors highlighted two main weaknesses of the most face verification methods: (1) most methods assume that the training and the test data are drawn from the same feature space and follow same distribution; (2) most existing methods require some assumption about the structure of the data to be made. To overcome these weaknesses, they proposed the GaussianFace algorithm, which is a reformulation based on the Gaussian Processes, which is a non-parametric Bayesian kernel method. Therefore, the model can adapt its complexity flexibly into the complex real-world data. However, reformulating Gaussian Processes for large-scale multi-task learning is a non-trivial problem, therefore to simplify calculations, they also proposed a more efficient equivalent form of Kernel Fisher Discriminant Analysis. GaussianFace model can be optimized by the Scaled Conjugate Gradient technique, however, this technique was too slow for large-scale data, so authors had to make some additional adjustments.

GaussianFace model can be applied in two ways: (1) as a binary classifier; (2) as a feature extractor, see Figure 7.6. Each face is first normalized to 150x120 pixels size by an affine transformation based on five landmarks (eyes, nose, mouth corners). When is GaussianFace model used as a binary classifier, the face image is divided into overlapped patches. Then, from each patch is extracted multi-scale LBP feature. Then the similarity vector is regarded as the input data point of the Gaussian model. and its output y is $y = 1$ for same identities and $y = -1$ for different ones. When is GaussianFace used as a feature extractor, method regards the joint feature vector (to enhance the robustness, the flipped form of the joint

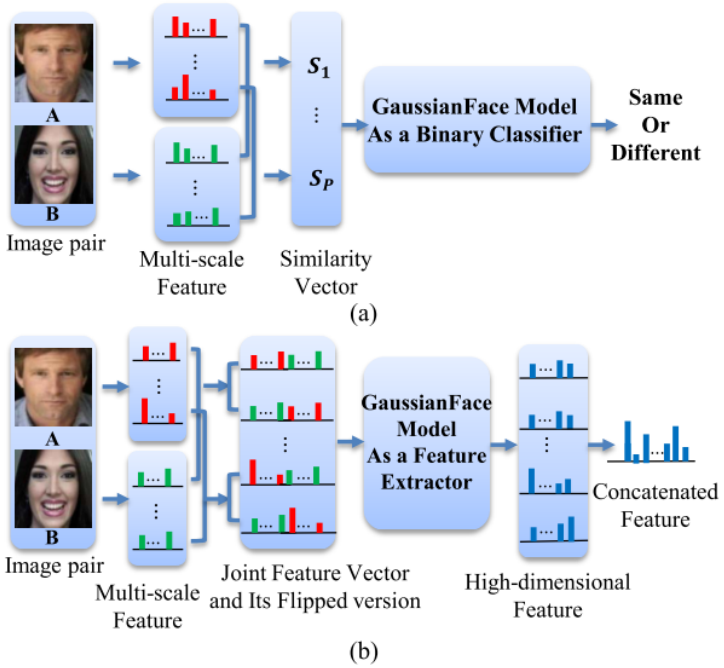


Figure 7.6: Two tested approaches based on GaussianFace model: (a) as a binary classifier; (b) as a feature extractor. Taken from [142].

vector is also included) as the input data point of the GaussianFace model. For an unseen pair of images, it is therefore first computed its joint feature vector for each pair of patches, and it is estimated its latent representation. Then first-order and second-order statistics are computed, and this latent representation is used as the input of the GaussianFace model. By concatenating all of the new high-dimensional features obtained from the GaussianFace model, authors get the final high-dimensional feature vector. Testing showed that this second approach reaches better results. The algorithm was tested on the LFW dataset, reached recognition accuracy 98.52%, surpassed the human performance, and advanced state-of-the-art results significantly. The main disadvantage of this approach is the very long training of the GaussianFace model, even after training speed-up adjustments.

7.2.2 AI methods

AI methods employ machine learning techniques or neural networks to FR. Zhang et al. [143] proposed an approach based on LBP features and the AdaBoost learning algorithm [144]. Adaboost learning is applied to extracted LBP features to select the most efficient one. Moreover, Adaboost aims to obtain the similarity function in the form of the linear combination of LBP. It should be noted that the Adaboost algorithm is more often used for face detection tasks. In 2011, Susheel and Tripathi [145] presented another method using Adaboost. In their approach is Adaboost applied with Haar cascade to real-time face detection and fast PCA algorithm to FR.

The most popular and the most promising approaches of recent years are based on deep neural networks. The first solution using NN was introduced in 1989 by Kohonen [2]. Let's

name at least a few other attempts. In 1993, Weng et al. [146] used a hierarchical neural network. In 2005, Eleyan and Demirel [147] used PCA to obtain feature projection vectors, which was then classified by a feed-forward neural network.

Unfortunately, despite the age of this first approach, the real renaissance of neural networks didn't come till 2012, when Krizhevsky et al. published his article [48] proposing their network AlexNet. However, since then, the NN approaches showed their superiority, and nowadays, most of the state-of-the-art results in FR were reached by using method utilizing NN.

In 2014, Taigman et al. [13] proposed an innovative approach, which advances the state-of-the-art significantly, see Fig. 7.7. Their work had two main contributions: (1) development of effective alignment system based on explicit modeling of 3D faces; (2) development of effective deep neural network architecture and learning method. Work utilizes a face alignment system with two stages. In the first stage, there is perform 2D alignment according to 6 fiducial points (eyes, nose, corners of the mouth). Points are extracted by a Support Vector Regressor (SVR) trained to predict point configuration from image descriptor based on LBP histograms. This first stage should solve scale, in-plane rotation, and translation, however, there is still out-of-plane rotation to be solved. To align face undergoing out-of-plane rotation, they used a generic 3D shape model with manually placed 67 anchor points and 67 fiducial points, which are detected on the face image by SVR again. A 3D model is then fitted using the generalized least-squares solution. After that, they can use a piece-wise affine transformation to perform the frontalization of the image. Invisible parts of the original image can be replaced using image blending with symmetrical counterparts.

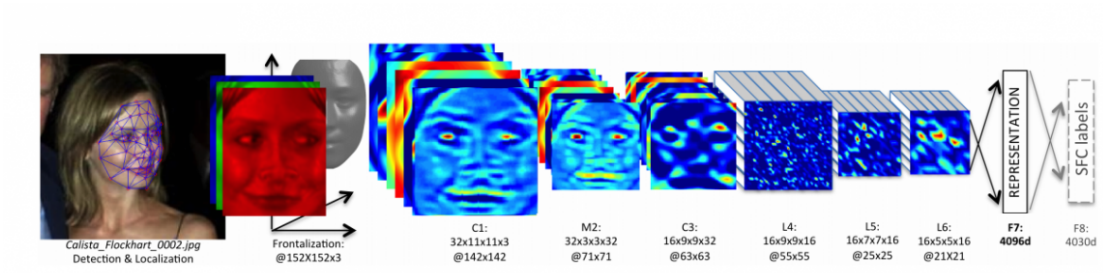


Figure 7.7: Outline of the DeepFace architecture. C = convolutional layer; M = max-pooling layer; L = locally connected layer, F = fully-connected layer. Taken from [13].

Thus preprocessed 3D aligned RGB image with the unified size is an input of CNN. They used an eight-layer deep neural network and train it on a multi-class face recognition task by minimizing cross-entropy loss L_{ce} for each training sample. For updating NN's parameters W they employed standard SGD optimization method. Due to the large training set (the SFC dataset includes 4.4 million labeled faces from 4030 people, 5 % left for testing), they did not observe any significant overfitting during the training phase. As a final stage, they normalized the features to be between zero and one in order to reduce the sensitivity to illumination changes. Thus normalized feature vectors were used to calculate the weighted-χ^2 similarity, which was for two representations f_1 and f_2 defined as follows:

$$\chi^2(f_1, f_2) = \sum_i \omega_i \frac{(f_1[i] - f_2[i])^2}{f_1[i] - f_2[i]}, \tag{7.4}$$

where $\mathbf{W}$ is a weight parameter trained by a linear SVM. The authors also tested Siamese networks with sufficient results. The DeepFace achieved accuracy 97.25% on the LFW dataset,

which reduced the error of the previous state-of-the-art method by more than 25% and almost reached human-level performance (97.53%). Moreover, DeepFace reached a reasonable speed of 0.33 seconds per image on a device with a single-core Intel 2.2 GHz CPU. To conclude, the biggest strength of DeepFace is robust and precise face alignment, which is also confirmed by their testing, when they tried to train CNN without any alignment, and they reached accuracy 87.9% only.

Later in 2014, Sun et al. presented [148] method using CNN to learn effective features for FR. Many previous FR approaches based on CNN use a classification layer, however, Sun et al. claim that this approach has two main disadvantages: indirectness - one has to hope, that the bottleneck representation generalizes well for new faces, and inefficiency - by using bottleneck layer the representation per face is usually very large (and very sparse, as in [13]). One has two options to prevent this result - use some method for dimensionality reduction or train to this purpose one layer of the NN. Moreover, Sun et al. claim, that it is essential to learn such features by using two supervisory signals simultaneously - face identification and face verification signal. They named these features Deep IDentification-verification (DeepID2) features. They argue that usage of only identification signal stems into a relatively weak constraint on two different images of the same person, so they add verification signal, which requires that every image of the same person are in the feature space close to each other. And oppositely, when only the verification signal is used, distinguishing different identities becomes very difficult. The structure of the used CNN can be seen in Figure 7.8. The last layer is the DeepID2 layer, which is fully connected to both the third and the fourth convolutional layers. The reason for this is that authors want to capture multi-scale features, so they utilize the fourth layer, which extracts more global features than the third one. The CNN extracts 160-D DeepID2 vector at this last layer.

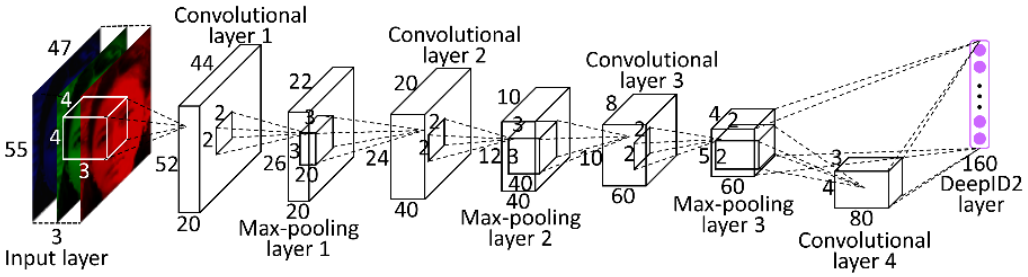


Figure 7.8: Structure of the neural network for DeepID2 extraction. Taken from [148].

The extraction process is defined as function $\boldsymbol{f} = \text{Conv}(x, \boldsymbol{\theta}_c)$, where Conv(.) is feature extraction function, x is the input and $\boldsymbol{\theta}_c$ are parameters learned by CNN. As already stated, learning is done, by two supervisory signals. The face identification signal classifies each image into one of n different identities. This is done by an n-way softmax layer following the DeepID2 layer. The network is trained to minimize the cross-entropy loss, which authors called the identification loss, defined as follows:

$$L_{id}(\boldsymbol{f}, t, \boldsymbol{\theta}_{id}) = -\log \hat{p}_t, \tag{7.5}$$

where $\boldsymbol{f}$ is a DeepID2 vector, t is target class and $\boldsymbol{\theta}_{id}$ are softmax layer parameters. The face verification signal directly regularizes the DeepID2 layer, and two different loss functions

(verification losses) were tested. The first is the L2 loss function and is defined as follows:

$$L_{ve}(\boldsymbol{f}_1, \boldsymbol{f}_2, y_{12}, \boldsymbol{\theta}_{ve}) = \begin{cases} \frac{1}{2}||\boldsymbol{f}_1 - \boldsymbol{f}_2||_2^2 & \text{if } y_{12} = 1, \\ \frac{1}{2}\max(0, m - ||\boldsymbol{f}_1 - \boldsymbol{f}_2||_2)^2 & \text{if } y_{12} = -1, \end{cases} \tag{7.6}$$

where $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are DeepID2 vectors, $y_{12} = 1$, if $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are from the same identity and $y_{12} = -1$ if are from different one, and $\boldsymbol{\theta}_{ve} = m$ is verification function parameter and represents margin between two different identities. The second loss function is cosine similarity loss defined as follows:

$$L_{ve}(\boldsymbol{f}_1, \boldsymbol{f}_2, y_{12}, \boldsymbol{\theta}_{ve}) = \frac{1}{2}(y_{12} - \sigma(wd + b))^2, \tag{7.7}$$

where d is cosine similarity, $\boldsymbol{\theta}_{ve} = \{w.b\}$ are verification function parameters and σ is sigmoid function. The goal is to train the parameters $\boldsymbol{\theta}_c$, while $\boldsymbol{\theta}_{id}$ and $\boldsymbol{\theta}_{ve}$ are the only parameters to back-propagate during the training, whereas $\boldsymbol{\theta}_c$ are the only parameters used during the testing. For parameter update is used SGD algorithm, however, it should be noted that the verification gradient is weighted by a hyperparameter λ. Testing showed best results for $\lambda = 0.05$. Authors used for training CelebFaces+ dataset (which they divided into training and testing set), i.e., approximately 160 thousand images from 8192 identities. Thus learned DeepID2 features are used to the learning of the Joint Bayesian model [149] for face verification. The trained Joint Bayesian model this way over-performs [13] and advances state-of-the-art results.

Sun et al. continued in their research and presented some upgrades of their system in the article [150]. In this work, they not only improved their features (DeepID2+) and reached better results, but the main contribution of this article lay primary in the deepening of understanding of neural activations. The improvement of state-of-the-art was achieved by three improvements: (1) Enlarging feature vector from 128-D to 512-D; (2) enriching the training data by merging CelebFaces+ dataset with WDRef dataset [149]; (3) Enhancing the supervision by connecting a fully-connected layer to each of four convolutional layers, see Figure 7.9.

The second part of the work was focused on an empirical evaluation of the properties of deep neural activations. They discovered three very important properties: sparsity, selectiveness, and robustness. They claim that all these properties are owned by DeepID2+ after large-scale training, without any extra regularization. More details about these properties of neuron activations are given bellow.

- Sparsity - Approximately half of the neurons in the top hidden is active for each image, and each neuron is active for approximately half of the images. Such sparsity makes neurons to have maximum discrimination abilities. Moreover, the authors discovered that for different identities are active different neurons, whereas the same ones have very similar activation patterns. In one of the experiments, authors tried to binarize the neuron activations and compare face verification abilities with the original representation. The recognition rate dropped only by approximately 1%. This implies that binary activation patterns are more important than activation magnitudes.

- Selectiveness - Neurons in higher layers are selective to identities and identity-related attributes. There can be identified a subset of neurons, which is constantly excited or inhibited for a given attribute. These neurons have strong discrimination abilities.

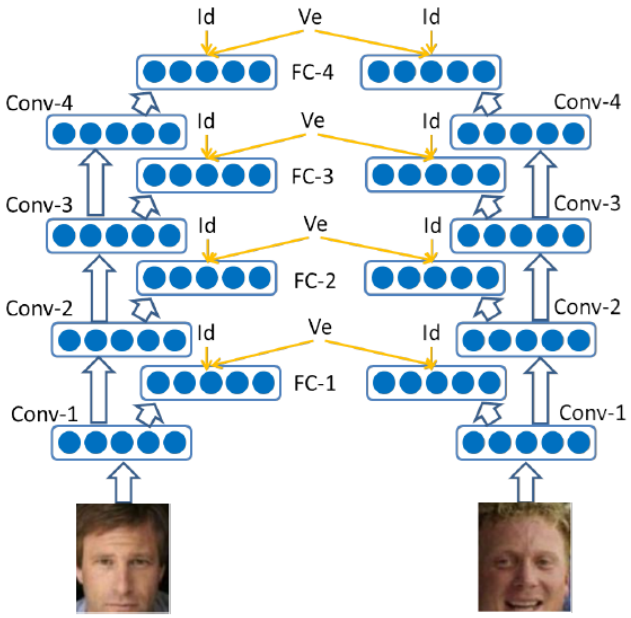


Figure 7.9: Structure of the neural network for DeepID2+ extraction. Id = identification supervisory; Ve = verification supervisory; FC-n = n-th fully connected layer. Conv-n = n-th convolutional layer. Taken from [150].

- Robustness - Experiments showed that neurons in higher layers are more robust than engineered features or neurons in lower layers. Face verification accuracy of DeepID2+ for 40% occlusion was still above 90%.

In conclusion, works from Sun et al. showed the real discriminate potential of learning-based features. Moreover, these works can help with the understanding of processes in deep neural networks.

In 2015, Schroff et al. [64] came up with another approach based on DNN, see Fig. 7.10. Their work not only advanced state-of-the-art significantly, but among the main contributions can be count (1) development of effective mapping from face images to a compact Euclidean space; (2) developing of efficient representation; (3) and most importantly, usage of the very effective triplet-loss function. In contrast with [13], this method use only rough alignment (translation and scale). Furthermore, FaceNet is directly trained to provide 128-D embedding using a triplet-based loss function based on Large Margin Nearest Neighbor [151]. The main motivation is that the triplet loss encourages all faces of one identity (intra-class) to be projected so that the margin between them is smaller than their margin from other person's faces (inter-class), and therefore is more suitable for FR tasks.

