Frontal to Profile Face Recognition with Rank Lists

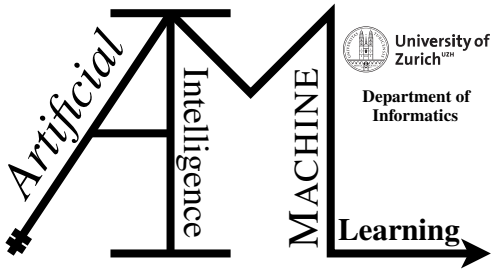
Bachelor Thesis

Tom Marco Wartmann

16-929-630

Submitted at
March 22 2021

Thesis Supervisor
Prof. Dr. Manuel Günther



Bachelor Thesis

Author: Tom Marco Wartmann, tom.wartmann@uzh.ch

Project period: 28.09.2020 - 28.3.2021

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

Acknowledgements

I would like to take this opportunity to thank Prof. Dr. Manuel Günther for supervising this bachelor thesis. His experience, helpful advice and constructive criticism were invaluable for this work. I have appreciated the inspiring and motivating nature of the numerous personal conversations. I am also very grateful for the support and patience of my family and friends during this time.

Abstract

Accuracy in face recognition, especially with machine learning and artificial neural networks, has reached new records in recent years. However, the direct verification of two face images with a large difference in pose is still a big challenge even for these methods. An earlier developed method avoids the direct comparison of two images by ranking similarities to reference images and then determining the similarity of two faces by the similarity of their rank lists. This method is combined in this thesis with state-of-the-art neural networks to evaluate the value of rank lists and reference images, especially in the recent context. The results are demonstrated on the two datasets *Celebrities in Frontal-Profile* (CFP) and *Cross-Pose Labeled Faces in the Wild* (CPLFW) and with four different convolutional neural networks. In this bachelor thesis it could be shown that this combination, applied on CPLFW, is not significantly different than the most accurate neural network by its own. Additionally, two new methods were developed that can calculate the similarity of two rank lists in this context more accurately than all known methods from the literature. Furthermore, it was shown that in addition to the low ranks, i.e. similar reference persons, the high ranks, i.e. dissimilar reference persons, are also extremely informative about a person's identity.

Zusammenfassung

Die Genauigkeit in der Gesichtserkennung, im Speziellen mit maschinellem Lernen und künstlichen neuronalen Netzwerken, erreichte in den letzten Jahren neue Rekorde. Die direkte Verifikation zweier Bilder mit Gesichter mit einem grossen Unterschied in der Ausrichtung, sind jedoch auch für diese Methoden immer noch eine grosse Herausforderung. Eine ältere Methode vermeidet den direkten Vergleich zweier Bilder, indem sie Ranglisten von Ähnlichkeiten zu Referenzbildern erstellt und die Ähnlichkeit zweier Gesichter dann durch die Ähnlichkeit deren Ranglisten bestimmt wird. Diese Methode wird in dieser Arbeit mit modernsten Neuronalen Netzwerken kombiniert um den Mehrwert der Ranglisten und Referenzbildern, insbesondere im aktuellen Kontext, zu evaluieren. Die Resultate werden auf den zwei Datensätzen *Celebrities in Frontal-Profile* (CFP) und *Cross-Pose Labeled Faces in the Wild* (CPLFW) und mit vier verschiedenen Neuronalen Netzwerken demonstriert. In dieser Bachelorarbeit konnte gezeigt werden, dass sich diese Kombination bei der Posen übergreifenden Verifikation auf dem CPLFW Datensatz nicht signifikant von dem genausten Neuronalen Netzwerk alleine unterscheidet. Ausserdem wurden zwei neue Verfahren entwickelt, welche die Ähnlichkeit zweier Ranglisten in diesem Zusammenhang genauer berechnen können, als alle bekannten Methoden aus der Literatur. Ferner wurde gezeigt, dass neben den tiefen Rängen, also ähnlichen Referenzpersonen speziell auch die hohen Ränge, also unähnliche Referenzpersonen am aussagekräftigsten über die Identität einer Person sind.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It was possible early on for computers to solve mathematical equations faster than an adult human, and at some point they were also able to play chess better than humans. However, it was and still is incredibly difficult for them to achieve the perception or mobility of a human. The research field of computer vision tries to teach computers an understanding of images and videos that the human visual system already has in its early years of development.

Due to the wide range of commercial and law enforcement use, face recognition is one of the best explored and documented image analysis fields (Zhao et al., 2003). During the research on facial recognition methods, a great variety of different data sets with different framework conditions were experimented with. It started with small data sets of about 100 images from a controlled laboratory environment and is now moving towards billions of images of people in real world environments. The unconstrained setting of images from the real world is much more difficult to handle, due to different illumination, expressions or variation of the pose. Pose variation makes the recognition task particularly difficult when large parts, in extreme cases half of the face, are not visible.

One application of face recognition is verification, meaning to decide if two presented images of persons have the same or different identity. There is a wide variety of approaches that try to solve this problem. Some early methods extract facial features like the eyes, mouth and nose and then compares the geometric relations between them to the ones of another image (Brunelli and Poggio, 1993). Some more sophisticated methods compare images by converting the faces into labelled graphs, with prominent facial points being a node. For this conversion into a graph, different filters are used, whereby a frequently used filter is for example a set of Gabor wavelets (Wiskott et al., 1997). These image graphs are then compared by a similarity function, which then determines the original similarity of the two images.

Nowadays, neural networks are being used more and more for face recognition. A neural network is a statistical classification method and usually consists of a number of layers that weight the input in a certain way. This weighting is optimised by training data so that the neural network can change the input as accurately as possible into the desired output. This idea was inspired by the human brain and has its beginnings in the middle of the 20th century (Kroese et al., 1993). The accuracy of this method depends very heavily on the size and quality of the training data. It is therefore not surprising that with the general increase in computational power and the accumulation of larger amounts of data, the accuracy of neural networks also increased in recent years. A convolutional neural network is a subcategory of neural networks that has at least one convolutional layer and is often used in image processing. The exact functionality of a convolutional layer is very well explained in O'Shea and Nash (2015), but can be thought of as a filter that extracts, for example, the edges and corners of an image. Current face recognition methods using convolutional neural networks, such as described in Sun et al. (2015) or Deng et al. (2019), can determine with over 99% accuracy whether two people are the same or different identities in

frontal images in an unconstrained environment. In fact, in the verification task on two frontal images, neural networks reach even a higher accuracy than humans, while humans still perform better when comparing a profile to a frontal image (Sengupta et al., 2016). This approach still has room for improvement under large variations in pose, scale or illumination (Jafri and Arabnia, 2009). There are some methods that have been developed especially for face recognition between pose variation to explicitly compensate for this deficit. One of these is the Ranked List method (Müller et al., 2013), which avoids the direct comparison of images in different poses by using a cohort where the images have the same orientation. Another, newer method called PF-cpGAN (Taherkhani et al., 2020) projects the frontal and profile images onto a common subspace and bases the verification on that.

The Ranked List method, as explained in (Müller et al., 2013), uses a reference data set, also called cohort, and creates rank lists from it, which can then be compared and does not require any information about the orientation of the faces. The aim of this work is to combine a state-of-the-art neural network with this Ranked List method. An important step in this method is to determine the similarity of two rank lists. This combination of methods is compared in this work to face recognition using only neural networks without the use of rank lists. Furthermore, different algorithms for determining the similarities of the two rank lists are compared in this context. The performance is demonstrated and compared by verification experiments on two data sets with unconstrained images from the real world. Finally should the results of this work be publicly accessible and comprehensible.

Chapter 2

Related Work

The method for face verification presented in this work is basically based on two approaches from the literature. On the one hand the idea with the similarity of two rank lists, which were created by means of a cohort, which is also called model database or reference database. And on the other hand on face verification using deep learning. The literature relevant to this work is explained in the next two sections.

2.1 Ranked List

2.1.1 Overview

The idea of using rank lists, also called doppelgänger lists to reference data, to improve verification over different face orientations and illuminations is discussed in several papers. The first publication of this procedure was in Müller et al. (2007) followed by Müller (2010), Schroff et al. (2011) and Müller et al. (2013). These differ mainly in the types of rank list similarity functions and datasets that are used for the evaluation of the results.

This type of method is in the following called Ranked List method and it is based on two assumptions. The first assumption is that one person can be described in relation to other persons. An everyday example would be when it is possible to explain the appearance of a person by saying "she is very similar to person *A*, somewhat similar to person *B* but not at all similar to person *C*" and another person understands who is meant. The second assumption is that two people who are similar in one facial orientation are also similar in another facial orientation. Hence, if the face of person *A* is very similar to person *B* in two profile pictures, they are also very similar in two frontal pictures. These two assumptions can be used to indirectly compare two images with possibly different orientations of the face. To do this, two rank lists are created based on the similarities of each image to reference images. The similarity of the two rank lists can now be calculated by an appropriate rank list similarity function.

An illustration of the workflow of the Ranked List method can be seen in Figure 2.1. In all the papers mentioned, it basically consists of 3 steps.

1. The similarities of the two images to a cohort are calculated.
2. Rank lists are created, based on the similarities to the cohort.
3. The two rank lists are compared with each other.

In the following, these three steps are explained in more detail and how to distinguish the different works from the literature.

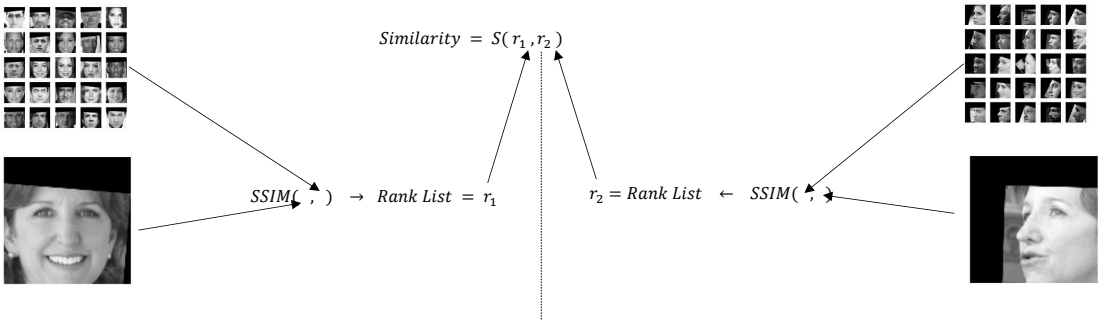


Figure 2.1: Illustration of image recognition method using rank lists, demonstrated on aligned images from the CFP data set. The similarities between the image and all images from a cohort are calculated directly (using SSIM) and from that a rank list is created. The similarity of the two rank lists is evaluated using a rank list similarity function $S()$. This illustration follows the rank list idea from Schroff et al. (2011).

The first step of the Ranked List algorithm is to calculate the similarities of the two test images, to each of the cohort images. In the first version of the Ranked List method in Müller et al. (2007), the image is converted into a model graph with nodes on the facial landmarks, like the eyes and nose. This conversion is achieved by an automatic bunch graph generator, which needs some manually labelled images at first. The structure of the facial landmark at each node then is represented using multiple Gabor wavelets, also called Gabor jets. A gabor wavelet is a function that is often used to simulate the image filter behaviour of simple cells in the primary visual cortex (Daugman, 1985). In Müller (2010) and Müller et al. (2013), they also used graph matching and they tried Gabor jets, and Local Gabor Binary Patterns (LGBPHS) as locale features. The similarity between different images are then calculated nodewise with different kinds of jet similarity functions (Müller et al., 2007), (Müller et al., 2013). When using graphs, images with different poses are not directly comparable because they may have different numbers of nodes and highly distorted features (Müller et al., 2013).

In Schroff et al. (2011) the similarity of two images is computed with the Structural Similarity Index (SSIM) (Zhou Wang et al., 2004), which calculates the similarity of each pixel of both images, with the mean and standard deviation of a window around this pixel. This pixel-wise measurement is inappropriate to use across poses, but works well on evenly aligned images (Schroff et al., 2011).

Using these similarity calculation methods, the two images are now compared with all reference images. When knowing about the orientation of the faces, test images are only compared to images from the reference data base with the same orientation. The reference images consist of more than one image with different poses of multiple identities (Müller et al., 2007). In addition, images with different lighting conditions were included in Müller (2010), Schroff et al. (2011) and Müller et al. (2013). In general they have high similarities to images with the same viewing direction and lower similarities with images with different viewing directions (Schroff et al., 2011). This follows the findings of Moses Yael (1994), where a different viewing direction has almost always a bigger influence on the similarity of two images than a change of identity. This behaviour is illustrated in the figure 2.2, where the picture of person A is more similar to a picture of person B with the same facial orientation than to a picture of himself with a different orientation. There-

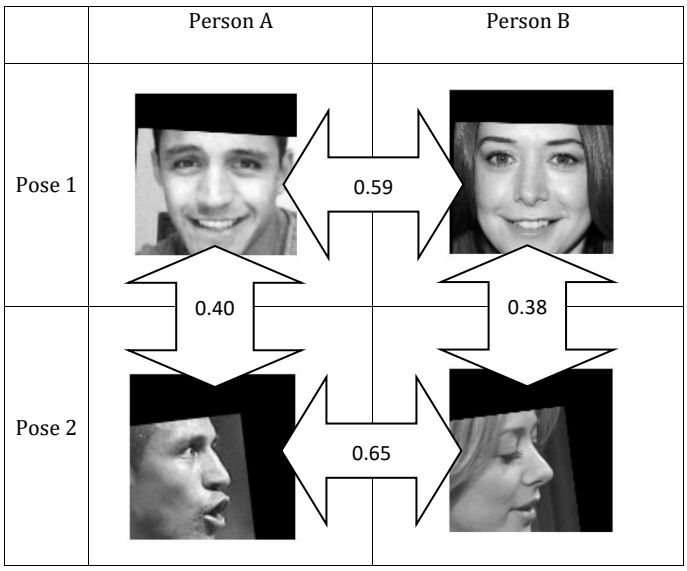


Figure 2.2: Profile and frontal image of two people from the CFP database. Possible similarities are shown in the arrows between the pictures.

fore, taking the image with the highest similarity of each identity, from the comparison with the cohort, have all a similar situation (Schroff et al., 2011).

In a second step, the list containing the highest similarity for each individual from the cohort is converted into a rank list. The reference person with the image having the highest similarity gets rank 0, followed by the person with the second most similar person with rank 1 and so on (Müller et al., 2013). This rank list now serves as a representation of the former image and can be compared with a rank list of another image.

