Face Recognition Aspects with DNNs

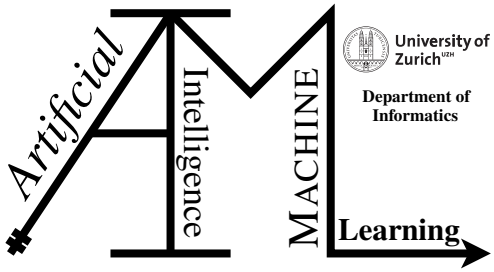
An Experimental and Reproducible Research Survey

Bachelor Thesis

Dominic Schmidli
17-710-807

Submitted on
June 1 2021

Thesis Supervisor
Prof. Dr. Manuel Günther



Bachelor Thesis

Author: Dominic Schmidli, dominic.schmidli@uzh.ch

Project period: 01.12.2020 - 01.06.2020

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

In the process of writing this thesis, I received a great deal of assistance and support. First, I would like to thank my supervisor Prof. Dr. Manuel Günther for formulating the research question. His highly competent supervision and guidance made me think and helped this thesis to shine. I would like to acknowledge the Idiap Research Institute for providing their resources. Many thanks to my supervisor Prof. Dr. Sébastien Marcel from the Idiap Research Institute for temporarily welcoming me onto his team. Furthermore, I thank Dr. Tiago de Freitas Pereira, who patiently introduced me to the Idiap system and was always available to offer a helping hand. Finally, I would like to thank everyone in my family for the correction suggestions and the mental support.

Abstract

Face recognition has become an indispensable part of life in today's world. In addition to unlocking mobile devices, it can also be used in public safety applications. Earlier, face recognition was accomplished using traditional algorithms. Today, face recognition has been dominated by deep learning, and deep convolutional neural networks can achieve impressive results. Unfortunately, these results can rarely be reproduced due to missing experimental details. The goal of this work is to compare state-of-the-art deep neural networks with respect to different aspects of face variations. For this purpose, four open-source networks from ArcFace and one from VGGFace2 were used. Experiments are performed on different databases to evaluate the influence of face variations. The results show that deep learning methods clearly outperform traditional face recognition algorithms, and the training database plays a crucial role in their performance. Most of the networks can handle occlusion and illumination well, but poses and facial expressions may still cause problems. Finally, recognizing faces at longer distances requires further improvement.

Zusammenfassung

Gesichtserkennung ist aus dem Alltag nicht mehr wegzudenken. Neben dem Entsperren von mobilen Geräten kommt sie auch bei der öffentlichen Sicherheit zum Einsatz. Früher wurde Gesichtserkennung noch mit traditionellen Algorithmen gemacht. Heute wird die Gesichtserkennung von Deep-Learning dominiert, wobei Deep Convolutional Neural Networks beeindruckende Ergebnisse erreichen. Leider können diese Resultate, aufgrund fehlender Angaben, selten reproduziert werden. Das Ziel dieser Arbeit ist es, moderne Deep Neural Networks in Bezug auf verschiede Aspekte von Gesichtsvariationen zu vergleichen. Dafür werden vier Netzwerke von ArcFace und eines von VGGFace2 benutzt. Es werden Experimente auf verschiedenen Datenbank durchgeführt, um den Einfluss von Gesichtsvariationen zu evaluieren. Die Resultate zeigen, dass Deep-Learning-Methoden die traditionellen Gesichtserkennungs-Algorithmen übertreffen. Ein entscheidender Faktor für die Performance stellt die Trainingsdatenbank dar. Mit teilweisen Gesichtsbedeckungen und unterschiedlichen Beleuchtungen können die meisten Netzwerke gut umgehen. Verschiedene Posen und Gesichtsausdrücke bereiten immernoch Probleme. Das Erkennen von Gesichtern auf grössere Distanzen funktioniert weiterhin schlecht.

Contents

Chapter 1

Introduction

Biometric recognition has attracted much attention in the past years. Commonly used examples of biometric recognition include methods of recognizing one's face, iris, voice, ear, palm print, gait or signature (Minaee et al., 2021). Face recognition is one of the most popular forms of biometric recognition and its development has made great progress in the last decade. Furthermore, its field of application is very versatile, as almost every mobile device, including laptops and smartphones, now offers the possibility to unlock its screen through face recognition. Moreover, the automatic grouping of images based on their subjects' identities in the gallery is a natural software component of such devices. Another popular field of application is that of video surveillance and security cameras (Masi et al., 2018). In this application, face recognition can help to identify criminals or find missing persons. In these and many other fields, the need for robust facial recognition systems has increased year over year (Guo and Zhang, 2019). Face recognition has achieved near human performance in a controlled and constrained environment for years. In some cases in which security is patrolled, as automatic border controls, frontal faces, and good lighting are enforced. However, such an environment with perfectly illuminated images of frontal and neutral faces in front of a white background can not always be found. Especially outdoors, the illumination from the sun is often not ideal for capturing faces. In such an uncontrolled environment, faces may feature different expressions and people may not even look into the camera. Furthermore, subjects may wear hats or glasses, or part of the face might even be occluded. Additionally, the quality and size of the image can vary greatly (Wang and Deng, 2021). All these scenarios can seriously interfere with the performance of face recognition.

Before the era of deep learning, traditional algorithms, such as local binary patterns (Heusch et al., 2006) or Gabor graphs (Günther et al., 2012), were used. These methods mostly used hand crafted features to describe faces. The disadvantage of these features, however, was that they were often not compact enough and could not be separated. Thus, one could not always rely on the algorithms. In particular, when conditions for capturing faces were not optimal, such as when there were different facial expressions, lighting conditions or poses, the performance of hand crafted features dropped significantly. All this changed with the development of artificial neural networks, especially deep neural networks. Due to lack of hardware, they could not be used for a long time. Only with the introduction of the Compute Unified Device Architecture (CUDA), which has made GPUs more easily accessible, can the data-intensive training of deep networks be afforded (Minaee et al., 2021). These new methods use multilayer constructs to extract features from images. They can be trained on a large data-set by minimizing a loss function in order to learn complex abstractions of faces (Wang and Deng, 2021). The most widely used network for face recognition is a convolutional neural network.

Deep learning made a breakthrough in 2012 when AlexNet (Krizhevsky et al., 2012) outperformed all other state-of-the-art algorithms by a wide margin (Wang and Deng, 2021). Since then, face recognition has been dominated by neural networks, and their state-of-the-art performance

has steadily improved. At first, researchers focused on creating deeper neural networks with the support of advanced network architectures. In recent years, the focus has been on creating more powerful loss functions. The main geometric idea is to separate the features such that the features of one identity are as close to each other as possible and have a large distance from all other identities. One of the methods that has demonstrated a state-of-the art performance is ArcFace (Deng et al., 2019), which uses an additive angular margin loss function to separate the features. Another main area of research is the creation of databases. These are not only important for training, but also essential for performance comparison. Many papers and several surveys show the performance of different deep neural networks in various challenging environments. The methods achieve impressive results on the LFW (Huang et al., 2007) and the IJB-C (Maze et al., 2018) database. However, they are not yet on the same level with human performance (Masi et al., 2018). Unfortunately, these results are hardly reproducible due to the lack of information about the experiments and the use of non-public databases. Under such different circumstances, it is almost impossible to compare the performance of a particular network with other networks or to build on previous research.

The present work aims to compare the performance of state-of-the-art deep neural networks in different challenging face recognition environments. To achieve this, four networks from ArcFace (Deng et al., 2019) and the VGGFace2 (Cao et al., 2018) are examined in more detail. ArcFace and VGGFace2 both belong to the current state-of-the-art in face recognition. Experiments are performed on eight different databases, such as AR face (Martínez and Benavente, 1998), Multi-PIE (Gross et al., 2010), CAS-PEAL (Gao et al., 2008), MOBIO (McCool et al., 2012), SCface (Grgic et al., 2011), GBU (Phillips et al., 2011), FRGC (Phillips et al., 2005) and LFW (Huang et al., 2007). The database protocol implementations provided by Günther et al. (2016) and Günther et al. (2017) allow the consideration of different aspects of face variations. Thus, the impact of individual influences on network performance can be observed in isolation. For example, the effect of different types of occlusion, facial expressions, or poses can be evaluated individually. Most of the existing papers evaluate the performance of face recognition algorithms on large benchmark databases, such as IJB-C (Maze et al., 2018). Therefore, an isolated evaluation of single challenging conditions on the performance of the mentioned networks has not been conducted. Thus, the present work attempts to fill this identified gap in the research. The performed experiments makes use of the software Bob (Günther et al., 2012; Anjos et al., 2012, 2017), which is an open-source software. In the present study, all the necessary information for conducting this experiments is explained, so that the traceability and reproducibility of the results can be ensured.

This work builds on the papers of Günther et al. (2016) and Günther et al. (2017), in which the authors evaluated the performance of several traditional algorithms on different databases and thereby considered individual challenging face recognition conditions in isolation. In doing so, they developed the database protocol implementations that are also relevant for the present work. They also used the open-source software Bob (Anjos et al., 2012) to perform their experiments. It was found that strong occlusion has a significant impact on performance. Furthermore, in environments with different poses, especially with faces turned away from the camera, the traditional algorithms almost completely failed. However, background variations caused only few difficulties (Günther et al., 2017). In summary, traditional algorithms at that time performed far worse than humans in unconstrained face recognition environments.

According to various surveys, deep learning-based methods have the potential to become more robust against such influences, which is also confirmed in the experiments conducted for this thesis. The results show that different types of illumination and occlusion do not constitute a major issue for deep neural networks. In addition, different facial expressions and ages do not affect their performance nearly as much as they affect that of traditional algorithms. The tested networks were mostly able to recognize poses from -45 to 45 degrees without any problems. As Günther et al. (2016) was already able to observe, the correct preprocessing has a decisive

influence on the performance of face recognition. It can further be reported that different network architectures and loss functions can improve performance, and the training database used plays a crucial role. All the experiments conducted in this thesis used the same databases, protocols and evaluation as Günther et al. (2016) and Günther et al. (2017) to better compare the performance of modern deep learning algorithms with traditional methods.

This thesis is organized as follows. In Chapter 2, an overview of deep learning-based face recognition methods and commonly used databases for training and performance evaluation is provided. In addition, a brief review of the papers of Günther et al. (2016) and Günther et al. (2017) is given. Chapter 3 presents the deep neural networks used in this study. Next, Chapter 4 introduces the experimental setup and its implementation in the software Bob. Furthermore, necessary information for the reproducibility of the experiments is given. Chapter 5 reveals the results of the experiments, which are subsequently discussed in Chapter 6. Finally, the thesis closes with a short conclusion in Chapter 7.

Chapter 2

Related Work

This chapter provides an overview of the current state of research regarding face recognition and deep learning. First, the common databases that have been used are described. Other than a few popular databases, the focus is placed on the databases used for the experiments in this thesis. The next section gives a brief summary of Günther et al. (2016) and Günther et al. (2017), on whose research this thesis is based. Finally, the state of the art in deep learning for face recognition is presented.

2.1 Databases

A major research interest in the area of deep learning lies in the development of new databases. There are a large number of databases that differ greatly in the number of images and identities, as well as in the diversity of the images. This section gives an overview of some common databases for face recognition tasks and training deep neural networks. Most of them are used in the experiments described in Chapter 5. The section is divided into databases for performance evaluation and databases for training.

2.1.1 Databases Used for Training

The VGGFace (Parkhi et al., 2015) database consists of over 2.6 million images from 2622 different celebrities. About five percent of these images are profiles, and the rest are frontal images. The founders of VGGFace believed that the availability of a large amount of training data was a critical factor for the success of neural networks. However, there was a lack of large, publicly available databases for the community. Thus, progress in this area has primarily been reserved for Internet giants such as Facebook and Google. The motivation behind VGGFace was to create a reasonably large data-set with limited human effort. The researchers developed a five-step guide to compile a large dataset. They applied these instructions to images of the Internet Movie Data Base celebrity list (Parkhi et al., 2015).

The VGGFace2 (Cao et al., 2018) database consists about three million images of 9131 identities. The images vary in pose, background, age and illumination. More information about this database can be found in Section 3.1.

MS-Celeb-1M (MS1M) from Guo et al. (2016) is a database with ten million images from celebrities from the Internet. It consists of 100000 identities in total with about 100 images per identity. A variety of professions are represented, such as politicians, actors, writers and singers. There is also a wide range of nationalities represented, the majority of which are Americans. Over three quarters of the images are female faces (Guo et al., 2016). Unfortunately, the database is no

longer available for download. Deng et al. (2019) created their own version of MS1M to train their networks. They removed inaccurate labels from the database to create a refined version of the database, which they called MS1MV2.

CASIA-WebFace (Yi et al., 2014) is a database that is also often used for face verification and face identification. It features about half a million images from 10000 identities. The images were collected from celebrities of various years of birth. It should be noted that this database was not used for the experiments in this study.

2.1.2 Databases Used for Evaluation

AR face (Martínez and Benavente, 1998) is an older database, that is still used today. It contains about 3312 images taken of 76 males and 60 females. The images vary in facial expressions, illumination, and occlusion in the form of scarves and sunglasses.