The embedding $f(x)$ is 128-D vector and it embeds an face image x into a 128-dimensional Euclidean space with additional constrain $||f(x)||_2 = 1$. Given an image x_i^a (anchor) of a specific person, any other image of the same person x_i^p and an image of any other person x_i^n (negative), we want to be true the following equation:

$$||x_i^a - x_i^p||_2^2 + \alpha < ||x_i^a - x_i^n||_2^2, \forall (x_i^a, x_i^p, x_i^n) \in T, \tag{7.8}$$

where α is margin (in experiments set on 0.2) that is enforced between positive and negative pairs, and T is the set of all possible triplets in the training set with cardinality N. The

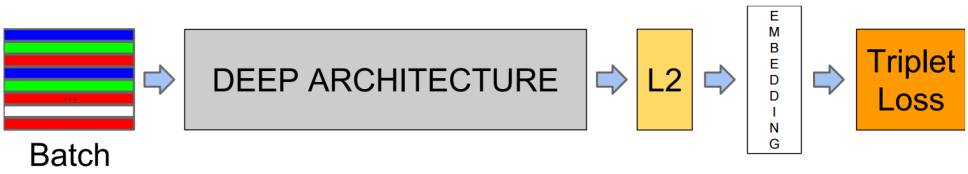


Figure 7.10: Structure of the FaceNet. The network consists of a batch input layer, a DNN, L_2 normalization, which results in the face embedding, and the triplet loss during training. Taken from [64].

triplet loss L_{tr}, that is being minimized during training is then defined as follows:

$$L_{tr} = \sum_{i}^{N} \left[||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha \right]_+ . \tag{7.9}$$

There exists a huge number of possible triplets, however many of them easily satisfy Eq. 7.8. These triplets would not contribute to the training and this result to slower convergence. The authors claim, that it is critical in order to fast convergence to select triplets, that violates this constraint. This means, that it is necessary to select an x_i^p such that $\text{argmax}_{x_i^p} ||f(x_i^a) - f(x_i^p)||_2^2$ (hard positive) and x_i^n such that $\text{argmin}_{x_i^n} ||f(x_i^a) - f(x_i^n)||_2^2$ (hard negative). Unfortunately, it is impossible to compute all hard triplets from the whole training set and additionally, the network might make poor conclusions, if we present it only hard positives and negatives. Therefore, authors generate triplets online using large mini-batch (a few thousand exemplars). Additionally, it needs to be ensured, that around 40 faces of exemplar of any identity are present in such mini-batch. Eventually, they used all positive pairs, instead of only hard positive, and hard negative pairs for the training. Square L2 distance thus trained feature vectors directly correspond to face similarity, therefore FR becomes k-NN classification problem, see Figure 7.11.

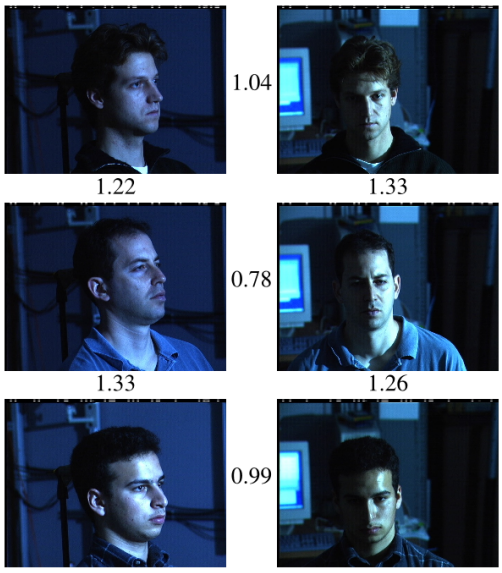


Figure 7.11: Euclidean distance between trained 128-D vectors for three different persons in two different illumination and pose conditions each. A threshold of 1.1 would classify every pair correctly. Taken from [64].

Authors tested two network architectures: (1) based on Zeiler&Fergus architecture with 22 hidden layers and 140 million parameters; (2) based on GoogLeNet Inception models architecture with only 6.6 - 7.7 million parameters. For training, they used standard SGD with standard back-propagation and AdaGrad. The training set contained 260 million face images. The method was evaluated on the LFW dataset, and Youtube Faces database [11]. On LFW, it achieved a classification accuracy of $98.87\% \pm 0.15$ using the fixed center crop and $99.63\% \pm 0.09$ when using the extra face alignment. This reduces the error reported in [13] by more than a factor of 7. On the Youtube Face database, authors achieved a classification accuracy of $95.12\% \pm 0.39$ using the first one hundred of frames, which reduced the state-of-the-art error rate by almost half. It can be seen that if one has a horrendous amount of data, face alignment step can be effectively discarded, unfortunately, it is not always possible to obtain such a big amount. The main advantage of this approach is that the trained embedding can be effectively used for all the tasks - for verification, identification, and even for face clustering.

In 2016, Masi et al. [152] proposed a method focused on the problem of extreme pose variations. The method assumes that in general, face pose distribution is not dominated by near-frontal faces. Authors observed that with detected landmarks on an image is easy to compensate roll when the face is near-frontal and for pitch when the face is near profile by using plane alignment. Because of these facts, their model is focused on compensating mainly yaw variations. In contrast with current techniques, which either expected a single model to learn pose invariance through the massive amount of training data [64], or which normalize images to a single frontal pose [13], their method explicitly overcome pose variations by using multiple Pose-Aware CNN Models (PAMs) (five models in total). The authors tested both AlexNet and VGGNet architecture. To be able to train PAMS on CASIA WebFace dataset [19], it was necessary to partition and augment the training dataset considering the training pose distribution. Moreover, the authors used a rendering technique to generate synthetic views using a generic 3D model. The authors revealed that their method reaches a better result if they used a co-training approach instead of training each CNN separately.

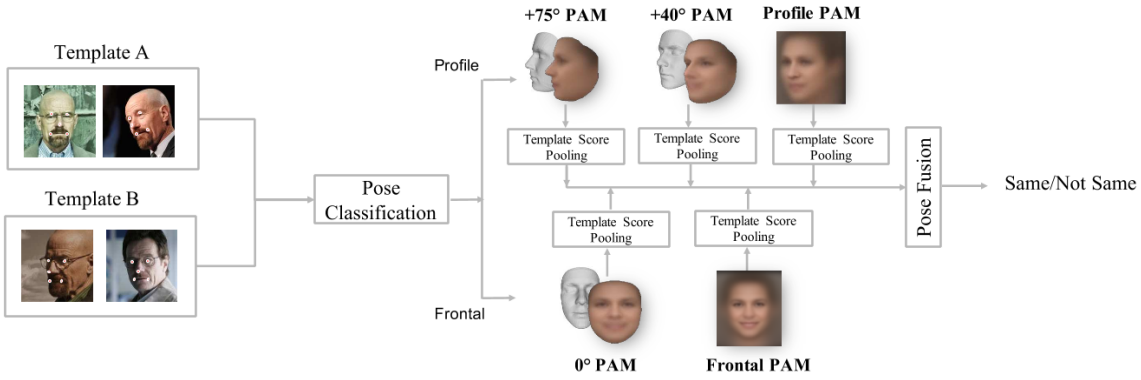


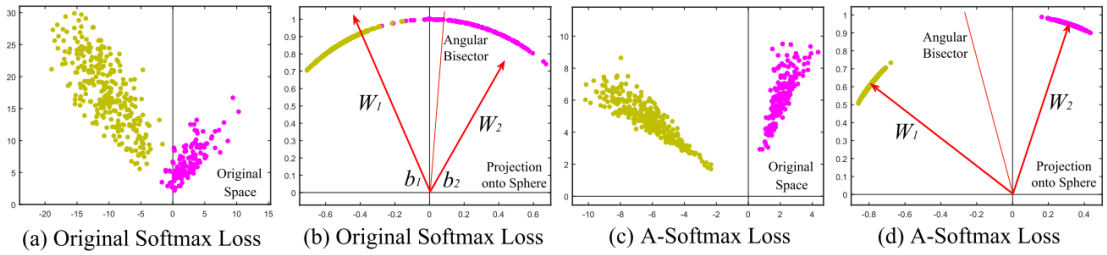
Figure 7.12: Pose-Aware FR - method overview. Taken from [152].

Given a testing image, the algorithm firstly detected landmarks and classify the pose either to near-frontal or near-profile. Additionally, the image is rendered into the half-profile view, and then, if the image is classified as near-frontal, into the frontal view, the image is rendered to the profile view otherwise. For all obtained view is generated a score, each by a specific PAM. All scores are pool by average, see Figure 7.12.

The matching process is then performed based on this final score. Authors tested their

approach on the IJB-A dataset and PIPA dataset and on both sets improved state-of-the-art results. For example, on the PIPA dataset PAMs method achieved 57.65% recognition rate, whereas DeepFace [13] only 47.97% (on IJB-A they reached 82.6% recognition rate). Authors plan to extend their work by incorporating the landmark confidence method into their future approach.

In 2017, Liu et al. [7] proposed a novel approach (SphereFace) with a focus on the open-set FR utilizing Angular Softmax (A-Softmax) loss. Authors reached 75.77% Rank1 identification accuracy on MegaFace challenge and improved state-of-the-art results. This approach also showed superiority over the previous approaches in two ways: (1) it was designed to handle open-set classification, see Figure 7.13; (2) in contrast with the triplet loss [64], it is not necessary to rearrange your training set (creating pairs, triplets, etc.), rather A-Softmax is very modular. Therefore, in theory, it is possible to take already existing neural network architecture, change Softmax for A-Softmax, train the architecture, and reached improved results. However, the following experiments show that A-Softmax training is very unstable, and generally, it needs to pretrain model with standard Softmax. Moreover, the implementation of A-Softmax is not a trivial task.



(a) Original Softmax Loss (b) Original Softmax Loss (c) A-Softmax Loss (d) A-Softmax Loss

Figure 7.13: Comparison of feature spaces (original and projection onto hypersphere) among standard Softmax and Angular Softmax in two class (classes are distinguished by different colors) classification task.

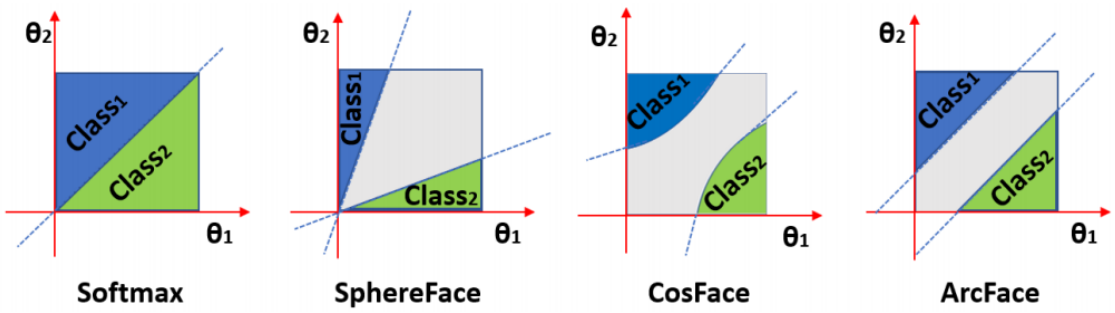


Figure 7.14: Decision margins of different loss functions under binary classification case. Taken from [68].

From 2017, novel methods are trying to follow modular fashion. Lets mention most important ones: NormFace [70], COCOFace [66], and CosFace [71]. In 2018, Deng et al. [68] presented their approach named ArcFace - a novel loss function based on additive angular margin. The loss stems from the A-Softmax loss, however, it more computationally efficient, much easier to implement, its training is stable (therefore there is no necessity to pretrain model with standard Softmax anymore), and it improves state-of-the-art results significantly. The

performance of different loss functions was directly compared to ArcFace, and despite the numerical similarity between them, ArcFace has better geometrical attributes as the angular margin has the exact correspondence to the geodesic distance. As illustrated in Figure 7.14, ArcFace has a constant linear angular decision margin throughout the whole interval. By contrast, SphereFace and CosFace only have a nonlinear angular decision margin. This minor difference in margin designs can have a big effect on model training. The authors utilized modified ResNet architecture and train it on the refines MS1M dataset and the Casia WebFace dataset. With this setting, they reached 83.27% Rank1 identification accuracy on the MegaFace challenge.

7.3 3D Face synthesis-based methods

Since directly matching two faces under different conditions is difficult, one intuitive method is 3D face synthesis. The synthesized faces are then transformed into the same conditions and therefore is comparison much more straightforward. The synthesis of the 3D morphable model (3DMM) can be done based on found feature points (local) or based on the whole image (holistic). This division and ideas behind it are very similar to two previous sections (obtaining face representation based on local facial features or based on the whole image), however, the usage of a 3D model has so many dissimilarities, that it deserves to have its own section. It should be noted that some approaches use a synthesized model only to face alignment (pose normalization, etc.) as a part of preprocessing, nevertheless, in this section would be visited only approaches, which directly use a model for face recognition.

Most 3DMM are PCA-based models, however, there are some other approaches, for example, 3D Generic Elastic model. In 1999 Blanz and Vetter proposed novel PCA-based 3DMM [153] (model is divided into two parts - shape and texture) and based on fitting this model, four years later, they proposed innovative face recognition approach [154]. It is a semi-automatic method that demands from six to eight manually labeled feature points on the face, such as corners of eyes, a tip of a nose, corners of the mouth. Based on computer graphics simulation of projection and illumination, it is estimated intrinsic shape and texture fully independent on extrinsic parameters. The iterative algorithm starts from the average face and standard rendering conditions (front view, frontal illumination, etc.). Then the user has to manually mark required feature points (depending on the visible part of the face). The fitting algorithm then optimize shape coefficients $\boldsymbol{\alpha}$ and texture coefficients $\boldsymbol{\beta}$ along with 22 rendering parameters concatenated into a vector $\boldsymbol{\rho}$ - pose angles, 3D translation, focal length, ambient light intensities, direct light intensities, two angles of direct light, light contrast, and gains and offsets of color channels. Given input image $\boldsymbol{I}_{in}(x, y)$ the primary goal in analyzing face is to minimize the sum of squared differences over all pixels and all color channels between this image and the synthetic reconstruction i.e.

$$E_i = \sum_{x,y} ||\boldsymbol{I}_{in}(x, y,) - \boldsymbol{I}_{model}(x, y)||^2. \tag{7.10}$$

In the first iteration are exploited the manually defined feature points $(q_{x,j}, q_{y,j})$ and the corresponding position on the model $(p_{x,j}, p_{y,j})$. That gives us additional error function:

$$E_f = \sum_j || \begin{pmatrix} q_{x,j} \\ q_{y,j} \end{pmatrix} - \begin{pmatrix} p_{x,j} \\ p_{y,j} \end{pmatrix} ||^2. \tag{7.11}$$

Minimization of these functions with respect to parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}$ may cause overfitting, therefore is employed a MAP estimator - find model parameters with maximum aposterior probability $p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} | \boldsymbol{I}_{in}, \boldsymbol{F})$ given the input image I_{in} and feature points $\boldsymbol{F}$. The fitting procedure is then maximizing this probability by minimizing the cost function, which is done with a stochastic version of Newton's method. Model fitting and identification were tested on the CMU-PIE dataset and FERET database. Testing showed that the algorithm adapts to different illuminations and poses well. For face verification purposes, the entire face was divided into four segments - eyes, mouth, nose, and the rest. For comparing two faces, the authors tested Mahalanobis distance, cosine distance, and distance based on the within-subject variation. The last one showed the most promising results. In conclusion, results were quite impressive (95% recognition rate), the method can handle different poses and illumination, however, the method is strongly dependent on the manual annotations and can not deal with different face expressions and ethnic groups. Moreover, the method ignores glasses, beards, and can have problems with occlusion.

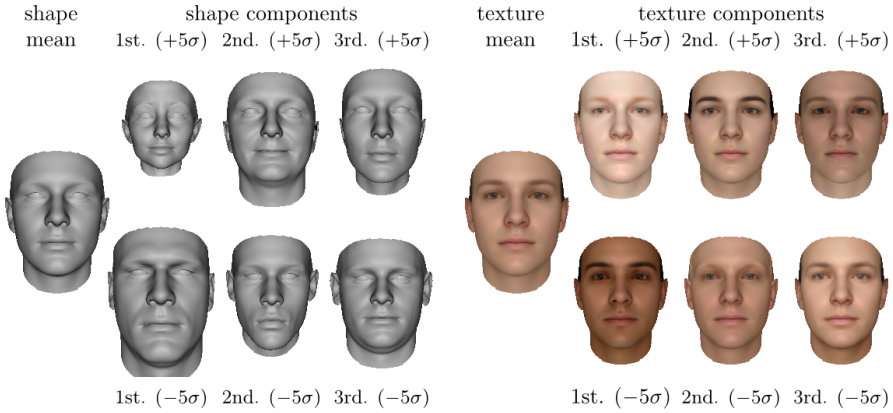


Figure 7.15: The mean μ together with the first three principle components of the shape (left) and texture (right) PCA model. Shown is the mean shape and texture plus/minus five standard deviations σ. Taken from [155].

In 2009, Paysan et al. [155] presented a novel PCA-based 3DMM named Basel Face Model (BFM) and demonstrated its application in face recognition tasks (the same model can be fit to 2D or 3D images under different external conditions). Older 3DMM has problems with recognition of faces from profiles and with some harder illumination conditions (like shadowing, etc.). Before BFM was common to use the same face dataset for both training and testing, however, such recognition systems have usually difficulties with generalization. Moreover, BFM is trained from high-resolution scans, and the innovative method of registration is used, which results in fewer correspondence artifacts.