In the third and final step, two rank lists are compared with each other. To compute the similarity of two rank lists, there are a great number of methods. There are different statistical approaches to determine the similarity of two rankings. The number of exchanges of ranks can be counted to get from one rank list to the other (i.e. Kendall Tau), or the difference in ranks can be summed for each reference person (i.e. Spearman rank). A larger number of substitutions, or a larger sum of differences respectively, lead to more dissimilar rank lists. However, in the literature on face recognition using the Ranked List method, comparison functions that do not consider all ranks equally important have been favoured. The basic idea is that identities with lower ranks tell more about the similarity of the test images than identities with higher ranks. Or in other words, dissimilarity to the reference images does not imply much about the similarity of the two images (Müller, 2010). Therefore, the functions in Müller et al. (2007) , Müller (2010) and Müller et al. (2013) weight higher ranks lower and in Schroff et al. (2011) only the lowest few ranks are used. In addition, rank list comparison with a spiking neural network was tested, which shows the exact same recognition rates as the other rank list comparison methods (Müller et al., 2013). The similarity of the two rank lists is after all the similarity of the initial two test images.

The evaluation of the performance has been done in the literature on the verification task with different standard data sets. Following some of the results are presented, which used data with

no label from the orientation of the face.

On the FRGC (Phillips et al., 2006) data set, Müller et al. (2013) reached a higher accuracy on face verification than the principal component analysis (PCA), as a reference method. But the results are below the median of commercial recognition systems, published in Phillips et al. (2006).

The paper of Schroff et al. (2011) uses a separate database (Multi-PIE (Gross et al., 2010)) as reference images and shows the performance on FacePix (Black Jr. et al., 2002) and LFW (Huang et al., 2008). The method was applied using SSIM as the image similarity score and the rank list comparison method which considers only the first 100 ranks. The Ranked List method performs on the LFW data set, with a accuracy of 70.8% about 1-2% better than their comparison methods TPLBP, FPLBP and SIFT (Schroff et al., 2011). Compared to FPLBP increased the accuracy in FacePix by 2.5% when comparing two images with face orientation between -30 and 30 degrees and increased up to 4.2% when comparing images with even larger pose difference. In conclusion, it can be said that the latest experiments using the Ranked List method increased the accuracy in comparison to some benchmarks.

2.1.2 Definitions

In this section, the mentioned methodology of the rank list creation is expressed in formulas and the rank list comparison functions from the mentioned papers stated. These notations and definitions will also be the basis for the method used in this work.

Rank List Creation

In the following, the two test images to be compared are called probe and gallery image and denoted as P and G, respectively. The cohort M contains N_M individuals, each having a number N_S of images in different situations. Each image of the cohort can be written as M_m^s where $m \in \{1, ..., N_M\}$ and $s \in \{1, ..., N_S\}$. First the similarity between P, respective G, and all images out of M are computed and stored in Π respective Γ:

$$\Pi_m^s = sim(P, M_m^s), m = 1...N_M, s = 1...N_S \tag{2.1}$$

$$\Gamma_m^s = sim(G, M_m^s), m = 1...N_M, s = 1...N_S \tag{2.2}$$

In this case, the similarity is calculated using the cosine function, but there are other calculation options which will be discussed later. The similarity function $sim(u, v)$ receives the extracted features of two images as one dimensional vectors and returns a similarity value $\in (-1, 1)$, where a value of 1 imply u is equal to v.

$$sim(u, v) = \frac{u \cdot v}{\|u\|\|v\|} \tag{2.3}$$

A list π, respective γ, is created, containing the maximal similarity over all situations s of every individual m.

$$\pi(m) = max(\Pi_m^1, ..., \Pi_m^{N_S}) \tag{2.4}$$

$$\gamma(m) = max(\Gamma_m^1, ..., \Gamma_m^{N_S}) \tag{2.5}$$

Finally the two lists π, respective γ, are converted into rank lists by replacing the highest value by 0, the second highest with 1, etc. If two values are the same, they get the following rank. For example, if the highest two similarities are both equal to 0.801, one of the two randomly gets rank 0 and the other rank 1. The ranks of the two rank lists range from 0 to $N_M - 1$.

Rank List Comparison

To compare two ranked lists with equal length, there are several functions used in literature. In the following we denote π and γ as the two rank lists and $S(\pi, \gamma)$ as the similarity function, which returns the measure of the similarity of the two rank lists. The higher the value of S, the more similar are lists π and γ .

The function used in Müller et al. (2013) fulfils three requirements. First the similarity value S lies within 0 and 1 and is equals 1 when both lists are equal. Second, the similarity value is higher when more ranks with the same model index are the same. And third, low ranks should have a bigger influence than high ranks. This decrease of the importance with higher ranks is reached with the parameter $\lambda \in [0.9, 1)$.

$$S_{2013muller}(\pi, \gamma) = \frac{1}{F} \sum_{m=0}^{N_M - 1} \lambda^{\pi(m)+\gamma(m)} \tag{2.6}$$

$$F = \sum_{m=0}^{N_M - 1} \lambda^{2m} \tag{2.7}$$

Where N_M is the length of the rank lists and F a normalization factor.

Another function (Müller, 2010) having the same three requirements as the one above but does not have a parameter is the following.

$$S_{2010muller}(\pi, \gamma) = \frac{1}{F} \sum_{m=0}^{N_M - 1} \frac{1}{\sqrt{\pi(m) + \gamma(m)}} \tag{2.8}$$

$$F = \sum_{m=0}^{N_M - 1} \frac{1}{\sqrt{2m + 1}} \tag{2.9}$$

The function used in Schroff et al. (2011) does not weight low ranks higher and thus doesn't fulfil the third requirement from above. Instead they only take the first k ranks into account. Where $[\bullet]_+$ is equivalent to $max(\bullet, 0)$.

$$S_{schroff}(\pi, \gamma) = \sum_{m=0}^{N_M - 1} [k + 1 - \pi(m)]_+ \cdot [k + 1 - \gamma(m)]_+ \tag{2.10}$$

2.2 Deep Learning

Using deep learning, an extremely high degree of accuracy in face recognition has been achieved in recent years. A simplified illustration of face recognition using neural networks is shown in the figure 2.3. The input of this network are the values of each pixel of an image and the output a vector, which can be interpreted as a representation of the person's identity. This vector of numbers is often also called *extracted features* of the image. To verify two images, the extracted features are compared with one another. In the last years and in the following networks, this is calculated with the cosine distance, where a small distance indicates for more similar identities.

The creation of such neural networks can be done using a variety of architectures, loss functions and training data. As the accuracy increases, so does the complexity and often the size of these networks.

One of the numerous networks for face verification is the publicly available VGG network presented in Parkhi et al. (2015). It consists of 36, among others convolutional, layers and was

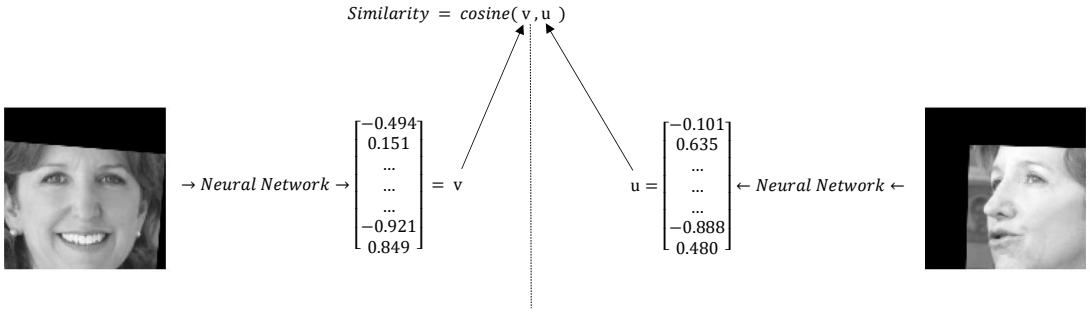


Figure 2.3: Calculation of the similarity of two images by means of a neural network. The features of the image are extracted by a neural network and then compared with a similarity function, in this example with the cosine. This is equivalent to the Baseline method in this work.

optimized by triplet loss training. It was trained on 2.6 million images of 2,622 individuals, of which 95% are frontal and 5% profile images. As a consequence, this network performs well in frontal to frontal comparisons and worse in cross-pose comparisons. The VGG network, despite its relatively simple architecture, achieves comparable results on frontal facial verification to state-of-the-art methods at that time. Applied to the LFW dataset (Huang et al., 2008), which is a standard benchmark for frontal face verification, the VGG-16 network achieves an accuracy of 98.95% (Parkhi et al., 2015).

Another, more sophisticated network for face verification was published in Deng et al. (2019) and is publicly available. This deep convolutional neural network differs from other modern networks mainly by the loss function used, which in this case is the Additive Angular Margin Loss (ArcFace). Such a loss function is needed to make decisions in a classification model, but its explanation is beyond the scope of this paper. They used a ResNet50 and a ResNet100 architecture for the neural network (He et al., 2016) and trained it on a refined version of MS1M data set (Guo et al., 2016) containing about 10 million images of 100 thousand people in a wide variety of poses, which is one of the largest public available databases.

With the ResNet100 and this optimized ArcFace loss function, all other state-of-the-art methods have been surpassed and very high accuracies are achieved on many test datasets. With an accuracy of 99.53% (Deng et al., 2019) on the LFW (Huang et al., 2008) dataset, the maximum of the verification task was reached and new datasets with greater pose variation had to be created, such as CPLFW (Zheng and Deng, 2018). Even on the cross-pose dataset CFP (Sengupta et al., 2016), accuracies of 95.56% (Deng et al., 2019) are achieved with this neural network.

Chapter 3

Approach

This section explains the Ranked List method in combination with neural networks and the terminology that is used in this work. The concept is illustrated in figure 3.1. The two test images to compare are called probe and gallery image. The cohort contains some persons, each having images in multiple poses. The used data is divided into a training and a test dataset. The gallery and probe image are taken from the test data set and the model images from the training data set. To receive a rank list, the probe and gallery image are first compared to all images of all identities from the cohort. This is done by means of a neural network that extracts the features and a distance function to compute the similarity of the extracted features. Only the most similar image of each individual from the cohort is further considered. The result are two lists of similarities between the probe/gallery image and the most similar image of every individual from the cohort. Those two lists are converted into rank lists and then compared to each other.

In this work we use for image comparison three state-of-the-art and one older pretrained Deep Convolutional Neural Networks (DCNN). To calculate the similarities between the rank lists, known functions from the literature as well as new developed methods are used. This chapter introduces first the used data (*Data*, section: 3.1), secondly defines the Ranked List method and the Baseline method exactly (*Method*, section: 3.2) and finally in the chapter *Experiments* (chapter: 4) shows the results of applied experiments.

3.1 Data

Data sets are generally divided into two groups, one to train the method and the other to calculate the performance of the method. The first of these is called the training data set and the second the test data set. Facial recognition data usually consists of a collection of images of a face and the associated identity of the person. Datasets specifically designed for cross-pose methods also may have the orientation of the face labelled. Since the aim of this work is to develop a method that works without knowing the orientation of a face, this information is ignored in the following.

The two datasets CFP and CPLFW, which are used in this work, are explained in more detail below.

3.1.1 Celebrities in Frontal-Profile

The Celebrities in Frontal-Profile (CFP) database (Sengupta et al., 2016) contains images of 500 famous celebrities, having 10 frontal and 4 profile labelled images each. Where *frontal* images have a yaw rotation under 10 degrees and *profile* images one over 60 degrees of the absolute frontal

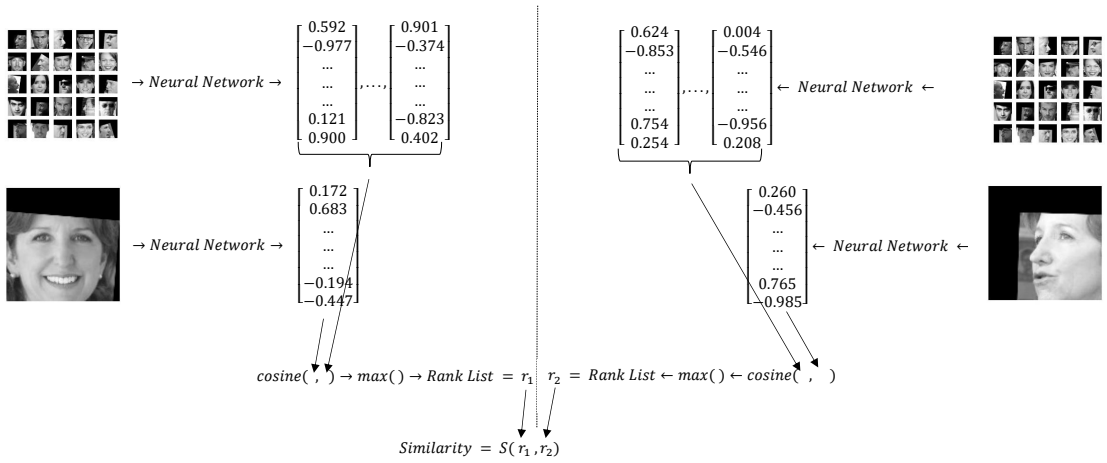


Figure 3.1: Illustration of the Ranked List method implemented in this work. The features are first extracted by a neural network, then the similarity between the test image and each image from the cohort computed, using the $cosine()$ function. Then, the rank lists are generated based on the maximum ($max()$) similarity value to each individual of the cohort. The final similarity is calculated by a rank list similarity function $S()$.

position. The pictures are taken in a real life environment, meaning different backgrounds, illuminations, expressions and accessories like glasses, earrings and headphones and have various sizes. This data set comes with 30 facial landmarks of every image, labelled by a combination of an automated key-point detector and Amazon Mechanical Turk workers, and two evaluation protocols, one for frontal to frontal and one for frontal to profile experiments. Both of them have their individuals divided into 10 splits and for every split 350 *same* and 350 *different* pairs. We follow this verification protocol and carry out a 10 fold cross-validation, taking nine splits as training and one as test set. Further we use the first split to optimize parameters if required.

3.1.2 CPLFW

The Cross-Pose Labeled Faces in the Wild (CPLFW) database was developed by Zheng and Deng (2018). It is an extension of the older LFW data base (Huang et al., 2008), which has been expanded to include images with greater pose difference. The data set contains 11'652 images of 3'930 individuals with 2 or 3 images per person, which were taken in an uncontrolled environment with a great variation of orientations, expressions and lightings. The official evaluation protocol of this dataset contains 3000 comparisons between pairs of images with people of the same identity (positives) and 3000 comparisons between pairs of images of different identities (negatives). Since this official protocol does not contain a defined training set, two new protocols are created, which contain independent training sets from the official defined comparisons.