Multi-PIE (Gross et al., 2010) is another database that has been used for performance comparison. It contains about 755370 images shot in four sessions from 337 different subjects. The abbreviation PIE stands for pose, illumination, and expression and therefore the faces vary in poses, facial expressions, and illumination. They were taken at 15 view points and in 19 different lighting conditions.

The CAS-PEAL (Gao et al., 2008) database consists of 99594 large-scale images of 1040 Chinese faces. 595 of the individuals are males, and 445 are females. Unlike other databases that are mainly composed of Caucasian people, this database consists only of Chinese people. The abbreviation PEAL stands for position, expression, aging, and lighting. Overall, the images include variations in pose, expression, accessory, lighting, background, distance, and time. The founders used 15 lamps over five azimuths and three elevations to simulate different lighting conditions. To test the influence of accessories, the participants were equipped with a variety of glasses and hats. The provided expressions were neutral, smiling, frowning, surprised, faces with closed eyes, and faces where the mouth is wide open. Subjects were photographed at three different distances to the camera. Different, plain blankets were used to provide variations in the background. To record different times, the participants were recorded in two separate sessions, half a year apart. For pose variation, images were taken from nine different angles from -90 to 90 degrees. Furthermore, there were also some images taken in mixed variations, such as when different expressions across multiple poses were captured (Gao et al., 2008).

The surveillance cameras face (SCface) database (Grgic et al., 2011) contains 4160 images from 130 subjects taken by five video surveillance cameras of different qualities. The authors were motivated by law enforcement person identification. Therefore, the cameras were installed slightly above the head position of a human being, just as it would be in reality. They took pictures of the participants from three different distances. SCface also includes infrared images taken in the dark. This equipped the database with a very special kind of uncontrolled lighting (Grgic et al., 2011).

The good, the gad & the ugly (GBU) database (Phillips et al., 2011), in the version in which it is used in this study, consists of 8638 frontal images from 782 different identities. It provides the three protocols called Good, Bad and Ugly. Each of the protocols contains 1085 images. As can be inferred from the name of the protocol that Ugly is the most difficult protocol, while Good is the easiest one (Phillips et al., 2011).

The Face Recognition Grand Challenge (FRGC) database (Phillips et al., 2005) contains about 50000 images from 466 identities. However, the database version used in the present work was composed of 33032 images and 466 identities. It consists of high-resolution images and also three-dimensional images. The database provides up to six different protocols (Phillips et al., 2005).

The MOBIO (McCool et al., 2012) database consists of 61 hours of video data taken from 150 identities via mobile phone or laptop. Using only mobile devices gives the data-set a special,

uncontrolled touch since the camera is not in a fixed position. Consequently, there is high variability in illuminations, poses, and background. All the data in this database was collected within 12 distinct sessions. In addition to face recognition, research also uses this database for speaker recognition (McCool et al., 2012).

The Labeled Faces in the Wild (LFW) database (Huang et al., 2007) is probably one of the most popular image databases for benchmarking face recognition algorithms in unconstrained environments. It was made to approximate conditions in everyday life for the purpose of creating a database. Therefore, the images contain special types of lighting, poses, and expressions, such as additional people and faces in the background or self-occlusion. The database consists of 13233 images from 5749 individuals, all of which are downloaded from the Internet. The images were separated into two sets, the first of which was made for algorithm development and the second of which was made for performance reporting (Huang et al., 2007).

The YouTube Faces (YTF) database (Wolf et al., 2011) is a benchmark for recognizing faces in challenging and unconstrained videos. It consists of 3425 videos of 1595 identities. The duration of the videos varies between 48 and 6070 frames (Wolf et al., 2011).

The IARPA Janus Benchmark C (IJB-C) database (Maze et al., 2018) is currently the most widely used benchmark for face recognition. It improves on its predecessor IJB-B (Whitelam et al., 2017) by adding 1661 more identities. Thus, IJB-C has more diversity in occlusion, occupation, and geographic origin to better represent as much of the world's population as possible. The database consists of a total of 31334 images and 11779 videos of 3531 identities (Maze et al., 2018). It should be noted that no experiments were performed on this database within the scope of the present work.

2.2 Face Recognition in Challenging Environments

Before deep learning, face recognition was accomplished through traditional face recognition algorithms. At that time, many face recognition algorithms were published. Due to the lack of information in research papers, it was almost impossible to compare the performance with the state of the art. Additionally, because of databases that are not publicly available and did not have published protocols, experiments could not be reproduced either (Günther et al., 2017). This motivated Günther et al. (2016) to do a study on state-of-the-art face recognition algorithms that was completely based on open-source material. This study was further extended in Günther et al. (2017).

Günther et al. (2016) and Günther et al. (2017) evaluated the performance of traditional algorithms in unconstrained face recognition environments. For their experimental setup, they used the open-source software Bob (Anjos et al., 2012), which is also used in the present study. First, they preprocessed the data by aligning the face mostly with hand-labeled eye locations and then removing background information. In the next step, the features were extracted from the preprocessed image. Then, the extracted features of the probes were compared with the models. The database protocol specified which probe image should be compared with which model. Each comparison was assigned to a score, which was then used to evaluate the performance (Günther et al., 2017). The authors made their implementation of the database protocols publicly available. According to Günther et al. (2016), they used a total of five open-source algorithms and one commercial algorithm for their experiments. The first algorithm is called Linear Discrimination Analysis (Zhao et al., 1998). It basically projects the input data to a new space, so that the class separation is maximized. Gabor grid graphs (Günther et al., 2012) is another algorithm that was used. The method makes use of the Gabor jet to compare the similarity of the input faces. A further algorithm is called local Gabor binary pattern histogram sequence (Zhang et al., 2005). Inter-Session Variability (Wallace et al., 2011) is the fourth used open-source algorithm. Last but

not least, there is an algorithm called LDA-IR (Lui et al., 2012) which is also known as Cohort-LDA. The Commercial Of-The-Shelf (COTS) algorithm is not freely available and therefore no specific information about its implementation is known (Günther et al., 2016). In the work of Günther et al. (2017) the PCA (Turk and Pentland, 1991) and the LR-PCA (Phillips et al., 2011) algorithm were included into the evaluation.

As a first step, Günther et al. (2016) optimized the performance of the algorithms. In doing so, the authors investigated the influence of different preprocessing variants and image resolutions of the input images on the algorithms. The preprocessing algorithms used followed the concept of reducing the illumination in the images (Günther et al., 2016). One of these algorithms is called Histogram Equalization (Ramírez-Gutiérrez et al., 2010), which basically adapts the gray values of the image. Self Quotient Image (Wang et al., 2004) is another preprocessing technique that follows the idea of dividing the image by a smoothed version of itself. Finally, a multistage preprocessing technique by Tan and Triggs (2010) and preprocessing with local binary patterns (Heusch et al., 2006) were also tested.

Subsequently, experiments were performed on different images and video databases to test the algorithms in unconstrained and mobile environments. To evaluate the performance on occlusion, illumination, pose, and facial expressions, the AR face and the Multi-PIE database were used. For evaluation on unconstrained image and video databases, LFW and MOBIO, as well as YouTube Face were added (Günther et al., 2016). Experiments were also extensively performed on the CAS-PEAL, FRGC, GBU, and SCface databases (Günther et al., 2017).

From the results on the AR face database, it can be seen that most algorithms cope well with illumination. Occlusion, especially when a scarf is added, has a significant impact on their performance. Facial expressions can also cause problems for the algorithms. The screaming expression seems to confuse them the most. Regarding the different poses, it is striking that none of the tested algorithms is able to recognize non-frontal faces. With a face turned more than 45 degrees to the left or right to the camera, the performance of the algorithms can more or less be described as guessing (Günther et al., 2016). For the CAS-PEAL database, different illuminations caused the most difficulties. However, the use of different backgrounds usually has a minor influence. It should also be indicated that female faces on the MOBIO database were not recognized as well as the male faces (Günther et al., 2017). Furthermore, the authors concluded that the choices made in preprocessing played a crucial role. The Inter-Session Variability algorithm performed much better than the others in many experiments. Overall, however, the COTS algorithm achieved the best results (Günther et al., 2017).

2.3 Deep Learning

Deep learning has dominated and revolutionized the field of face recognition in recent years. Current face recognition surveys and reviews have been full of deep learning methods. These algorithms have advanced face recognition to a level that traditional methods can no longer keep up with (Wang and Deng, 2021). Therefore, deep learning has changed the process of conducting face recognition. When deep learning methods are employed, the first step in the face recognition pipeline is to preprocess the images. The faces are detected, cropped, and optimally prepared for the corresponding network. The next step is the extraction of features. The network takes the preprocessed faces as input and extracts descriptive features from them (Guo and Zhang, 2019). These features are then compared to models of previously enrolled images in the database gallery. These models are mostly made from one or more of a person's neutral faces and are intended to represent the identity of that individual. In the subsequent matching process, one can distinguish between face verification and face identification (Wang and Deng, 2021). Face verification, on the one hand, tries to find out if two face images belong to the same identity. On

the other hand, face identification is a one-to-many process. In the case of open-set identification, it is examined whether a face belongs to an identity that has already been enrolled in the database or not (Guo and Zhang, 2019). However, closed-set identification involves finding the correct person in the database gallery. As a last step, matches and mismatches are evaluated. There are different evaluation metrics, which will be explained later in Chapter 4.

The traditional face recognition process follows a similar procedure, but involves other steps as well. The whole process also begins with preprocessing. This is followed by feature extraction. In contrast to deep learning, the extraction is conducted using hand-labeled features (O' Mahony et al., 2019). Finally, for many algorithms, the extracted features are fed into a classifier, which performs the recognition (Minaee et al., 2021).

To understand the value of deep learning, one must first examine the original idea of artificial neural networks. An artificial neural network can be described as a mathematical model inspired by the structure of the human brain (Guresen and Kayakutlu, 2011). It consists of neurons that process information and are connected to each other through weighted connections. Basically, three different types of layers can be distinguished. Data is fed to the network through the input layer. Hidden layers process this data, and the output layer then returns the result of this process. Networks with more than two hidden layers are called deep neural networks (Wang and Deng, 2021). There are many different kinds of deep neural networks. The convolutional neural network is one of the most widely used in face recognition (Yi et al., 2014). It consists of three different types of hidden layers, convolutional layers, fully-connected layers, and pooling layers (Wang and Deng, 2021). The convolutional layer extracts essential features from the input through filters. The pooling layer is used to remove superfluous information (Guo and Zhang, 2019).

Returning to the intuition of the human brain, the neural networks also determine whether a neuron fires or not. This function is called an activation function and basically provides some non-linear projections of data. Different activation functions were developed in the past. First, one used the sigmoid activation function, which transformed a value into a value ranging from 0 to 1. Since this was not zero centered, one took the tanh activation function, which placed a value in the range of -1 to 1. The most widely used function today is the ReLU activation function. It transforms a negative value into zero and a positive into itself. ReLU is inexpensive to operate and, by its design, can differentiate the informative data from noisy data. There are variations of ReLU, such as the LReLU, but they are not as often used as the ReLU (Guo and Zhang, 2019).

Other than the creation of new databases for training, there are two main research directions in the academic community that have tried to improve the performance of neural networks, especially in unconstrained face recognition environments. One is the development of new network architectures and the other is the creation of new loss functions. In 2012, the AlexNet (Krizhevsky et al., 2012) network was the first network to achieve state-of-the-art results. Made of five convolutional layers and three fully-connected layers, the AlexNet network used ReLU as an activation function (Wang and Deng, 2021). Two years later, Simonyan and Zisserman (2014) presented VGGNet. This consists of several convolutional layers followed by fully-connected layers. By adding more convolutional layers, this network architecture can reach 16 to 19 layers. Another special feature of this architecture is that the convolutional filter number is doubled after each pooling (Wang and Deng, 2021). Again, a year later, the GoogLeNet was introduced by Szegedy et al. (2015). This consisted of 22 layers and had the particularity of executing several convolutional layers with different filter sizes in parallel. In 2016, He et al. (2016) published ResNet, which is still one of the most popular network architectures today. ResNet has the particularity of having shortcuts between the layers, that allow layers to be skipped. This is a great advantage when training a very deep ResNet, as the shortcuts can help train the deeper layers as well. In this network, 18 up to 152 layers are common. One of the newest architectures is the SeNet from Hu et al. (2018), who introduced a special block for squeeze and excitation that can be integrated into current network architectures (Wang and Deng, 2021). This block allows for weightings of individual

channels (Hu et al., 2018). The networks used today are unsuitable for use on mobile devices due to their size. For this reason, so-called lightweight network architectures have recently been developed. One of these is the MobileNet of Howard et al. (2017), which uses depth wise separable convolutions, leading to a considerable reduction of parameters compared to other networks with similar depth. As a result, it is usually only several megabytes large and does not require a significant amount of memory. In addition, MobileNet does not need much processing power and can therefore be used on mobile devices (Wang and Deng, 2021).

Research in recent years has mainly focused on the development of new loss functions. Since there are many loss functions, only a few popular ones will be discussed in this section. The basic idea of the loss functions is similar for the most of them. Basically, it is about having the features of an identity as close to each other as possible, which is referred to as intra-class similarity. Between them, however, one would like to have as large distance as possible. In other words, the features of one identity should be as separate as possible from the features of another. This is called inter-class variance.