The training set contained 100 scans of men and 100 women, mostly Europeans. Each person was scanned three times, and it was selected the scan with the most natural look. To establish the correspondence of the raw data was used a modified version of the Optimal Step Nonrigid ICP algorithm [156]. After registration, the faces are parameterized as triangular meshes with 53490 vertices and shared topology. Each vertex has an associated color. Applying PCA are created two models, one for shape $\boldsymbol{s}(\alpha)$ and one for texture $\boldsymbol{t}(\beta)$:

$$\boldsymbol{s}(\alpha) = \boldsymbol{\mu}_s + \boldsymbol{U}_s \ \mathrm{diag}(\sigma_s)\alpha, \tag{7.12}$$

$$\boldsymbol{t}(\beta) = \boldsymbol{\mu}_t + \boldsymbol{U}_t \ \mathrm{diag}(\boldsymbol{\sigma}_t)\beta, \tag{7.13}$$

where $\mu_{\{s,t\}}$ are the mean, $\sigma_{\{s,t\}}$ are the standard deviations and $U_{\{s,t\}} = [u_1, ..., u_n]$ are orthonormal basis of principal components of shape and texture. As a fitting algorithm was chosen nonrigid ICP algorithm again. The method was tested on the CMU-PIE dataset and FERET database. It advanced state-of-the-art results. In conclusion, this work, except advancing state-of-the-art, has one main contribution: it provides a powerful tool to generate precise 3D face models in any kind of pose and light variation. Unfortunately, the model can generate a face with only neutral facial expressions and can have problems with beards and glasses.

From newer approaches, let's mention the method proposed by Prabhu et al. [157]. The authors first constructed a 3D model for each subject in their database using a single 2D image by applying the 3D Generic Elastic model (3DGEM) approach. All obtained 3D models are saved into the gallery dataset. They choose the GEM approach because of its efficiency and effectiveness. However, 3D models generated by the GEM approach are derived from a single canonical depth map, which is an only strong approximation of the true depth map. This fact can cause problems for faces with atypical features. The authors proposed a method to mitigate these problems by learning different depth maps for a different ethnicity, ages, and genders. Before matching with a new face, an initial estimate of the pose of the input test image is obtained using a linear regression framework based on automatic facial landmark detection. Subsequently, each 3D model from the gallery dataset is rendered at the estimated pose, and a 2D projection is extracted. Then are aligned positions of the rendered and test images. Finally, the l-cosine distance is computed. The algorithm was tested on the images and the video input. The testing showed that the method is able to handle big pose variations and mild variations in illumination and expression.

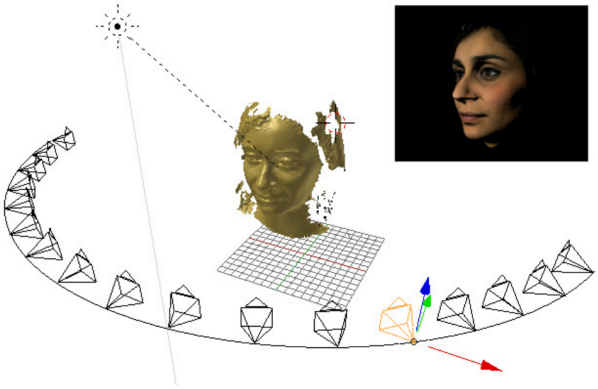


Figure 7.16: Synthetic data generation with given a 3D model. Taken from [158].

In 2013, Masi et al. [158] presented an approach to hybrid 2D/3D FR. In the first step is a high-resolution 3D model for each individual is acquired using 3D data from a scanner. From each model is artificially generated n (authors choose $n = 25$) 2D synthetic images across varying viewpoints, see Figure 7.16. The final representation of each individual in obtained as an unordered bag of SIFT descriptors calculated at salient image points identified using a Harris-Laplace corner detector. Given a probe image from unknown identity, SIFT descriptors are extracted on salient points again. For the classification task is employed a

sparse discriminative classifier. The main advantage of this approach is that no discriminative model is necessary to relearn after adding a new subject to the gallery. The experiments showed promising results.

Chapter 8

Face Recognition from Other Sensory Input

8.1 Video sequence

While traditional face recognition is based on still images, face recognition from a video sequence is very popular in recent years due to its significance for real-life applications (surveillance, etc.). A video-based FR system typically consists of three modules: face detection module, face tracking module, and FR module. In this section will be reviewed the FR module in more detail.

Part of video-based FR algorithms utilize approaches based on single image-based recognition, however, videos are capable of providing more information than still image [159]. There are four major advantages of video over the single image:

1. Multi-frame FR - the possibility of employing redundancy contained in the video sequence to improve the still images recognition rate. This can be done by a combination of classification results from several frames or by choosing the best frames and discarding the others.
2. Obtaining of more effective representation - 3D face model, super-resolution images, or multiple resolution-faces, can be obtained from a video sequence and used to improve the recognition rate.
3. Psychophysical and neural studies revealed that dynamic information is crucial in the human process of FR, especially when spatial image quality is low.
4. Video-based FR allows learning and updating the subject representation over time.

In light of these advantages, it may look like the video-based FR is easy, but there also exist some serious disadvantages:

1. Low video quality - this is a common problem of many real-life applications. It cannot be expected to use a full HD camera for surveillance purposes.
2. Cluttered background - the problem primary complicates face detection and tracking.
3. Large amount of data to process - this problem becomes less and less relevant with better computational devices.
4. Small face images - face image may be much smaller than the required size by the most conventional FR systems.

To all these drawbacks should be added the same drawbacks as for single image-based FR, however, some of them (occlusion, expression variations, etc.) are much easier to overcome thanks to the availability of multiple frames. Video-based FR methods can be divided into two groups: set-based methods, and sequential-based methods. Set-based methods consider videos as an unordered collection of images and take advantage of the multi-frame observation. Sequence-based methods explicitly use temporal information to increase efficiency. In conclusion, video-based face recognition has great potential for real-life tasks, however, it also brings many unsolved problems.

8.2 Heterogeneous face recognition

Heterogeneous face recognition (HFR) is face recognition across different visual domains. Instead of working with just 2D images, it comprehends the problem of closing the semantic gap among faces captured using different sensor devices, different camera settings, or between sketch and photography. HFR includes comparing infrared and RGB images, 2D and 3D data, photographs and sketches, high-resolution and low-resolution images, or comparing imagery before and after plastic surgery (within-modality heterogeneity). With the progress of FR techniques, heterogeneous FR become popular in recent years because heterogeneous sets of face images must be matched in many practical applications, for example, in security or for identifying a wanted person from eyewitness sketches. HFR algorithms have to, except general FR problems, overcome differences between two different representations. To address these differences, HFR systems contain an additional step named the cross-modal gap allowing direct comparison. Most HFR studies focus their effort on developing improved strategies for this step. Common strategies can be broadly divided into four following groups:

- **Feature-based** - these strategies are focused on engineering or learning-based features that are invariant to the differences between two different representations (modalities), while simultaneously being discriminative enough for a person's identity. Typical strategies include SIFT, LBP, or HOG.
- **Synthesis** - synthesis based strategies focus on synthesizing one modality based on the other. Typical methods include eigentransform, MRFs, Local Linear Embedding (LLE). The synthesized model/image is then used directly for matching. These strategies are crucially dependent on precise synthesis.
- **Projection** - projection strategies project both modalities of facial images to a common subspace, where they are comparable. These strategies usually include LDA, CCA, or partial least squares.

- **Feature selection** - Feature selection strategies are a special case of projection-based strategies, which rather than mapping the whole image to a common subspace, discover which subset of input dimensions are most modality irrelevant to compare across domains and ignore the others. One typical example is AdaBoost.

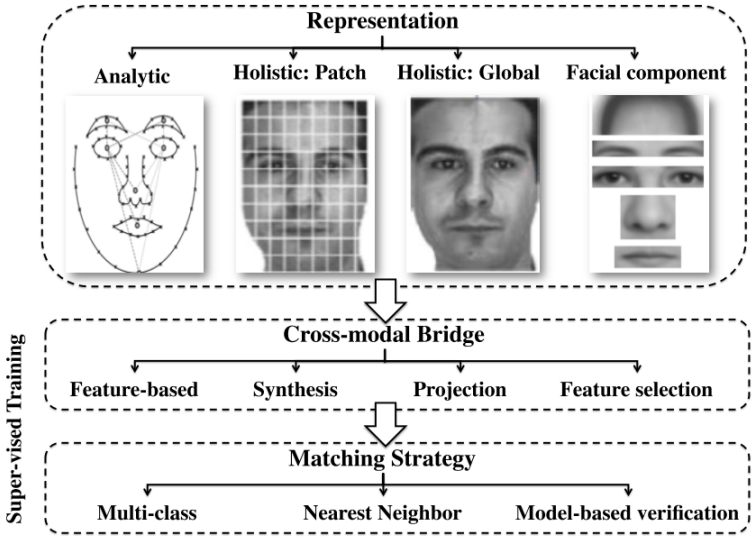


Figure 8.1: The diagram of HRF. Taken from [160].

HFR methods, therefore, usually contain three stages, see Figure 8.1: (1) obtaining face representation; (2) cross-modal bridge; (3) matching strategy. The first and the last one are very similar or identical, as in the traditional FR algorithms. Overall, there exist many different and relevant approaches used in the cross-modal bridge. A significant factor for HFR algorithms is the availability of the training data. Since large datasets of annotated cross-modal pairs are rare, methods that require no or very little training data are in big advantage. This fact especially slows down the modern homogeneous FR approaches based on neural networks. Very actual and elaborated survey about heterogeneous face recognition can be found in the article [160].

8.2.1 Facial Sketches

The problem of matching facial sketches to visible light images has an important application in assisting law enforcement in identifying subjects by retrieving their photos automatically from the existing police database. In most cases, the actual photo of the suspect is not available, only a sketch based on eyewitnesses description. Therefore, standard FR algorithms are out of the question. Sketches can be divided into four following categories:

- Viewed sketches - Sketches are drawn by an artist while looking at a corresponding photo.
- Forensic sketches - Sketches are drawn based on recollections of witnesses.
- Composite sketches - Sketches are produced by specific software.

- Caricature sketches - Sketches are generally hand-drawn, but facial features are exaggerated.

Most of the existing HFR algorithms are focused on recognizing viewed sketches because they are much more accessible. But this is not a realistic use case, nevertheless, viewed sketch performance should reflect performance for forensic sketches too, and their studying provides an essential step towards improving forensic sketch accuracy. Sketch-based FR algorithms can be broadly divided according to the cross-modal bridge strategy.

Feature-based approaches are trying to extract features invariant to the modality while preserving a person's identity. The most widely used image feature descriptors are SIFT, Gabor transform, HoG, and LBP. Once the features from both the sketch and the photo are extracted, classic FR algorithms are applied. Klare et al. [161] proposed a method based on invariant SIFT features. Euclidean distances between feature vectors are computed and then is used kNN matching. Galoogahi et al. [162] proposed a novel face descriptor based on LBP - Local Radon Binary Pattern. In this work are face images are first transformed into Radon space, then standard LBP is applied. The main advantage of this approach is low computational complexity and lack of any hyperparameters. Zhang et al. [163] introduced face descriptor based on coupled information-theoretic encoding. By maximizing the mutual information between photos and sketches in the quantized feature spaces, they obtained a coupled encoding using an information-theoretic projection tree. In 2011, Klare et al. [164] combined feature-based and projection-based approach. First, SIFT and LBP features are extracted. Second, LDA projection to minimize the distance between corresponding sketches and photos while maximizing the distance between identities. The main disadvantage is the necessity to design suitable features, which is a hardly solved task. Machine learning approaches provide a solution, however, there, unfortunately, is a problem with a lack of training data.

Synthesis-based approaches are trying to synthesize a sketch from a corresponding photo or vice-versa. After this step, traditional homogeneous FR algorithms can be applied again. Wang and Tang [165] proposed an eigensketch transformation, wherein a new sketch is constructed using a linear combination of training sketch samples, with linear coefficients obtained from corresponding photos via eigendecomposition. Obtained eigensketch features are then used for classification. Liu et al. [166] proposed a method inspired by a Local Linear Embedding to convert photos into sketches based on image patches. The nearest Neighbor from the training set is found for each converted image patch. Reconstruction weights of neighboring patches are then computed and used to generate the final synthesized patch. Wang and Tang [159] improved this approach by synthesizing local face structure at different scales using Markov Random Fields (MRF), see Figure 8.2.

Zhong et al. [167] modeled the nonlinear relationship between photo and sketch using embedded hidden Markov model. This model is then used to transform photos into sketches. On the other hand, Xiao et al. [168] using very similar approach synthesize sketches from photos. The main problem of sketch synthesis methods is that they hardly handle glasses and rare hairstyles. Moreover, the quality of synthesis is directly proportional to the amount of training data. In 2016, Ouyang et al. [29] presented model trained over their own database MGDB utilizing multi-task Gaussian process regression to synthesize facial sketches. Using the model, they addressed the memory problem (forgetting details by eye-witness after some time) and successfully reversed the forgetting process. They tested the model on IIIT-D dataset and reached state-of-the-art results.

Figure 8.2: Sketch synthesis using multi-scale MRF.

Projection-based approaches are trying to to find a lower-dimensional subspace in which the two modalities are directly comparable while preserving a person's identity again, where can we once again utilize standard FR approaches. Lin and Tang [169] based their method on linear transformation called common discriminant feature extraction. In 2011, Sharma et al. [170] presented two approaches - first based on CCA and second based on Partial Least Squares. The second one reached superior results.

8.2.2 3D data

The main reason in favor of using 3D information is the fact that this approach allows methods to use features based on the shape and the curvature of the face (such as the shape of the forehead, cheeks, etc.), without any distortion caused by illumination or pose variations. Mainly three following techniques are used to obtain 3D information: stereoscopic camera systems, structured light scanners, and laser scanners. There are three main tasks in 3D face recognition based on used data: (1) 3D to 3D recognition; (2) multi-modal 3D+2D recognition; (3) 2D to 3D recognition.

3D to 3D face recognition has great potential to provide the best possible results, because naturally overcome many FR problems, such as illumination or pose variations, scale, many types of noise, etc. But there are also some serious drawbacks of this approach - establishing correct alignment between two face surfaces is not a trivial task, devices to obtain 3D data are usually expensive, and the computational complexity of 3D FR methods is generally high. Moreover, common 3D scanners have problems to capture hairs and beard. One of the first attempts [171] utilizes the curvatures, and metric size of the face (shape of the forehead, jaw line, eye corner cavities, etc.) to the describe face. Nearest-neighbor matching is then done based on the distance between features. Tanaka et al. [172] also perform curvature-based segmentation and represent the face using an extended Gaussian image. An easy way to gain correct alignment is using of 3DMM [153][154][155]. Expression variations can be problematic for 3DMMs, so Amberg et al. [173] proposed a method based on fitting an identity/expression separated 3DMM to shape data. Another alternative to gain correct alignment is the iterative closest point (ICP) algorithm (which is sometimes also used for 3DMM fitting) used by Cook et al. in their work [174]. The ICP establishes a correspondence between 3D surfaces in order to compensate for problems due to the non-rigid nature of faces. Once the registration

is done, the face can be compared by, for example, the statistical model. Unfortunately, despite the strength of the ICP algorithm, it can not handle expression variations. Chua et al.[175] revealed that some regions of the face, such as the nose, eye socket, and forehead, are less sensitive to expression variations than the other. They perform face recognition based on these rigid parts with great success, however, it should be noticed that the testing set was very small. Then there exists a group of methods, which uses the same approaches as methods in the 2D domain. The most popular are probably PCA-based methods, for example, Achermann and Bunke [176] extend eigenfaces and HMM approaches to work in the 3D domain. From newer approaches, let's mention the method proposed by Lv et al. [177] using region-based extended LBP. The method extracts by binary mask different regions according to their distortion under facial expressions and represents them by the uniform pattern of extended LBP. Then, a sparse representation classifier is adopted for the classification of the single region.

Thanks to more information with which can multi-modal 3D+2D recognition, it has a natural advantage over pure 3D recognition., with just a small addition of purchase cost and computational complexity. The simplest methods fuse results obtained independently from the 3D data and the 2D data, for example, in 2003, Chang et al. [178] perform the PCA on the 2D and 3D data separately and then combines results obtained from both strategies. Moreover, they made the following conclusions: (1) combination of 2D and 3D results outperforms either 2D or 3D alone; (2) combination of multiple 2D images results outperforms a single 2D image; (3) combination of 2D and 3D results outperforms the multiple 2D images. Papatheodorou and Rueckert [179] presented a method based on the ICP algorithm. They modified it, and to the original three dimensions, they added intensity as the fourth dimension. 4D ICP method integrates shape and texture information at an early stage, rather than making a decision using each of them independently and combining them after. Tsalakanidou et al. [180] proposed an HMM approach to integrating depth data and intensity images. The experimental results were very promising, but the testing set was quite small again. In 2014, Hsu et al. [181] proposed a method for RGB-D face reconstruction and recognition. They used an RGB-D image of a face, and they reconstruct its 3D model and then perform classification via a Sparse Representation-based classifier. To conclude, the multi-modal 3D+2D domain shows promising results for the future research.

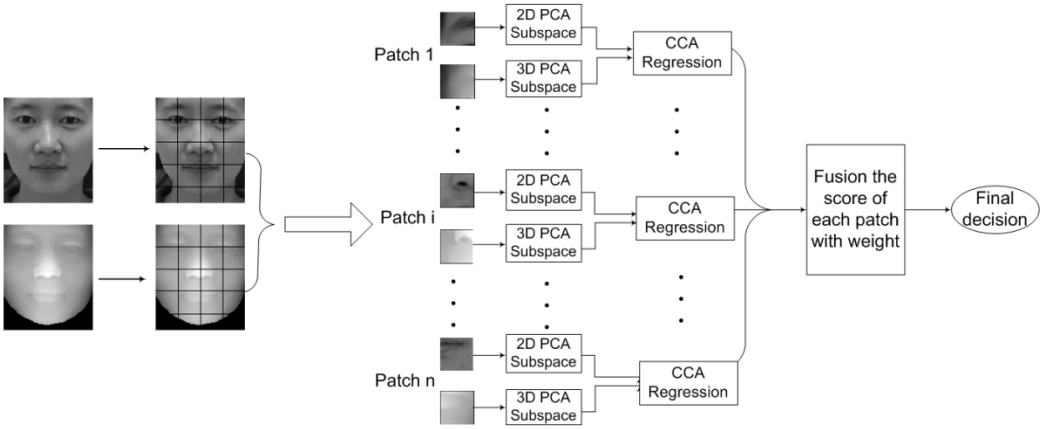


Figure 8.3: Patched based CCA in 2D-3D matching. Taken from [182].