The first one divides the officially defined positive and negative pairs into 10 equal subsets to perform a 10 fold cross-validation. For each subset, a training set is created that does not include any of the individuals from the comparisons. Thus, the test sets each contain about 20% of the individuals and the training set about 80%. The second one uses all officially defined comparisons as test and the CFP data set as training data.

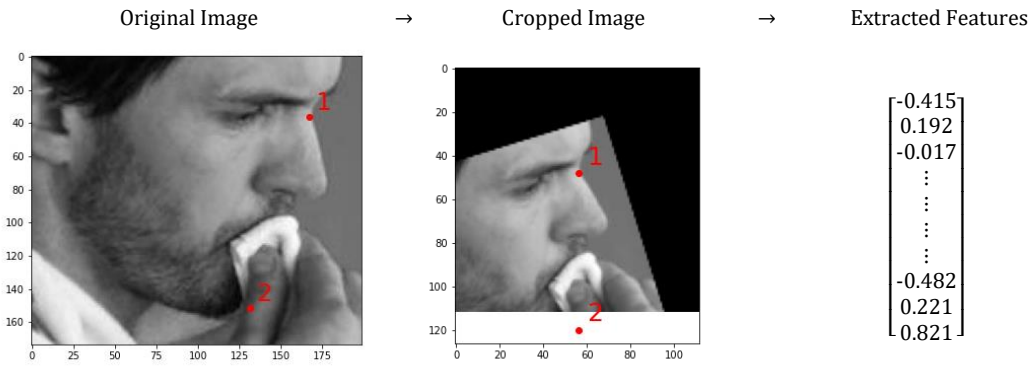


Figure 3.2: Visualisation of image cropping (according to the first group) and feature extraction on example profile image from the CFP data set. Extracted facial landmarks 1: *between the eyes* and 2: *chin* are annotated in red.

For each picture there are 5 marked points on the face, which are the eyes, nose and corners of the mouth. The points of the two eyes and the corners of the mouth were used to calculate the centres of the eyes and mouth, which were later used to align the images (section: 3.2.1).

3.2 Method

The following methods receive two images as input, called probe and gallery image, and return a similarity value. To implement those methods, the bob.bio framework (Günther et al., 2012) is used. In addition to the method used in this work, the Ranked List method, a comparative method, the Baseline method, was also applied to the same data sets to obtain a reference value. The structure of the following sections are leaned on the structure of the framework. It starts with image cropping followed by the extraction of the features. After that, the Baseline method or the Ranked List algorithm is applied, which will be explained in the sections below. Further are all important settings of our experiments mentioned, which allows reproducing of the results.

The pre-processing steps of image cropping and feature extraction are illustrated in figure 3.2.

3.2.1 Image cropping

In preparation for the following feature extraction, the faces need to be aligned and the size of the images adjusted. This is because the images should have similar orientation as the training images of the networks used for the feature extraction. The orientation and image size differs between the two groups of networks and the data sets used in this thesis. These networks will be explained in the next section, but for the pre-processing the first group consists of the MobileFaceNet, ResNet50 and ResNet100 and the second group consists of the VGG-16 network.

CFP

For the first group of networks are all faces of the CFP database cropped and resized to have a similar alignment and an image size of 112x112 pixels. The alignment is reached by using two of

(a) Original images.

(b) Cropped images.

Figure 3.3: Example images of the CPLFW dataset. Subfigure 3.3b shows cropped images according to the first group of neural networks. The images are aligned such that the labelled landmarks *between eyes* and *mouth* lie on the 52th pixel and the 90th pixel from the top, respectively, and both on the horizontal centre.

the already labelled facial landmarks, one lies between the eyes and the other on the chin. The face is rotated, translated and resized such that both facial landmarks lie in the horizontal middle and the first (i.e. between the eyes) lies on the 48th pixel from the top and the chin below the bottom (i.e. 120th pixel from the top). This way of cropping results in images with big empty space but showed in experiments (A.1.1) the best performance. The second group, on the other hand, requires a 224x224 image. The optimal position of the point between the eyes is on the 70th pixel and the chin on the 224th pixel from the top. Both points are again in the horizontal centre, i.e. on the 112th pixel from the left.

All images are converted from colour to grayscale, because the contribution of colour to face recognition is not sufficient explored, although colour probably improve the performance. Due to the analysis in section A.1.3 are the profile images not mirrored such that all their looking direction point to the same side.

CPLFW

For the first group of networks, which require a 112x112 image, the images are aligned based on experiments as follows. The images are rotated, translated and resized so that the point between the eyes is on the 52th pixel from the top and the point in the middle of the mouth is on the 90th point from the top. Again, both points are on the 56th pixel from the left. Following the proportions between the two groups from the CFP dataset, in the CPLFW dataset and the second group of networks, the point between the eyes is aligned with the 74th and the mouth with the 194th pixel from the top. Again are both points in the middle of the x-axis and all images converted to grayscale

3.2.2 Feature Extraction

The features are extracted using two pretrained Deep Convolutional Neural Networks (DCNN's) with the ArcFace loss function presented in Deng et al. (2019). These two networks are called ResNet100 and ResNet50 and differ mainly in the number of the layers and the resulting size. The ResNet50 has 170MB and ResNet100 255MB. Another used network, which is trained on the same data but has a much smaller size (4MB), is the MobileFaceNet (Chen et al., 2018). Larger networks are usually more accurate than smaller networks, but on the other hand they are slower in feature extraction. Those DCNN's receive the cropped grayscale image with the size of 112x112 (section 3.2.1) and return a list with the extracted features. The size of this list depends on the network, where ResNet50 and ResNet100 return 512 features and MobileFaceNet 128. These extracted features can be understood as values that should represent the identity of the person in the image.

Another, somewhat older neural network is the VGG-16, which was developed in Parkhi et al. (2015). This convolutional neural network receives a 224x224 pixel image. VGG-16 is used in this work to show the performance also with older neural networks. Furthermore, compared to the ArcFace (ResNet100, ResNet50 & MobileFaceNet) networks, this network is mostly trained on verification of frontal images, which allows an analysis of the potential influence of the type of training data for the neural network.

3.2.3 Baseline

This section explains the so called Baseline method, which will be used to compare the Ranked List method to. The Baseline method basically corresponds to face recognition with neural networks and is illustrated in figure 2.3.

To compare a probe with a gallery image, the two images go through the steps mentioned above. First are the images cropped according to section 3.2.1 and then the features extracted according to section 3.2.2. These steps lead to two lists of extracted features, one for the probe and one for the gallery image. The distance between those two lists of features is the similarity between the probe and gallery image, where a higher distance means a lower similarity. This can be calculated for example with the Euclidian distance or the cosine similarity. The cosine similarity is used here because it gives better results, as evaluated in section 4.5. Since the comparison between the probe and gallery image is done directly, this method does not need a training set.

3.2.4 Ranked List

Compared to the Baseline method, the Ranked List method does not compare the two images directly, but compares the similarity to reference images. This requires a cohort which contains multiple images of different identities. Again, all images go through the above steps of image cropping (section (3.2.1) and feature extraction (section 3.2.2) so that each image is represented by a list of extracted features. The rank list creation is done in the same way as in the literature, respectively mentioned in section 2.1.2. The similarity between the extracted features of the test images and the reference images is calculated. The most similar image of each reference person is used to create a rank list of the similarities between the test and the gallery images. Finally, these two rank lists are compared with each other, resulting in the similarity of the original two images.

In addition to the above mentioned rank list comparison methods from the literature in section 2.1.2, the following comparison methods are used in this work:

Additional Rank List Comparison Methods

The rank list comparison method Kendall Tau (Kendall, 1938) is a statistical measure of the correlation of two rank lists and was used according to my knowledge not yet in connection with the rank list method. Further, an implementation of a weighted version of the Kendall Tau method (Vigna, 2015) is used. The Weighted Kendall Tau weights the changes according to the rank, such that higher ranks have a smaller influence on the final similarity value. In this thesis, a fast version of the Kendall Tau (Knight, 1966) and Weighted Kendall Tau function is used, implemented by the python package scipy.stats and in this work denoted as S_{kt} and S_{wkt}.

Further, the most simple version of a similarity function is used, to compare the functions above to. It sums the absolute difference of each rank with the same model index up. The result in the end is negated, such that higher differences in ranks leads to a lower similarity value. The result is not normalized and thus can lie within $[-\sum_{i=0}^{\lfloor N_M/2 \rfloor} 4 \cdot i, 0]$ where the value 0 would result in comparing two equal rank lists.

$$S_{dif}(\pi, \gamma) = -\sum_{m=0}^{N_M - 1} |\pi(m) - \gamma(m)| \tag{3.1}$$

By analysing the correlation of the difference between the ranks to the rank (section: 4.1), three new methods to compute the similarity of two rank lists are found. These methods are S_{new}, S_{chisq} and S_{svc} and are explained in section 4.2 in the experiments..

Chapter 4

Experiments

In this chapter, various verification experiments are carried out and to be able to compare the results, values which represent the accuracy of a method are required. The performance is compared by looking at the receiver operating characteristic (ROC) curves and the accuracy of each method. The ROC curve plots the true match rate (TMR), which is equivalent to one minus the false non match rate (FNMR) versus the false match rate (FMR). As a side note, in literature are the terminology of *match* also known as *acceptance* and *non match* as *rejection*. In general a better detection method leads to a ROC curve that is higher and more to the left (Metz, 1978). Two concrete comparable values out of the ROC curve are the area under the curve (AUC) and the TMR for one specific FMR. The size of the data set determines how small FMR values can be reached, where bigger the data sets leads to smaller possible FMR values in the ROC plot. The accuracy of a method is computed as one minus the equal error rate (EER), which can be understood as the percentage of true classified images when having an equal FNMR and FMR.

4.1 Analysis Rank Difference

Rank list comparison methods preferred in Müller et al. (2013), Müller (2010), Schroff et al. (2011) and Vigna (2015) weight low ranks higher than high ranks. "It is expected that high image similarities are more informative about identity" (Müller et al., 2013). In the following, the CFP data set is used to verify this statement. For each probe/gallery image of the first split a rank lists is created, using the other nine splits as cohort, as explained in section 2.1.2. The frontal-profile protocol of the data sets define comparisons of probe to gallery images, where comparisons between images of the same person are called *positive* comparisons and with different persons *negative* comparisons. To compare two rank lists, one needs to compare the rank of each person from the model data base between the probe and the gallery rank list. Since the probe and gallery image can be used interchangeably, the absolute difference of those two ranks is used. This difference is here denoted as D_p in the positive comparison and D_n in the negative case. The difference d between D_p and D_n is the indicator how informative each difference of ranks is. A bigger d means one can distinguish clearer between the positive and the negative case. The assumption above would be correct, if d gets smaller with higher ranks. Looking at figure 4.1, where the absolute difference between the ranks is plotted versus the rank of the probe image, one can not observe such a relation. Starting from a low rank, d gets smaller then bigger again and is nearly symmetric around a rank of 225. This must lead to a rejection of the former assumption.

The violation of this assumption could explain the bad results in figure 4.5a, using rank list comparison function $S_{2013muller}$ with low lambda, meaning fast decaying function. It also would explain the result (figure 4.5c) of S_{wkt} being worse than S_{kt}. Although the d does not decay with

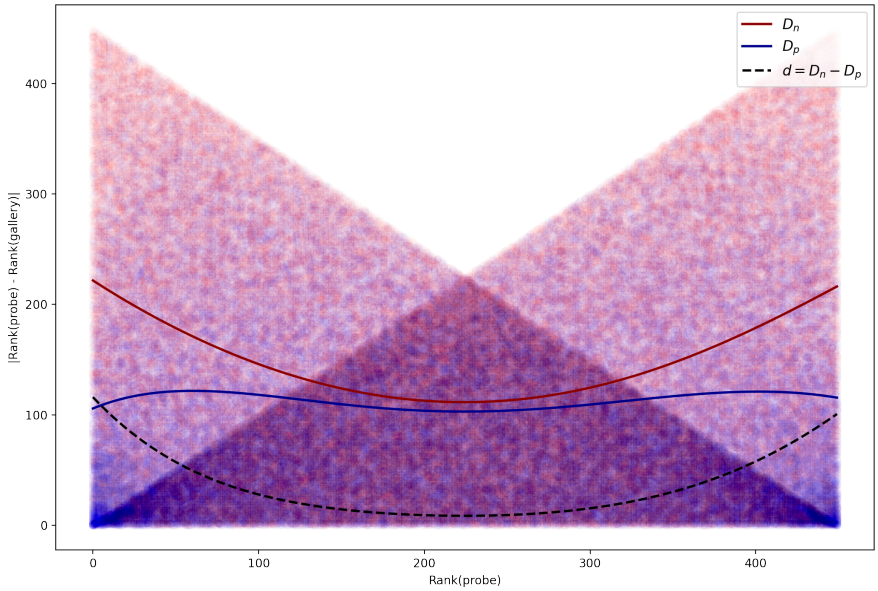


Figure 4.1: The absolute difference between the ranks of negative (red) and positive (blue) comparisons plotted against the ranks. Polynomial regression of degree 5, by minimizing the sum of squares, plotted as lines. Difference d of negative minus positive comparisons is plotted as dashed black line. Rank lists are created on the frontal-profile protocol of the CFP data set, where first split is used for probe and gallery images and other nine splits as cohort.

higher ranks, we can observe some sort of dependency. In the next paragraph we analyse this behaviour and try to make use of this knowledge to improve the rank list comparison method.

To get a better understanding of the plots in figure 4.2, let's imagine the most extreme case where all positive comparisons, compare two identical images and the negative case two images with random noise. The first case would result in two identical rank lists, which would be represented in this figure as all blue points lying on 0 on the y-axis for each rank of the probe image. The second case would result in two random rank lists and be visible as red points lying uniform distributed ([Rank(probe)-450, Rank(probe)]) for each rank of the probe image.

In figure 4.2 one can observe three things.

- The differences between the ranks in the negative case is nearly uniform distributed.
- The difference between the ranks is small for very low and very high ranks, for the positive case
- There are only a few extreme differences of ranks in the positive case.

Further, in the positive case, it can be observed that at rank 0 and rank 450, the differences are not exactly horizontally aligned. This probably has to do with the second and third point above. I can not explain the relative hard cut, but at rank 0 (respectively 450) the probability of obtaining a high rank difference is very low and therefore there are less, up to no points the higher the difference gets at this point.

