IDIAP COMMUNICATION REPORT

FACE RECOGNITION SYSTEMS: PERFORMANCE EVALUATION AND BIAS ANALYSIS

Yannick Dayer

Idiap-Com-04-2020

AUGUST 2020

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

FACE RECOGNITION SYSTEMS:

PERFORMANCE EVALUATION AND BIAS ANALYSIS

Student:	DAYER Yannick
Student number:	12-687-067
Project supervisor:	Dr. MARCEL Sébastien
Company supervisor:	TARSETTI Flavio

Master's Dissertation presented on June 15, 2020

Martigny, Idiap Research Institute

Acknowledgements

I would like to express my very great appreciation to Dr Sébastien Marcel for his valuable and constructive suggestions during the development of this project.
I am grateful for the assistance given by Mr. Flavio Tarsetti.
I would also like to express my special thanks to my colleagues of the Biometrics Security and Privacy group as i came to know about so many new things.
Finally I would also like to thank my parents and friends who helped me a lot with their continuous support throughout the duration of this project.

Martigny, June 15, 2020 Y. D.

Abstract

User authentication is a crucial part of data security, and biometrics is an advantageous way of achieving this. Face images capture being minimally invasive and easy to acquire makes face recognition a good contender for being used in a lot of applications that require to know if the user is really who he claims he is.
In this thesis, I compare the performance of multiple existing face recognition systems on different datasets.
I then present how a convolutional neural network system works, and show the performance results of such a system trained from scratch for face recognition. I show that training a big neural network with few images is detrimental, and a big training dataset is required.
An experiment on racial bias evaluation is then presented with methods to reduce the disparity between ethnicity in the products of a face recognition system.

Keywords: Artificial Intelligence, Facial Recognition, Neural Network, Convolutional Neural Network, bias

Résumé

Authentifier une personne est une partie cruciale dans la protection des données, et la biométrie apporte une solution avantageuse pour ce problème. Comme la reconnaissance faciale est un moyen peu invasif de s'assurer de l'identité d'une personne, c'est un bon candidat pour être utilisé dans de multiples applications demandant de s'assurer que l'identité qu'une personne est vraiment celle qu'elle prétend.
Dans ce document, je compare la performance de plusieurs systèmes de reconnaissance faciale existants à l'aide de différentes bases de données.
Ensuite je présente comment fonctionne un réseau neuronal convolutionel et évalue la performance d'un tel système entraîné de zéro. Je montre qu'entraîner un grand réseau neuronal avec une petite base de donnée est nuisible, et qu'une base de donnée conséquente est nécessaire.
Finalement, j'expérimente sur le biais ethnique de plusieurs systèmes et présente des méthodes pour réduire ces disparités.

Mots-clés : Intelligence Artificielle, Reconnaissance Faciale, Réseaux Neuronaux, Réseaux Neuronaux Convolutionnels, biais

Contents

Acronyms

CNN	*Convolutional Neural Network*
DET	*Detection Error Tradeoff*
EER	*Equal Error Rate*
FAR	*False Acceptance Rate*
FMR	*False Match Rate*
FN	*False Negative*
FNMR	*False Non-Match Rate*
FP	*False Positive*
FRR	*False Rejection Rate*
GMM	*Gaussian Mixture Model*
HTER	*Half Total Error Rate*
ISV	*Inter-Session Variability*
MEDS	*Multiple Encounter Dataset*
PCA	*Principal Component Analysis*
ROC	*Receiver Operating Characteristic*
TN	*True Negative*
TP	*True Positive*

List of Figures

1 Introduction

Nowadays, defining the identity of a person can be crucial, to ensure that this person is authorized to have access to a restricted area (physical or virtual), or to verify that they are really the person they claim to be. If a system of password or physical key can be used, there are different ways of confirming a claimed identity.