The whole idea of 2D to 3D FR is based on the fact, that 3D data are more expensive to obtain so that these data will be stored only as trained images, however, probe images

obtained during utilization of the system will be only 2D as it is cheaper and 2D sensors are more available. 2D to 3D matching can potentially outperform 2D to 2D matching if the heterogeneity problem is effectively solved (matching between different coordinate systems). Yang et al. [182] used CCA (learned per patch) to correspond to the 2D and 3D face modalities and deal with their heterogeneous dimensionality. After projecting into a common subspace, cosine distance is applied, see Figure 8.3. Huang et al. [183] proposed a scheme to improve results by fusing 2D and 3D matching. 2D LBP features are extracted from both the 2D image and the 2D projection of the 3D image. Then they are compared by Chi-squared distance. LBP features are also extracted from the 3D image. These features from the 3D image are with LBP features from the 2D image then mapped into a common space using CCA and compared with cosine distance again. Obtained distances are fused at the decision level. Toderici et al. [184] proposed a novel 2D-3D FR method based on a bidirectional algorithm. A personalized 3D annotated model is obtained by using a generic 3D model and 2D texture. With such a model are 2D images projected into a normalized image subspace, where classification is performed.

8.2.3 Infrared light

Near-infrared (NIR) FR gets some attention in recent years because of its natural illumination invariance associated with decreasing cost of NIR acquisition devices. Furthermore, NIR facial images reveal veins and tissue structure of the face, which should be unique to each individual. Despite these undisputed advantages, NIR FR also brings many disadvantages. First, glasses are very problematic, because infra-red radiation is opaque to glass. Indeed, traditional FR systems working with intensity images can also have problems with glasses, however, they are still able to obtain some meaningful data. Second, infra-red images are sensitive to changes in ambient temperature and wind, which is hardly a problem for 2D FR systems. Third, testing showed that metabolic processes of captured subjects could also affect the infrared images. Fourth, in comparison with the 2D FR dataset, datasets for NIR FR are quite small and not very diverse.

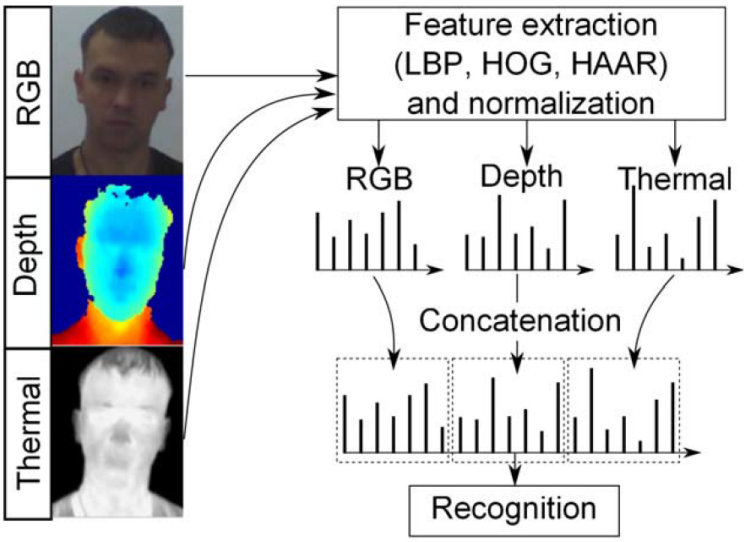


Figure 8.4: The block diagram of multi-modal FR method from article [185].

Socolinsky et al. [186] performed PCA on a database of visible and infrared images of 91 distinct subjects captured under variable illumination, with variable expression and with or without glasses. Infrared imagery significantly outperformed the visible one during all performed experiments. However, the superiority of the infrared approach probably stems from the fact that data did not contain sufficiently challenging situations in the infrared part, whereas it did so in the visible images. In 2001, Chen et al. [187] captured a dataset of images both in infrared and in the visible spectrum. Subjects were captured with variable expression and illumination during a period of 10 weeks. Studies revealed that despite approximately the same recognition rate for both sets captured the same day, visible images outperformed the infrared, when there was a significant amount of time between which were images captured. This is probably caused by the metabolic processes of the subjects.

In 2014, Nikisins et al. [185] proposed a multi-modal FR algorithm based on RGB-D-T data (it uses RGB images, depth data, and thermal images), see Fig 8.4. The authors tested various engineered-based features (LBP, HOG, HAAR-like) with many different classifiers (SVM), and based on these testing, they made a few important conclusions. First, capturing scenarios can be prioritized based on the complexity for the FR as follows: rotation (most difficult), illumination (less difficult), expression (the most simple one). Second, the importance of each modality is dependent on the capturing scenario, however, thermal data constantly holds a high impact in the FR recognition regardless scenario. Third, LBP features provide the best recognition rate in most cases, nevertheless, this can be caused by the fact that the tested LBP features had the highest dimensionality.

Chapter 9

X-Bridge based heterogeneous face recognition system

This chapter presents a detail description of my proposed heterogeneous face recognition system. The system is composed of two main parts: (1) cross-modal bridge; (2) feature extractor. At the end of this chapter is introduced a novel metric for measuring the performance of cross-modal bridge in the heterogeneous face recognition task.

9.1 Cross-modal bridge

In an ideal world, there would exist a dataset of image-sketch data pairs big enough to train face recognizer directly without using any cross-modal bridge. However, such a dataset is nonexistent, and its creation would be very complicated due to the necessity of handmade creation of all the sketches.

There are basically three different strategies of cross-modal bridges, however, in this work are utilized synthesis-based approaches. There are two main reasons for this decision. First, thanks to the fast development, GANs reach photo-realistic results in the synthesis of new facial images. I believe this makes them perfect candidates for synthesis-based cross-modal bridges. Second, after a synthesis from one domain to another, the traditional FR algorithm can be utilized. This gives me a very powerful tool to work with.

In this thesis, a novel supervised approach called X-Bridge is presented. X-Bridge is my method designed specifically as a cross-modal bridge in the heterogeneous face recognition tasks. The structure of X-Bridge stems from Pix2pix approach, however, inspired by UNIT, it assumes shared-latent space across two different domains. The main attribute of shared-latent space is that a pair of corresponding images $(x, \hat{x})$ from two different domains $\boldsymbol{X}, \hat{\boldsymbol{X}}$ can be mapped to a same latent code z in a shared-latent space $\boldsymbol{Z}$.

X-Bridge is composed of five main parts: encoder, two generators, and two discriminators. Each part is a different convolutional neural network. These parts create two main paths of the method: translation path, and reconstruction path. Each path can be imagined as

separated GAN and has its own generator and discriminator, whereas both of them share one shared encoder.

The task of the translation path is to translate an input image x from domain $\boldsymbol{X}$ to the other domain $\hat{\boldsymbol{X}}$ and therefore generate image $\hat{x}$ from this second domain. The whole process can be divided into a few steps. First, the encoder encodes important information from an input real image x_r into the shared-latent space $\boldsymbol{Z}$. Second, the generator decodes this information and generates a translated fake image $\hat{x}_f$ from the second domain $\hat{\boldsymbol{X}}$. Third, during the training, same as in the traditional GAN, the fake images with real images $\hat{x}_r$ from the second domain are introduced to the discriminator, which distinguishes between the real and the fake ones. The translation path in X-Bridge approach utilizes a conditional discriminator, which means there is additionally original x_r on the input of the discriminator. This prevents mode collapse and forces the generator to generate the corresponding pair image. The translation path is, in principle, the same as the Pix2pix method. The loss function of the translation path can be expressed as:

$$L_{TR}(E, G_1, D_1) = E_{x_r,\hat{x}_r}[\log D_1(x_r, \hat{x}_r)] + E_{x_r,z}[\log(1 - D_1(x_r, G_1(x_r, z)))]. \tag{9.1}$$

The task of reconstruction path is to encode original image x_r into the shared-latent space $\boldsymbol{Z}$ and then to reconstruct it in the original domain as x_f. During the training, the discriminator is utilized again, however, the standard one this time. Conditional discriminator would be too demanding for the reconstruction generator to overcome because, for the real image, the conditional input and the real input are identical. The addition of the reconstruction path motivates the shared encoder to preserve information about important facial features, to generalize better, and to learn important regularities across both domains. The loss function of the reconstruction path can be expressed as:

$$L_R(E, G_2, D_2) = E_{x_r}[\log D_2(x_r)] + E_{x_r,z}[\log(1 - D_2(G_2(x_r, z)))]. \tag{9.2}$$

Testing proves the benefit of mixing the traditional GAN objective with some metric loss to further motivate the generators to produce an image corresponding with the original input. X-Bridge employs L_1 distance in both paths. Additional losses are defined as follows:

$$L_{11}(E, G_1) = E_{x_r,\hat{x}_r,z}[||\hat{x}_r - G_1(x_r, z)||_1], \tag{9.3}$$

$$L_{12}(E, G_2) = E_{x_r,z}[||x_r - G_2(x_r, z)||_1]. \tag{9.4}$$

The final loss is then defined as follows:

$$L_F = \min_{E,G_1,G_2} \max_{D_1,D_2} L_{TR}(E, G_1, D_1) + \lambda_1 L_{11}(E, G_1) + \lambda_R[L_R(E, G_2, D_2) + \lambda_2 L_{12}(E, G_2)], \tag{9.5}$$

where λ_1 and λ_2 are proportional constants affecting the amount of influence of additional metric loss. During my experiments, both constants are heuristically set to the value = 100. λ_R is proportional constant affecting the learning speed of the reconstruction path. During experiments, λ_R is heuristically set to the value = 0.1. That means the shared encoder is affected ten times less by the reconstruction path than by the translation path. It's because the reconstruction is generally easier, and also good translation is the primary goal of the X-Bridge method. For the X-Bridge pipeline, see Fig. 9.1.

Inspired by Pix2pix, X-Bridge also employs Markovian discriminators in both of its paths. This discriminator tries to classify if each $N \times N$ patch in an image is real or fake. The

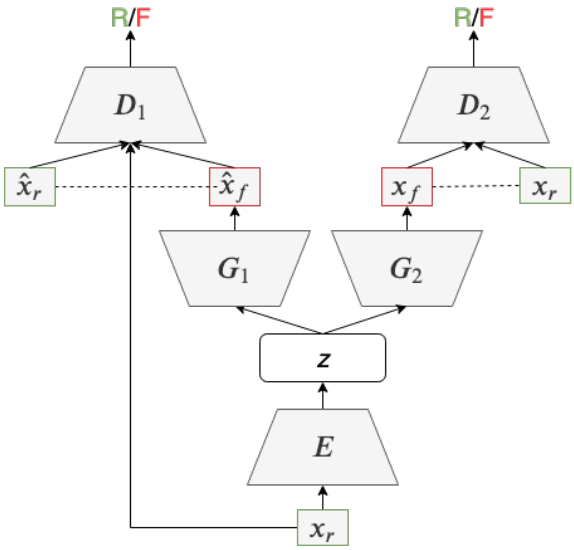


Figure 9.1: X-Bridge pipeline. E = encoder, G_1, G_2 = generators, D_1, D_2 = discriminators, z = latent space. Dotted line indicates L_1 loss. $\boldsymbol{x}_r$ is real input from the first domain, $\hat{\boldsymbol{x}}_f$ is reconstructed fake image from the first domain, $\hat{\boldsymbol{x}}_f$ is translated fake image from the second domain, $\hat{\boldsymbol{x}}_r$ is corresponding real image from the second domain. The translation path is on the left, whereas, the reconstruction path on the right.

discriminator is run across all the images, averaging all responses to provide the final output. N is heuristically set to the value = 70.

In X-Bridge, the shared-latent space is primarily enforced by the shared encoder. To further enforce it, the first four layers (high-level layers) of generators are sharing their weights. Different weights in low-level layers allow generators to specialize in the specific domain.

To improve important features propagation, skip connections between the last four layers of the encoder and the first four layers of the generators are added. Specifically, the skip connections are implemented as channel concatenation of all channels between each ith layer and $(n-i)$th layer, where n is a total number of layers. Element-wise addition (residual skip connection) was also tested, however, the current implementation provides better results.

9.2 Feature Extractor

A choice of suitable neural network architecture and good design of loss function are important parts of modern FR approaches. Because of the lack of any sketch-based face recognition dataset, it is important the feature extractor works in the open-set testing protocol, i.e., obtained features are not only linearly separable, but they also create compact class clusters with a margin between them. Therefore, chosen neural network architecture has to be effective enough to be able to transfer data from image space to such feature space. Moreover, the chosen loss function should motivate the creation of the class clusters by direct inclusion of the required margin during the feature extractor training process.

Combining state-of-the-art knowledge with my experiments, I decide to utilize DenseNet-121 with Arc loss. As a training data, I use preprocessed Casia-WebFace dataset and retrain architecture pretrained on ImageNet challenge. The feature extractor expects the input to be a facial image. For face detection in an unknown image can be utilized Haar Cascade detector, for example, however, the task of the face detection is outside the scope of this work.

In the open-set testing protocol, the classification of the image is based on the distance between the features of a tested image and an anchor. If the distance is lower than the chosen threshold, the tested image is classified into the anchor's class, otherwise, a new anchor from the database is chosen. If no untested anchor left, the person on the tested image is declared to be an unknown subject.

9.3 Pipeline of the system

The pipeline of the whole X-Bridge based heterogeneous face recognition system can be found on Fig. 9.2.

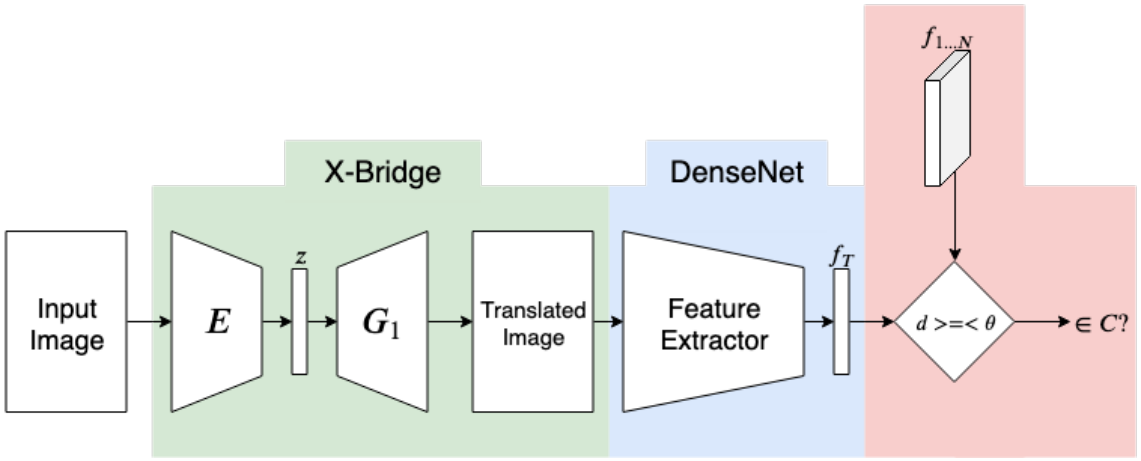


Figure 9.2: X-Bridge based heterogeneous face recognition system pipeline. The system is composed of three main parts: (1) Cross-modal bridge (green); (2) Feature extractor (blue); (3) Final decision based on a distance of features (red).

The input image x with a human face is firstly encoded by the encoder E into the shared-latent space $\boldsymbol{Z}$. In the next step, based on latent code z, the generator generates translated image $\hat{x}$ from the required modality. $\hat{x}$ is further sent into the feature extractor, which produces feature vector f_t on the output. In the last step, a distance between f_t and anchor's feature vector f_a is calculated, where f_a is a feature vector of an image from a tested database (containing N feature vectors). If the distance is lower than the threshold θ, x is classified into the class C_a (anchor's class), a new anchor from the database is chosen otherwise. If no anchor left, the person on the image x is declared to be an unknown subject.

9.4 Facial Features Preservation Score

For GAN evaluation, there are used two different performance metrics - Inception Score, and Fréchet Inception Distance.

Inception Score (IS) [188] is named after Inception classifier [50], which is classification network from Google trained on the ImageNet challenge. There are two criteria in measuring the performance of GAN: (1) The quality of the generated images; (2) The diversity of the generated images. The Inception network is used to classify the generated images and predict $P(y|x)$, where y is the class label, and x is the generated image. The IS has the lowest value of 1, and the highest value of the number of classes supported by the classification model, i.e., the highest possible IS is 1000.

Fréchet Inception Distance (FID) is also using the Inception net, although, only for extraction features from an intermediate layer this time. Using a multivariate Gaussian distribution, the data distribution of extracted features is modeled. The FID between the real images and generated images is then computed. The main advantage of FID over IS is its better robustness to noise.

However, despite the fact the used cross-modal bridge is based on GAN, I argue that none of the above-listed metrics is suitable to evaluate cross-modal bridge qualities fully. The most important thing for the cross-modal bridge in heterogeneous FR pipeline is the ability to preserve important facial features. Such features are essential for the correct decision of face classifier and, therefore, critical for the proper functioning of the whole heterogeneous face recognition system. To further elaborate, GAN's higher IS or FID does not directly ensure its better performance while using it as the cross-modal bridge.