A successful list comparison method must be able to differentiate as good as possible between the positive and the negative case. The list comparison method S_{dif} sums up the absolute differences between two ranks of comparisons to the same individual from the cohort. This is independent to the level of the rank. The next section attempts to improve S_{dif}, by applying the knowledge of the dependency between rank differences and rank.

4.2 New Rank List Comparison Methods

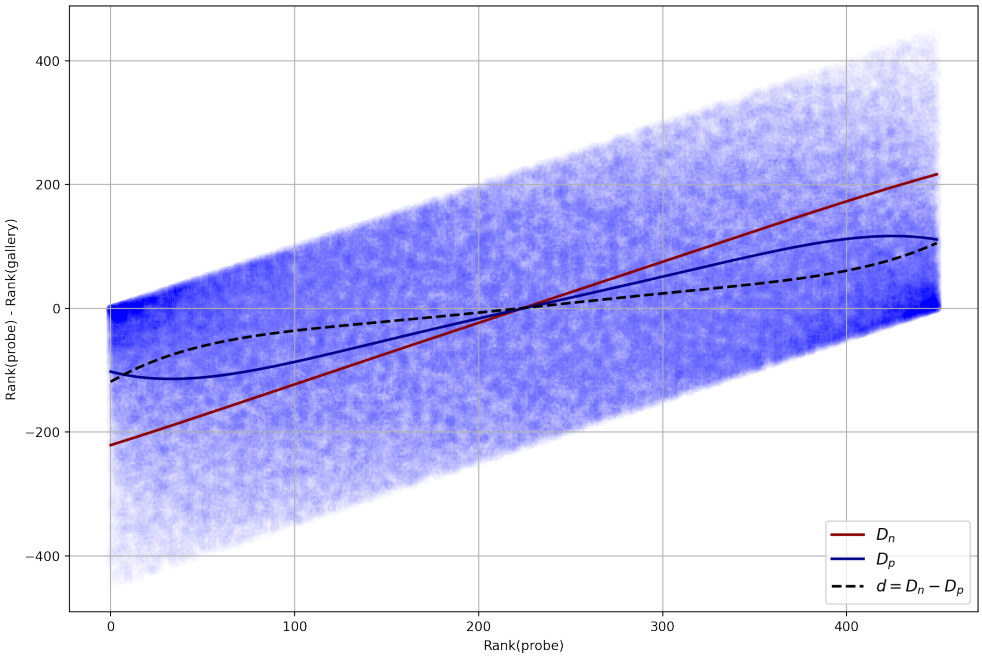
Using the knowledge of section 4.1, a rank list comparison method, which is able to distinguish the negative case as good as possible from the positive one, must have the following characteristics.

1. The rank differences of low and high ranks need to be weighted stronger.
2. Smaller absolute differences of ranks must lead to more similar rank lists.
3. Very high absolute differences of ranks should make the two rank lists even more dissimilar.

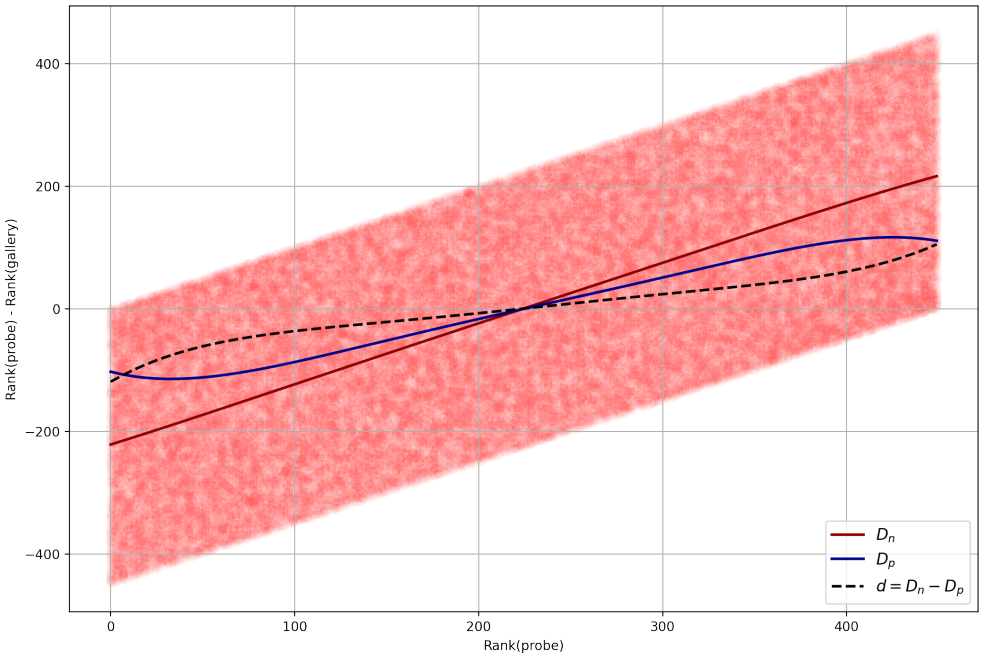
4.2.1 New Parametric Similarity Function

Rank Weighting: S_{new}

The list comparison function (4.1) has those mentioned characteristics and can be understood as a composition of two parts. In the first parentheses the difference of each two ranks is calculated and multiplied by the second bracket as weighting. In the first part, the difference is divided by the highest possible rank difference to obtain a value between 0 and 1. With an exponent $\alpha \in [1, \infty)$, high differences are exponentially weighted more than small ones, which covers point 2 and 3 in the enumeration above. The second part weights high and low ranks more heavily and therefore covers the first point of the enumeration above. Where this weight is the sum of two parabolas (one for each rank list) open upwards, with the minimum at half of the total ranks. The



(a) Positives.



(b) Negatives.

Figure 4.2: Difference of ranks vs. rank for positive (a) and negative (b) comparisons. Polynomial regression of degree 5, by minimizing the sum of squares, plotted as lines. Frontal-profile protocol of CFP data set, where first split is used for probe and gallery images and other nine splits for the model data base.

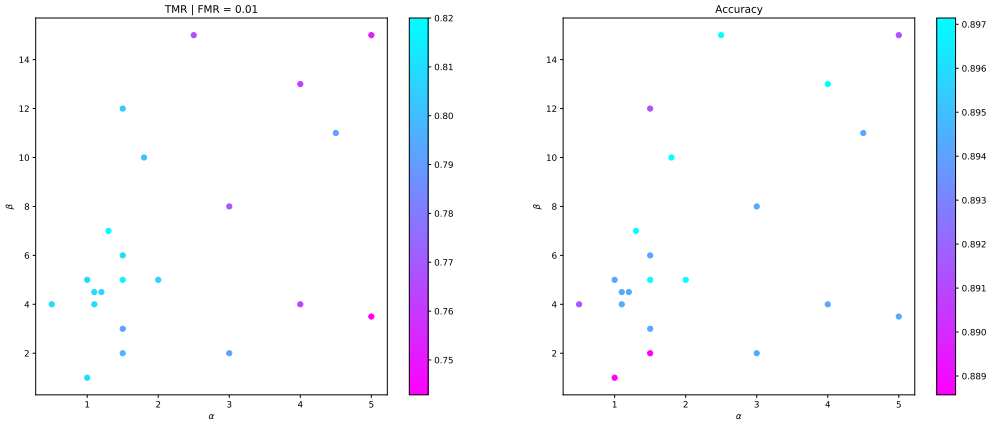


Figure 4.3: TMR and Accuracy (colors: pink = low, turquoise = high) for Ranked List algorithm using S_{new} with different combinations of α and β. Applied on the CFP data set (FP Protocol), using the first split as test set and other nine splits as training sets.

higher the exponent β ($\in [1, \infty)$), the more extreme (very high or very small) the rank must be to result in a high weights. Similarly, a higher β leads to a larger range of very low weights for medium ranks. Finally, the sum is negated, so that high differences between the individual ranks mean a lower similarity of the rank lists.

$$S_{new}(\pi, \gamma) = -\sum_{m=0}^{N_M-1} \left(\frac{|\pi(m) - \gamma(m)|}{N_M} \right)^{\alpha} \cdot \left(\left| \frac{\pi(m)}{N_M \cdot 0.5} - 1 \right|^{\beta} + \left| \frac{\gamma(m)}{N_M \cdot 0.5} - 1 \right|^{\beta} \right) \tag{4.1}$$

The Ranked List algorithm using S_{new} as similarity function with random parameter combinations of α and β, is applied on the CFP data. The results of the experiments are shown in figure 4.3. Looking at both, the accuracy and TMR, S_{new} has best performance with $\alpha = 1.5$ and $\beta = 5$.

4.2.2 New Non-Parametric Comparison Mehtod

Chi-squared: S_{chisq}

In section 4.1, it was found that when comparing images of the same identity, the differences between ranks are not uniformly distributed. This is especially the case for low and high ranks. This knowledge is also incorporated in the similarity function S_{new}, where a parabolic function is used to give more weight to low and high ranks. Since in that case a parameter optimization is necessary, in the following a non-parametric method is presented, to obtain those weights.

For all training set images, rank lists are created, as explained in section 2.1.2, using the same training set images as cohort. For the rank list calculation the same cohort is used, because all ranks (0-449 in this case) must be contained in the definition range of the function. Logically, this means that each individual is compared with itself once and thus receives rank 0 with itself. By means of those rank lists, the rank difference of all positive comparisons are computed. Using a Chi-squared test, one determines for each rank how much the rank differences differ from a uniform distribution. The resulting test statistic value is denoted as $f_{\chi^2}(r)$ for each rank r and

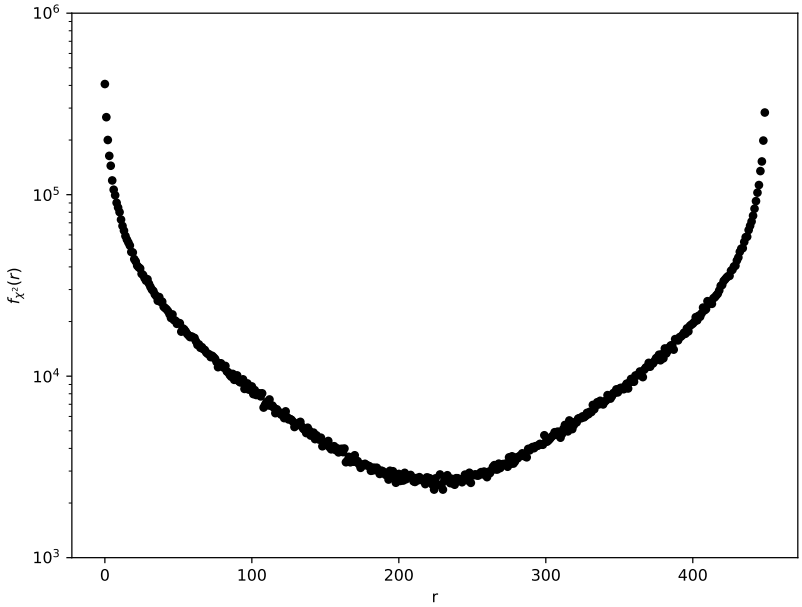


Figure 4.4: Example of computed Chi-squared test values $f_{\chi^2}(r)$ for each rank r. All positive comparisons (Frontal-Profile and Frontal-Frontal) of splits 2 to 10 of the CFP dataset (FP protocol) are used.

used in equation 4.2 for the weighting, when comparing two images from the test set. Due to the above mentioned behaviour, rank 0 is highly overrated. Therefore, this value is removed and so that the rank list has the original length again, the minimum of all values is inserted at the median rank.

$$S_{chisq}(\pi,\gamma) = -\sum_{m=0}^{N_M-1}\left(\frac{|\pi(m)-\gamma(m)|}{450}\right) * \left(f_{\chi^2}(\pi(m)) + f_{\chi^2}(\gamma(m))\right) \tag{4.2}$$

Figure 4.4 shows large Chi-squared values for low and high ranks. It can also be seen that the function increases exponentially in the direction of low and high ranks. Assuming that differences between the ranks of negative comparisons are uniformly distributed, it can be stated that positive and negative comparisons can be distinguished especially at low and high ranks.

As a side note, it should be noted that in this example, by creating the rank lists and calculating the Chi-squared values, this method takes 4.2 (Enrolment + Scoring: 1 hrs 24 min vs. 20 min) times longer than similarity functions that do not need training.

Support Vector Classification: S_{svc}

The rank list comparison methods presented so far basically sum up the differences of the ranks for each individual in the cohort. Depending on the method, the differences are weighted according to the height of the ranks. In this section, a rank list comparison method is presented that takes a step back and classifies the differences of rank lists using a supervised learning method, called support vector machine (SVM). Basically, the SVM is trained from a large number of classified data points, which can have any number of dimensions. After training, the SVM is able

RL_1	RL_2	$RL_1 - RL_2$	sorted by RL_1	$RL_2 - RL_1$	sorted by RL_2
4	0	4	-2	-4	-4
3	4	-1	-2	1	-1
0	2	-2	-1	2	2
1	3	-2	-1	2	2
2	1	1	4	-1	1

Table 4.1: An example of computing the difference between two rank lists (RL_1 and RL_2) and sorting it according to one of the rank lists.

to classify an unknown data point, by using different sorts of kernel types like *linear*,*polynmial* or *radial basis function* (rbf). The more detailed explanation of this method is beyond the scope of this paper, but the SVM can be thought of as trying to draw a decision line, or hypersurface, between two groups of data points. When deciding whether a point belongs to a certain class, the SVM also returns the *signed distance* in addition to the classification, which is positive if the point belongs to the class and negative otherwise. With a value of 0, the point to be classified would lie exactly on the decision line. In our case, the data point is the difference of the two rank lists and the classification whether the two rank lists are from images of the same person or from different persons. Respectively, the *signed distance* as the similarity value of the two images, where a high positive value indicates a high probability that the two images are from the same person.

For this purpose, the difference in the ranks to each reference individual is calculated and sorted according to the height of one of the two rankings, see the example in table 4.1. These two sorted lists of differences are further used as the input of the SVM. The problem that the rank lists of the training images get always a rank of 0 when they are compared with themselves is avoided by assigning a random similarity in this case. All positive and an equal number of negative comparisons of the model databank serve as training data for the SVM. Not all negative comparisons are used because their number is very high and the SVM prefers a balanced number of both classes. The similarity of the rank lists of the probe and gallery image from the test dataset are then determined using the trained SVM classifier. This rank list comparison method is called S_{svc} and is a relatively slow variant due to the training.