Probably the most common loss function used for classification is softmax loss (Cao et al., 2018). The big disadvantage of this loss function is that it does not explicitly encourage higher variety for inter-class samples and similarity for intra-class samples. This makes it difficult to maintain optimal performance on data-sets with many varying images (Deng et al., 2019). DeepFace (Taigman et al., 2014) can be cited as one of the networks that performed quite well with softmax loss. Another approach is triplet loss, which became very popular through FaceNet (Schroff et al., 2015). It takes a sample from the database and compares it with a positive match and a negative match. In doing so, it tries to minimize the relative distance of the sample's features to the positive match and maximizes their relative distance to the negative match (Wang and Deng, 2021). The drawback of this function is that the selection of the single triplets significantly influences the training. It is also very computationally intensive (Wang and Deng, 2021). VGGFace of Parkhi et al. (2015) achieved state-of-the-art performance with triplet loss as a loss function. Center loss (Wen et al., 2016) is a function that favors intra-class similarity, learning a center for each class and penalizing the distance of features to the corresponding center. This approach pulls the features from the same class together (Hsu et al., 2020). However, the disadvantage of this function is that it requires much of GPU memory and many balanced training data (Wang and Deng, 2021). Due to frequent updates to the class centers, the training process can also be very unstable (Hsu et al., 2020). So far, only Euclidean-based loss functions have been discussed. For a few years, cosine-margin-based loss functions, which try to separate features by a larger cosine distance, have been developed. Large-margin loss (Liu et al., 2016), abbreviated L-Softmax, was one of the first loss functions that extended softmax loss with a margin. In contrast to triplet loss, this function tries to separate the classes with an angular margin. In 2017, Liu et al. (2017) created SphereFace which uses angular softmax loss which is similar to L-Softmax. The advantage of angular softmax is that through geometric interpretation, the features lie on a hypersphere (Minaee et al., 2021). Two other promising methods are CosFace (Wang et al., 2018) and ArcFace (Deng et al., 2019). CosFace, also called large margin cosine loss, learns features by maximizing the inter-class cosine (Hsu et al., 2020). ArcFace uses an additive angular margin to maximize intra-class similarity and inter-class variance. AdaCos (Zhang et al., 2019a) and PS2Grad (Zhang et al., 2019b) are two of the most recent loss functions. Over time, there has also been much variation in softmax loss, such as ring loss (Zheng et al., 2018), which will not be further discussed in the present work.

After the introduction of different loss functions, transfer learning should be briefly mentioned, as it is very popular today. A network cannot always be trained from scratch because there are not always enough training data available and because of the great depth of the network. Transfer learning allows a network trained for a related task to be adapted for the actual task, which can be especially useful when there is very little training data available (Minaee et al., 2021).

Chapter 3

Deep Neural Networks

This chapter introduces the deep neural networks used in the experiments of this study. These deep neural networks are taken from two different research papers, and each network is described in its own section. Details about the network architectures, the loss functions used, and the database used for training are provided.

3.1 VGGFace2

One network that is later used in the experiments, is called VGGFace2. Originally, only the training database was called VGGFace2. In the present work, however, the term is also used for the network itself. When VGGFace2 was introduced, people were already aware of the importance of inter-class diversity and intra-class similarity. However, the founders of VGGFace2 felt that no existing database had been developed specifically for testing pose and age variation. Therefore, they created guidelines to collect images of faces that exhibit significant variance in pose, illumination, ethnicities, and age. These guidelines followed similar procedures to those that Parkhi et al. (2015) used, but with a special focus on pose and age variations (Cao et al., 2018).

To create the VGGFace2 database, the researchers applied their guidelines to celebrities and public figures from Google image search. This resulted in a new data-set of over three million images of 9131 people. There are 80 to 843 different images per identity. The images differ substantially in terms of poses, age, background, and lighting. The database also showed a wide range of ethnicities and professions. It contained many more Asian faces than the VGGFace database (Parkhi et al., 2015). In addition, with about 60% percent male faces, it was more or less gender balanced (Cao et al., 2018).

Cao et al. (2018) trained a ResNet50 (He et al., 2016) on the self-developed database with softmax loss. The resulting VGGFace2 network was compared to two other ResNet50s on several benchmarks. VGGFace2 outperformed the state of the art in this comparison. With the use of an advanced network architecture, such as the SeNet50 (Hu et al., 2018), its performance could further be improved. Additionally, pretraining with MS1M and subsequent fine tuning with the VGGFace2 data-set had a positive effect on the performance (Cao et al., 2018). To attain the best possible performance, the present work uses VGGFace2[1] as a SeNet50, pretrained with MS1M and fine-tuned with the VGGFace2 database.

[1] https://www.robots.ox.ac.uk/~albanie/pytorch-models.html

3.2 ArcFace

Many researchers have developed new loss functions in the past years, as described in Section 2.3. One of these methods is the additive angular margin loss, which the developers also called ArcFace (Deng et al., 2019). It is still considered a state-of-the-art loss function today. One of the most common loss functions is softmax loss. However, this loss function has weaknesses because it does not explicitly promote high similarity in intra-class and high diversity in inter-class variation. As a consequence, performance decreases when intra-class variation increases, which occurs when a database contains a large variety of poses or facial expressions. To resolve this problem, Deng et al. (2019) added an additive angular margin penalty. This margin penalty improves the intra-class similarity and inter-class diversity. Other research papers have also developed loss functions with other types of margin penalties, such as SphereFace (Liu et al., 2017) and CosFace (Wang et al., 2018). From a numerical point of view, both favor inter-class diversity and intra-class compactness, but ArcFace features better geometric attribution (Deng et al., 2019).

Deng et al. (2019) list four advantages of their loss function. First of all, it is engaging since their loss function directly optimizes the geodesic distance margin. Second, it achieves state-of-the-art performance. The performance is evaluated on ten different databases. ArcFace competes with a RestNet100 architecture trained on the MS1MV2 database against several other methods. For more information on MS1MV2, please see Section 2.1. The new loss function outperforms any state-of-the-art algorithm. Next, it is easy to implement in several deep learning frameworks, as it requires only a few lines of code. Finally, it does not require significant computational power (Deng et al., 2019).

The developers of ArcFace have provided a model zoo with pretrained networks on their Github[2] repository. Among these are a ResNet34, ResNet50, ResNet100, and a MobileFaceNet, which are used for the experiments in this work. All four networks were trained by ArcFace on their own refined version of MS1M. MobileFaceNet is the smallest network used in the present work. The architecture for this small network was taken from Chen et al. (2018). Throughout the rest of this work, the four networks from the model zoo are called ArcFace-34, ArcFace-50, ArcFace-100, and ArcFace-Mobile.

[2] https://github.com/deepinsight/insightface/wiki/Model-Zoo

Chapter 4

Experimental Structure

This chapter first describes the setup of the experiments and the software used. Section 4.2 discusses the preprocessing in more detail. Challenges in alignment and the resolutions for these challenges by the networks used are shown. The last section describes the extraction of features from the images. A separate extractor for deep neural networks had to be implemented for the task of extraction. Additionally, parts of the source code are provided for better understanding. Finally, challenges and problems are explained in detail.

4.1 Software Bob

The experiments described in the present work all rely on the software Bob (Günther et al., 2012). This is a free, open-source signal processing and machine learning toolbox (Anjos et al., 2012). It was developed by the Idiap Research Institute in Switzerland to encourage reproducible research. According to Anjos et al. (2017), a paper is considered to be reproducible if it is repeatable, shareable, extensible, and stable. Unfortunately, many papers still involve experiments that are not reproducible with the information they provide for several reasons. First, the requirements for software are different for everyone (Anjos et al., 2012). Most often, one software does not meet all requirements. Therefore, a bundle of software packages has to be used. In addition, the installation of frameworks is often not easy and the experimental setup consists of several steps. Furthermore, current research papers are bound by a relatively short text limit, which makes it difficult to show the details of the implementation. Moreover, the complexity of research challenges does not make it simpler to reproduce a study. Therefore, it can be quite difficult to make a research paper reproducible (Anjos et al., 2017). Bob is designed to eliminate the problem of reproducible research by providing an all-in-one and transparent open-source software. It has a Python programming interface, which makes it easy to start using it (Günther et al., 2012). Some bottlenecks are implemented in C++ to maintain efficiency in processing large quantities of multimedia data. Additionally, the software is actively maintained and well documented. Furthermore, the whole biometric framework provides several implementations for preprocessors, as well as features extractors, databases, recognition algorithms, and evaluation metrics. The most important thing about Bob is probably its extensibility, which allows researchers to extend the software for their own purposes (Anjos et al., 2017). Lastly, Bob can easily be installed in a Conda environment. The software currently runs only on Linux-based systems, but the founders plan to add support for Windows (Anjos et al., 2012).

This work uses the same version of Bob (8.0.0.) that Günther et al. (2016) and Günther et al. (2017) used in their papers. The experimental setup provided by Bob consists of four main steps. Figure 4.1 illustrates the steps in the face recognition process according to the Bob documenta-

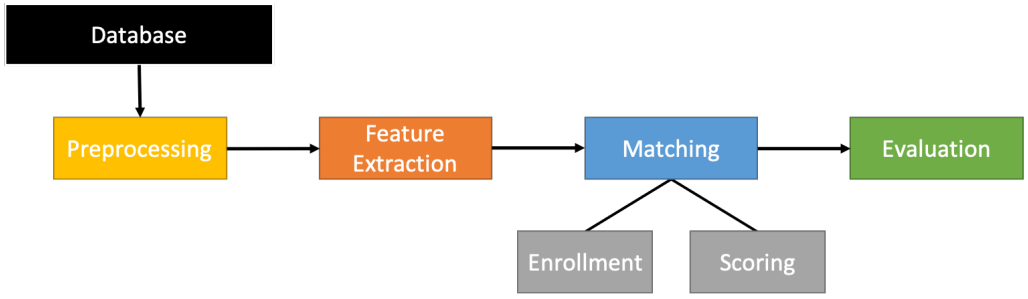


Figure 4.1: FACE RECOGNITION PROCESS. This figure displays the main face recognition process used for the experiments.

tion,[1] which are explained in the following sections. First, a preprocessing step is executed on the data-set. This step usually cleans up the raw data. In the case of the experiments in this work, the image is first aligned by hand-labeled eye positions, resized, and finally cropped. Section 4.2 describes this step and the preparatory efforts in greater detail. At this point, it must be said that the communication between the main steps is file-based and takes place through reading and writing. This means that the intermediate products are stored after each process and then processed by the next step. The preferred format for this read and write process is called `HDF5`. After preprocessing the images, feature extraction is executed on the preprocessed data. This is basically the step where the face recognition algorithm extracts features from the preprocessed image that describe the face. Section 4.3 provides more information about feature extraction and the implementation details used in this step. Next comes matching, which includes the three sub-steps of projection, enrollment, and scoring. The first of these, projection, is optional and would serve to project the extracted features into a lower dimensional subspace. However, in the experiments of this work, projection is never needed. The second sub-step, enrollment, takes several samples of the extracted features of a person to create a model. This model represents the identity of the corresponding person. The database protocol determines the images that are utilized for the enrollment. The last sub-step is the scoring, in which the models are compared with several probe samples. The database protocol defines which probes are associated with which model. During this comparison a score, which describes the similarity between the model and the probe is calculated. The present work uses the cosine similarity for this comparison. A higher score signifies high similarity, and a lower score means low similarity (Anjos et al., 2017). The scores are then saved in text files. Finally, the evaluation process decides whether a score indicates a match or no match with regard to face verification. There are several evaluation metrics to report performance, which are discussed later in this section.

What has not yet been discussed is that the identities in the database are initially divided into groups called the development set and the evaluation set. The development set is basically used to define certain parameters that later become important for the evaluation phase. Within the scope of this work, the development set is used to define the threshold above which a higher similarity score can be interpreted as a match. It is usually set at the intersection of genuine score and impostor scores (Günther et al., 2016). The score that compares two feature vectors of the same person is called the genuine score. The impostor score is the score when the feature vectors of different persons are compared. Figure 4.2 shows one of the genuine and impostor scores with the respective threshold (Anjos et al., 2017).

Before running face recognition experiments in Bob, one has to create a configuration file, to specify all required parameters. Listing 4.1 shows a part of an example configuration used in the

[1] https://www.idiap.ch/software/bob/docs/bob/bob.bio.base/v4.1.1/index.html

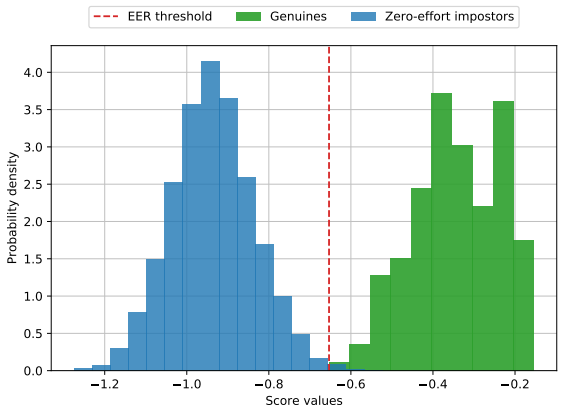


Figure 4.2: GENUINE-IMPOSTOR. This figure shows an example histogram of genuine and impostor attempts, executed on the on the protocol occlusion of the AR face database with ArcFace-Mobile as feature extractor.

experiments in this work. First, all parameters that are indispensable for the experiment must be defined. This includes the database, the preprocessor, the feature extractor, the algorithm for comparing the score files, and the name for the sub-directory of the results. In the example given, an already registered database could be set as a simple string. Furthermore, the algorithm for the cosine similarity is already implemented and can be selected from a list. Additionally, the protocol, development set, evaluation set, and more precise directories are defined. The verbosity can also be specified, simplifying the process of debugging.