From the above-mentioned reasons, I propose a novel performance metric directly for the evaluation of a cross-modal bridge called Facial Features Preservation Score (FFPS). FFPS is defined as follows:

$$FFPS = \frac{RR(\text{translated})}{RR(\text{original}) + \xi}, \tag{9.6}$$

where ξ is a very small number, RR is recognition rate on the *original* dataset, *translated* respectively. For the open-set classification protocol, the recognition rate is substitute by the F1 Score. To obtain recognition rates (F1 Scores), I propose to employ the pretrained ArcFace network [68], which is publicly available and provides state-of-the-art results across multiple benchmark datasets. The same network is utilized for both - RGB and Sketch domain. I believe that within one modality feature extractor should be robust and precise enough.

The FFPS has the lowest value 0, which can occur in the case recognition rate of *translated* dataset is 0. The highest value of the FFPS is not limited, nevertheless, I argue there should be the most information in the original data, therefore, with their translation is some information lost. This means the recognition rate of *translated* dataset should always be equal or lower than the original one.

Chapter 10

Experiments

In this chapter are presented various experimental results divided into two main parts which correspond to the two main parts of the system - comparison of feature extractors, and comparison of cross-modal bridges. Each part starts with a description of the used dataset and its preprocessing. In the following subsections are introduced tested methods, experimental settings, and reached results. Each main section is ended with result comparison and their discussion. All the tested methods were implemented in Python using Keras [189] or Pytorch [190] deep learning frameworks.

10.1 Feature extractor comparison

In this section are described experiments with different feature extractors in the standard face recognition task. Based on these experiments is chosen final feature extractor used in the pipeline of my own system described in the previous Chapter. The outline is as follows: First, the training data are described. Second, the comparison of state-of-the-art architecture is listed. Third, different loss functions are tested on two different datasets. Fourth, the obtained results are discussed, and conclusions are drawn.

10.1.1 Training data

For the following experiments, I use the Casia-WebFace database as a training set. If it does not say otherwise, it is also used as a testing set. Casia-WebFace contains 494414 RGB images of 10575 subjects, each with the resolution of 250×250 pixels. Persons are captured in variable external conditions, including pose, illumination, occlusion, age variations, haircut changes, sunglasses, etc. For exemplary images, see Fig. 10.1.

For the training of tested neural network architectures, I decided to use only identities, which have at least 100 images presented. With this step, I largely alleviate a problem with unbalanced classes for the training, when it is easier for the method just to ignore rare classes and focus on the common ones during the training. This flaw generally has two main

Figure 10.1: Exemplary images from Casia-WebFace database. Taken from [191].

solutions. First, the usage of weighted loss during the training, where the loss from the rare classes has introduces a larger penalty than from the common ones. The second solution is to balance the frequency of classes via a change of the training set.

If it does not say otherwise, all the training data are preprocessed in the following way. The resolution of all images is decreased to 128×128 pixels. To enrich the training data, I generated a horizontally-flipped version of each image. To further enrich the training set and address the rest of the unbalances, there are performed data augmentations. To be more specific, I modified images with Gaussian blur, noise, and brightness transformations. This leads to 908953 images in total. The data are split into three subsets - training, validation, ad testing set, in portion 70-15-15. All the image values are normalized from 0 to 1.

10.1.2 Comparison of state-of-the-art architectures

There is presented a comparison of a baseline NN architecture and three most important neural network architectures usable for image classification in this subsection. All the networks in this chapter are trained on a multi-class closed-subset face recognition using standard cross-entropy loss function, and their last layer is a fully-connected layer with 925 neurons (one for each class). For updating NN's parameters is used standard SGD method.

As a baseline method is used simple CNN containing three convolutional layers (32 filters each), each followed by ReLU non-linearity and max-pooling layer. After convolutional layers, a fully-connected layer with 1024 neurons is employed, followed by ReLU non-linearity and dropout with drop-rate 50%. CNN was trained with a mini-batch size of 128 images during 400 iterations.

As the second tested architecture, it is utilized VGG16, which belongs to the golden standard

among classification networks nowadays. Its main drawback is a huge number of parameters and, therefore, very slow training. The main advantage is the possibility to download a model which was pretrained on ImageNet challenge and only fine-tune its weights because I presume features extracted by initial convolutions to be the same or very similar for face recognition as for general image classification. VGG was fine-tuned with a mini-batch size of 64 images during 150k iterations.

The third tested architecture is based on the Deep Residual network ResNet-50. Despite its much bigger depth, its number of parameters is approximately only one-quarter of their number in VGG16 architecture. I again use the ImageNet pretrained model, which I fine-tuned with a mini-batch size of 64 images during 150k iterations.

The last architecture is Dense Convolutional network DenseNet-121, which is again fine-tuned with a mini-batch size of 64 images during 150k iterations. The reduction of parameters comparing to the ResNet-101 is approximately 70%. Comparison of results of classification is showed in Table 10.1.

Table 10.1: Comparison of classification recognition rates for tested state-of-the-art architectures. Partially taken from [191].

Architecture	**Development set**	**Test set**	**Number of parameters**
Baseline CNN	72.5%	71.1%	2148189
VGG16	85.2%	84.4%	132863336
ResNet-50	91.6%	90.9%	25636712
DenseNet-121	96.6%	96.2%	7901056

DenseNet architecture decreased the recognition error by more than 5% on both development and test set, comparing to the second-best tested architecture - ResNet-50. This is approximately 60% of relative error decrease. These results are even more significant from the point of view of parameters, because, as it was already said, DenseNet spares around 70% of parameters comparing to ResNet-50 and approximately 94% spare comparing to VGG16. This fact indicates a huge boost of parameter efficiency across the two newer architectures. Both of them significantly surpassed the baseline architecture and VGG16 while also spare a huge amount of parameters and, therefore, also computational time.

10.1.3 Comparison of loss functions

There is presented with a comparison of different designs of loss functions in this subsection. All the results in the following table are taken from the original articles and/or from the MegaFace challenge result table. Unfortunately, I was unable to replicate these results, because there is no longer possible to access the dataset due to the inactivity of authors of the challenge. Moreover, only results from the five most important loss functions are listed, see Tab. 10.2.

All the tested loss functions significantly outperform standard Softmax loss function, whereas loss functions based on angular and cosine margin reach superior results.

Table 10.2: Comparison of the MegaFace challenge results for different loss functions.

Method	Rank1 - Identification
Softmax loss	78.89%
Triplet loss	80.60%
SphereFace	82.95%
Coco loss	80.57%
Arc loss	83.57%

In the following experiment, I implemented chosen loss functions and use them for fine-tuning of pretrained ResNet-50. The classification protocol and the training setup are the same as in the previous subsection. The results are listed in Tab. 10.3.

Table 10.3: Comparison of classification recognition rates for chosen loss functions.

Method	Recognition rate
Softmax loss	90.9%
Contrastive loss	91.9%
Triplet loss	92.6%
Arc loss	95.3%

It can be seen that a change of the loss function without any changes in the NN's architecture can significantly improve classification recognition rates. All the tested loss functions outperform standard Softmax. Arc loss reaches the best results once again. Moreover, Arc loss can be trained the same way as Softmax loss, i.e., without any changes of labels of the training set, which is a significant advantage in comparison with Triplet loss, for example.

10.1.4 Discussion

In this section is presented a comparison of different NN architectures and different classification loss functions. The comparison of state-of-the-art architectures shows superior results of DenseNet over the other candidates. Moreover, the training time of the network is lowest, thanks to the significant saving of parameters.

Arc loss reaches the best results in the comparison of loss functions. By combining the reached results with the ability to use original labels for the training of the network, Arc loss becomes potentially the best option among tested losses.

For the final heterogeneous face recognition system, I retrain pretrained DenseNet using Arc loss. During the training are the first few layers fixed. As the training data, I use preprocessed Casia-WebFace again. This FR approach reaches 98.9% recognition rate on the development set and 98.6% recognition rate on the test set.

10.2 Cross-modal bridge comparison

In this section are described experiments with methods potentially useful as a cross-modal bridge. First, the training and testing data are described. Each tested method is evaluated for the sketch-to-image, and also for the image-to-sketch task. Qualitative results are provided for all the tested methods. For chosen methods, quantitative methods are listed afterward. The results of state-of-the-art methods are compared with a novel proposed approach. At the end of this section, conclusions are made and discussed.

10.2.1 Training data

There basically exist two suitable datasets for the training of facial photo-to-sketch translation systems. First, a CUFS dataset, which has three main subparts and should contain 606 photo-sketch pairs in total. However, the photos from the second and the third part are no longer available on the Internet. This means there remains only a CUHK dataset containing 188 images of students from Honk Kong university, see Fig. 10.2. Despite its small size, it is a very popular dataset. All the photos are in the frontal pose, normal lighting conditions, and with a neutral expression. All the sketches are drawn by an artist. Apart from the small size, the main disadvantage is the fact there are only Asians in the dataset. This is a very limiting factor for the training of the system because thanks to the total omission of the other races, there is a high probability that the fully-trained system would have problems with generalization and, therefore, with the translation of the different data.

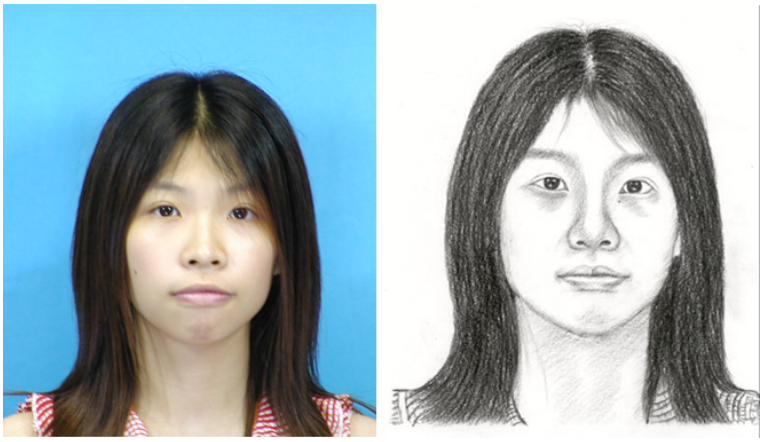


Figure 10.2: Exemplary image pair from CUHK database.

The second possible dataset is CUFSF. CUFSF includes 1194 sketches drawn by an artist, which corresponds to images from the color-FERET dataset. Unfortunately, there are some inaccuracies in pair filenames, therefore, only 895 pairs can be easily constructed. Sketches are provided in two versions - original version, and a cropped version. All the corresponding RGB images can also be cropped according to coordinates of the center of the eyes and the tip of the nose, which are provided by the authors of the dataset. For the exemplary image pair, see Fig. 10.3. The preprocessed images have a resolution of 426×372 pixels.

Apart from the bigger size of this dataset, the main advantage is the presence of different nationalities among the drawn subject, therefore, I decide to use the CUFSF dataset instead

Figure 10.3: Exemplary of preprocessed image pair from CUFSF database.

of the first one. In all the following experiments are used cropped versions of the sketches. To enrich the training data, a horizontally-flipped version of each image is generated again. Unfortunately, there is no possibility to use very popular augmentation - Gaussian blur because it would the generator allow learning to produce blurred images, and moreover, thanks to the presence of the blurred images in the real data, there would not be any way for the discriminator to reveal them. All the data values are normalized from 0 to 1. The data are split into three subsets again (training, validation, testing) in portion 80-10-10.

10.2.2 Testing data

To obtain quantitative results for different cross-modal bridges and also for the testing of the performance (for details see the next Subsection) of the whole heterogeneous face recognition system in the image-to-sketch translation task is chosen a color-FERET dataset. The FERET dataset was created to support the development of automatic FR systems assisting security and law enforcement. Unfortunately, obtaining the original dataset is very complicated nowadays. Luckily, since its original release, there was an update of the original FERET with new additional RGB images. The updated dataset was named color-FERET, for exemplary images see Fig. 10.4. For each identity, there are at least six different images with big pose variations captioned in controlled lighting conditions.

For modern homogeneous FR approaches, it is an easy dataset, and there is a trend to test novel approaches on more challenging ones. Nevertheless, there is important to point out that in law enforcement tasks or wanted-person database searching tasks, i.e., tasks relevant in heterogeneous FR, it is expected photos of the subject will be taken in a controlled or a semi-controlled environment. This makes the color-FERET ideal benchmark dataset for heterogeneous FR.

The color-FERET dataset is preprocessed in the following way: (1) Images, used for the training of cross-modal bridges are completely removed because otherwise, they would positively affect the recognition rate of the whole system. That leaves me 11336 images for 994 identities; (2) Each image is cropped. If coordinates of eyes, mouth, and nose are provided from the authors of the dataset, the crop will be based on these coordinates. Otherwise, the crop is based on the Haar-Cascade face detector. All the crops are converted to gray-scale

Figure 10.4: Exemplary images from color-FERET database.

because colored images are not presented in the cross-modal training set. The preprocessed images are resized to a resolution of 426×372 pixels to correspond to the cross-modal bridge training set.

Unfortunately, to my best knowledge, there does not exist any suitable dataset for testing heterogeneous FR system in the sketch-to-image task. On the one hand, there exist some facial sketches dataset, on the other hand, they always contain maximally one image per person, i.e., are unusable for FR task. For this reason, only qualitative results for sketch-to-image translation are listed in this work.

10.2.3 Quantitative-results testing protocol

To objectively compare the results of different cross-modal bridges, I propose the following testing protocol. In the first step, using the tested method, I translate the whole FR benchmark dataset to the sketch modality. In the second step, there is an applied chosen facial feature extractor. The results are evaluated under the open-set setting protocol. To obtain a quantitative result for each cross-modal bridge is calculated FFPS. To further compare and also to provide better picture and comparison with already existing systems, precision, recall, and F1 Score of the proposed heterogeneous systems using tested cross-modal bridge are calculated. Accuracy is omitted due to a large number of true negatives comparing with true positives.

It would be ideal, if there exists some dataset directly designed for image-sketch recognition, however, to my best knowledge, there exists none to this day. Therefore, color-FERET is

chosen as the benchmark dataset. On the one hand, it is a relatively easy dataset for state-of-the-art FR algorithms. On the other hand, it is still challenging for heterogeneous FR tasks.

For facial features, extraction is chosen DenseNet trained with ArcFace loss. This decision is based on the experiments from the previous section and justified in Section 10.1.4.

10.2.4 PG-GAN

The first tested system is PG-GAN - a high-resolution generator. I follow the same assumption as UNIT - a pair of corresponding images from two different domains can be mapped to the same latent code in a shared-latent space. The pipeline of the experiment is following: (1) Use PG generator G_{PG} to generate huge amount of synthetic image $\boldsymbol{x}_{gen}$- latent code $\boldsymbol{z}_{gen}$ pairs using original PG-GAN; (2) Train encoder E_I with mirrored structure of the generator using synthetic pairs; (3) Utilize trained encoder E_I to obtain latent codes $\boldsymbol{z}_C$ for images from CUFSF dataset; (4) Utilize sketch $\boldsymbol{x}_s$ - latent codes $\boldsymbol{z}_C$ pairs to train encoder E_S; (5) Use encoder E_S and generator G_{PG} as a cross-modal bridge.

In the first step, I generate 400k synthetic image $\boldsymbol{x}_{gen}$ - latent code $\boldsymbol{z}_{gen}$ pairs. This new dataset is split into three subsets - training (320k pairs), validation (40k pairs), and testing (40k pairs). Each synthetic image is generated from randomly generated latent code with normal probability distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma^2 = 1$. For the purpose of the training, all the generated images are resized from the original resolution 1024×1024 pixels to the resolution 128×128 pixels.

In the second step is utilized CNN as an encoder with the mirrored architecture of the generator G_{PG}, for the detail see Tab. 10.4. I tested two different setup of the architecture - one with max-pooling and one with average-pooling as the last layer. The architecture with average-pooling reached slightly better results. The encoder's goal is to encode the synthetic image into the original latent code correctly, i.e., it is trained for the regression task using L_2 squared norm. For updating CNN's parameters is used standard SGD with initial learning rate 0.1 and step decay 0.1 every 60 epochs. The network is trained during 300 epochs with a mini-batch size of 64 images. I also tested other optimizers, but SGD provides the best results.

The fully-trained encoder provides very good results for the synthetic data during testings. After encoding the synthetic image, I make its reconstruction and compare the result with the original image. Generally, the pose of the reconstructed image is almost identical, and most of the facial features too. There are appearing some small differences in facial details, nevertheless, the reconstructed person is very similar to the original subject, see Fig. 10.5.

Unfortunately, this is not a case for real images. In the next step, I take images from the Casia-WebFace dataset and try to encode and reconstruct them the same way as the synthetic data. Despite the same preprocessing and very similar difficulty of the data, results are much worse. The preservation of pose is very reliable, however, the resemblance of the reconstructed person is very vague, and facial features are generally very different, see Fig 10.6.

I argue this can be caused by two possible reasons. First, different distribution between the synthetic and the real data. The encoder sees only the synthetic data during training,

Table 10.4: Structure of proposed encoder E_{PG}. All convolutions are implemented with stride 1. There are two possibilities for the last pooling layer - average-pooling, or max-pooling. The output of the last layer has size 512×1, which is also the size of the latent space $\boldsymbol{z}$.