In the experiments, the SVM implementation of the python package scikit-learn was used, called svm.SVC and svm.LinearSVC, respectively. The *rbf* kernel is used because it reaches an accuracy of 0.8743 ($C = 1$) in comparison to 0.6886 with the *linear* kernel, when applied on the first split of the CFP data set as test and the other nine as training sets and the ResNet50 as feature extraction method. In comparative experiments (see table 4.2) one found the highest accuracy of S_{svc} with parameter $C = 0.1$. With the parameter C equals 0.01, one achieves a very high TMR, but a lower accuracy. It should also be noted here that further values for the parameter C, especially with the combination with other kernels, could not be tested due to time limitations and could eventually provide better results.

4.3 Rank List Comparison Methods

This section compares the different lists comparison methods, explained in section 2.1.2, by applying them on the frontal to profile protocol of the CFP data set. The first split is used as test and the other nine splits as training set. The ResNet50 was used for feature extraction and the cosine

	Accuracy	AUC	TMR
$C = 0.01$	0.8886	**0.9672**	**0.8143**
$C = 0.1$	**0.8914**	0.9664	0.7914
$C = 1$	0.8743	0.9492	0.4542
$C = 10$	0.8885	0.9565	0.7743
$C = 100$	**0.8914**	0.9614	0.7771

Table 4.2: The accuracy (at EER) and TMR (FMR = 10^{-2}) of the Ranked List algorithm with S_{svc} as rank list comparison method, using the *rbf* kernel and different types of parameter C. Applied on the first split of CFP data set as test and the other nine as training sets. ResNet50 is used as feature extraction method.

to calculate the similarity of the features.

4.3.1 Muller 2013 and Muller 2010

Using the rank list similarity $S_{2013muller}$ the result in table 4.3 shows higher values of accuracy and AUC for higher λ values. This is also true looking at the ROC curves in figure 4.5 a. The non-parametric method $S_{2010muller}$ performs better than $S_{2013muller}$ with low λ but worse than the ones having $\lambda > 0.95$.

4.3.2 Schroff

Looking at figure 4.5 b one can say that $S_{schroff}$ performs better, the higher k is. This is also reassured in table 4.3 where the accuracy, AUC and TMR value increase with the k value. It performs the best with $k = 450$, with a accuracy of 0.894, an AUC of 0.967 and a TMR of 0.806. A value of k equal to 450 means that not only the values of the first few ranks are used, but all of them.

4.3.3 Kendall Tau and Weighted Kendall Tau

The Kendall Tau (S_{kt}) and the Weighted Kendall Tau similarity measure (S_{wkt}) perform according to accuracy and AUC nearly the same, but the weighted version has a much lower TMR. Since the ROC curve of S_{kt} in figure 4.5 c is for most values of FMR above the curve of the weighted version S_{wkt}, one would assume it has a higher accuracy. But the accuracy of those two methods are nearly the same, and the AUC is even a bit higher for S_{wkt}. This is because the figure is plotted in logarithmic scale and therefore small differences in high FMR values have big influences on the accuracy and AUC values.

4.3.4 Rank Difference

The reference method for rank list similarity computation S_{dif} performs nearly the same as S_{kt}. With a accuracy of 0.888, an AUC of 0.963 and a TMR of 0.808, these results are surprisingly good, considering that this method is the simplest possible.

Method	Accuracy (FP)	AUC (FP)	TMR @ FMR = 10^{-2} (FP)
$S_{2013muller}$, $\lambda = 0.9$	0.817	0.909	0.534
$S_{2013muller}$, $\lambda = 0.95$	0.860	0.943	0.659
$S_{2013muller}$, $\lambda = 0.99$	0.894	0.964	0.762
$S_{2013muller}$, $\lambda = 0.999$	0.894	0.967	0.811
$S_{2013muller}$, $\lambda = 0.9999$	0.894	0.966	0.805
$S_{2010muller}$	0.891	0.962	0.751
$S_{schroff}$, $k = 25$	0.826	0.913	0.571
$S_{schroff}$, $k = 50$	0.874	0.944	0.669
$S_{schroff}$, $k = 100$	0.883	0.956	0.691
$S_{schroff}$, $k = 200$	0.894	0.963	0.766
$S_{schroff}$, $k = 300$	0.889	0.964	0.791
$S_{schroff}$, $k = 450$	0.894	0.967	0.806
S_{kt}	0.891	0.966	0.800
S_{wkt}	0.891	0.960	0.711
S_{dif}	0.888	0.963	0.808
S_{new}, $\alpha = 1.5$, $\beta = 5$	0.897	**0.969**	**0.817**
S_{chisq}	**0.900**	0.967	0.789
S_{svc}, kernel = rbf, $C = 0.1$	0.891	0.966	0.791

Table 4.3: Performance of different rank list comparison methods on first split of CFP data set (frontal-profile (FP) protocol). The maximum of each block is underlined and the maximum of the entire table highlighted in **bold** font.

4.3.5 New List Comparison Methods

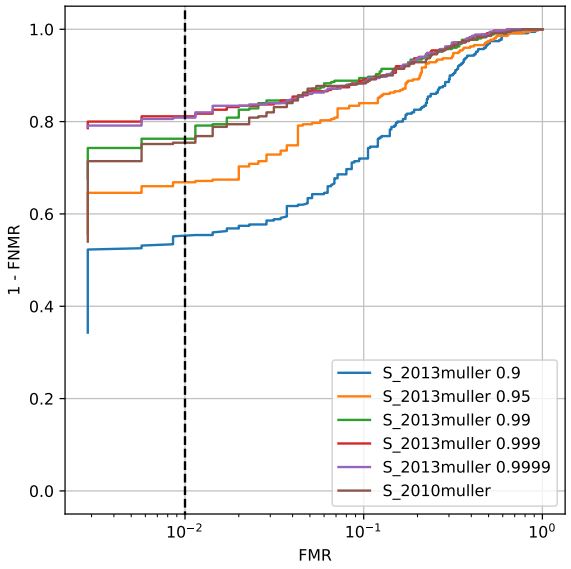
The rank list similarity function S_{new}, created and optimised in section 4.2.1, results in a high accuracy and TMR of 0.897, 0.817 respectively. Using the non-parametric S_{chisq}, explained in section 4.2.2, shows the highest accuracy of 0.900, but has a relative low TMR of 0.789. The rank list comparison method S_{svc}, which uses a trained support vector classifier, achieves an accuracy of 0.891 and a TMR of 0.791 and thus performs well but not the best.

4.4 Dependence of the size of the cohort

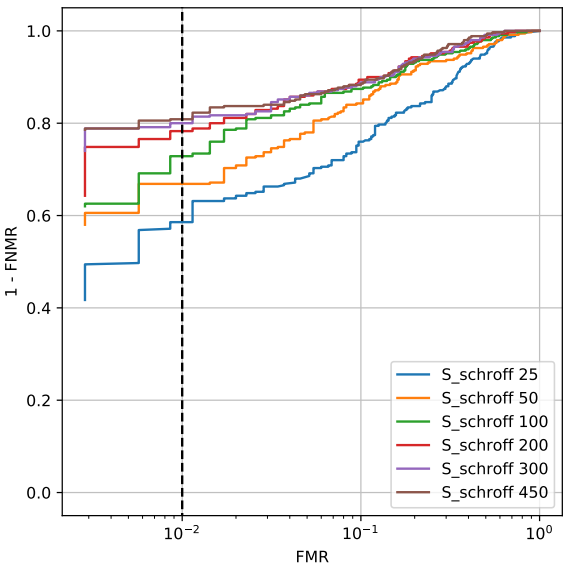
The Ranked List algorithm needs a cohort and the number of individuals and images per individual in the database intuitively have a influence on the performance. In order to evaluate the relationship between the size of the cohort and the accuracy of the algorithm, two experiments are applied.

In the first, the Ranked List algorithm (S_{dif}) was used with different numbers of individuals in the cohort and in the second different numbers of images per individual. It is tested on the first split of the CFP data set (frontal-profile protocol) and ResNet50 as feature extraction method. In the cohort of the first experiment are splits 2-10, 2-8, 2-6, 2-4 and 2 and therefore 450, 350, 250, 150, 50 and 25 individuals. In the second experiment contains the cohort all 450 individuals from splits 2-10, each having 14 (10 frontal, 4 profile), 11 (8 frontal, 3 profile), 8 (6 frontal, 2 profile), 6 (4 frontal, 2 profile), 4 (2 frontal, 2 profile), 2 (1 frontal, 1 profile) and only 1 frontal and only one 1 profile image per individual.

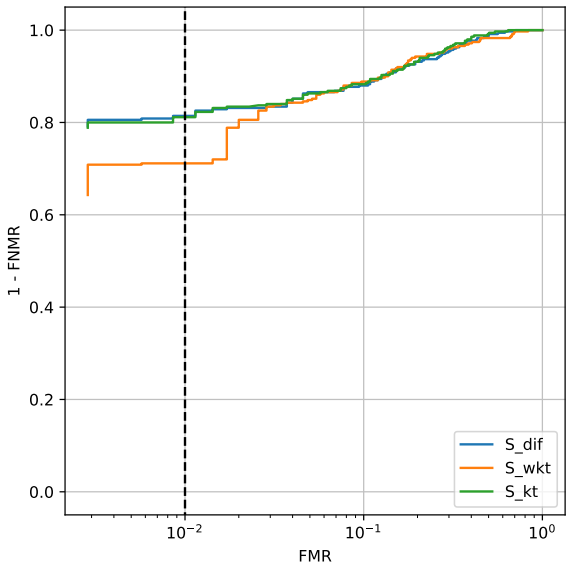
Accuracy and TMR strictly increase with more individuals in the reference data, but saturate



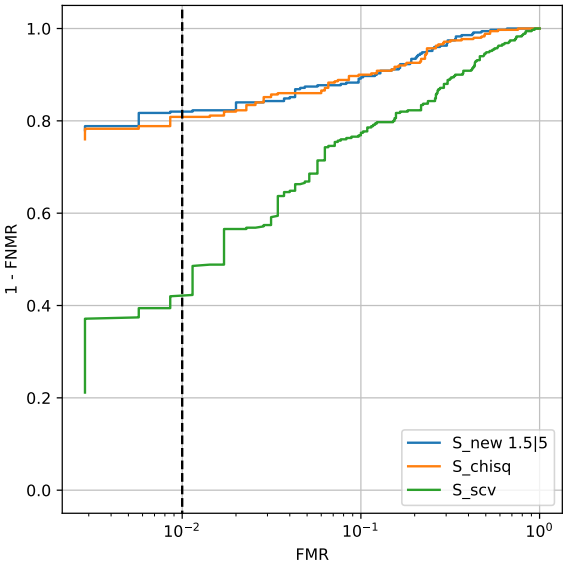
(a) $S_{2013muller}$ with $\lambda = 0.9/0.95/0.99/0.999/0.9999$ and $S_{2010muller}$.



(b) $S_{schroff}$ with $k = 25, 50, 100, 200, 300, 450$.

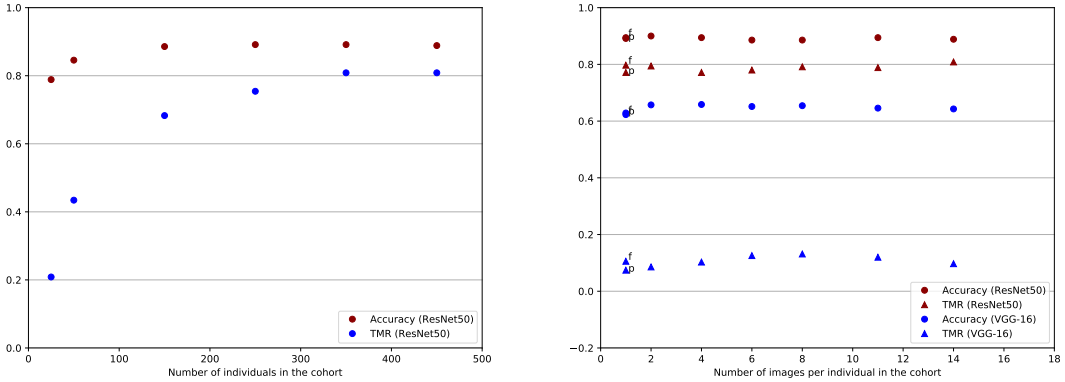


(c) S_{dif} , S_{wkt} and S_{kt}.



(d) S_{new} with $\alpha = 1.5$ and $\beta = 5$, S_{chisq} and S_{svc} with rbf kernel.

Figure 4.5: ROC plots of the Ranked List algorithm using different rank list comparison methods. Applied on first split of CFP data set and frontal-profile protocol. The dashed line marks a FMR of 10^{-2}.



(a) First experiment.

(b) Second experiment.

Figure 4.6: Ranked List algorithm with S_{dif}, applied on CFP data (FP-protocol). The accuracy and TMR (FMR = 10^{-2}) for different number of individuals (a) and images per individual (b) in the cohort. The dashed lines represent the tendency of the points. In subfigure 4.6b is only one frontal image per individual annotated with a small **f** and only one profile image per individual with a small **p**.

above 300-400 individuals at 88.8% and 80.8% respectively (see subfigure: a). The TMR drops sharply below 200 people in the cohort and reaches a TMR of 20.9% for 25 individuals, whereas the accuracy falls at 78.9%. With an increasing number of images per individual in the cohort, the accuracy is almost constant and the TMR slowly increasing, but having outliers at one and two images per individual, when using the more accurate ResNet50 (see subfigure: b). With only one profile and one frontal image, the method with ResNet50 reaches the highest accuracy of 90.0% and a relatively high TMR of 79.4%. In fact, only one frontal image achieves an accuracy of 89.4%, which is also higher than 88.9% for 14 images per individual. With the less accurate VGG-16 network, accuracy and TMR decrease with very few images. However, the drop is only a few percentage points and with a large number of images in the cohort, the accuracy and TMR also seems to decrease somewhat. Only a frontal image in the cohort works better than just a profile image. It should also be mentioned that the VGG-16 network is mainly trained on frontal images, however, also with the ResNet50, a frontal image in the reference images seems to be more useful than just a profile image.

4.5 Distance Measure

Earlier methods for face recognition by means of neural networks used the Euclidean distance to calculate the similarity of two feature vectors. All recent approaches using DNN's for face recognition, for example such as Wang et al. (2018), use the cosine similarity because it seems to work much better. This is also confirmed by experiments on the frontal-profile protocol of the CFP dataset. The similarity using the Euclidean distance ($d_{euclidean}$, used implementation: scipy.spatial.distance.euclidean) is called $sim_{euclidean}$ and is computed as in equation 4.3. The values of $sim_{euclidean}$ are between minus infinity and 1, whereas the comparison of two identical vectors would results in 1. The similarity using the cosine distance is explained in section 2.1.2 above.