After the experiments are successfully run, the results should be formatted in a way that allows performance comparisons. Therefore, Bob provides the functionality to calculate different evaluation metrics from the final score files. This functionality is particularly helpful for plotting the results. The following paragraph briefly explains the key metrics that are relevant for the present work. For more detailed information on the calculations and the mathematical formulas, the reader is referred to Günther et al. (2016) and Günther et al. (2017). Many evaluation metrics build on the false acceptance rate (FAR) and the false rejection rate (FRR), which are also known as false match rate and false non-match rate. The FAR basically shows the relative number of impostor attempts above a certain threshold and the FRR measures the number of genuine attempts below a threshold. One metric that is frequently used is the equal error rate (EER). To calculate the ERR, the threshold must be calculated first. As can be seen in Figure 4.2, the scores are divided into two categories, the intersection of which is the threshold (Günther et al., 2016). This work takes the scores of the development set for this task. The EER is then defined by the summing up half of the FAR and FRR from the development set. The half total error rate (HTER) is calculated in the same way as the ERR, but the evaluation set is used instead of the development set. However, the threshold value is still based on the scores of the development set (Günther et al., 2016). The receiver operating characteristic (ROC) is another common evaluation metric that plots the correct acceptance rate (CAR) over the FAR. The CAR results from 1-FAR (Minaee et al., 2021).

4.2 Image Preprocessing

Preprocessing plays a crucial role in the face recognition process. If not done correctly, it can affect the performance of face recognition algorithms. Therefore, a great emphasis is put on the correct

```
# Need to be set
database = 'arface'
preprocessor = bob.bio.face.preprocessor.FaceCrop(cropped_image_size=(H,W),
   cropped_positions={"reye":(y,x),"leye":(y,x)},color_channel='rgb')
extractor = DNNExtractor(model="dnn.onnx")
algorithm = 'distance-cosine'
sub_directory = 'ArcFace-Mobile'

# Optional arguments
groups = ['dev', 'eval']
protocol = 'occlusion'
temp_directory = 'temp/'
result_directory = 'results/'
verbose = 3
```

Listing 4.1: Example Configuration File

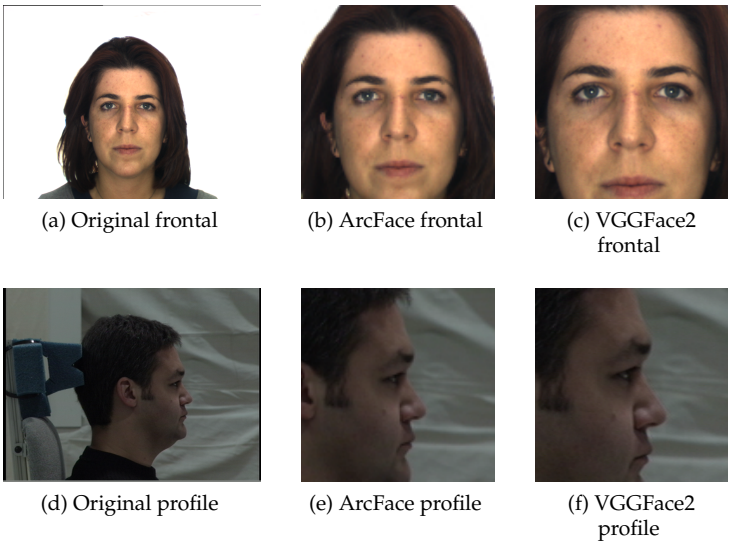
preprocessing. This allows the networks used to achieve the best possible performance. In most cases, the face is detected in the image aligned and cropped to a given size. For all databases used in the present work, hand-labeled eye positions are available (Günther et al., 2016). Thus, the faces do not have to be detected first. They can be cropped directly according to the desired size and the eye positions.

Unfortunately, many research papers lack detailed information on how preprocessing was performed. Often, the description of the alignment is omitted completely, and the focus centers on the results. The lack of these details makes it particularly difficult for others to build upon this research. Reproducing the experiments becomes almost impossible. This was also a major challenge in the present work. Exact details were not provided by any of the networks used. Deng et al. (2019) provided only the required dimension of the input data of 112×112 for the ArcFace networks. There were also some scripts to align faces based on landmarks detected with a multitask cascaded convolutional networks (Zhang et al., 2016), also known as MTCNN. However, since this work only uses eye positions, it was not entirely clear how to achieve alignment. Six pictures, which indicate some kind of alignment, could be taken from the GitHub[2] repository. This work conducts preprocessing according to these sample images since they are the only source for the correct alignment. The eye positions were taken from these images by hand, and a mean value was generated, which was then used for the experiments.

Cao et al. (2018) provides more information about the alignment of VGGFace2. The research paper itself only shows the cropped dimensions of the input data of 224×224. However, a webpage,[3] which also provides a copy of the network, still offers notes from one of the authors of VGGFace2. According to this information, the alignment could be reproduced for some example images. The faces were first detected with MTCNN. Subsequently, the created bounding box was extended by a factor of 0.3. Next, the shorter side was resized to 256 pixels, and finally, a central piece of 224×224 pixels was cropped. Some example images for the performed preprocessing can be found in Figure 4.3.

The preprocessing for almost all experiments in this work could have been done with the eye positions. The experiments on the Multi-PIE database required an additional alignment point since the images do not always provide two visible eyes (Gross et al., 2010). The visible eye and

[2] https://github.com/deepinsight/insightface
[3] https://www.robots.ox.ac.uk/~albanie/pytorch-models.html



(a) Original frontal (b) ArcFace frontal (c) VGGFace2 frontal

(d) Original profile (e) ArcFace profile (f) VGGFace2 profile

Figure 4.3: PREPROCESSING EXAMPLES. This figure shows for each network some preprocessed example images from the AR face and Multi-PIE database.

	VGGFace2	ArcFace
Right Eye	(100, 65)	(52, 38)
Left Eye	(100, 159)	(52, 74)
Eye	(100, 112)	(52, 56)
Mouth	(199, 112)	(91, 56)

Table 4.1: PREPROCESSING. This table shows a summary of the used eye and mouth positions for the experiments. All parameters are given in `(y,x)` order because Bob requires this order for alignment. The right and left eye were used for frontal faces and the eye and mouth for profile faces.

the respective corner of the mouth served as a reference point for these images. Example images can be found in Figure 4.3. Table 4.1 shows a summary of the preprocessing specifications that were ultimately used.

4.3 Feature Extraction

A new extractor had to be implemented to connect the deep neural networks to the biometrics framework. The feature extractor needed to inherit from the existing extractor class and to implement the `__init__` and the `__call__` methods. OpenCV,[4] an open-source computer vision and machine learning software library that includes various algorithms and frameworks, was used to load the networks and forward propagate through them. The software has a module for deep neural networks, which can load networks from various frameworks. At a minimum, the model must be specified in the constructor to be able to use the networks. Optionally, a config file and the name of the framework of the network can be provided. There is also the possibility to set a mean that should be subtracted from the preprocessed images. Listing 4.2 shows a snippet of the

[4] https://opencv.org

```
def __init__(self, model, config=None, framework=None,
   mean=None, swapRB=False, **kwargs):

   Extractor.__init__(self, kwargs)
   self.model = model
   self.config = config
   self.framework = framework
   self.mean = mean
   self.swapRB = swapRB

   # load model
   self.net = cv2.dnn.readNet(model=self.model,
      config=self.config, framework=self.framework)
```

Listing 4.2: Constructor of the created extractor

```
def __call__(self, data):
   # 1. transpose the image CHW -> HWC
   image = numpy.transpose(numpy.uint8(data), (1,2,0))
   # 2. create blob
   blob = cv2.dnn.blobFromImage(image, mean=self.mean, swapRB=self.swapRB)
   # 3. set the blob as input to the network
   self.net.setInput(blob)
   # 4. perform a forward-pass on the network
   features = self.net.forward()
   # 5. return the features
   return features[0]
```

Listing 4.3: Feature extraction method of the created extractor

`__init__` method of the implemented extractors. If all necessary parameters are provided, the network has been loaded.

The main part of the feature extraction takes place in the `__call__` method. One of the advantages of OpenCV, besides the fact that it can load various frameworks, is that its implementation can be kept very simple. As can be seen in Listing 4.3, it does not even take ten lines of code to extract the features from an image. Since Bob provides the images in a different channel order than OpenCV requires, this must be changed in a first step from CHW to HWC. Here, C stands for the color channel, and H and W represent the height and width of the images. Next, a blob of the image is created. A blob is basically nothing more than the image after the preprocessing step defined in the blob. The present work uses this step only for mean subtraction since the images have already been preprocessed, as described in Section 4.2. The blob is set as input for the corresponding network, which is then forward-passed in the fourth step. Finally, the resulting features are returned.

One of the most substantial challenges in implementing the extractor was loading the networks from the framework MXNet.[5] After some research, it turned out that OpenCV cannot load

[5] https://mxnet.apache.org/versions/1.8.0

this framework. Consequently, the networks from ArcFace had to be converted to the Open Neural Network Exchange[6] format. This was done with the help of the Apache MXNet-Incubator Github[7] repository. Additionally, VGGFace2 had to be converted in the Open Neural Network Exchange format since OpenCV is not able to load networks from PyTorch.[8] Therefore, the PyTorch Github[9] repository was used.

[6] https://onnx.ai
[7] https://github.com/apache/incubator-mxnet
[8] https://pytorch.org
[9] https://github.com/pytorch/tutorials

Chapter 5

Results

The following chapter presents the results of all experiments performed using the five networks ArcFace-34, ArcFace-50, ArcFace-100, ArcFace-Mobile, and VGGFace2. In addition to visualizations with graphics, information about the experimental settings is also provided. For more information on the databases used, please refer to Section 2.1.

5.1 Face Variations

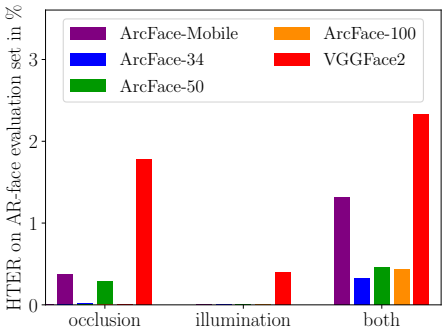
In this section, the introduced neural networks were tested against three types of face variations, more precisely partial occlusion, different expressions and poses. Each type of face variation and its experimental setting and results are described in the following subsections.

5.1.1 Partial Occlusions

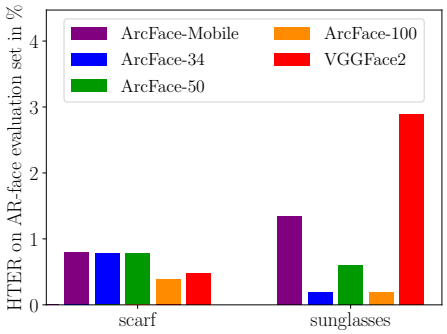
Partial occlusion is a common issue in unconstrained face recognition environments, which makes the verification of identities harder. Especially during the COVID-19 pandemic, when this work was written, many people wore masks that covered their faces from nose to chin. The AR face database (Martínez and Benavente, 1998) is used to evaluate the performance of the neural networks with respect to different partial occlusions. The database consists of four protocols expression, occlusion, illumination, and occlusion_and_illumination. The protocol expression is not used in the experiments. Figure 5.1(a) displays some example images from the used protocols. For all experiments on this database, only images with neutral facial expressions were used to observe the influence of occlusion and illumination as isolated as possible. The identities were split up in to 24 males and 19 females for each the development and the evaluation set. For model enrollment, two images per identity that featured neutral expressions and illumination, as well as no form of occlusion, were used.

As can be seen in Figure 5.1(b), most of the neural networks were not affected by occlusion or illumination. None of the used networks from ArcFace (Deng et al., 2019) had any trouble with the presence of illumination. ArcFace-Mobile, which is the smallest network used in this work, was able to handle occlusion quite well, but had slightly more trouble when occlusion and illumination are combined. ArcFace-34 coped nearly perfectly with occlusion and even a bit better than ArcFace-50 and ArcFace-100. It showed the best performance over all networks as far as the protocol occlusion_and_illumination is concerned. ArcFace-50 was minimally affected by occlusion and performed very similarly to ArcFace-100 in the combination of occlusion and illumination. VGGFace2 (Cao et al., 2018) had more issues with illumination and occlusion than the networks from ArcFace (Martínez and Benavente, 1998). Taking a closer look at VGGFace2,

(a) neutral, occlusion, illumination, illumination and occlusion



(b) Effect of illumination and occlusion



(c) Effect of different occlusion types

Figure 5.1: PARTIAL OCCLUSION. This figure shows example images of the AR face database and the effect of partial occlusion of the face on the tested neural networks.

occlusion affected this network more than illumination. When it comes to the two different types of occlusion in this study, ArcFace-100 showed the best performance with both subjects wearing a scarf and subjects wearing sunglasses. The networks from ArcFace handled both occlusion types quite well. ArcFace-Mobile and VGGFace2 were somewhat more affected by the subjects wearing sunglasses. All in all, the eyes seemed to have a similar effect on the performance of the neural networks than the region around the mouth.