Conv2D(32, 3× 3)	Conv2D(128, 3×3)	Conv2D(256, 3×3)	Conv2D(256, 3×3)
Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)
Conv2D(32, 3× 3)	Conv2D(128, 3×3)	Conv2D(256, 3×3)	Conv2D(256, 3×3)
Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)
MaxPool(2×2)	MaxPool(2×2)	MaxPool(2×2)	MaxPool(2×2)
Conv2D(64, 3×3)	Conv2D(256, 3×3)	Conv2D(256, 3×3)	Conv2D(512, 3×3)
Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)
Conv2D(64, 3×3)	Conv2D(256, 3×3)	Conv2D(256, 3×3)	Conv2D(512, 3×3)
Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)	Leaky ReLU(0.2)
MaxPool(2×2)	MaxPool(2×2)	MaxPool(2×2)	Pooling(2×2)

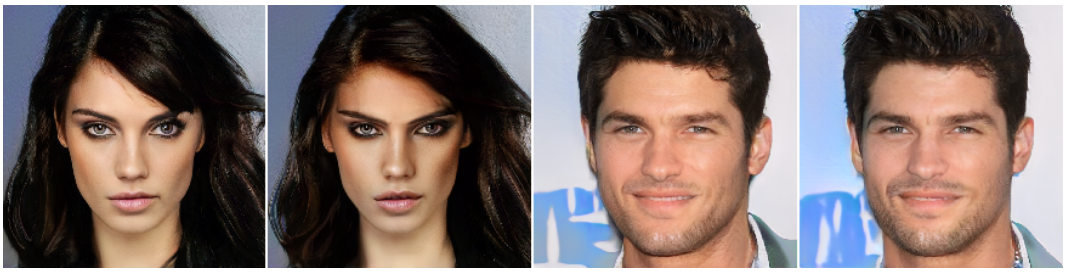


Figure 10.5: Results of the encoder E_{PG} for the testing synthetic data. In each pair is an original encoded image on the left and reconstructed image from the obtained latent code on the right.

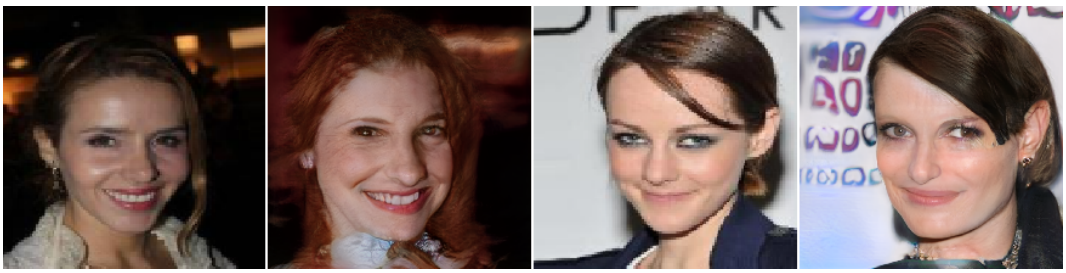


Figure 10.6: Results of the encoder E_{PG} for the testing real data (Casia-WebFace). In each pair is an original encoded image on the left and reconstructed image from the obtained latent code on the right.

and it is learned their distribution. The encoding of data with different distribution can be very problematic for it. Second, an insufficient mapping of the latent space. It is possible the generator can handle only an "incomplete" representation of latent space, i.e., can not generate reasonable results for some parts of it, because these parts were not introduced to it during its training. In combination with the possibility, the encoder encodes the never seen real data to these parts of the latent space, the generator can have serious problems to

reconstruct the data correctly.

Unfortunately, without the ability to correctly encode and reconstruct the real data, this method can not be used as a cross-modal bridge. I propose two possible solutions. First, a comparison of the distribution of the synthetic and the real data. If there exist any differences, it is necessary to adjust the synthetic data generation. Second, training of the encoder with the generator with fixed parameters together. By this approach, it is possible to calculate the loss of the encoder not as the difference between latent codes, but as a difference between original and reconstructed image. Moreover, if I assume that the generator can also reconstruct the real data, it is no longer necessary to use the synthetic data for the training. I plan to address the problem in my future work.

10.2.5 VAEGAN

VAEGAN can be a very potent solution for image-to-image translation. VAEGAN is composed of three main parts - encoder, generator, and discriminator. For the training of the method is used the CUFSF dataset. All the images are resized to the resolution of 64×64 pixels. The best-tested architecture can be found in Tab. 10.5.

Table 10.5: Structure of the best-tested architecture of VAEGAN. All convolutions have stride 2. Each layer except the last one in the discriminator (which is the classification layer) is followed by instance normalization and ReLU activation function. The latent space is represented by two fully-connected layers with 256 neurons.

Encoder	**Generator**	**Discriminator**
Conv2D(64, 5× 5)	FC(1024)	Conv2D(32, 5×5)
Conv2D(128, 5× 5)	Deconv2D(256, 5×5)	Conv2D(128, 5×5)
Conv2D(256, 5×5)	Deconv2D(128, 5×5)	Conv2D(256, 5×5)
FC(1024)	Deconv2D(32, 5×5)	FC(256)
		FC(1)

Figure 10.7: Results of the reconstruction of images using VAEGAN.

I perform three different experiments with VAEGAN. First, image reconstruction to verify the

potency of the approach. I use data from two facial datasets - Casia-WebFace and CelebFaces. I do not make any augmentation, I just resize all the images to the size of 128×128 pixels and then randomly cropped an area of size 64×64 pixels around the center. Moreover, all image values are normalized between 0 and 1. VAEGAN is trained during 50 epochs with mini-batch size 64 and initial learning rate $l = 3 \times 10^{-3}$ using RMSProp optimizer. During the training, there are always performed two updates of the encoder and the generator for each update of the discriminator. This heuristic approach is employed to reach better training stability. The method reaches promising results, see Fig. 10.7.

In the second experiment, there is VAEGAN used as a cross-modal bridge for image-to-sketch translation. The architecture of the network is unchanged. As training data, it is used the preprocessed CUFSF dataset, whereas all the images are resized to the size of 64×64 pixels. The training setup is identical to the one in the first experiment, for the qualitative results, see Fig. 10.8.

Figure 10.8: Results of the image-to-sketch translation using VAEGAN.

It can be seen, the results are not good. The sketches are very blurred, and there apparently occurs mode collapse of the GAN. Moreover, the training is very unstable, even with the usage of heuristic training tricks. To overcome mode collapse, I employ Wasserstein loss instead of the standard adversarial. This leads to slightly better results, however, the network is still unable to produce sharp images with preserved facial features.

In the last experiment, I train VAEGAN to sketch-to-image translation, which is arguably a harder task. The network is unable to converge to the meaningful results and generate only noise. To conclude all the listed experiments, reached results are not good enough to be successfully used as a part of the heterogeneous FR pipeline.

10.2.6 Pix2pix

Pix2pix [83] is a method designed for image-to-image translation task, for its pipeline see Fig. 10.9. As far as I know, it is the best existing supervised method for such a task. Originally, it was trained on the Cityscapes dataset and CMP Facades dataset. Both datasets have a very similar concept - both contain pairs consist of RGB image and its semantic (per-pixel) segmentation. The method was later also tested on a day-night photo translation task and a pose transfer task. The method generally provides very good results for all of the above-mentioned tasks, therefore, it has the big potential for usage as a cross-modal bridge.

In my experiments, I test many modifications of the standard Pix2pix architecture, however, I reach best results with two following modifications: (1) Different size of Markovian discriminator - 16×16 in low-resolution experiments, 70×70 in high-resolution experiments - by this

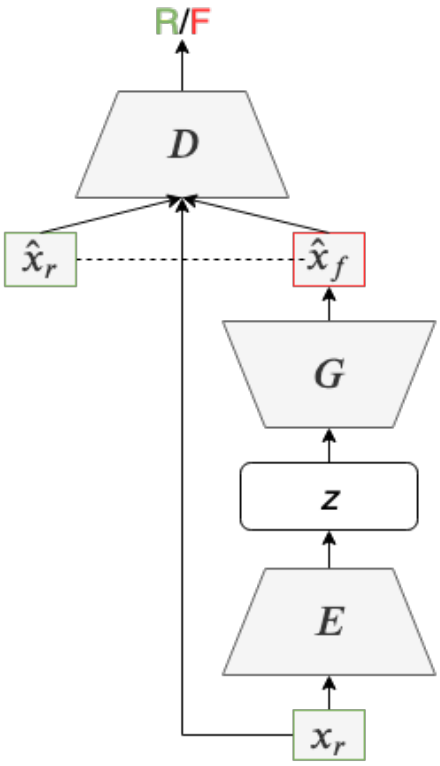


Figure 10.9: Standard Pix2pix pipeline. E = encoder, G = generator, D = discriminator, z = latent space. Dotted line indicates L_1 loss. $\boldsymbol{x}_r$ is real input from the first domain, $\hat{\boldsymbol{x}}_f$ is corresponding fake image from the second domain, $\hat{\boldsymbol{x}}_r$ is corresponding real image.

setting I get sharper results; (2) Instance normalization instead of the batch normalization - I observe better training stability and faster convergence while using instance normalization. Otherwise, the architecture is unchanged and can be found in the original paper. I would like to mention that I also test using L_2 distance instead of the original L_1 distance. While it leads to satisfactory results, the results using L_1 distance are sharper and preserve more facial details.

I perform four experiments in total to verify the sufficiency of the Pix2pix approach - two with low-resolution images (sketch-to-image translation, and image-to-sketch translation) and two corresponding ones with high-resolution images. As training data, I use the CUFSF dataset. During the experiments, all the images are resized to resolution of 64×64 pixels (low-resolution), 256×256 respectively (high-resolution). All models are trained during 150 epochs with mini-batch size 1 using Adam optimizer.

The method reaches very good results in all of the experiments, whereas, qualitatively speaking, results from sketch-to-image are slightly more realistic and precise in facial details, see Fig. 10.10. I argue, that for some cases, the translated sketches are even more realistic, precise and correspond better to the real image than the ground-truth data (see the second row of Fig. 10.10.)

The qualitative results from image-to-sketch translation are also promising, see Fig. 10.11. However, there arises a small problem with generalization, i.e., the system has problems to

Figure 10.10: Image-to-sketch translation using Pix2pix method. There are real images in the first column, generated corresponding sketches in the second column, and ground-truth sketches in the third column.

generate glasses, earnings, etc. This flaw stems from a very small amount of training data.

Figure 10.11: Sketch-to-image translation using Pix2pix method. There are original sketches in the first column, generated corresponding images in the second column, and ground-truth images in the third column.

To conclude, the Pix2pix approach provides very realistic and precise results. Its main disadvantages are the necessity of image pairs for the training, small problems with generalization stem from the lack of the data, and the necessity of two different networks (one for each direction of the translation) in the two-directional cross-modal bridge task.

10.2.7 UNIT

UNIT [85] is another method designed for image-to-image translation tasks. UNIT is one of the best unsupervised (i.e., does not need data pairs for the training) methods available. Its main advantage is the ability to use the same model for both image-to-sketch and sketch-to-image translation. It stems from shared-latent space assumption, i.e., it is assumed that two

corresponding images from two different domains can be mapped to the same latent code in a shared-latent space. Such an assumption also implies cycle consistency, i.e., the result of a translation of already translated image should be the original image. The method was tested on the Cityscapes dataset and also on a day-night photo translation task. For a reason UNIT provided very realistic results in both of these experiments, I decided to test it as a cross-modal bridge. It is worth to mention, the decision to use the UNIT method instead of MUNIT [86] is based on the fact that only translation between two different domains is necessary.

To verify the efficiency of the UNIT method, I train on the CUFSF dataset again. I use the unchanged architecture of UNIT from the original article. During the experiment, there is performed random crop around the center of an image, and this crop is resized to the resolution of 256×256 pixels. The method is trained during 150 epochs with mini-batch size 1 using Adam optimizer.

Qualitatively speaking, in the sketch-to-image translation experiment, UNIT reaches very good results, see Fig. 10.12. In comparison with Pix2pix, I argue, the UNIT method is better at generalization. On the other hand, Pix2pix provides more detailed outputs.

Figure 10.12: Sketch-to-Image translation using Unit method. There are real images in the first row, generated sketches in the second row.

Figure 10.13: Image-to-sketch translation using Unit method. There are original in the first row, generated images in the second row.

In the image-to-sketch translation experiment, UNIT also provides very good results, see Fig. 10.13. Comparing it with Pix2pix, the generalization is better again, however, UNIT is unable to learn to generate sharp images, and therefore there occurs lack of details. Another disadvantage is the inconsistency of results. It is not very obvious for sketch-to-image

translation, but it is much more visible for the reverse process, see flipped image pairs.

10.2.8 X-Bridge

X-Bridge is a novel method designed specifically as a cross-modal bridge in the heterogeneous face recognition task. Same as UNIT, X-Bridge assumes shared-latent space among two different domains. Same as Pix2pix, it is a supervised method, therefore, it needs data pairs for the training. In contrast with Pix2pix, it contains two main branches - translation, and reconstruction branch. Each of the branches contains its own generator and its own discriminator. Both branches share one latent space and one encoder. For the X-Bridge structure, see Fig. 9.1.

In experiments, I follow the same protocols as for previous methods, i.e., firstly is X-Bridge trained on the CUFSF dataset. During the experiments, all the images are resized to the resolution of 256×256 pixels, and the model is trained during 150 epochs with mini-batch size 1 using Adam optimizer. For the X-Bridge detail architecture, see Tab. 10.6.

Table 10.6: Structure of the X-Bridge architecture. Both generators have the same structure, whereas sharing parameters between Deconv2D-R and all Deconv2D-U layers. Both discriminators have the same structure, except the translation discriminator is conditional, whereas, reconstruction discriminator is not. If not say otherwise, all the convolutions and the deconvolutions have stride 2. Conv2D-L denotes 2D convolution, followed by Leaky ReLU. Conv2D-IL denotes 2D convolution, followed by instance normalization and Leaky ReLU. Conv2D-U denotes encoder's Conv2D-IL with an additional skip-connection between Conv2D-U and corresponding Deconv2D-U layers in the generators. Conv2D-R denotes convolution, followed by ReLU. Deconv2D-IR denotes deconvolution, followed by instance normalization and ReLU. Deconv2D-U denotes Deconv2D-IR with additional skip-connection input from the corresponding encoder's layer. Deconv2D-T denotes deconvolution, followed by the Tanh activation function. Conv2D-IR1 denotes 2D convolution with stride 1, followed by instance normalization and Leaky ReLU. Conv2D-1 denotes 2D convolution with stride 1.

Encoder	**Generator**	**Discriminator**
Conv2D-L(64, 4×4)	Deconv2D-R(512, 4×4)	Conv2D-L(64, 4×4)
Conv2D-IL(128, 4×4)	Deconv2D-U(512, 4×4)	Conv2D-IR(128, 4×4)
Conv2D-IL(256, 4×4)	Deconv2D-U(512, 4×4)	Conv2D-IR(256, 4×4)
Conv2D-IL(512, 4×4)	Deconv2D-U(512, 4×4)	Conv2D-IR1(512, 4×4)
Conv2D-U(512, 4×4)	Deconv2D-IR(256, 4×4)	Conv2D-1(1, 4×4)
Conv2D-U(512, 4×4)	Deconv2D-IR(128, 4×4)	
Conv2D-U(512, 4×4)	Deconv2D-IR(64, 4×4)	
Conv2D-C(512, 4×4)	Deconv2D-T(3, 4×4)	

For the results, see Fig. 10.14 and Fig. 10.15. It can be seen that X-Bridge reaches almost flawless results in reconstructing of the original images in both tested tasks. It should be noted that reconstruction results are usually a little bit brighter than the original image.

Figure 10.14: Image-to-sketch translation using X-Bridge. There are real images in the first column, reconstructed images in the second column, translated corresponding sketches in the third column, and ground-truth sketches in the fourth column.

Figure 10.15: Sketch-to-image translation using X-Bridge. There are original sketches in the first column, reconstructed sketches in the second column, translated corresponding images in the third column, and ground-truth images in the fourth column.

As for the image-to-sketch translation results, X-Bridge reaches very similar results as Pix2pix. I argue they are slightly better in terms of generalization. In the sketch-to-image task, for the most sketches, X-Bridge produces more detailed and precise results than Pix2pix. Moreover, I argue X-Bridge generalizes a little bit better than the original method, i.e., it has smaller problems with the translation of earrings, glasses, etc., see Fig. 10.16.

In the next experiment, I test X-Bridge's ability to handle real images taken in a semi-controlled environment, see Fig. 10.17. The photo is taken using a mobile camera, and the face is detected automatically using the Haar-Cascade face detector. The generated sketch preserves the pose of the subject almost flawlessly, the translation of hair is also very realistic and detailed. The facial features preservation is worse than for the images from the testing database, however, I argue the mutual resemblance is still satisfactory for purposes of the face recognition task.

Figure 10.16: Comparison of translating subject with glasses (original image on the left) using Pix2pix (the second one), UNIT (the third one) and X-Bridge (the last one). All methods correctly drew glasses into the sketch, however, Pix2pix was unable to correctly draw the right eye behind the glasses, and the sketch generated by UNIT is low quality overall, which suggest the glasses cause problems. On the other hand, X-Bridge was able to preserve sketch details quite well and generates .

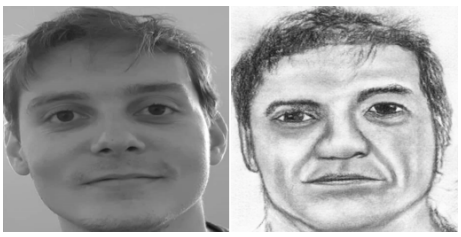


Figure 10.17: Image-to-sketch translation using X-Bridge. There is an original real image on the left and generated sketch on the right.

To test X-Bridge's ability to handle sketches drawn in a different style than the training sketches, I perform two quick experiments. In the first experiment, I utilize data from the CUHK dataset. The sketches are cropped and resized to a resolution of 426×372 pixels. The comparison of exemplary results generated by Pix2pix, Unit, and X-Bridge can be found in Fig. 10.18. I argue the decrease in the quality (the biggest one for Pix2pix) of the translated images is caused by the change of the drawing style. Especially the fact that faces on the sketches from the original dataset are much darker and shaded than the CUHK sketches seems problematic for the translation.

Figure 10.18: Sketch-to-Image translation using Pix2pix (the second image) UNIT (the third image) and X-Bridge (the fourth image). There is a sketch from CUHK dataset on the left, and the corresponding real image on the right.