	Baseline		Ranked List: S_{dif}		Ranked List: S_{chisq}		Ranked List: S_{new}	
	Accuracy	TMR	Accuracy	TMR	Accuracy	TMR	Accuracy	TMR
Euclidean	0.9057	0.6314	**0.8943**	0.3429	0.8829	0.5257	0.8886	0.5171
Cosine	**0.9143**	**0.8800**	0.8886	**0.8086**	**0.9000**	**0.7886**	**0.8971**	**0.8171**

Table 4.4: Accuracy (at EER) and TMR (FMR = 10^{-2}) of Baseline and Ranked List algorithm with S_{dif}, S_{chisq} and S_{new} ($\alpha = 1.5$, $\beta = 5$) as rank list comparison methods. Similarity of extracted features (ResNet50) are computed by Euclidean or cosine distance. Results of first split of CFP data set as test and other nine splits as training set.

$$sim_{euclidean}(u, v) = 1 - d_{euclidean}(u, v) = 1 - \sqrt{\sum (u_i - v_i)^2} \tag{4.3}$$

The baseline method, applied with the similarity using $sim_{euclidean}$, achieves an accuracy of 0.9057 with a TMR of 0.6314 (FMR = 10^{-2}). With the cosine similarity, however, it achieves an accuracy of 0.9143 with a TMR of 0.8800. Also with the Ranked List algorithm, the cosine as a distance function generally has a higher accuracy and TMR than the Euclidean distance. An exception is the accuracy of the rank list algorithm with S_{dif} which is higher with the Euclidean distance than with the Cosine distance, but since the TMR is very low, it can be determined that the cosine is the more accurate function with both methods.

4.6 Ranked List Performance

This section shows the results of the Ranked List method, using the best rank list comparison methods from section 4.3, applied on the CFP and CPLFW data set.

4.6.1 CFP

In general, the methods applied on the whole data set, have a slightly higher accuracy than on the first split. However, they perform in the same proportion when compared to each other. Looking at the frontal-profile case in figure 4.7 and table 4.5, it is obvious that the Baseline method outperforms the Ranked List algorithm, regardless of the used rank list comparison method. The Baseline method has an accuracy of 0.9303 and a TMR of 0.814, with an FMR of 10^{-3}. The best Ranked List method is the one using S_{chisq} as the rank list comparison method, which is consistent with the findings from section 4.3, but it performs better on all three metrics (Accuracy, AUC, TMR). An important observation is that the accuracy and AUC of the different methods are all within the standard deviation from each other. A one-way ANOVA test applied to each of the three measures shows that at a significance level of 5%, only the TMR (p-value = 0.0019) of the different methods differs significantly. The accuracy has a p-value of 0.1037 and the AUC one of 0.7438 and can therefore not be considered significantly different. The p-value can be understood as the probability of obtaining these data, assuming that the mean of the different groups does not differ.

Using a two-sided t-test, it can be stated that the TMR of the Baseline method is significantly higher (p-value = 0.0005) than the one of the Ranked List algorithm (S_{chisq}), applied on the frontal-profile protocol of the CFP data.

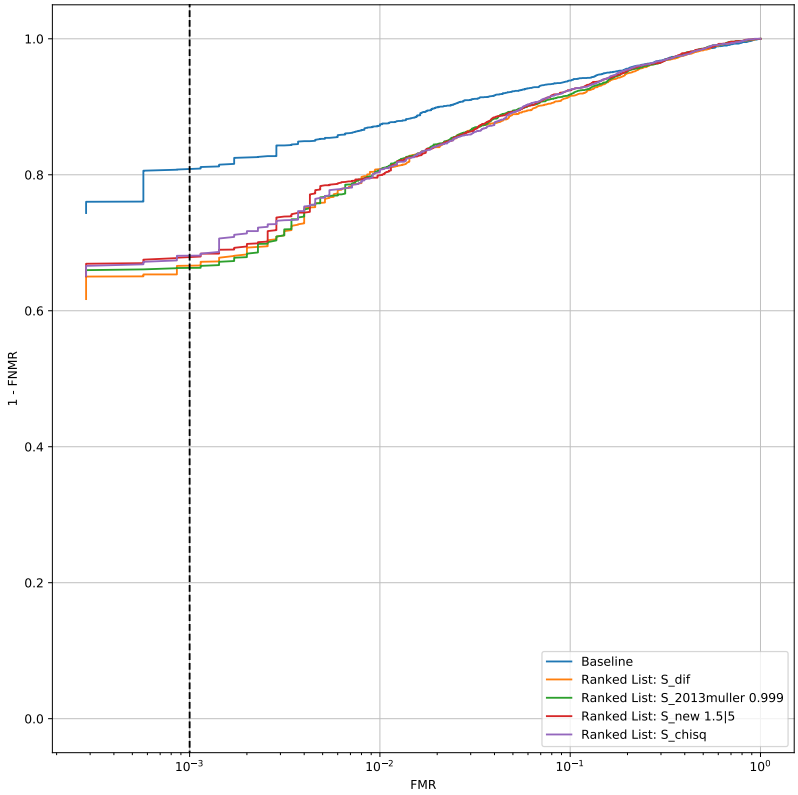


Figure 4.7: Baseline and Ranked List algorithm with some of the best performing rank list comparison methods (ResNet50). ROC plot of results on CFP data set, following frontal-profile protocol with 10 fold cross-validation. The dashed line marks a FMR of 10^{-3}

The results of frontal-frontal comparisons in table 4.6 show a similar behaviour. The Baseline method is the most accurate as well, but only with a small gap to the Ranked List methods. S_{new} as the rank list comparison method, shows in frontal to frontal comparisons and the entire data set the best performance of all Ranked List methods. However, the results of those frontal-frontal comparisons are so close that they are not statistically different from each other.

The accuracy decreases with the Ranked List (S_{chisq}) method by 0.03% in the frontal-frontal protocol and by 1.47% in the frontal-profile protocol compared to the Baseline.

Performance of Different DCNN's

Table 4.7 shows the result of the Baseline method and the Ranked List algorithm using S_{chisq} and $S_{2013muller}$ as rank list comparison method, using different deep convolutional neural networks (DCNN) to extract the features of the images.

First, it is obvious that the performance of the DCNN's increases from the top to the bottom, for both measurements and all applied methods. With the neural networks MobileFaceNet, ResNet50 and ResNet100, the Baseline algorithm has a higher accuracy than all the Ranked List methods. Using the VGG-16 network, the Ranked List ($S_{2013muller}$, $\lambda = 0.99$) method achieves an accuracy of 77.23%, which is an 4.28% increase of the 74.06% from the Baseline method. The

Method	Accuracy (FP)	AUC (FP)	TMR @ FMR = 10^{-3} (FP)
Baseline	**0.9303** (0.0142)	**0.9739** (0.0087)	**0.8140** (0.0501)
Ranked List: S_{dif}	0.9083 (0.0180)	0.9681 (0.0096)	0.7043 (0.0696)
Ranked List: $S_{2013muller}$, $\lambda = 0.999$	0.9137 (0.0186)	0.9697 (0.0094)	0.7111 (0.0695)
Ranked List: S_{new}, $\alpha = 1.5$, $\beta = 5$	0.9166 (0.0174)	0.9703 (0.0087)	0.7143 (0.0666)
Ranked List: S_{chisq}	0.9166 (0.0165)	0.9707 (0.0093)	0.7160 (0.0486)

Table 4.5: Performance of Baseline and Ranked List algorithm with different rank list comparison methods on CFP data set (frontal-profile protocol). Mean and standard deviation (in parentheses) over all 10 folds. Feature extraction with ResNet50.

Method	Accuracy (FF)	AUC (FF)	TMR @ FMR = 10^{-3} (FF)
Baseline	**0.9957** (0.0048)	**0.9998** (0.0004)	**0.9951** (0.0057)
Ranked List: S_{dif}	0.9943 (0.0048)	0.9997 (0.0005)	0.9906 (0.0083)
Ranked List: $S_{2013muller}$, $\lambda = 0.999$	0.9949 (0.0042)	0.9997 (0.0005)	0.9917 (0.0066)
Ranked List: S_{new}, $\alpha = 1.5$, $\beta = 5$	0.9954 (0.0034)	0.9997 (0.0005)	0.9920 (0.0075)
Ranked List: S_{chisq}	0.9954 (0.0037)	0.9997 (0.0004)	0.9917 (0.0069)

Table 4.6: Performance of Baseline and Ranked List algorithm with different rank list comparison methods on CFP data set (frontal-frontal (FF) protocol). Mean and standard deviation (in parentheses) over all 10 folds. Feature extraction with ResNet50.

CNN	Accuracy		
	Baseline	S_{chisq}	$S_{2013muller}$
VGG-16	0.7406 (0.0208)	0.7197 (0.0216)	**0.7723** (0.0212)
MobileFaceNet	**0.8809** (0.0219)	0.8603 (0.0281)	0.8540 (0.0208)
ResNet50	**0.9303** (0.0142)	0.9166 (0.0165)	0.9117 (0.0176)
ResNet100	**0.9794** (0.0067)	0.9683 (0.0081)	0.9671 (0.0106)
	TMR @ FMR = 10^{-3}		
VGG-16	**0.1083** (0.0378)	0.0926 (0.0562)	0.0860 (0.0319)
MobileFaceNet	**0.5863** (0.0849)	0.4851 (0.0870)	0.4474 (0.1114)
ResNet50	**0.8140** (0.0501)	0.7160 (0.0486)	0.6626 (0.0655)
ResNet100	**0.9549** (0.0165)	0.8883 (0.0681)	0.8754 (0.0429)

Table 4.7: Accuracy and TMR of Baseline and Ranked List algorithm with S_{chisq} and $S_{2013muller}$ ($\lambda = 0.99$) rank list comparison method on CFP data set (frontal-profile (FP) protocol) using different deep convolutional neural networks (DCNN) for feature extraction. Mean and standard deviation (in parentheses) over all 10 folds.

accuracy of the Ranked List method ($S_{2013muller}$, $\lambda = 0.99$) decreases for ResNet100 1.26%, for ResNet50 2.00% and for MobileFaceNet 3.05% in comparison to the Baseline method.

The TMR is for every DCNN higher using the Baseline than using the Ranked List method, in which S_{chisq} reaches in every case a higher TMR than $S_{2013muller}$ as rank list comparison method.

The Baseline method together with ResNet100 for feature extraction achieves the highest accuracy of 0.9794 and highest TMR of 0.9549.

4.6.2 CPLFW

Performance of Different DCNN's

The table 4.8 shows the results of the baseline and rank list methods with different neural networks for feature extraction on the CPLFW dataset. The VGG-16 network shows very low accuracy on the CPLFW dataset, which contains very few frontal images. With around 55%, the values are so low that a comparison of the methods is not meaningful. With the ResNet100, the accuracy of the ranked list method is slightly higher with S_{new} than with S_{chisq} or the Baseline. On the other hand, the TMR of the Baseline is slightly higher than of the Ranked List methods. $S_{2013muller}$ as a rank list comparison method has the lowest accuracy as well as TMR. However, the accuracy and TMR of the Baseline and Ranked List method (S_{new}) are not statistically significantly different, with a two-sided t-test showing a p-value of 0.867 and 0.450, respectively.

The Ranked List method with the ResNet50 also has a not statistically significantly higher accuracy than the Baseline, with a p-value of 0.423. However, the TMR is statistically significantly lower (p value = 0.0059). With the network MobileFaceNet and VGG-16, the Baseline has a higher accuracy as well as TMR than all Ranked List methods.

In the figure 4.8 you can see that the TMR of the Ranked List method is higher than the one of the Baseline with large FMRs, and becomes lower with smaller FMRs. Furthermore, with an FMR of 10^{-3}, the difference between the two methods is smaller for the ResNet100 than for the ResNet50.

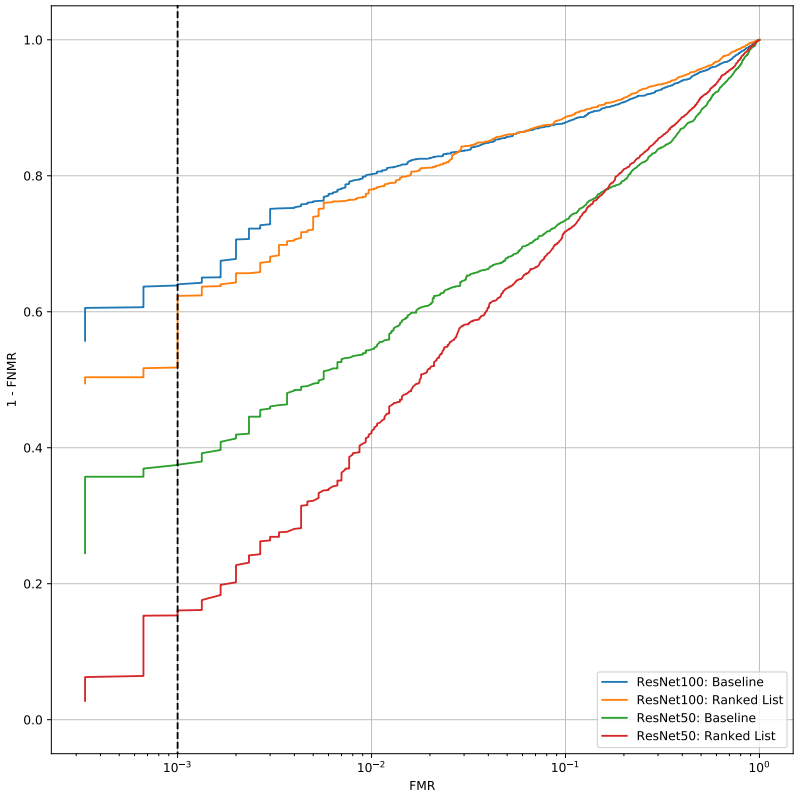


Figure 4.8: Baseline and Ranked List algorithm with S_{new} ($\alpha = 1.5$, $\beta = 5$) using neural networks ResNet100 and ResNet50 on CPLFW dataset. One split as test and nine as training sets, where results of the 10 folds are taken together for this ROC curve.