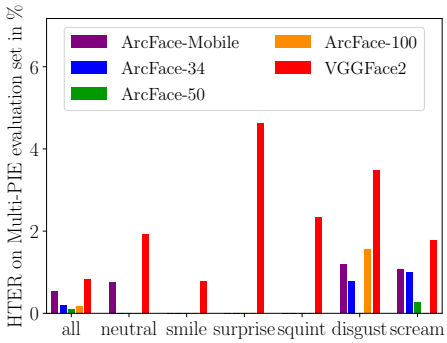
5.1.2 Facial Expressions

A neutral facial expression is usually not the reality in practical face recognition. Humans are emotional beings and tend to show their emotions intensely through facial expressions. This has a great visual impact on facial features (Guo and Zhang, 2019). Therefore, modern face recognition algorithms must be able to handle a wide range of facial expressions. The Multi-PIE (Gross et al., 2010) database is used to test neural networks against a variety of expressions. Günther et al. (2016) published different protocol implementations, such as protocol P containing pose variations, protocol E containing facial expressions, and protocol U containing non-frontal illumination. In this experiment, only protocol E is used. Sixty-four identities were applied for the development set, and the evaluation was composed of 65 identities. According to the protocol used, five faces per identity were considered for model enrollment.

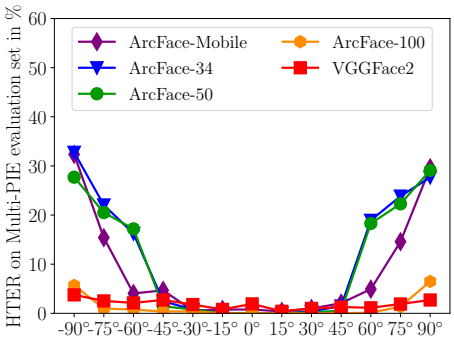
The plot of Figure 5.2(c) reveals that most networks can handle facial expressions well. The best recognized expression was the smiling expression. ArcFace-50 performed best across all algorithms and only had minor problems with screaming faces. ArcFace-34 struggled with the disgusted and screaming expressions. ArcFace-Mobile performed slightly worse than ArcFace-34 and, additionally, could not classify all neutral faces correctly. The large ArcFace-100 performed the worst out of all ArcFace algorithms on the disgusted expression. VGGFace2 showed the

(a) Poses examples from -90 to 90 in steps of 15 degrees

(b) Facial expressions examples: neutral, smile, surprise, squint, disgust, scream



(c) Effect of different expressions



(d) Effect of different poses

Figure 5.2: EXPRESSION AND POSE. This figure shows the effect of different expressions and poses of the face on the tested neural networks. Example images for poses and facial expressions are displayed in subfigure (a) and subfigure (b).

worst performance across all tested facial expressions. Especially when faces with surprising expressions were included, VGGFace2 failed compared to the others. All networks from ArcFace could correctly classify faces with smiles, surprise, or squints.

5.1.3 Face Poses

Another aspect that challenges face recognition is the presence of different facial poses. It is known that the performance of neural networks significantly drops when faces are no longer visible from the front (Sengupta et al., 2016). Protocol P from the Multi-PIE (Gross et al., 2010) database is used to observe the performance of neural networks on pose variations. This protocol provides faces rotated from left to right in steps of 15 degrees. The facial expressions are neutral, without any type of occlusion nor illumination. Since both eyes are still visible from -45 to 45 degrees, the hand-labeled eye positions could be used for alignment. For the poses from -90 to -60 and from 60 to 90 degrees, only one eye is still visible. Therefore, for these poses, the alignment was made according to the still visible eye and the corner of the mouth on the relevant side of the face. As in the experiment on facial expressions, 64 identities were used for the development set and 65 for the evaluation set. Five frontal images per identity were used for model enrollment.

Figure 5.2(d) shows the results for different pose rotation angles. It can be observed that all networks are able to handle poses from -45 to 45 degrees very well. From a rotation angle of more than 60 degrees to the right and the left, the performance of ArcFace-34 and ArcFace-50 started to drop abruptly. ArcFace-Mobile showed remarkable capabilities from -60 to 60 degrees and could almost keep up with VGGFace2 and ArcFace-100. From -75 to 75, ArcFace-100 and VGGFace2

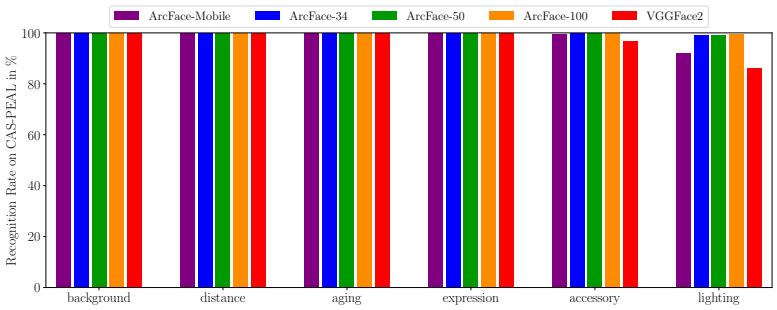


Figure 5.3: CAS-PEAL. This figure displays the recognition rates of the tested neural networks on the CAS-PEAL database.

could classify faces almost perfectly. Even with faces pointing at 90 degrees, their performance was still much better than that of all the others. ArcFace-34 performed the worst over the whole experiment, while VGGFace2 showed the best performance.

5.2 Experiments on Other Databases

This section provides the results for some common databases that were also used by Günther et al. (2017), including CAS-PEAL, MOBIO, SCface, FRGC, GBU, and LFW.

5.2.1 CAS-PEAL

The experiments on CAS-PEAL used only a subset of 9031 images from the original database, which was also used by Günther et al. (2017). Using this setting, the database provides the protocols aging, accessory, background, distance, expression, and lighting to test different variations, as described in Section 2.1. One image per identity was used for model enrollment.

The recognition rate of the networks on the CAS-PEAL database is shown in Figure 5.3. This metric basically identifies how many correct classifications an algorithm has achieved on a particular protocol. It can be observed that all four networks of ArcFace have a recognition rate of 100% on faces with variations in background, distance, age, or expression. Moreover, VGGFace2 can perfectly handle variations in background, distance and age. The performance drops for all networks when accessories and lighting vary, but ArcFace-34, ArcFace-50, and ArcFace-100 still offer a nearly correct classification. Handling variations in accessories works better with ArcFace-Mobile than with VGGFace2. Overall, lighting conditions cause most of the problems for the networks, while VGGFace2 has the worst performance on that particular protocol.

5.2.2 MOBIO

The MOBIO (McCool et al., 2012) database consists of video data taken by mobile phone or laptop. Therefore, the images varied in illuminations, poses, and background. For each video, a frame was extracted after each second to obtain images. MOBIO provides two gender-dependent protocols male and female, which are used for the experiments. The development set consists of 18 females and 24 males with 1890 and 3520 images respectively. However, the evaluation set consists of 20 females and 38 males with 2100 and 2990 images (Günther et al., 2017). Similar to Günther et al. (2016), five images per person are used for the model enrollment.

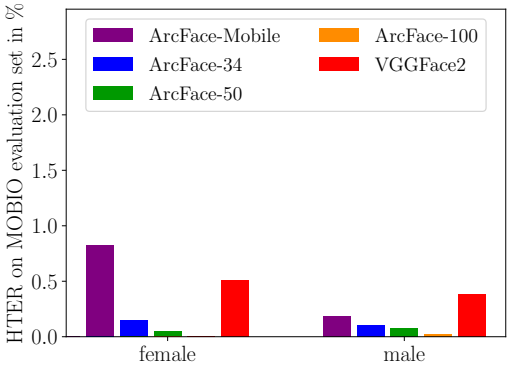


Figure 5.4: MOBIO. This figure shows the performance of the tested neural networks for protocol female and protocol male of the MOBIO database.

Figure 5.4 shows the results from the MOBIO database. ArcFace-100 outperformed all other networks on both protocols. It is able to recognize female faces perfectly. ArcFace-Mobile showed the worst performance on the protocol female. Both ArcFace-34 and ArcFace-50 performed better than VGGFace2, with ArcFace-50 showing the best performance of all three on both protocols. Across all networks, VGGFace2 performed the worst on the MOBIO database. Overall, a slight tendency that female faces are more difficult to recognize than male faces can be seen. This trend has also been observed in other experiments on this data-set (Khoury et al., 2014).

5.2.3 Surveillance camera face database

The surveillance camera face (SCface) database (Grgic et al., 2011) contains images taken by different video surveillance cameras in three different distances. The four different protocols combined, close, medium, and far were used to evaluate the performance on different camera distances. The first protocol is provided by the database, and the latter three protocol implementations were taken from Günther et al. (2017). For each protocol, images of 44 identities were used for the development set, and 43 identities were used for the evaluation set. One frontal image per identity is used for model enrollment. Unlike all probe images, the enrollment image is taken with passport-quality illumination.

Figure 5.5 shows the HTER on the SCface database, which was one of the most challenging databases in these experiments. Short distances did only slightly affect the networks. The performance decreases with increasing distance of the face from the camera. At long distances, only ArcFace-50 and ArcFace-100 showed a HTER below 20%. The resolution was correspondingly low for distant faces, which seemed to have a substantial impact on the performance of the networks. ArcFace-100 outperformed all the other networks on all four protocols. ArcFace-Mobile showed the worst performance of all, except at short distances, where VGGFace2 performed even worse. All algorithms performed worse on protocol combined than on close and medium put together.

5.2.4 Face recognition grand challenge

The FRGC (Phillips et al., 2005) database provides six different protocols. Protocols 2.0.1, 2.0.2, and 2.0.4 were the only ones that contained only 2D images. In protocol 2.0.1, one image at a

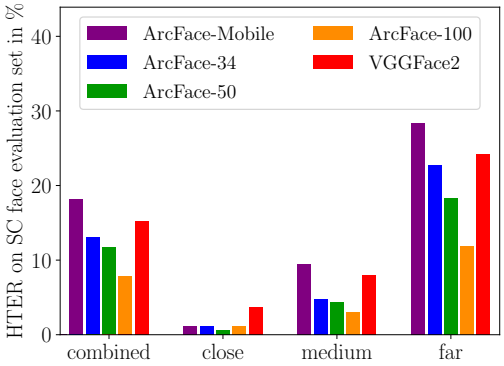
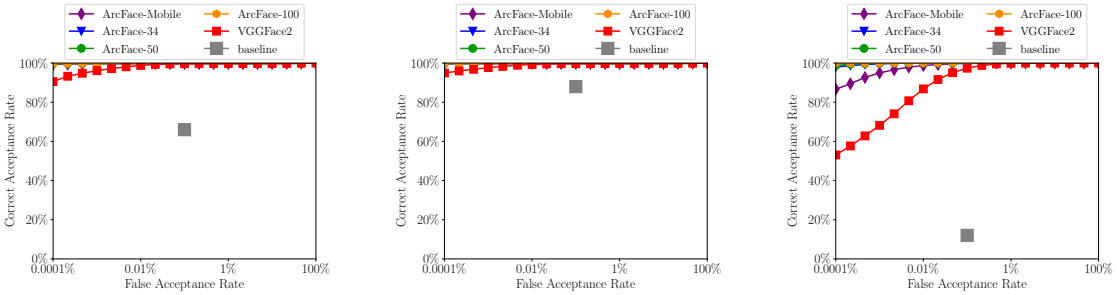


Figure 5.5: SCFACE. This figure displays the performance of the tested neural networks on SCface.

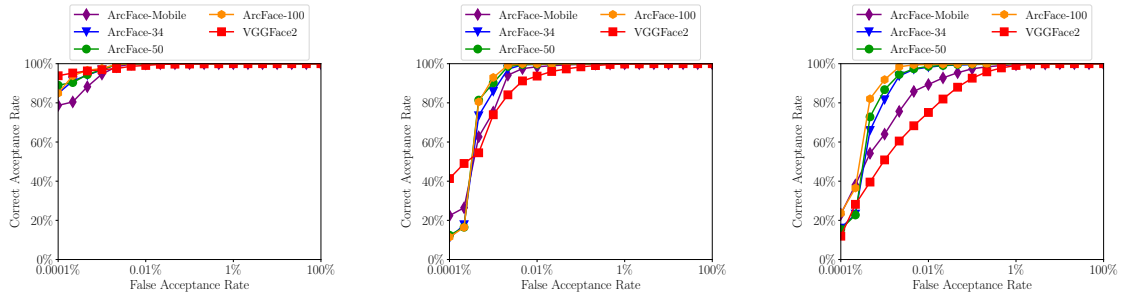


(a) 2.0.1 (b) 2.0.2 (c) 2.0.4

Figure 5.6: FRGC. This figure shows ROC curves for the tested networks on protocol 2.0.1, 2.0.2 and 2.0.4 of the FRGC database. Additionally, the baseline of 66%, 88% and 12% CAR at 0.1%FAR is displayed.

time was used for model enrollment, which was then compared to a single probe. Protocol 2.0.2 tested the effect of using multiple images per person. Accordingly, four images per identity were used for model enrollment, and a probe also consisted of four images. Averaging was then used to threaten the multiple images for building the probe and model enrollment. Protocol 2.0.4 used the same models as 2.0.1. Its difference with protocol 2.0.1 is that in 2.0.4, the probe images consist of images with uncontrolled illumination (Phillips et al., 2005).