In the second experiment, I utilize a sketch drawn by an amateur, see Fig. 10.19. Unfortunately, the obtained result is very low quality. The generated object remotely reminds the human face, however, the result is unusable for the face recognition task. I believe the amateur-drawn sketch is not enough contrasting, therefore, the X-Bridge's encoder is unable

to detect important facial features to encode them into the latent space. Also, the style of the sketch is very different from the original one.

Figure 10.19: Sketch-to-Image translation using X-Bridge. There is an amateur-drawn sketch on the right, translated image in the middle and the corresponding real image on the right.

To overcome such a problem can be utilized in X-Bridge's reconstruction path. The reconstruction generator reconstructs the amateur-drawn sketch in the style of the training sketches, and then a translated image is generated from this sketch, see Fig. 10.20. In comparison with the direct translation approach, the final result is much better in terms of the similarity between sketch and the translated image. However, the translated image still does not resemble the original subject. There are multiple reasons: (1) The original sketch is of bad quality overall. This directly affects the original-style sketch reconstruction; (2) The trained networks are not robust enough to the drastic change of the sketch style, because, in the training set, there is only one style. To obtain more accurate and robust results, it is necessary to enrich the training, which is a complicated task due to the lack of such data available.

Figure 10.20: Sketch-to-sketch reconstruction using X-Bridge. There is the amateur-drawn sketch on the right, reconstructed sketch on the middle, and translated image on the right.

In the last experiment, I test robustness in translation. As a testing data, I use the color-FERET dataset utilizing non-frontal images, for exemplary results, see Fig. 10.21. For both tested methods, a decrease in quality occurs in terms of detail preservation. Moreover, Pix2pix is unable to overcome the fact that the right ear of the subjects is occluded and try to model it in both cases. In the top row, Pix2pix believes the glasses are "weird ear", whereas, in the bottom row, it models at least small remnants of the right ear. On the other hand, UNIT and X-Bridge "understand" the fact ear is occluded and is not forced to model it.

In conclusion, I argue qualitative results provided by X-Bridge overcome other state-of-the-art methods in terms of similarity between translated and corresponding images, robustness, generalization capacity, and translated facial features preservation.

Figure 10.21: Comparison of translation of a facial photo in non-frontal pose. Original photo on the left, Pix2pix translation the second from left, UNIT translation the third from left, and X-Bridge translation on the right.

10.2.9 Quantitative results comparison

In this section are presented quantitative results and their comparison for all the tested methods. All tests follow the quantitative results testing protocol described in Subsection 10.2.3. For each of the tested methods, a graph for different distance thresholds with precision, recall, and F1 score while using ArcFace feature extractor is listed. Moreover, the optimal F1 score using my own feature extractor is calculated. The comparison of FFPS for all the methods can be found at the end of this subsection in Tab. 10.7.

First, all the statistics are calculated for the color-FERET dataset, see Fig. 10.22. ArcFace classifier reached F1 Score 0.75. I argue, such a low score is caused by the difficulty of profile images, which makes approximately 30% of the whole dataset. My classifier reached F1 Score = 0.80 for distance threshold = 0.49.

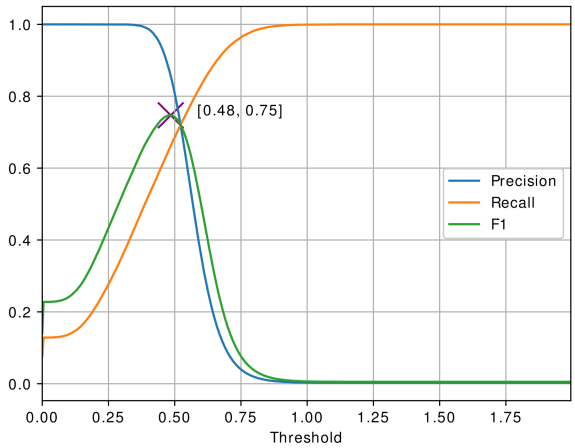


Figure 10.22: Precision, Recall, and F1 Score for color-FERET dataset. Optimal F1 Score is marked with purple cross.

Second, the Pix2pix method is tested, see Fig. 10.23. From the graph, it is obviously a big performance drop of the ArcFace classifier on the translated dataset, to be more specific, optimal F1 Score = 0.27 for distance threshold 0.32. I argue there are two main reasons for such a decrease of performance: (1) ArcFace classifier was trained to classify facial photos, not sketches. However, approximately the same drop can be expected for every transfer method; (2) Due to their complete omission in the training set, the Pix2pix method has very big problems to transfer images with non-frontal poses or with non-neutral expression. Moreover, even for the frontal face with a neutral expression, the result's quality is dramatically decreased by the presence of earrings, glasses, etc. While using my classifier optimal F1 Score = 0.31 for threshold = 0.35.

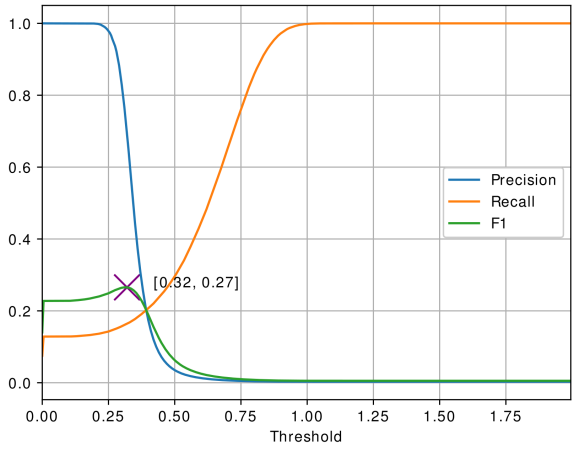


Figure 10.23: Precision, Recall, and F1 Score for dataset translated by Pix2pix method. Optimal F1 Score is marked with purple cross.

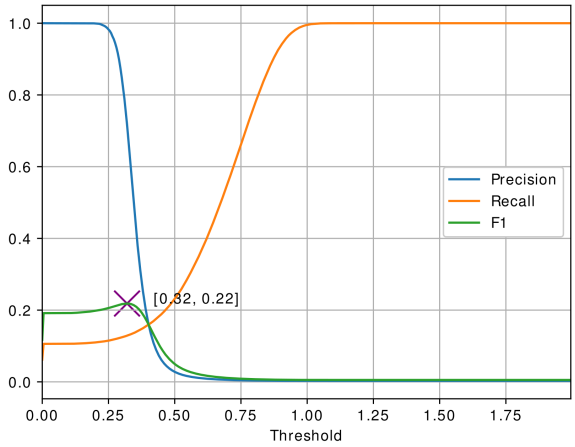


Figure 10.24: Precision, Recall, and F1 Score for dataset translated by UNIT method. Optimal F1 Score is marked with purple cross.

Third, I perform the quantitative test using the UNIT method, see Fig. 10.24. Unfortunately, UNIT reaches even worse results than Pix2pix. On the one hand, UNIT is more robust to pose changes and can generalize better than Pix2pix, on the other hand, the quality of translated

sketches is generally lower. I believe this flaw outweighs the advantages of UNIT, and it causes a decrease in the performance. While using my classifier optimal F1 Score = 0.24 for threshold = 0.33.

The last tested method is X-Bridge. I perform two experiments, the first is the same as with other methods - translation, and the second while using reconstruction branch of X-Bridge. In the first experiment, X-Bridge outperforms other methods by a large margin while it reaches F1 Score = 0.57 for threshold = 0.48, see Fig. 10.25. I believe the dramatical increase of performance compared to the other methods is caused by the combination of accurate translation of frontal and near-frontal images (Pix2pix level of accuracy) with good generalization and good robustness in pose and expression (UNIT level of generalization and robustness). While using my classifier optimal F1 Score = 0.60 for threshold = 0.48.

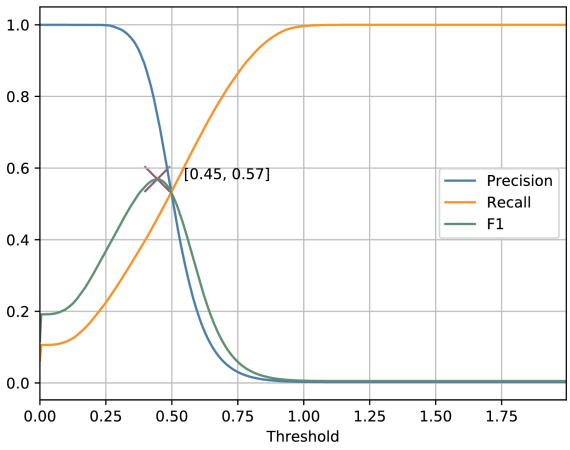


Figure 10.25: Precision, Recall, and F1 Score for dataset reconstruction using X-Bridge method. Optimal F1 Score is marked with purple cross.

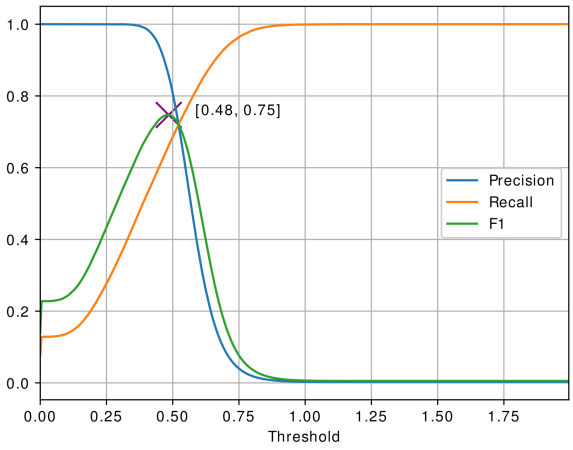


Figure 10.26: Precision, Recall, and F1 Score for dataset translated by X-Bridge method. Optimal F1 Score is marked with purple cross.

In the second experiment with X-Bridge, I calculated statistics over the reconstructed data.

Despite small reconstruction inaccuracies, it can be seen obtained statistics over the reconstructed data are identical with the statistics over the original one, see Fig. 10.26.

A comparison of the results can be found in Table 10.7, whereas the listed values of the F1 Score are while using my feature extractor.

Table 10.7: Comparison of the tested Cross-modal bridges.

Method	**F1 Score**	**FFPS**
Original	0.80	-
Pix2pix	0.31	0.39
UNIT	0.24	0.30
X-Bridge-T	0.60	0.75
X-Bridge-R	0.80	1.00

10.2.10 Discussion

In this section are described experiments with different cross-modal bridges and with the whole heterogeneous face recognition system. Both qualitative and quantitative results are provided for various testing settings.

My proposed method X-Bridge, provides superior qualitative results in all relevant areas. Moreover, X-Bridge reaches superior quantitative results and outperform other tested methods by a significant margin. I believe such huge success is caused by two main factors: (1) Method is based on supervised learning, which generally provides better results, if some training data are available; (2) The addition of the reconstruction path, which motivates the encoder to preserve and encode important facial features into the shared-latent space.

Chapter 11

Conclusion

This chapter is divided into three main parts. In the first part, there is a summary of the thesis, and conclusions are drawn. The second part is dedicated to the evaluation of the dissertation goals. And the last part quickly outlines my future work and possible improvements to the presented system.

11.1 Thesis summary

This thesis proposes a novel heterogeneous face recognition system based on a novel synthesis-based cross-modal bridge method named X-Bridge. In heterogeneous face recognition task, the system has to overcome differences between two recognize modalities using cross-modal bridge before traditional face recognition approaches can be utilized. Precisely for this task, I develop and present a novel method based on generative adversarial networks named X-Bridge. The main purpose of X-Bridge is to translate the input image from the first modality into the second modality, i.e., to generate the corresponding image from the second modality while preserving important facial features. In the first step, X-Bridge encodes the input image into the shared-latent space. In the second step, based on the obtained latent code, it is generated the translated image.

Facial feature extractor based on DenseNet is then applied to the translated image. DenseNet is trained on the Casia-WebFace dataset while using Arc loss to produce compact class clusters with a margin between them. Comparing it with the usage of traditional Softmax, Arc loss dramatically improves the separability of class clusters, especially in the open-set classification protocol. During the testing, features provided by DenseNet are compared with anchor features taken from the testing database and are classified according to their distance and calculated threshold.

Both parts of the system, cross-modal bridge, and feature extractor are compared with other state-of-the-art methods and reach superior results. Moreover, a novel metric named Facial Feature Preservation Score (FFPS) is presented. FFPS is designed to objectively measure the performance of the cross-modal bridge in the heterogeneous face recognition task.

11.2 Dissertation goals

In this section, the evaluation of the dissertation goals set in Section 1.4 is presented. Each goal definition is repeated and followed by the evaluation of my work on it and by the discussion of the results.

11.2.1 Face recognition methods

Modern face recognition approaches have three main attributes: (1) Training data; (2) Neural network architecture; and (3) Design of the loss function. This goal aims to analyze existing face recognition datasets, state-of-the-art methods, and available loss functions.

In this thesis can be found a quick review of face recognition datasets in Chapter 3. Based on the review, the suitable dataset for training and testing both main parts of the system, cross-modal bridge, and feature extractor, are chosen. The review of different neural network architectures and loss functions can be found in Chapters 4 and 5. The most important ones are tested, and the best ones are utilized in the final solution of the heterogeneous face recognition system.

This allows me to consider this dissertation goal as completed.

11.2.2 Cross-modal bridge comparison

Each heterogeneous face recognition system needs a cross-modal bridge part to overcome differences between two different modalities. This goal of the dissertation aims to analyze existing methods potentially usable as the cross-modal bridge. With the quick development of generative adversarial networks, they offer huge potential for synthesis-based cross-modal bridges. The quick review of generative adversarial networks can be found in Chapter 6.

The most promising ones are tested and compared. Moreover, a novel method named X-Bridge is proposed. X-Bridge addresses some of the problems of the existing methods and reaches state-of-the-art results. Both qualitative and quantitative results are provided for all the tested cross-modal bridges.

This allows me to consider this dissertation goal as completed.

11.2.3 Heterogeneous face recognition system

The traditional heterogeneous face recognition system is composed of two main parts: (1) Cross-modal bridge; and (2) Classifier. This goal aims to apply methods from previous sub-sections and combine them in a novel heterogeneous face recognition system while addressing some of their flaws.

In Chapter 9 is presented a novel heterogeneous face recognition system based on X-Bridge and DenseNet. X-Bridge overcomes other state-of-the-art methods in terms of similarity

between translated and corresponding images, robustness, generalization capacity, and facial features preservation. Moreover, the system using X-Bridge reaches superior results in comparison with systems using other cross-modal bridges.

This allows me to consider this dissertation goal as completed.

11.3 Future work

During the development of the heterogeneous face recognition system, I have discovered a few problems. First, I was unable to train a cross-modal bridge based on PG-GAN successfully, however, I believe this approach can be very promising for future research.

Second, X-Bridge has problems to translate facial images in profile poses. Third, X-Bridge also has problems with sketches drawn in different styles than the style of the training set. I propose three different solutions to address these problems in the future. First, enriching the training set with data with large pose variations and drawn in different styles. This can be problematic due to the fact that obtaining facial sketches in good quality is not a trivial task. Second, I propose to utilize the reinforcement learning approach, i.e., find the similar task of translation between two modalities with a big amount of data available, train the X-Bridge method on this task and then perform fine-tuning of the method with the image-sketch pair data. Third, the development of a better unsupervised method for image translation. This allows us to utilize unpair data, which is much easier to obtain.

The last problem is a significant performance drop of the X-Bridge method for images in the real world conditions. This problem was not addressed in this work because, in tasks of person database investigations, it is not expected the suspect persons would be described in such conditions. However, for example, in the task of surveillance, while using thermal-cameras, the ability to recognize a person in arbitrary conditions can be beneficial. For the reasons mentioned above, I would like to investigate the solutions to this problem more in my future work.

Bibliography

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[2] G. Kohonen, *Self-organization and Associative Memory*, Berlin, Germany, 1994.

[3] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.

[4] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan. 1991.

[5] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, pp. 37:1–37:42, Feb. 2015.

[6] Biometrics 101: Verification vs identification. [Online]. Available: http://www.eyeverify.com/blog/biometrics-101-verification-vs-identification

[7] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," *CoRR*, vol. abs/1704.08063, 2017.

[8] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[9] K. Hasan, M. S. Moalem, and C. Pal, "Localizing facial keypoints with global descriptor search, neighbour alignment and locally linear models," *Computer Vision Workshops (ICCVW), IEEE International Conference*, pp. 362–369, 2013.

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[11] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 529–534.

[12] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, "Multi-pie," in *Proceedings of The Eighth IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[13] 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*, June 2014, pp. 1701–1708.

[14] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations."

[15] Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, and X. Chen, "A benchmark and comparative study of video-based face recognition on cox face database."

[16] J. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, P. J. Flynn, and S. Cheng, "The challenge of face recognition from digital point-and-shoot cameras," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013, pp. 1–8.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[18] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," *Technical Report*, 2019.

[19] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[20] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1931–1939.

[21] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large scale face recognition," in *European Conference on Computer Vision*. Springer, 2016.

[23] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *CoRR*, vol. abs/1710.08092, 2017.

[24] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev, "Beyond frontal faces: Improving person recognition using multiple cues," 2015.

[25] S. Sengupta, J. Cheng, C. Castillo, V. Patel, R. Chellappa, and D. Jacobs, "Frontal to profile face verification in the wild," in *IEEE Conference on Applications of Computer Vision*, February 2016.

[26] A. Martinez and R. Benavente, "The ar face database," *CVC Technical Report 24*, 1998.

[27] 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.

[28] H. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa, "Memetic approach for matching sketches with digital face images," *IIITD-TR-2011-006*, 2011.