CNN	Accuracy			
	Baseline	S_{chisq}	$S_{2013muller}$	S_{new}
VGG-16	**0.5800***	0.5233*	0.5466*	0.5267*
MobileFaceNet	**0.7733***	0.7633*	0.7100*	0.7700*
ResNet50	0.7957 (0.0278)	0.8033*	0.7503 (0.0294)	**0.8060** (0.0256)
ResNet100	0.8877 (0.0217)	0.8877 (0.0205)	0.8640 (0.0210)	**0.8893** (0.0199)
	TMR @ FMR = 10^{-3}			
VGG-16	**0.0367***	0.0400*	0.0167*	0.0100*
MobileFaceNet	**0.2633***	0.0167*	0.1433*	0.0233*
ResNet50	**0.4300** (0.1010)	0.1467*	0.1217 (0.1278)	0.2340 (0.1591)
ResNet100	**0.7043** (0.0896)	0.5650 (0.1879)	0.3427 (0.2406)	0.6693 (0.1024)

Table 4.8: Accuracy and TMR of Baseline and Ranked List algorithm with S_{chisq}, $S_{2013muller}$ ($\lambda = 0.99$) and S_{new} rank list comparison method on CPLFW data set using different deep convolutional neural networks (DCNN) for feature extraction. Mean and standard deviation (in parentheses) over all 10 folds. Results marked with * are only of the first split as test and other nine as training sets.

Performance on CPLFW with CFP as cohort

In this section, the performance of the Ranked List method on the CPLFW dataset is shown, using the entire CFP dataset as cohort data. This requires a uniform alignment of the faces. From the labelled positions between the eyes and the mouth in the CPLFW dataset, the position of the chin is calculated, which is approximately half the distance between the eyes and the mouth below the mouth. Using the already labelled facial landmarks from the CFP dataset, all images for these experiments are aligned so that between the eyes is on the 51st pixel from the top and the chin is on the 108th pixel from the top and both are in the horizontal centre. For the verification evaluation, the 3000 positive and 3000 negative pre-defined comparisons are used, that came with the CPLFW dataset.

The results in the table 4.9 show that the Ranked List method is less accurate than the Baseline with all ranked list comparison methods. S_{new} shows the best performance in all three measurements of the Ranked List methods. However, the results of the Baseline and Ranked List method, using S_{new}, are not statistically significantly different. This was decided by dividing the verification comparisons into 10 independent subsets and using a t-test with a significance level of 5%. A side note, the accuracy of 88.24% of the Baseline is much lower than the published accuracy of 92.08% on the same dataset in the paper (Zheng and Deng, 2018), where they probably used the same network (called ResNet100 in this work). This is most likely due to the different preprocessing of the images.

Method	Accuracy	AUC	TMR @ FMR = 10^{-3}
Baseline	**0.8824**	**0.9402**	**0.5601**
Ranked List: $S_{2013muller}$, $\lambda = 0.99$	0.8633	0.9255	0.5097
Ranked List: S_{new}, $\alpha = 1.5$, $\beta = 5$	0.8723	0.9354	0.5526
Ranked List: S_{chisq}	0.8717	0.9317	0.5090

Table 4.9: Performance of Baseline and Ranked List algorithm with different rank list comparison methods on entire verification protocol of CPLFW. All images of the CFP dataset are used for the cohort. Features are extracted with ResNet100.

Chapter 5

Discussion

The CFP dataset clearly shows, through the two protocols, how a method works differently in accuracy when comparing two frontal images or one frontal to a profile image. In addition, it would be interesting to evaluate how comparisons of two profile images perform. The modern neural network, called ResNet100 in this work, reaches almost the maximum accuracy of 99.57% for the frontal-frontal protocol, which leads to very small differences between the different rank list comparison methods with the Ranked List method. On the other hand, generally lower accuracies are achieved on the CPLFW dataset. This is probably due to the larger pose variations and more other difficulties in the CPLFW dataset than in the CFP dataset. These difficulties are, for example, helmets, hands or other objects that cover parts of the face.

The developers of the publicly available neural networks (MobileFaceNet, ResNet50 & ResNet100) used in this work have published the results of their networks in Deng et al. (2019). With the ResNet50, an accuracy of 95.56% is achieved on the CFP dataset (FP protocol) and an accuracy of 92.08% on the CPLFW dataset with the ResNet100. But on the website where the networks were published, the accuracy of the ResNet50 on the CFP dataset (FP protocol) is given as 92.74%. Due to the different pre-processing, the accuracy of 93.03% of the Baseline method (ResNet50) of this work differ from the published results in that paper and also the website. The alignment of the faces and the conversion of the images from colour to greyscale in this work differs from other papers. Therefore, it only makes sense to compare the results within this work and not with results published elsewhere.

There is still room for improvement in the optimisation of pre-processing, because the implementation of the Baseline algorithm from this work performs worse than the results of the same network in other papers. This is probably due to the conversion of color images to grayscale images, which is not the required input for these neural networks. In addition, the large black parts of the cropped images could be filled with content of the image, which could also increase the accuracy. It should also be mentioned that the position of the face in the image has a substantial influence on the accuracy of the networks; shifting the face by one pixel leads to a change in accuracy of over one percent (section: A.1.2). Also, when optimising the alignment, it was often the case (see figure: 4.3) that a very high accuracy comes with a lower TMR, so there was a trade-off between accuracy and TMR.

The Ranked List method, where the features are extracted by neural networks, generally achieves a lower accuracy than the networks by themselves on the CFP data. In the case of frontal to profile image comparison, the accuracy drops even more than in the case of frontal to frontal comparison. Nevertheless, the accuracy is not statistically significantly lower, as all values are very close to each other, only the TMR of the Baseline is statistically significantly higher then the one of the Ranked List algorithm (see section: 4.6.1).

However, it has also been shown on the CFP data that the less accurate a network is, the more helpful a cohort and rank lists are. The VGG-16 network achieved a higher accuracy with the

Ranked List method than with the Baseline method on the CFP dataset (section: 4.6.1). This could be due to the fact that the MobileFaceNet, ResNet50 and ResNet100 networks are already trained on faces across poses, or because in the area of very high accuracies, using the similarities to many model images adds a certain amount of uncertainty. For example, a perfect network that always returns a maximum similarity for positive comparisons and a minimum similarity for negative comparisons would not achieve perfect verification with the Ranked List algorithm, because the rankings would still differ strongly. This is also where it comes into play that the networks are trained to decide whether two images have the same or different identities and not how *similar* two people are in the sense of what people mean by similarity. Moreover, such a perfect network that can verify between all poses equally well would violate the basic idea of the Ranked List method, where the benefit of the method is that a face has a higher similarity to a face of the same orientation. From these considerations, I would propose that the Ranked List method has an advantage only as long as the pose variation in the test images is smaller than in the training images of the used neural network.

Furthermore, it was shown in the section 4.6.1 that the more precise the networks of the ArcFace group become, the less poorly the Ranked List method performs in comparison to the Baseline in percentage terms. Therefore, one could assume that with even more precise networks, the Ranked List method could work even better than the Baseline. However, it is more likely that the networks with very high accuracies reach a saturation value and the Ranked List method will not outperform the Baseline, also with more accurate networks.

On the CPLFW dataset, although not statistically significant, the Ranked List method performs slightly more accurate than the Baseline, also with the ResNet50 and ResNet100 networks (see section: 4.6.2). The TMR on the other hand is lower, but also not statistically significant. It is interesting to see that the Ranked List method works more accurately relative to the Baseline on the CPLFW dataset than on the CFP with the frontal-profile protocol (compare: section 4.8 and 4.5). This could be due to the comparison over larger pose differences, the generally lower reached accuracy or simply due to chance.

Using the images from the CFP dataset as reference images (see section: 4.6.2), the Ranked List method on the CPLFW dataset shows lower accuracy, AUC and TMR than the Baseline. This is a different behaviour than when a part of the CPLFW images are used as reference data, as in the paragraph before. My hypothesis is therefore that if the images in the cohort have a different or smaller variance in the orientation of the faces, the Ranked List method will become a less useful method.

It has been shown (see experiments in section 4.3) that rank list comparison methods that weight lower ranks higher or include only the lowest few ranks perform worse than no weighting when using current neural networks for feature extraction. Furthermore, analyses of the rank lists (see section: 4.1) have shown that in this case low and high ranks are more meaningful, and that it is useful to give more weight to very large differences in the ranks. Especially the former finding raises an interesting implication. Does dissimilarity to certain people say as much about a person's identity as the similarity to certain people? This led to the development of two new methods, called S_{new} and S_{chisq}, that surpass all published rank list comparison methods from the literature, when using ResNet50 as extraction method. In addition, good results could be achieved with the developped rank list comparison method S_{svc}. With the VGG-16 network on the other hand, the rank list comparison method $S_{2013muller}$, where the higher ranks are weighted more heavily than lower ranks, achieves a more favourable result than the other methods (see section: 4.6.1). This could lead to the hypothesis that our developed rank list similarity functions works better with modern networks and the earlier published functions with older networks. However, the differences between the various methods are not statistically significant.

It was also shown (section: 4.4) that the Ranked List method works better the more individuals there are in the cohort, although the accuracy does not increase beyond a certain number of

individuals. Unexpectedly, the *number* of images per individual in the cohort has only a relatively small influence on performance. In addition, it also does not have a very big influence on the performance, *which* of the images of a reference person is used for the rank list creation (see section: A.5).

Finally, the runtime of this Ranked List method is also an additional point of interest. When creating the rank lists by comparison to the cohort, this algorithm takes longer than the direct comparison of two images using neural networks. However, as soon as the number of images to be compared is larger than the cohort, the ranked list algorithm becomes faster in comparison, since the individual rank lists do not have to be created each time. The speed of the ranked list comparisons depends strongly on the function, where for example S_{dif} is multiple times faster than S_{svc}, where a classification model has to be trained. However, the runtime analysis could be further investigated in a subsequent study.

Chapter 6

Conclusions

In this bachelor thesis, the idea of a face recognition system was adopted, which facilitates changes of poses and illumination differences, and combined it with neural networks. It has been demonstrated that using rank lists of similarities to reference images is an acceptable alternative to comparing two images directly, especially with state-of-the-art neural networks for feature extraction. Actually, 4 different neural networks were used, so that their different influences on the ranked list method could also be taken into account. Applied to the two datasets CFP and CPLFW, which were specifically constructed for cross-pose face recognition, this Ranked List method showed statistically not significantly different results compared to the direct comparison of the images. In addition, within the framework of this work, two new rank list comparison methods (S_{new}, S_{chisq}) were developed which can determine the similarity of two rank lists more precisely than all methods from the literature, to the best of my knowledge. This work also transparently describes the type of pre-processing of the images for the neural networks used (including VGG-16, MobileFaceNet, ResNet50 and ResNet100), which has potential for improvement, but can be used as a point of reference.

It would be very interesting to explore the quality of this approach in relation to the type of images used to train the neural networks and the type of images used for the cohort. And it would also be of interest to evaluate this method on databases with even larger pose variations of the faces, for example, images from surveillance cameras that show the face slightly from above, or with other objects such as plants or animals. It would also be interesting to extensive optimize the parameter C in combination with the kernel type of S_{svc} to evaluate the maximum potential of this rank list comparison method, but this is very time-consuming due to the training of the SVM. Further it would also be good to optimise the alignment of the used datasets in connection with these neuronal networks. And finally, the same method should be applied to colour images instead of grey scale images, which would increase the overall verification accuracy and TMR, which would make this method more comparable to face recognition methods from the literature.

Finally, it can be said that the face recognition system implemented in this work also works well with the latest neural networks with very high recognition rates. Therefore, this method should also be considered for future neural networks, especially as the images to be compared are exposed to larger variations than the training images of the neural networks. In particular, if re-training of the network with additional images is too time-consuming or impossible, a cohort with rank lists may be a very good alternative.

Appendix A

Attachments

A.1 Image cropping

A.1.1 Crop Position CFP

Figure A.1 shows some of the results applying the Baseline method on the CFP data, using ResNet50 for feature extraction, with different crop types. For example 15 | 105 is the ROC curve when cropping the original image such that the point between the eyes lies on pixel 15 from the top and the chin on pixel 105 from the top. Both points lie in the horizontal middle. We chose the cropping of 48 | 120 as the best, since this has the highest TMR (= 1 - FNMR) at a FMR of 10^{-2}.

A.1.2 Crop Position CPLFW

Figure A.2 shows the performance of the baseline method on the CPLFW dataset for different crop types. A high accuracy sometimes comes with a lower TMR and vice versa, which is why the decision for the best choice is not trivial. When cropping the images such that the point between the eyes lies on the 52th pixel from the top and the middle of the mouth on the 90th the accuracy and the TMR are both relatively high, but it is neither the highest accuracy nor TMR. The accuracy is 91.00% (TMR = 83.67%) which is lower than the 92.08% achieved in the paper Zheng and Deng (2018) on the entire data set, where they probably also used the ResNet100 (Deng et al., 2019).

A.1.3 Profile Face Orientation

This subsection evaluates the influence of face orientation of the profile images to the performance of the recognition. It is assumed that comparison between profile images with the same orientation, meaning looking all to the left or all to the right, improves the accuracy of the method. To test this assumption, a two sided T-test is made with the following null hypothesis. The average accuracy of the Baseline method, explained in section 3.2.3, is the same for the case of all individuals looking to the right and all having the initial orientation. To achieve a comparable distribution of the accuracy, the first split of the frontal-profile protocol from the CFP data set was split into ten subsets. Due to this small size of 70 comparisons per subsets, the significance of this statistic is compromised. The mentioned Baseline method applied on each of those subsets results in a p-value of 0.7978, between the case of all profile images oriented to the right and no change of the orientation. Having a significance value of 0.05, the null hypothesis can not be rejected. The same test applied to the case of all images oriented to the left and no change of the orientation,

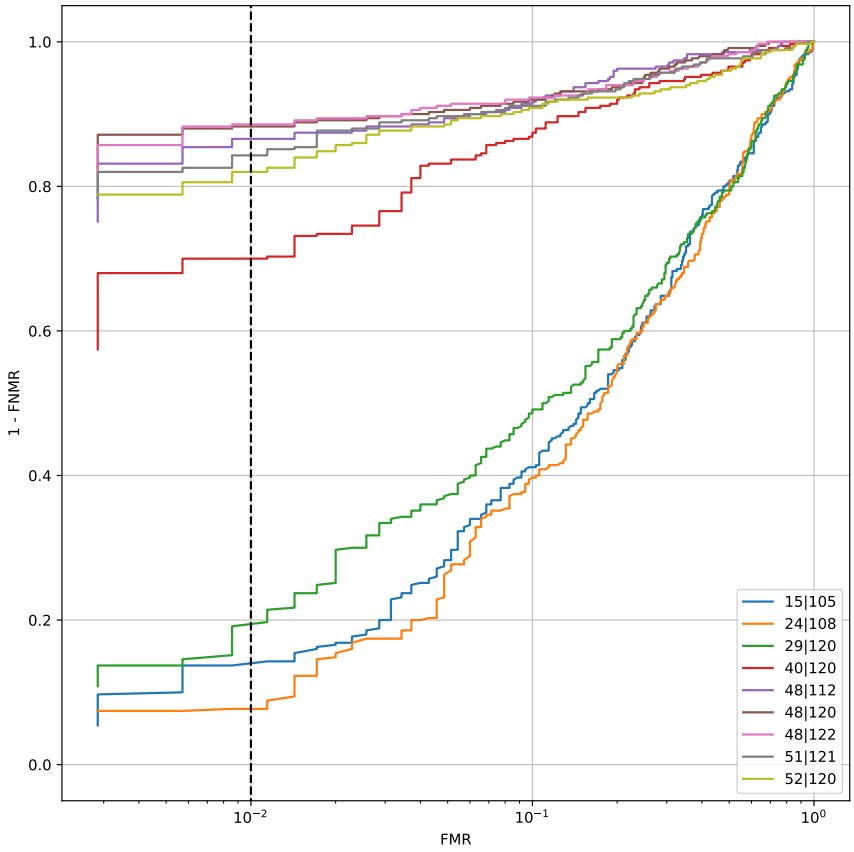


Figure A.1: Result of the Baseline algorithm on the first split of the frontal-profile protocol of the CFP dataset. The cosine distance is used to compare the features in the Baseline algorithm.