Figure 5.6 shows the ROC curves for each experiment on each protocol of this database. The baseline shows the results reported by Phillips et al. (2005) at 0.1% FAR. ArcFace-100 showed the best performance on protocol 2.0.1 compared to all other networks. The other three networks from ArcFace also showed a similar performance. VGGFace2 performed slightly worse. ArcFace-100 also outperformed its competitors on protocol 2.0.2 although all ArcFace networks showed similarly good performance. The protocol 2.0.4 was probably the most difficult experiment on the FRGC database. As with the other protocols, ArcFace-100 achieved the best results. It is interesting that ArcFace-Mobile performed considerably better than VGGFace2 despite its size. All the networks used by far outperformed the baseline on all protocols.



(a) Good (b) Bad (c) Ugly

Figure 5.7: GBU. This figure shows ROC curves for the tested neural networks for the protocols Good, Bad and Ugly of the GBU database.

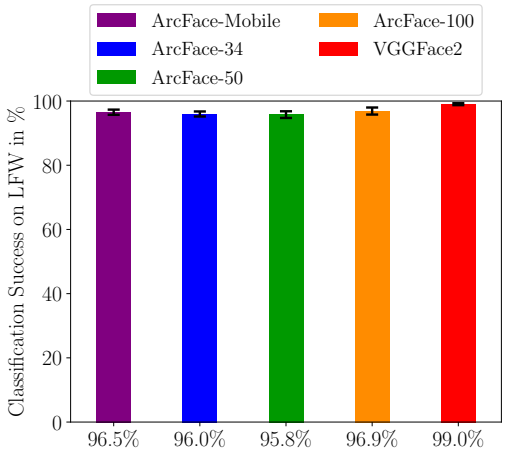


Figure 5.8: LFW. This figure shows the average classification success rates and standard deviations of the tested neural networks on LFW.

5.2.5 The good, the bad and the ugly

The experiments on the GBU (Phillips et al., 2011) database were performed on all three protocols Good, Bad, and Ugly. The model enrollment uses only one image per identity, but there are several models per identity. As described in Günther et al. (2017), the ROC curve is calculated by comparing all probes with all models. All networks performed well on protocol Good. The best performance was achieved by VGGFace2. For the protocol Bad, ArcFace-100 performed the best. The other ArcFace networks performed similarly. For the protocol Ugly, ArcFace-100 also achieved the best results. VGGFace2 could not keep up with the ArcFace networks on this protocol and did not even come close to the performance of ArcFace-Mobile.

5.2.6 Labeled Faces in the Wild

Labeled Faces in the Wild (Huang et al., 2007), also known as LFW, is a very popular image database for benchmark comparisons, allowing for the evaluation of the used networks with dif-

ferent types of lighting, expressions, and poses, as well as with the influence of various backgrounds. The experiments used the same protocol implementation as described in Günther et al. (2017). In this experiments, the second set, also called the second view, of the LFW database was used. In this set, the images were divided into 10 different folds, which were then used to determine a classification success. This indicates the percentage of all impostor and genuine attempts that were correctly classified below or above the threshold value. The development set, from which the threshold is calculated, consists of two folds.

Figure 5.8 shows the average classification success rate over the 10 folds. The best performance was achieved by VGGFace2 with a classification success rate of 99.0%. Interestingly, ArcFace-Mobile was almost on par with ArcFace-100 and only differed by 0.4%. Thus, it achieved a classification success rate of 96.5%. ArcFace-50 fared the worst out of all the networks from ArcFace.

Chapter 6

Discussion

This work provides an overview of the performance of state-of-the-art deep neural networks in challenging face recognition environments. Experiments were performed on eight different databases. The available evaluation protocols allowed an isolated examination of single face recognition aspects. In particular, the influence of face variations, such as occlusion, illumination, and different poses, on the performance of the deep neural networks could be investigated individually. Additionally, their performance could be reported on the LFW database. Unlike most current research, the experiments were conducted entirely with open-source software. Moreover, the pretrained deep neural networks used can be freely downloaded from the Internet. Furthermore, all relevant information was disclosed so that the results can be understood and reproduced. In addition, the database protocol implementations and evaluation were taken from Günther et al. (2016) and Günther et al. (2017), allowing for easy comparison of modern deep learning methods with traditional face recognition algorithms. In the previous chapters, much has been discussed about the experiment and results. This part of the thesis will examine and explain the results in detail.

The influence of partially occluded faces is the first face recognition aspect that was evaluated. This was mainly tested on the AR face database. The results showed that the ArcFace networks, in particular, were not affected by occlusion. VGGFace2 had somewhat more trouble, but still showed good performance. This could be explained by the fact that the additive angular loss implies inter-class diversity and intra-class similarity better than triplet loss does. This can also be interpreted from the paper of Deng et al. (2019), in which triplet loss was outperformed by far. Upon first glance, the different training database does not lead to any further explanation. Taking a closer look at the two types of occlusion, it is noticeable that faces with sunglasses cause more trouble for VGGFace2 than faces with a scarf in front of the mouth. This trend goes in the opposite direction for the ArcFace networks. Such differences are difficult to explain through the training databases since both consist of celebrities and do not explicitly address the issue of occlusion. A look at the evaluation on the development set could indicate a slightly better performance for faces occluded with a scarf. A general statement, such as the one of Günther et al. (2016), that sunglasses affect face recognition more, can not be identified in this study. ArcFace-50 and VGGFace2 both have a ResNet50 network architecture. It is interesting that ArcFace-Mobile handles occlusion worse than ArcFace-50, but still better than VGGFace2. This argues against the assumption that larger and deeper networks perform better on average, at least as far as occlusion is concerned. The results on the protocol accessory of the CAS-PEAL database support the results from the AR face database. With a variety of hats and glasses, the occlusions on CAS-PEAL are not the same as on AR face. The ArcFace networks performed even better under these conditions. On the one hand, this can be attributed to the preprocessing, in which a part of the upper head has been cut off, making the influence of different hats no longer significant. On the other hand, the CAS-PEAL database showed glasses and not sunglasses, which allowed the networks to extract

features around the eyes.

Another aspect that has been studied using the AR face database is the variation of illumination conditions. The ArcFace networks performed impressively under these circumstances and were not affected at all. VGGFace2 performed slightly worse. It is surprising that VGGFace2 had minor problems with illumination since the authors of the training database explicitly addressed this problem. The protocol lighting of the CAS-PEAL database tested the influence of different illuminations as well. On this database, the networks from ArcFace did not perform as well as on AR face. In particular, VGGFace2 and ArcFace-Mobile struggled, but ArcFace-100 was also unable to reach a recognition rate of 100%. This may be due to the fact that in the CAS-PEAL database, not only the direction of illumination, but also the types of lighting vary as the lighting ranges from ambient to bright white lighting. Thus, in the enrollment process, faces with different illumination types are selected, which may increase the difficulty of conducting face recognition. The fact that VGGFace2 performed worse than the ArcFace networks may also have to do with the different loss functions. Regarding the influence of the different training databases, no clear statement can be made.

The influence of different poses was specifically studied with the Multi-PIE database. When the head is slightly turned towards the camera, all networks still show impressive results. VGGFace2 performed best across all poses. This can be explained by the training database. The authors of VGGFace2 made an effort to provide their data-set with a variety of poses. ArcFace-100 performed similarly well. A possible reason for this might be its deep network architecture. The performance of ArcFace-Mobile is difficult to explain since it outperformed ArcFace-34 and ArcFace-50 at a pose rotation angle of -75 to -60 and 60 to 70 degrees. Such behavior might be justified by the different network architectures. Taking a closer look at the results, it can be seen that the HTER increases steadily from 0 to 45 and 0 to -45 degrees before decreasing again at 60 and -60 degrees, respectively. This phenomenon can be explained by the preprocessing. The alignment from -45 to 45 degrees is made on the basis of both eye positions. When the head turns, the eyes move closer to each other in a two-dimensional perspective. Since the defined eye poses remain the same, a kind of zoom-in effect takes place, affecting the performance. Günther et al. (2016) already emphasized the importance of preprocessing in traditional face recognition algorithms. A similar effect can be observed for deep neural networks.

The Multi-PIE database allowed the testing of the deep neural networks for performance on different facial expressions. Overall, the networks showed good results across all expressions. The networks from ArcFace were able to correctly classify all images except for the faces with disgusted or screaming expressions. Since these two expressions change the face significantly, they cannot always be recognized correctly. A possible explanation could be that the training database consists of celebrities who mostly offer friendly expressions as they look into the camera. Therefore, these two rather negative facial expressions are underrepresented in the training of the networks. VGGFace2 cannot correctly classify any of the protocols with 100% accuracy. This can be explained by the evaluation. A closer look at the ERR of the development set reveals that there are large performance differences between the development set and the evaluation set. The present work calculates the threshold using the development set to obtain a more realistic scenario. Thus, the threshold may not be very optimal for the evaluation set, which may cause this decrease in performance. The protocol expression of the CAS-PEAL database also evaluated facial expressions. In contrast with the Multi-PIE database, the ArcFace networks achieve a recognition rate of 100%, and VGGFace2 also performs very well. The reason for this behavior may be that the facial expressions differ between Multi-PIE and CAS-PEAL.

Another face recognition aspect studied in this work is the influence of different distances. The SCface database contains images of faces at different distances taken by surveillance cameras. The networks have no problem recognizing the faces at a short distance. However, with increasing distance, the HTER values increase rapidly. Since the two training databases did not

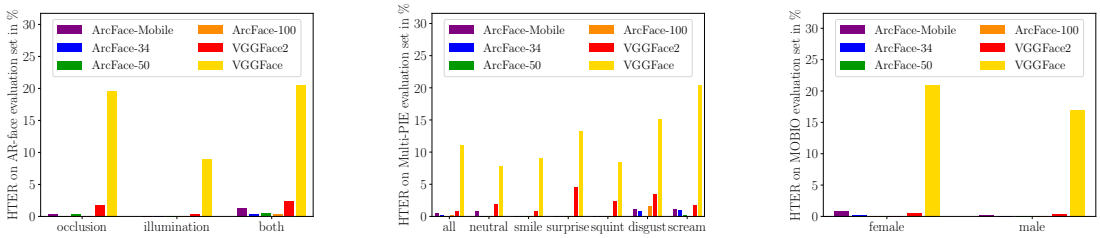
explicitly train the networks at different distances, this could be a possible reason for such a behavior. A more plausible explanation would be that the faces become too small to extract useful features at long distances. For example, in the experiments with the protocol far, the detected faces usually had a size of merely 20×20 pixels. After resizing the faces to the input size of the corresponding network, the faces were too pixelated. Examining the results on the CAS-PEAL database, one might think that there is a contradiction to the results on the SCface database. For its protocol distance, all networks achieved a recognition rate of 100%. This performance difference may have been for two main reasons. On the one hand, the CAS-PEAL database took frontal images with good lighting, while the images of SCface were taken from surveillance cameras with different qualities and at a higher angle. On the other hand, the range of distances recorded is very different. The SCface database contains images varying in distances from 1 to 4.2 meters to the camera. The distances provided by the CAS-PEAL database are much shorter and vary only between 0.8 to 1.2 meters. Therefore, the SCface database can be considered to be more difficult for the networks.

The experiments on the MOBIO database show the impact of female and male faces on the performance of the networks in unconstrained face recognition environments. Overall, male faces were easier for the networks to recognize. ArcFace-100 performed the best on both protocols. Additionally, the other ResNets from ArcFace performed similarly well on both protocols. The training database of the ArcFace networks contained many more female faces, while the VGGFace2 training database was fairly gender-balanced. Therefore, it was surprising to observe such a good performance on male faces, especially from the ArcFace networks. It can be concluded that the gender does not affect performance very much even though the training database does not contain an equal amount of female and male faces. In addition, it must be said that the images of the MOBIO database are often not frontal images. Thus, the ArcFace networks may have had an advantage.

Two challenges that could be tested on the CAS-PEAL database but have not yet been discussed are the influence of different backgrounds and ages. For the protocol background, as well as for the protocol aging, all networks of ArcFace and VGGFace2 achieved a recognition rate of 100%. These are all plausible results that can easily be explained. The backgrounds most likely have no influence on the performance since they might not even be noticed by the networks. The reason for this is the necessary preprocessing, which crops the faces so tightly that the background is no longer visible on the input image. The explanation for the good results on "aging" images lies in their collection of the database images. The images were taken at intervals of half a year. Most people hardly show visible aging within half a year. Thus, it is questionable how well this protocol handles the aspect of aging.