1.1 Biometrics

Biometrics is the science of defining and recognizing the identity of a person based on physical attributes (face, fingerprints, voice, iris, etc.) or behavior (gait, key presses rhythm, etc.) Biometric traits are great for identification, because they are always available on a person, and cannot be lost like a password, a key or a card. We always have our fingerprints on person (unless a severe accident happens).

However, there are also disadvantages with biometric identification. Where a password or a key will always be exactly the same when used, a biometric trait can change over time. The voice can change in pitch according to the stress or emotions of the person, make-up can be applied to a face, behavior can change. This means that biometric recognition cannot always be 100% accurate. Another point of biometric traits is that they cannot be replaced. If a good enough copy of a person trait is made, like a mold of a fingerprint, there is no way to issue a new finger like we would do with a password. Once a trait is compromised, it is compromised for life. Of course there are ways to design a system so it is not susceptible to presentation of such copies. The field of Anti-spoofing is responsible to design such systems.

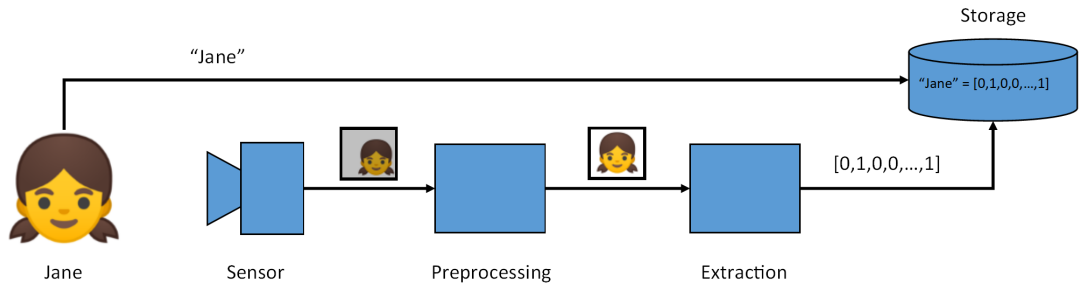


Figure 1.1 – Depiction of the face recognition workflow for enrollment.

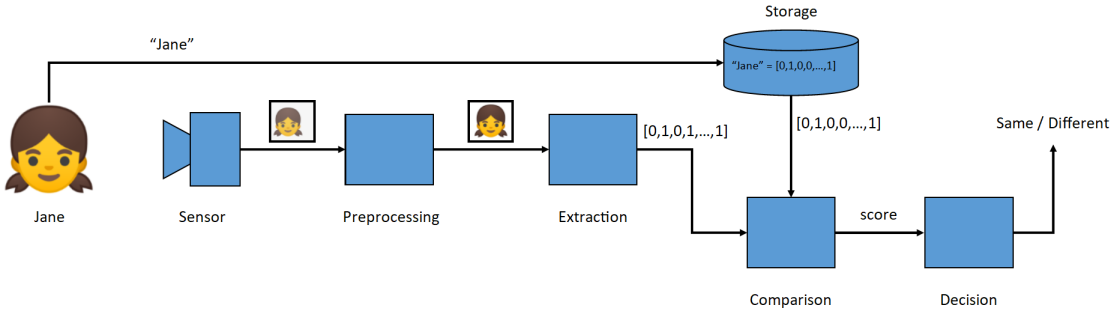


Figure 1.2 – Depiction of the face recognition workflow for verification.

1.2 Face Recognition

The face is the most convenient trait to use to determine the identity of a person and is done every day naturally by people to recognize others. The capture of this trait for identity recognition is advantageous for being non-intrusive and natural, and can be done at a distance, and even without the person knowing about it.

Face recognition systems have been refined and improved to allow machines to differentiate people almost as well as humans do (Phillips and O'toole [2014]), using different approaches. For a machine to recognize a face, the first step is to capture an image of it with a sensor (camera), then process the resulting image to remove variabilities due to sensor, face position, location or luminosity differences. Then, an algorithm takes this face image and extracts a set of features representing the identity of the person. This set of feature can be saved to a database to memorize this identity (enrollment, Figure 1.1), or it can be compared to a previously saved reference to decide if the person in front of the sensor is the person in claims to be (verification, Figure 1.2).