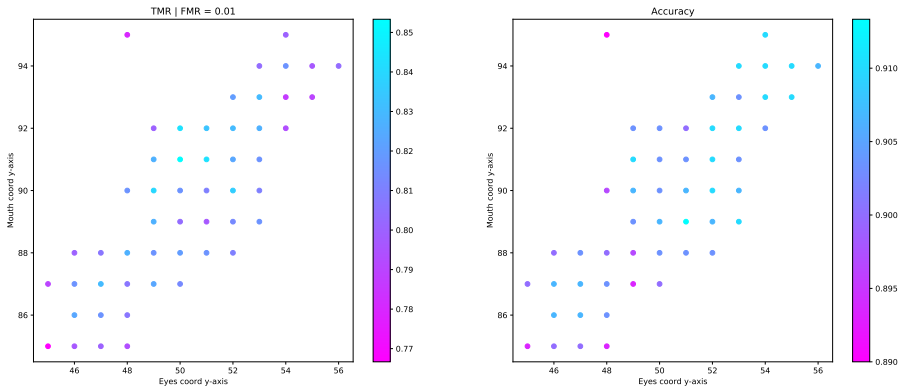


Figure A.2: Auccuarcy (at EER) and TMR (FMR = 10^{-2}) for different positions on y-axis of point between the eyes and middle of the mouth. The Baseline algorithm with ResNet100 and cosine distance on the first split of CPLFW data set.

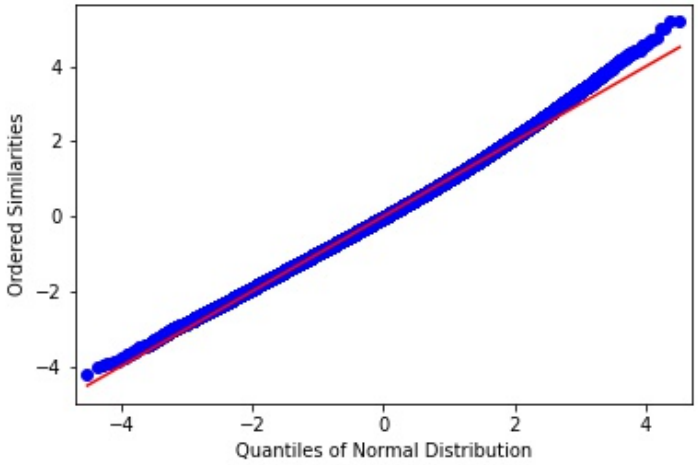


Figure A.3: Quantile-Quantile plot of similarities between probe/gallery images and cohort images from the CFP dataset.

results in a p-value of 0.8311 and therefore also does not reject the null hypothesis. In both comparisons, the average accuracy decreases by 0.00767 and 0.00607, respectively. Hence, changing all profile images to have the same direction of orientation does not have a significant influence on the Baseline method.

A.2 Analysis drop of similarity

Other rank list comparison methods from the literature, like in equation 2.10, only uses the lowest few ranks. This is motivated by the fact that they *"[...] discovered empirically that the similarity value between the probe and the Library instances drops significantly at some point."* (Schroff et al., 2011). Such a strong drop of the similarity with our data could be relevant to analyze the favoring of rank lists over the exact values similarity values. This would mean that the similarities, instead of being normally distributed, become very rare below a certain similarity value. Such data, where the mean is higher than the median, is also called *right skewed*. To find such a behaviour of the data in the CFP data set, the similarities between each probe/gallery image from the first split and all images from the other nine splits are analysed. The similarities are normalized with the mean and standard deviation of each probe/gallery image, in order to consider potential different distributions of different probe/gallery images. To show whether these similarities are normally distributed, a QQ-plot is used. The sorted similarities are plotted with the quantiles of the normal distribution, which would lie on a diagonal line if the similarities are also normally distributed. This works independently of the mean and standard deviation of the distribution, since it is not the absolute value of the distribution that is relevant, but its order. Figure A.3 shows that the distribution of the similarities is very slightly *right skewed*.

This fact could be an explanation of the results shown in figure 4.5b, where the results do not improve by only using the lowest few ranks.

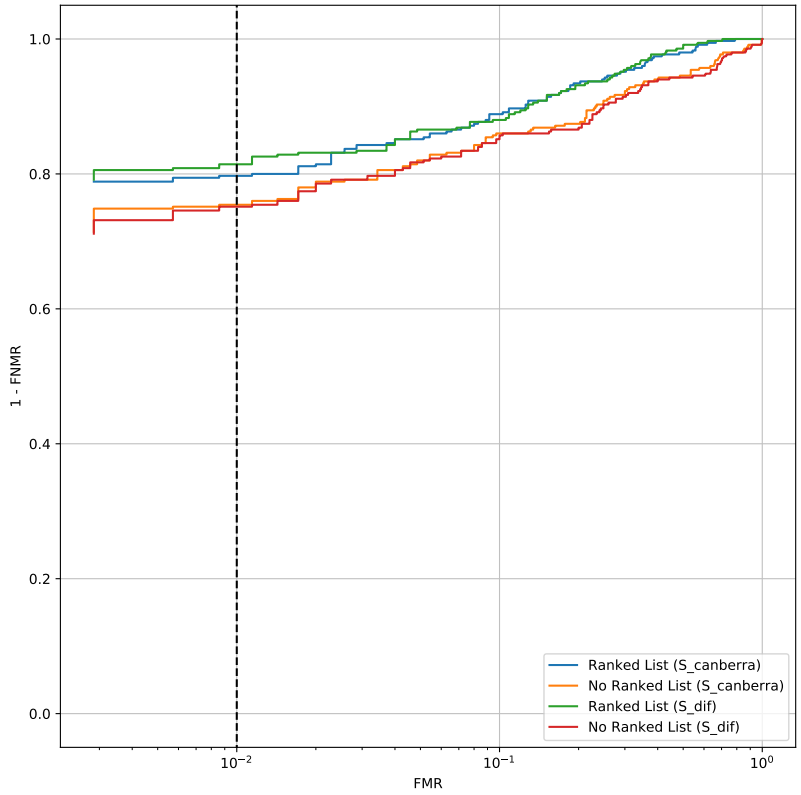


Figure A.4: ROC plot of method with and without ranked list creation step. Applied on first split of CFP data set.

A.3 No Rank List

This section shows the results when applying the Ranked List algorithm but without the step of converting the list of maximal similarities, between the probe/gallery image and the cohort, to a rank list. The lists of similarities are compared using the idea of the Canberra distance, first mentioned in G. N. Lance (1966). The distance $d_{canberra}$ between the two lists of similarities is computed and subtracted from 1 to receive a similarity value again. This is shown in equation A.1, where π and γ are the two lists containing the maximal similarity value to each of the individuals from the model data base, and $S_{canberra}$ the similarity value of those two lists.

$$S_{canberra}(\pi,\gamma) = 1 - d_{canberra}(\pi,\gamma) = 1 - \sum_{m=0}^{N_M-1} \frac{|\pi(m) - \gamma(m)|}{|\pi(m)| + |\gamma(m)|} \tag{A.1}$$

By not creating rank lists and computing the similarity value with $S_{canberra}$ the method achieves an accuracy of 0.8657 and a TMR of 0.7514 at a FMR of 10^{-2}. The comparison to the same method, but using rank lists and the list comparison function $S_{canberra}$ is shown in figure A.4. In addition, the result of the comparison method S_{dif} with and without the rank list creation step is shown.

This comparison suggests strongly that using rank lists is beneficial to the performance of the entire method.

	ResNet50		VGG-16	
	Accuracy	TMR	Accuracy	TMR
max	**0.8886**	**0.8086**	**0.7629**	**0.2943**
min	0.8857	0.7971	0.7400	0.2457

Table A.1: Accuracy and TMR (FMR = 10^{-2}) of Ranked List algorithm with S_{dif} as rank list comparison method. The rank list is created once with the maximum (max) of the similarities to all reference images of one individual and once with the minimum (min). Applied on the first split of CFP data set as test and the other nine as training sets. ResNet50 and VGG-16 as feature extraction methods.

A.4 Analysis of images with highest similarity

When using a model data base with no labels of orientation, the rank list method mainly benefits from the following findings of Moses Yael (1994): *" [...] the variations between the images of the same face due to illumination and viewing directions are almost always larger than image variations due to a change in face identity.* Also Schroff et al. (2011) observed: *"[...] although the comparison was not restricted to specific pose, [...] the SSIM measure returns look-alikes that are imaged in very similar conditions."* The SSIM measure is a similarity value of two images. This means in words used in this thesis, when comparing a probe image with profile orientation to images of one identity from the cohort, having different orientations, it will be most of the cases most similar to an image as well in profile orientation. And vice versa, a frontal probe image would be most similar to a frontal image of the cohort.

To show this observations in the CFP data set, the frontal and profile images from the first split are compared to all individuals of the other nine splits. In 80.14% of the comparisons they are most similar to an image with same orientation. In comparisons of frontal to profile images has the frontal image in 6.87% of the cases the highest similarity to the same image than the profile image.

A.5 Maximal vs. Minimal Similarity

A key aspect of creating the rank lists is that for each identity, the similarity of the image with the highest similarity to the test image is used. An experiment was carried out to quantify the magnitude of this importance. The Ranked List algorithm was carried out once as described in section 3.2.4 and then once where the image with the smallest similarity of each individual was used for the rank list creation.

In table A.1 one can see that the accuracy and TMR is higher when using the maximum instead of the minimum. However, it should be noted that this step only makes the method 0.29% more accurate and a 1.15% higher TMR in the case of using ResNet50. When using VGG-16 as feature extraction method, the accuracy increases 2.29% and the TMR 4.86%.

Bibliography

Black Jr., J. A., Gargesha, M., Kahol, K., Kuchi, P., and Panchanathan, S. (2002). Framework for performance evaluation of face recognition algorithms. In Smith, J. R., Panchanathan, S., and Zhang, T., editors, *Internet Multimedia Management Systems III*, volume 4862, pages 163 – 174. International Society for Optics and Photonics, SPIE.

Brunelli, R. and Poggio, T. (1993). Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052.

Chen, S., Liu, Y., Gao, X., and Han, Z. (2018). Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In Zhou, J., Wang, Y., Sun, Z., Jia, Z., Feng, J., Shan, S., Ubul, K., and Guo, Z., editors, *Biometric Recognition*, pages 428–438, Cham. Springer International Publishing.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

G. N. Lance, W. T. W. (1966). Computer programs for hierarchical polythetic classification ("similarity analyses"). *The Computer Journal*, 9(2):60–64.

Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-pie. *Image and Vision Computing*, 28(5):807 – 813. Best of Automatic Face and Gesture Recognition 2008.

Günther, M., Wallace, R., and Marcel, S. (2012). An open source framework for standardized comparisons of face recognition algorithms. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *European Conference on Computer Vision (ECCV) Workshops and Demonstrations*, volume 7585 of *Lecture Notes in Computer Science*, pages 547–556. Springer.

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 Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 87–102, Cham. Springer International Publishing.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. (2008). Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille, France. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

Jafri, R. and Arabnia, H. R. (2009). A survey of face recognition techniques. *journal of information processing systems*, 5(2):41–68.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Knight, W. R. (1966). A computer method for calculating kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61:436–439.

Kroese, B., Krose, B., van der Smagt, P., and Smagt, P. (1993). An introduction to neural networks.

Metz, C. E. (1978). Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. WB Saunders.

Müller, M. K., Tremer, M., Bodenstein, C., and Würtz, R. P. (2013). Learning invariant face recognition from examples. *Neural Networks*, 41:137 – 146. Special Issue on Autonomous Learning.

Moses Yael, Adini Yael, U. S. (1994). Face recognition: The problem of compensating for changes in illumination direction. In Jan-Olof, E., editor, *Computer Vision — ECCV 94*, pages 286–296, Berlin, Heidelberg. Springer Berlin Heidelberg.

Müller, M. K. (2010). *Lernen von Identitätserkennung unter Bildvariation*. doctoralthesis, Ruhr-Universität Bochum, Universitätsbibliothek.

Müller, M. K., Heinrichs, A., Tewes, A. H. J., Schäfer, A., and Würtz, R. P. (2007). Similarity rank correlation for face recognition under unenrolled pose. In Lee, S.-W. and Li, S. Z., editors, *Advances in Biometrics*, pages 67–76, Berlin, Heidelberg. Springer Berlin Heidelberg.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks.

Parkhi, O., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. pages 1–12. British Machine Vision Association.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., and Worek, W. (2006). Preliminary face recognition grand challenge results. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 15–24.

Schroff, F., Treibitz, T., Kriegman, D., and Belongie, S. (2011). Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In *2011 International Conference on Computer Vision*, pages 2494–2501.

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 (WACV)*, pages 1–9.

Sun, Y., Liang, D., Wang, X., and Tang, X. (2015). Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

Taherkhani, F., Talreja, V., Dawson, J., Valenti, M. C., and Nasrabadi, N. M. (2020). Pf-cpgan: Profile to frontal coupled gan for face recognition in the wild.

Vigna, S. (2015). A weighted correlation index for rankings with ties. In *Proceedings of the 24th International Conference on World Wide Web*, page 1166–1176, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wiskott, L., Krüger, N., Kuiger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.

Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458.

Zheng, T. and Deng, W. (2018). Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Number 18-01.

Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.