On the FRGC database, the impact of probe images on performance could be evaluated. Overall, ArcFace-100 showed the best results. The other ArcFace networks also seemed to work well. VGGFace2 had some difficulties in recognizing images within uncontrolled illumination conditions. The problems with illumination can also be observed in the experiments on AR face and CAS-PEAL. The reason is that training data from VGGFace2 does not deal enough with the aspect of illumination. It can clearly be seen that protocol 2.0.2, where four probe images are used to compute a score, worked best for all networks. Thus, it can be observed that the usage of multiple images for probing and model enrollment has a positive impact on face recognition performance. Protocol 2.0.4, which uses probe images with uncontrolled illumination, causes the most difficulties for the networks.

The GBU database contains frontal images at three difficulty levels. VGGFace2 achieved the best results on the easiest database protocol. The ArcFace networks had the upper hand on the two more difficult protocols. One possible explanation is that since the database contained only frontal images, the ArcFace networks did not benefit from their non-frontal training. However, when conditions worsened and different illuminations came into play, they benefited from their



(a) Occlusion and Illumination (b) Expression (c) MOBIO

Figure 6.1: VGGFACE. This figure reports the poor performance of VGGFace compared to the other tested networks. Subfigure (a) shows occlusion and illumination on AR face, subfigure (b) shows expression on Multi-PIE and subfigure (c) shows the performance on MOBIO.

training on illumination.

Last but not least, the results on the LFW database will be discussed. VGGFace2 outperformed all other networks on this benchmark. The largest of the ArcFace networks showed a classification success rate of 96.9%, which is not quite what ArcFace reported in their paper. They claimed to have achieved a performance of over 99% for ArcFace-100 (Deng et al., 2019). The differences can be explained by the calculation of the threshold. Deng et al. (2019) calculate their threshold based on the evaluation set. This implicitly calculated the threshold that maximizes the performance on the evaluation set. In reality, however, face recognition is generally performed on previously unknown faces, where the threshold must already be known. Therefore, the present work calculated the threshold on the basis of the development set and used it for evaluating the evaluation set. Thus, a more realistic environment was created in this study.

An important point to mention is that experiments with VGGFace (Parkhi et al., 2015) were also performed as part of the present work. VGGFace was developed in 2015 and was one of the first convolutional neural networks that was very useful for face recognition. It makes use of a VGGNet-16 (Simonyan and Zisserman, 2014) network architecture and was trained with the VGGFace database and triplet loss. With this configuration, researchers achieved comparable state-of-the-art performance against FaceNet (Schroff et al., 2015) and DeepFace (Taigman et al., 2014). Unfortunately, it was not possible to obtain useful results in the shape of the present work. Figure 6.1 reports the poor performance of VGGFace compared to the other networks used. As can be seen, VGGFace is outperformed by far and cannot even keep up with traditional algorithms. This is most likely due to misalignment, but it might also be caused by other unknown factors. Since the poor performance can hardly be true, VGGFace was completely dropped from all experiments.

At this point, the opportunity is taken to summarize the key findings of this work. First, it can be said that occlusion of different types is not a significant problem for the tested networks. Faces with sunglasses are the most difficult for ArcFace-Mobile, but also for VGGFace2. The presence of different illuminations has hardly any influence on the performance of the networks. However, when different types of illumination came into play, the performance could be improved. Poses with a rotation angle of -45 to 45 degrees are no problem for state-of-the-art deep neural networks. Recognizing more challenging poses requires a deep network architecture or balanced training on different poses to achieve an acceptable performance. Many facial expressions can already be recognized well. The performance dropped significantly as soon as an extreme expression, such as a disgusted or screaming facial expression comes into play. Face recognition at long distances does not work at all because the faces become too small. Gender has a slight influence

on the performance of today's deep neural networks. Preprocessing also plays an important role in the face recognition process. Small changes in the alignment might have a significant impact on performance. The training database is a critical factor for performance in challenging environments. The choice of a deep network architecture can be helpful in extreme unconstrained environments. All deep neural networks used in the present work outperformed the traditional algorithms used by Günther et al. (2016) in all experiments. The dominance of deep learning in face recognition appears to be justified.

Although the results look promising, the limitations of this work must also be discussed. First, it must be said that based on these findings, one is not capable of naming a winner as far as deep neural networks are concerned. The experiments purely focus on challenging environments in face recognition. Other aspects, such as runtime and memory consumption, which also indicate performance, are not addressed. As already mentioned, the preprocessing of the images plays a very important role in the face recognition process. However, there were hardly any precise instructions for the required alignment of the faces. The determination of the eye positions was mostly done from a few sample images, from which an average value was calculated. Although much time was invested in finding the best eye positions for performance, there is a possibility that someone else will find others that perform better. Another point is the limitation to image databases. In Günther et al. (2017), experiments were also performed on the YouTube Face (Wolf et al., 2011) database, which consists of video data. However, mainly due to time constraints, no video database could be included into the experimental framework of this study. The evaluation of video databases involves some complex challenges. The faces have to be detected first, and the issue of failed detection must be considered. Otherwise, there would be the risk that a poor face detector would filter out challenging images in advance. Furthermore, it should be noted that despite isolated consideration of the individual face recognition aspects, biases of the database may be included in the results. For instance, the CAS-PEAL database only contains images of Asian people. It is indeterminate how large the influence of such a bias on face recognition performance might be. Finally, some of the databases used in this work may no longer be publicly available. For some experiments, it was necessary to rely on the resources of the research institute Idiap,[1] which still has a copy of many databases. Thus, the reproducibility of the experiments is limited.

For further research, it is recommended to extend the present experiments to other databases. Thus, face recognition aspects should be further investigated. Moreover, the experiments should be performed with databases that test similar things to support the findings of this work. Furthermore, the influence of peculiarities of the networks can be investigated in greater detail. So far, it has been very difficult to attribute positive or negative performance to the loss function or the network architecture. Gaps in the training database can quickly be used as a basis for argumentation, while other properties of deep neural networks are more speculative. Most importantly, future research should build on the results of this work. Today's face recognition algorithms are not capable of recognizing faces at long distances. In addition, the recognition rate of illumination or facial expressions can still be refined. Also, the influence of aging should be investigated in more detail since it is one of the main challenges of face recognition. This work could only consider aging within six months, which is a very short period of time. Researchers are already trying to make algorithms more robust against aging (Guo and Zhang, 2019). A long-term study on age intervals could bring new insights. Another important point to mention is the usage of hand-labeled landmarks. Dutta et al. (2015) already investigated facial landmark errors for some traditional algorithms. They found that the problem with alignment with eye positions is that there is no clear definition of the eye center and that this might affect performance. How much incorrectly detected landmarks affect the performance of deep learning algorithms is an area left open for future research. All in all, there is still room for improvement with deep learning methods in the area of face recognition.

[1] https://www.idiap.ch

Chapter 7

Conclusion

This work aimed to investigate the performance of pretrained state-of-the-art deep neural networks in challenging face recognition environments. First, an overview of some commonly used databases and deep learning was provided. Additionally, the reader was given a brief review of traditional methods before deep learning. Freely available pretrained state-of-the-art deep neural networks were selected for the experiments. Four networks were taken from ArcFace (Deng et al., 2019), and one was ultimately selected from VGGFace2 (Cao et al., 2018). The main implementation in the open-source software Bob (Anjos et al., 2012) was then explained. Before the experiments could start, the most ideal alignment for the networks had to be found. For this purpose, the eye positions of the input images were determined for each network. First, the networks were tested for performance on different face variations. These included occlusion, illumination, facial expressions, and poses. The experiments were also extended to other databases. Thus, further challenging conditions could be investigated in isolation, such as the influence of different distances, gender, age, and background variations. In addition, the performance of the networks was evaluated on the popular LFW benchmark. Finally, the results were examined and discussed in greater detail.

It was shown that occlusion has hardly any influence on the performance of the networks. Even if a large part of the face, such as the mouth or the eyes, is not visible, faces are correctly recognized in most cases. Illumination from different directions is no longer an obstacle for today's deep neural networks. However, different illumination types still constitute a gap for further research. Diversity of poses is also no problem, especially with small rotations of the face to the camera. For a rotation angle of up to 90 degrees, the network needs special training on facial poses to achieve an acceptable performance. Different facial expressions are usually not a problem. However, challenging expressions, such as screaming, may still affect the performance of the networks. An influence of different backgrounds could not be found within the scope of this work. Female faces are more difficult to recognize than male faces. When recognizing faces at longer distances, all networks failed almost completely.

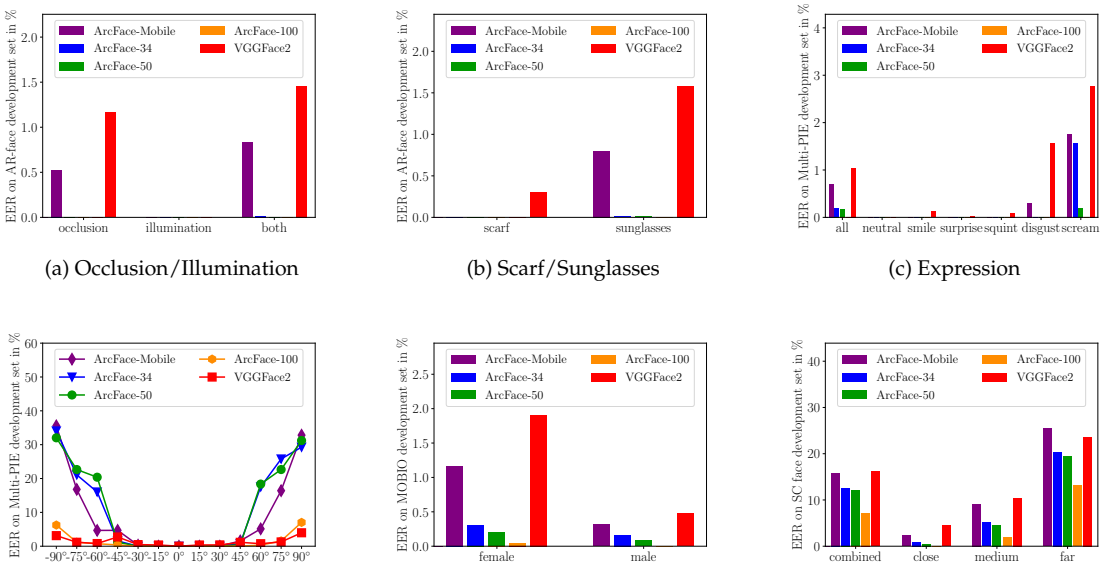
In summary, it can be said that the performance of deep neural networks depends more on the training database than on the network architecture or the loss function. If the training data does not contain a certain face variation, the network has no chance of recognizing the corresponding faces. ArcFace-Mobile performed better under illumination conditions than VGGFace2, which has a larger network architecture. Likewise, ArcFace-100 was outperformed by VGGFace2 in some experiments, especially for different poses. The structure of the training database may offer an explanation for such phenomena. Finally, the performance of state-of-the-art deep neural networks is well above that of traditional face recognition algorithms. However, there are still some challenges for current networks in unconstrained face recognition environments that may be resolved in future research.

Appendix A

Attachments

A.1 Evaluation on Development Set

Figure A.1 shows the results of the experiments on the development set of AR face, Multi-PIE, MOBIO, and SCface. Unlike the plots in Chapter 5, ERR values are reported. Since the threshold is calculated on the basis of the development set, the networks usually showed a slightly better performance on the development set than on the evaluation set. A gradation of performance can often be recognized in the ResNets of ArcFace. In other words, ArcFace-50 usually performs better than ArcFace-34, and ArcFace-100 performs better than ArcFace-50.



(a) Occlusion/Illumination (b) Scarf/Sunglasses (c) Expression

(d) Pose (e) MOBIO (f) SCface

Figure A.1: DEVELOPMENT SET. This figure reports the results on the development set.

A.2 Exact Numbers

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
occlusion	0.53%	0.01%	0.00%	0.00%	1.16%
illumination	0.00%	0.00%	0.00%	0.00%	0.00%
both	0.83%	0.01%	0.00%	0.00%	1.45%
scarf	0.00%	0.00%	0.00%	0.00%	0.30%
sunglasses	0.80%	0.01%	0.01%	0.00%	1.58%

Table A.1: AR FACE DEV. This table reports the ERR values for the AR face development set.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
occlusion	0.37%	0.02%	0.29%	0.00%	1.78%
illumination	0.00%	0.00%	0.00%	0.00%	0.39%
both	1.32%	0.33%	0.46%	0.44%	2.33%
scarf	0.80%	0.78%	0.78%	0.39%	0.48%
sunglasses	1.34%	0.19%	0.60%	0.19%	2.88%

Table A.2: AR FACE EVAL. This table reports the HTER values for the plots in Figure 5.1.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
all	0.69%	0.20%	0.17%	0.00%	1.04%
neutral	0.00%	0.00%	0.00%	0.00%	0.01%
smile	0.00%	0.00%	0.00%	0.00%	0.12%
surprise	0.00%	0.00%	0.00%	0.00%	0.02%
squint	0.01%	0.00%	0.00%	0.00%	0.09%
disgust	0.31%	0.01%	0.01%	0.00%	1.56%
scream	1.76%	1.56%	0.20%	0.00%	2.77%

Table A.3: EXPRESSION DEV. This table reports the ERR values for the Multi-PIE development set on protocol E.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
all	0.54%	0.20%	0.09%	0.17%	0.82%
neutral	0.77%	0.00%	0.00%	0.00%	1.92%
smile	0.00%	0.00%	0.00%	0.00%	0.79%
surprise	0.00%	0.00%	0.00%	0.00%	4.63%
squint	0.01%	0.00%	0.00%	0.00%	2.34%
disgust	1.19%	0.78%	0.00%	1.55%	3.49%
scream	1.07%	1.00%	0.28%	0.00%	1.79%

Table A.4: EXPRESSION EVAL. This table reports the HTER values for the plot in Figure 5.2(c).