[29] S. Ouyang, T. M. Hospedales, Y. Song, and X. Li, "Forgetmenot: Memory-aware forensic facial sketch matching," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5571–5579.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[31] Cs231n convolution neural networks for visual recognition. [Online]. Available: http://cs231n.github.io/

[32] Neuronové sítě. [Online]. Available: http://www.kky.zcu.cz/cs/courses/neu

[33] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, 3rd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[35] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018.

[36] M. Lin, Q. Chen, and S. Yan, "Network in network," *International Conference on Learning Representation*, 2014.

[37] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[39] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *CoRR*, vol. abs/1602.07868, 2016.

[40] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[41] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[43] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

[44] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-24, Mar 2010.

[45] Overview of mini-batch gradient descent. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[47] T. Dozat, "Incorporating nesterov momentum into adam," 2015.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1106–1114.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[51] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. arXiv preprint arXiv:1312.4400, 2013.

[52] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015.

[53] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016.

[54] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *CoRR*, vol. abs/1505.00387, 2015.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[56] ——, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016.

[57] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016.

[58] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[59] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," *CoRR*, vol. abs/1610.02915, 2016.

[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017.

[61] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *The 2nd International Conference on Learning Representations*, 2013.

[62] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, June 2006, pp. 1735–1742.

[63] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *CoRR*, vol. abs/1406.4773, 2014.

[64] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[65] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 499–515.

[66] Y. Liu, H. Li, and X. Wang, "Learning deep features via congenerous cosine loss for person recognition," *CoRR*, vol. abs/1702.06890, 2017.

[67] ——, "Rethinking feature discrimination and polymerization for large-scale recognition," *CoRR*, vol. abs/1710.00870, 2017.

[68] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CoRR*, vol. abs/1801.07698, 2018.

[69] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *CoRR*, vol. abs/1703.09507, 2017.

[70] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L_2 hypersphere embedding for face verification," *CoRR*, vol. abs/1704.06369, 2017.

[71] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," *CoRR*, vol. abs/1801.09414, 2018.

[72] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *CoRR*, vol. abs/1801.05599, 2018.

[73] A beginner's guide to generative adversarial networks (gans). [Online]. Available: https://skymind.com/wiki/generative-adversarial-network-gan

[74] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

[75] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, pp. 5769–5779.

[76] A. B. L. Larsen, S. K. Sønderby, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *CoRR*, vol. abs/1512.09300, 2015.

[77] I. Gruber, "Generating facial images using vaegan," in *Studentská vědecká konference ZČU-FAV*, 2018, pp. 38–39.

[78] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[79] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1283–1292.

[80] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, "Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 821–830.

[81] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *CoRR*, vol. abs/1710.10196, 2017.

[82] (2018) Generating custom photo-realistic faces using ai. [Online]. Available: https://blog.insightdatascience.com/generating-custom-photo-realistic-faces-using-ai-d170b1b59255

[83] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[84] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[85] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *CoRR*, vol. abs/1703.00848, 2017.

[86] X. Huang, M. Liu, S. J. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," *CoRR*, vol. abs/1804.04732, 2018.

[87] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.

[88] X. Tan, S. Chen, Z. H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.

[89] "A Survey of Recent Advances in Face Detection," *Learning*, no. June, p. 17, 2010.

[90] N. Wang, X. Gao, D. Tao, and X. Li, "Facial Feature Point Detection: A Comprehensive Survey," 2014.

[91] B. S. Manjunath, R. Chellappa, and C. von der Malsburg, "A feature based approach to face recognition," in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1992, pp. 373–378.

[92] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, Oct. 1996.

[93] L. Wiskott, N. Krüger, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, July 1997.

[94] P. Campadelli and R. Lanzarotti, *A Face Recognition System Based on Local Feature Characterization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 147–152.

[95] S. Biswas, G. Aggarwal, and P. J. Flynn, "Pose-robust recognition of low-resolution face images," in *CVPR 2011*, June 2011, pp. 601–608.

[96] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[97] H. Shin, S.-D. Kim, and H.-C. Choi, "Generalized elastic graph matching for face recognition," *Pattern Recognition Letters*, vol. 28, no. 9, pp. 1077 – 1082, 2007.

[98] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using hog–ebgm," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537 – 1543, 2008.

[99] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[100] B. Kepenekci, F. B. Tek, and G. B. Akar, "Occluded face recognition based on gabor wavelets," in *Proceedings. International Conference on Image Processing*, vol. 1, 2002, pp. I–293–I–296 vol.1.

[101] L. Lenc, "Face recognition under real-world conditions," Doctoral Thesis, University of West Bohemia, Faculty of Applied Sciences, Department of Computer Science and Engineering, 2014.

[102] Y. Gao and Y. Qi, "Robust visual similarity retrieval in single model face databases," *Pattern Recognition*, vol. 38, no. 7, pp. 1009 – 1020, 2005.

[103] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul 2002.

[104] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, *Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 61–70.

[105] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 3025–3032.

[106] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *CoRR*, vol. abs/1401.5311, 2014.

[107] A. Li, S. Shan, X. Chen, and W. Gao, "Maximizing intra-individual correlations for face recognition across pose differences," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 605–611.

[108] D. Yi, Z. Lei, and S. Z. Li, "Towards pose robust face recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3539–3545.

[109] T. D. Alter, "3d pose from three corresponding points under weak-perspective projection," Cambridge, MA, USA, Tech. Rep., 1992.

[110] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with som and soft k-nn ensemble," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 875–886, July 2005.

[111] H. S. Le and H. Li, "Recognizing frontal face images using hidden markov models with one training image per person," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, Aug 2004, pp. 318–321 Vol.1.

[112] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 4, no. 3, pp. 519–524, March 1987.

[113] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, September 1936.

[114] A. M. Martínez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 748–763, Jun. 2002.

[115] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3499–3506.

[116] J. Wright and G. Hua, "Implicit elastic matching with random projections for pose-variant face recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 1502–1509.

[117] T. Kanade and A. Yamada, "Multi-sub region based probabilistic approach toward pose-invariant face recognition," in *IEEE International Symposium on Cumputational Inteligence in Robotics and Automaticon (CIRA)*, July 2003, pp. 954–959.

[118] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 12 1966.

[119] F. Samaria, "Face segmantation for identification using hidden markov models," in *British Machine Vision Cinference*, 1993, pp. 399–408.

[120] S. Rahimzadeh Arashloo and J. Kittler, "Energy normalization for pose-invariant face recognition based on mrf model image matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1274–1280, Jun. 2011.

[121] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 84–91.

[122] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2707–2714.

[123] T. Ahonen, A. Hadid, and M. Pietikäinen, *Face Recognition with Local Binary Patterns*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 469–481.

[124] K. Simonyan, O. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," *Procedings of the British Machine Vision Conference 2013*, pp. 1–11, 2013.

[125] R. J. Baron, "Mechanisms of human facial recognition," *International Journal of Man-Machine Studies*, vol. 15, no. 2, pp. 137 – 178, 1981.

[126] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[127] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[128] S. J. D. Prince, J. H. Elder, J. Warrell, and F. M. Felisberti, "Tied factor analysis for face recognition across large pose differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 970–984, Jun. 2006.

[129] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Regularized latent least square regression for cross pose face recognition," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 1247–1253.

[130] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[131] B. Moghaddam, C. Nastar, and A. Pentland, "A bayesian similarity measure for direct image matching," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 2, Aug 1996, pp. 350–358 vol.2.

[132] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[133] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 585–591.

[134] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003.

[135] X. He, S. Yan, Y. Hu, and H.-J. Zhang, "Learning a locality preserving subspace for visual recognition," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 385–392 vol.1.

[136] J. Zhang, S. Z. Li, and J. Wang, "Nearest manifold approach for face recognition," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, May 2004, pp. 223–228.

[137] Y. Wu, K. L. Chan, and L. Wang, "Face recognition based on discriminative manifold learning," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 4, Aug 2004, pp. 171–174 Vol.4.

[138] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, March 2005.

[139] S. Yan, Y. Hu, D. Xu, H. J. Zhang, B. Zhang, and Q. Cheng, "Nonlinear discriminant analysis on embedded manifold," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 468–477, April 2007.

[140] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[141] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1450–1464, Nov 2002.

[142] C. Lu and X. Tang, "Surpassing human-level face verification performance on lfw with gaussian face," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 3811–3819.

[143] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (lbp)-based face recognition," in *Proceedings of the 5th Chinese Conference on Advances in Biometric Person Authentication*, ser. SINOBIOMETRICS'04. Berlin, Heidelberg: Springer-Verlag, 2004, pp. 179–186.

[144] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proceedings of the Fourteenth International Conference on Machine Learning*, ser. ICML '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 211–218.

[145] K. S. Kumar, V. B. Semwal, and R. C. Tripathi, "Real time face recognition using adaboost improved fast PCA algorithm," *CoRR*, vol. abs/1108.1353, 2011.

[146] J. J. Weng, N. Ahuja, and T. S. Huang, "Learning recognition and segmentation of 3-d objects from 2-d images," in *1993 (4th) International Conference on Computer Vision*, May 1993, pp. 121–128.

[147] A. Eleyan and H. Demirel, *Face Recognition System Based on PCA and Feedforward Neural Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 935–942.

[148] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *CoRR*, vol. abs/1406.4773, pp. 1–9, 2014.

[149] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 566–579.

[150] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *CoRR*, vol. abs/1412.1265, pp. 2892–2900, 2014.

[151] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Jun. 2009.

[152] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, "Pose-aware face recognition in the wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4838–4846.

[153] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[154] ——, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, Sept 2003.

[155] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sept 2009, pp. 296–301.

[156] B. Amberg, "Optimal step nonrigid icp algorithms for surface registration," in *In CVPR'07*, 2007.

[157] U. Prabhu, J. Heo, and M. Savvides, "Unconstrained pose-invariant face recognition using 3d generic elastic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1952–1961, Oct 2011.

[158] I. Masi, G. Lisanti, A. D. Bagdanov, P. Pala, and A. D. Bimbo, "Using 3d models to recognize 2d faces in the wild," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2013, pp. 775–780.

[159] H. Wang, Y. Wang, and Y. Cao, "Video-based face recognition: A survey," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 3, no. 12, pp. 2809 – 2818, 2009.

[160] S. Ouyang, T. Hospedales, Y.-Z. Song, X. Li, C. C. Loy, and X. Wang, "A survey on heterogeneous face recognition," *Image Vision Comput.*, vol. 56, no. C, pp. 28–48, Dec. 2016.

[161] B. Klare and A. K. Jain, "Sketch to photo matching: A feature-based approach," 2010.

[162] H. Kiani Galoogahi and T. Sim, "Face sketch recognition by local radon binary pattern: Lrbp," in *2012 19th IEEE International Conference on Image Processing*, Sep. 2012, pp. 1837–1840.

[163] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *CVPR 2011*, June 2011, pp. 513–520.

[164] B. Klare, Z. Li, and A. K. Jain, "Matching forensic sketches to mug shot photos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 639–646, March 2011.

[165] Xiaoou Tang and Xiaogang Wang, "Face photo recognition using sketch," in *Proceedings. International Conference on Image Processing*, vol. 1, Sep. 2002, pp. I–I.

[166] 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)*, vol. 1, June 2005, pp. 1005–1010 vol. 1.

[167] J. Zhong, X. Gao, and C. Tian, "Face sketch synthesis using e-hmm and selective ensemble," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, April 2007, pp. I–485–I–488.

[168] B. Xiao, X. Gao, D. Tao, and X. Li, "A new approach for face recognition by sketches in photos," *Signal Process.*, vol. 89, no. 8, pp. 1576–1588, Aug. 2009.

[169] D. Lin and X. Tang, "Inter-modality face recognition," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 13–26.

[170] A. Sharma and D. W. Jacobs, "Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch," in *CVPR 2011*, June 2011, pp. 593–600.

[171] G. G. Gordon, "Face recognition based on depth maps and surface curvature," in *SPIE Geometric methods in Computer Vision*, 1991, pp. 234–247.

[172] H. T. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation. principal directions for curved object recognition," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr 1998, pp. 372–377.

[173] B. Amberg, R. Knothe, and T. Vetter, "Expression invariant 3d face recognition with a morphable model," in *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, Sept 2008, pp. 1–6.

[174] J. Cook, V. Chandran, S. Sridharan, and C. Fookes, "Face recognition from 3d data using iterative closest point algorithm and gaussian mixture models," in *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, Sept 2004, pp. 502–509.

[175] C.-S. Chua, F. Han, and Y.-K. Ho, "3d human face recognition using point signature," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 233–238.

[176] B. Achermann and H. Bunke, "Classifying range images of human faces with hausdorff distance," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, 2000, pp. 809–813 vol.2.

[177] S. Lv, F. Da, and X. Deng, "A 3d face recognition method using region-based extended local binary pattern," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 3635–3639.

[178] K. I. Chang, K. W. Bowyer, and P. J. Flynn, "Face recognition using 2d and 3d facial data," in *ACM Workshop on Multimodal User Authentication*, 2003, pp. 25–32.

[179] T. Papatheodorou and D. Rueckert, "Evaluation of automatic 4d face recognition using surface and texture registration," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, May 2004, pp. 321–326.

[180] F. Tsalakanidou, D. Tzovaras, and M. G. Strintzis, "Use of depth and colour eigenfaces for face recognition," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1427–1435, Jun. 2003.

[181] G.-S. J. Hsu, Y.-L. Liu, H.-C. Peng, and P.-X. Wu, "Rgb-d-based face reconstruction and recognition," *Trans. Info. For. Sec.*, vol. 9, no. 12, pp. 2110–2118, Dec. 2014.

[182] W. Yang, D. Yi, Z. Lei, J. Sang, and S. Z. Li, "2d - 3d face matching using cca," in *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, Sept 2008, pp. 1–6.

[183] D. Huang, M. Ardabilian, Y. Wang, and L. Chen, "Asymmetric 3d/2d face recognition based on lbp facial representation and canonical correlation analysis," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 3325–3328.

[184] G. Toderici, G. Passalis, S. Zafeiriou, G. Tzimiropoulos, M. Petrou, T. Theoharis, and I. A. Kakadiaris, "Bidirectional relighting for 3d-aided 2d face recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2721–2728.

[185] O. Nikisins, K. Nasrollahi, M. Greitans, and T. B. Moeslund, "Rgb-d-t based face recognition," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 1716–1721.

[186] D. A. Socolinsky, L. B. Wolff, J. D. Neuheisel, and C. K. Eveland, "Illumination invariant face recognition using thermal infrared imagery," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–527–I–534 vol.1.

[187] X. Chen, P. J. Flynn, and K. W. Bowyer, "Visible-light and infrared face recognition," in *in: Proceedings of ACM Workshop on Multimodal User Authentication*, 2003, pp. 48–55.

[188] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *CoRR*, vol. abs/1606.03498, 2016.

[189] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[190] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[191] I. Gruber, M. Hlaváč, M. Železný, and A. Karpov, "Facing face recognition with resnet: Round one," in *Interactive Collaborative Robotics*, A. Ronzhin, G. Rigoll, and R. Meshcheryakov, Eds. Cham: Springer International Publishing, 2017, pp. 67–74.

Authored and Co-authored Works

Ivan Gruber. Rozpoznávání lidské tváře využitím hloubkových dat a 3D modelu. Master Thesis. 2013.

Ivan Gruber. Vliv polohy na nalezení 3D modelu lidské tváře. SVK FAV. 2014.

Ivan Gruber. Detekce významných bodů na lidské tváři pomocí neuronové sítě. SVK FAV. 2015.

Ivan Gruber, Miroslav Hlaváč, Marek Hrúz, Miloš Železný, and Alexey Karpov. An analysis of visual faces datasets. In *International Conference on Interactive Collaborative Robotics*, pages 18–26. Springer, Budapest, 2016.

Ivan Gruber. Detekce klíčových bodů pomocí konvoluční neuronové sítě. SVK FAV. 2016.

Ivan Gruber, Miroslav Hlaváč, Miloš Železný, and Alexey Karpov. Facing face recognition with resnet: Round one. In *International Conference on Interactive Collaborative Robotics*, pages 67–74. Springer, London, 2017.

Miroslav Hlaváč, Ivan Gruber, Miloš Železný, and Alexey Karpov. Semi-automatic facial key-point dataset creation. In *International Conference on Speech and Computer*, pages 662–668. Springer, London, 2017.

Ivan Gruber. Shooting target detection using particle filters. SVK FAV. 2017.

Ivan Gruber, Dmitry Ryumin, Marek Hrúz, and Alexey Karpov. Sign language numeral gestures recognition using convolutional neural network In *International Conference on Speech and Computer*, pages 68–75. Springer, Leipzig, 2018.

Miroslav Hlaváč, Ivan Gruber, Miloš Železný, and Alexey Karpov. LipsID using 3D convolutional neural networks. In *International Conference on Speech and Computer*, pages 209–214. Springer, Leipzig, 2018.

Ivan Gruber. ResNet vs DenseNet: Comparison of the State-of-the-Art Architectures for Face Classication. ITMO Student Conference. 2018.

Ivan Gruber. Generating Facial Images using VAEGAN. SVK FAV. 2018.

Ivan Gruber, Miroslav Hlaváč, Marek Hrúz and Miloš Železný. Semantic Segmentation of Historical Documents via Fully-Convolutional Neural Network. In *International Conference on Speech and Computer*. Springer, Istanbul, 2019.

Marek Hrúz, Petr Salajka, Ivan Gruber and Miroslav Hlaváč. Identity Extraction From Clusters of Multi-modal Observations. In *International Conference on Speech and Computer*. Springer, Istanbul, 2019.

Ivan Gruber. Quick comparison of state-of-the-art architectures for face classification. SVK FAV. 2019.