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
-90	35.57%	34.06%	32.03%	6.25%	3.12%
-75	16.79%	21.09%	22.66%	1.17%	1.17%
-60	4.67%	16.03%	20.39%	0.73%	0.79%
-45	4.69%	1.55%	1.17%	0.38%	2.73%
-30	0.31%	0.00%	0.00%	0.00%	0.47%
-15	0.00%	0.00%	0.00%	0.00%	0.37%
0	0.00%	0.00%	0.00%	0.00%	0.01%
15	0.00%	0.00%	0.00%	0.00%	0.36%
30	0.00%	0.00%	0.00%	0.00%	0.39%
45	1.56%	0.78%	0.46%	0.06%	1.15%
60	5.13%	17.53%	18.36%	0.30%	0.74%
75	16.41%	25.78%	22.66%	1.56%	1.26%
90	32.81%	29.27%	31.23%	7.03%	3.97%

Table A.5: POSE DEV. This table reports the ERR values for the Multi-PIE development set on protocol P.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
-90	32.35%	32.77%	27.71%	5.68%	3.75%
-75	15.40%	22.05%	20.50%	0.97%	2.53%
-60	4.04%	16.23%	17.19%	0.75%	2.12%
-45	4.68%	2.47%	1.45%	0.38%	2.72%
-30	0.53%	0.77%	0.77%	0.38%	1.78%
-15	0.77%	0.00%	0.00%	0.00%	0.80%
0	0.77%	0.00%	0.00%	0.00%	1.92%
15	0.38%	0.00%	0.00%	0.00%	0.41%
30	0.96%	0.38%	0.19%	0.00%	1.00%
45	2.06%	1.48%	0.53%	0.03%	1.29%
60	4.92%	18.92%	18.30%	0.11%	1.12%
75	14.59%	23.85%	22.28%	1.36%	1.90%
90	29.61%	27.65%	29.09%	6.48%	2.71%

Table A.6: POSE EVAL. This table reports the HTER values for the plot in Figure 5.2(d).

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
background	100.00%	100.00%	100.00%	100.00%	100.00%
distance	100.00%	100.00%	100.00%	100.00%	100.00%
aging	100.00%	100.00%	100.00%	100.00%	100.00%
expression	100.00%	100.00%	100.00%	100.00%	99.94%
accessory	99.61%	99.82%	99.87%	99.91%	96.81%
lighting	92.20%	99.02%	99.29%	99.60%	86.22%

Table A.7: CAS-PEAL. This table reports the exact numbers for the plot in Figure 5.3.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
female	1.16%	0.31%	0.20%	0.04%	1.90%
male	0.32%	0.16%	0.08%	0.00%	0.48%

Table A.8: MOBIO DEV. This table reports the ERR values for the MOBIO development set.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
female	0.83%	0.15%	0.05%	0.00%	0.51%
male	0.18%	0.11%	0.08%	0.03%	0.39%

Table A.9: MOBIO EVAL. This table reports the HTER values for the plot in Figure 5.4.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
combined	15.76%	12.42%	12.12%	7.09%	16.19%
close	2.27%	0.91%	0.44%	0.00%	4.50%
medium	9.09%	5.07%	4.61%	1.82%	10.38%
far	25.45%	20.35%	19.55%	13.18%	23.64%

Table A.10: SCFACE DEV. This table reports the ERR values for the SCface development set.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
combined	18.18%	13.02%	11.71%	7.84%	15.19%
close	1.09%	1.10%	0.65%	1.16%	3.69%
medium	9.42%	4.76%	4.34%	3.00%	8.00%
far	28.33%	22.67%	18.26%	11.89%	24.25%

Table A.11: SCFACE EVAL. This table reports the HTER values for the plot in Figure 5.5.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
2.0.1	99.67%	99.75%	99.76%	99.76%	96.10%
2.0.2	99.74%	99.76%	99.76%	99.76%	97.68%
2.0.4	94.97%	99.59%	99.87%	99.98%	68.19%

Table A.12: FRGC. This table reports the CAR @ 0.001% FAR for the plots in Figure 5.6.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
Good	94.69%	96.88%	97.00%	97.60%	96.91%
Bad	75.28%	85.81%	90.17%	92.96%	73.92%
Ugly	64.03%	81.59%	86.75%	91.84%	50.86%

Table A.13: GBU. This table reports the CAR @ 0.001% FAR for the plots in Figure 5.7.

	ArcFace-Mobile	ArcFace-34	ArcFace-50	ArcFace-100	VGGFace2
fold1	96.83%	96.50%	95.50%	98.67%	99.33%
fold2	96.67%	97.33%	96.83%	97.67%	98.83%
fold3	97.00%	95.00%	95.17%	95.50%	99.00%
fold4	96.67%	95.83%	96.83%	97.00%	98.67%
fold5	94.50%	95.00%	95.67%	95.00%	98.50%
fold6	95.67%	95.17%	94.50%	96.50%	99.00%
fold7	96.50%	96.67%	95.50%	96.17%	98.83%
fold8	96.67%	96.83%	97.17%	97.00%	99.17%
fold9	97.33%	95.50%	96.67%	98.33%	99.33%
fold10	97.33%	95.83%	93.83%	97.00%	99.50%
mean	96.52%	95.97%	95.77%	96.88%	99.02%
std	0.81%	0.78%	1.04%	1.10%	0.30%

Table A.14: LFW. This table reports the exact numbers for the experiments on LFW.

List of Figures

List of Tables

List of Listings

Bibliography

Anjos, A., Günther, M., de Freitas Pereira, T., Korshunov, P., Mohammadi, A., and Marcel, S. (2017). Continuously reproducing toolchains in pattern recognition and machine learning experiments. In *International Conference on Machine Learning (ICML)*.

Anjos, A., Shafey, L. E., Wallace, R., Günther, M., McCool, C., and Marcel, S. (2012). Bob: a free signal processing and machine learning toolbox for researchers. In *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition*.

Chen, S., Liu, Y., Gao, X., and Han, Z. (2018). MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. In *Biometric Recognition*, pages 428–438, Cham. Springer International Publishing.

Deng, J., Guo, J., Niannan, X., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.

Dutta, A., Günther, M., El Shafey, L., Marcel, S., Veldhuis, R., and Spreeuwers, L. (2015). Impact of Eye Detection Error on Face Recognition Performance. *IET Biometrics*.

Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., and Zhao, D. (2008). The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38:149–161.

Grgic, M., Delac, K., and Grgic, S. (2011). Scface–surveillance cameras face database. *Multimedia Tools and Applications*, 51(3):863–879.

Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, 28(5):807–813.

Günther, M., Haufe, D., and Würtz, R. P. (2012). Face Recognition with Disparity Corrected Gabor Phase Differences. In *Artificial Neural Networks and Machine Learning – ICANN 2012*, pages 411–418, Berlin, Heidelberg. Springer.

Günther, M., Shafey, L. E., and Marcel, S. (2016). *Face Recognition in Challenging Environments: An Experimental and Reproducible Research Survey*, pages 247–280. Springer International Publishing, Cham.

Günther, M., Shafey, L. E., and Marcel, S. (2017). 2D face recognition: An experimental and reproducible research survey. Technical Report Idiap-RR-13-2017, Idiap.

Günther, M., Wallace, R., and Marcel, S. (2012). An open source framework for standardized comparisons of face recognition algorithms. In *Computer Vision - ECCV 2012. Workshops and Demonstrations*, volume 7585, pages 547–556. Springer Berlin Heidelberg.

Guo, G. and Zhang, N. (2019). A survey on deep learning based face recognition. *Computer Vision and Image Understanding*, 189:102805.

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *European Conference on Computer Vision*, page 87–102. Springer.

Guresen, E. and Kayakutlu, G. (2011). Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, 3:426–433. World Conference on Information Technology.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Heusch, G., Rodriguez, Y., and Marcel, S. (2006). Local binary patterns as an image preprocessing for face authentication. In *7th International Conference on Automatic Face and Gesture Recognition*, pages 9–14.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Hsu, G.-S. J., Wu, H.-Y., and Yap, M. H. (2020). A Comprehensive Study on Loss Functions for Cross-Factor Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 826–827.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. (2007). Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical report, University of Massachusetts, Amherst.

Khoury, E., El Shafey, L., McCool, C., Günther, M., and Marcel, S. (2014). Bi-Modal Biometric Authentication on Mobile Phones in Challenging Conditions. *Image and Vision Computing*, pages 1147–1160.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, pages 1097–1105.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). SphereFace: Deep Hypersphere Embedding for Face Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 212–220.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-Margin Softmax Loss for Convolutional Neural Networks. *International Conference on Machine Learning*, 48(3):507–516.

Lui, Y. M., Bolme, D., Phillips, P. J., Beveridge, J. R., and Draper, B. A. (2012). Preliminary studies on the Good, the Bad, and the Ugly face recognition challenge problem. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–16.

Martínez, A. and Benavente, R. (1998). The AR face database. Technical Report 24, Computer Vision Center.

Masi, I., Wu, Y., Hassner, T., and Natarajan, P. (2018). Deep Face Recognition: A Survey. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images*, pages 471–478.

Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., Jain, A. K., Niggel, W. T., Anderson, J., Cheney, J., and Grother, P. (2018). IARPA Janus Benchmark - C: Face Dataset and Protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165.

McCool, C., Marcel, S., Hadid, A., Pietikäinen, M., Matejka, P., Cernocký, J., Poh, N., Kittler, J., Larcher, A., Lévy, C., Matrouf, D., Bonastre, J., Tresadern, P., and Cootes, T. (2012). Bi-modal person recognition on a mobile phone: Using mobile phone data. In *2012 IEEE International Conference on Multimedia and Expo Workshops*, pages 635–640.

Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., and Zhang, D. (2021). Biometrics Recognition Using Deep Learning: A Survey.

O' Mahony, N., Campbell, S., Carvalho, A., Krpalkova, L., Velasco-Hernandez, G., Harapanahalli, S., Riordan, D., and Walsh, J. (2019). Deep Learning vs. Traditional Computer Vision. *Science and Information Conference*, pages 128–144.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*.

Phillips, P. J., Beveridge, J. R., Draper, B. A., Givens, G., O'Toole, A. J., Bolme, D. S., Dunlop, J., Lui, Y. M., Sahibzada, H., and Weimer, S. (2011). An introduction to the good, the bad, & the ugly face recognition challenge problem. In *2011 IEEE International Conference on Automatic Face Gesture Recognition*, pages 346–353.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Jin Chang, Hoffman, K., Marques, J., Jaesik Min, and Worek, W. (2005). Overview of the face recognition grand challenge. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 947–954.

Ramírez-Gutiérrez, K., Cruz-Pérez, D., and Pérez-Meana, H. (2010). Face Recognition and Verification Using Histogram Equalization. In *ACS*, page 85–89. World Scientific and Engineering Academy and Society.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823.

Sengupta, S., Chen, J., Castillo, C., Patel, V. M., Chellappa, R., and Jacobs, D. W. (2016). Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision*, pages 1–9.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper With Convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Tan, X. and Triggs, B. (2010). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Wallace, R., McLaren, M., Mccool, C., and Marcel, S. (2011). Inter-session Variability Modelling and Joint Factor Analysis for Face Authentication. In *2011 International Joint Conference on Biometrics*, pages 1–8.

Wang, H., Li, S., and Wang, Y. (2004). Face recognition under varying lighting conditions using self quotient image. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 819– 824.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

Wang, M. and Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429:215–244.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A Discriminative Feature Learning Approach for Deep Face Recognition. In *European conference on computer vision*, pages 499–515. Springer International Publishing.

Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A. K., Duncan, J. A., Allen, K., Cheney, J., and Grother, P. (2017). IARPA Janus Benchmark-B Face Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 592–600.

Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning Face Representation from Scratch.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

Zhang, W., Shan, S., Gao, W., Chen, X., and Zhang, H. (2005). Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 786– 791.

Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. (2019a). Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 10823–10832.

Zhang, X., Zhao, R., Yan, J., Gao, M., Qiao, Y., Wang, X., and Li, H. (2019b). P2sgrad: Refined gradients for optimizing deep face models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 9906–9914.

Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D. L., and Weng, J. (1998). Discriminant Analysis of Principal Components for Face Recognition. In *Face Recognition: From Theory to Applications*, pages 73–85, Berlin, Heidelberg. Springer.

Zheng, Y., Pal, D. K., and Savvides, M. (2018). Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 5089–5097.