1.3 Bob

During this project, the toolbox developed and used by the Biometric Security and Privacy group, Bob (Anjos et al. [2012] and Anjos et al. [2017]) was extensively used. This tool is a set of software packages aiming at producing reproducible research. It contains standardized access to various databases, plenty of different machine learning algorithms, mainly in the field of biometrics.

1.4 Scope

This master thesis focuses on performance evaluation of face recognition systems, and estimation of recognition bias that occur given different covariates, like gender or race, and how to reduce these existing bias.

1.5 Organization of the Master's Dissertation

This dissertation contains four more chapters.

Chapter 2 describes how to evaluate face recognition systems and the evaluation methods and metrics that allows to compare their performance. This chapter also presents the evaluation results of different systems on various datasets.

Chapter 3 defines the basics of convolutional neural networks and their use for face recognition, and presents the results of a trained network.

Chapter 4 presents the results of the study of bias in face recognition systems, as well as a method to counteract the effect of those bias.

Finally, chapter 5 presents the conclusion and future work.

2 Evaluation of face recognition systems

This chapter will present how to evaluate the performance of biometric systems (particularly face recognition systems) in order to compare and define which one is the best for a given application. The methods and results will be described, then examples of different systems evaluation results will be presented.

2.1 Verification

A face recognition system used for verification needs to decide if the identity of a person (probe) presented to it corresponds to a claimed identity (reference or model). To do so, the system needs to know the representation of the identity of each user, which is given during the enrollment process.

Each probe can either be of the same identity as the reference, this group of probe and reference will be designed as *genuine*, or the probe can be of different identity than the reference, and is then considered as *impostor* (also called zero-effort impostor, if no spoofing effort is involved). see Figure 2.1.

A face recognition system used for verification takes two inputs, an image of the face of a person, and a claimed identity that corresponds to a previously enrolled identity.

2.2 Evaluation

To evaluate the performance of a system a great quantity of comparisons is required with different people identities, preferably representing the population using the system in its normal usage scenario. A significant amount of identities are enrolled in the system, and probes are tested against it. To facilitate the process, face picture

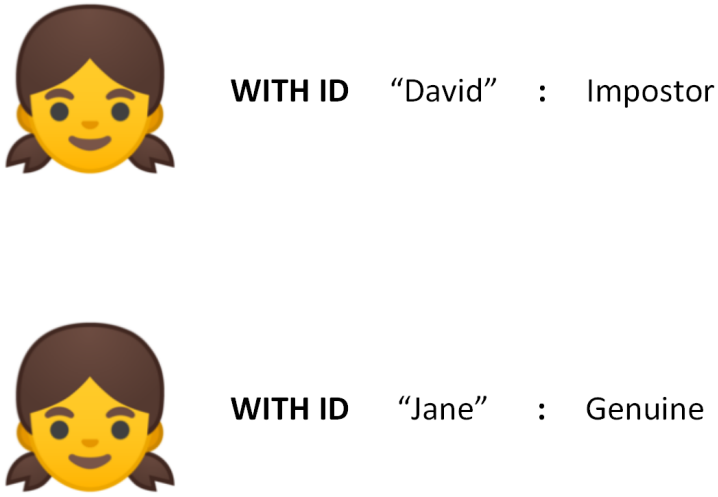


Figure 2.1 – Example of impostor and genuine pairs.

datasets were assembled, consisting of pictures of the face of different people labeled with their respective identity. In order to evaluate a system, a part of these datasets pictures is used to enroll the identity of all the subjects, then the rest of the pictures is used as probe. Each probe will be presented against every known identity to generate the most scores possible. It is also possible to define pairs that will be tested, instead of testing every possible combinations.

Due to the nature of biometrics data, being variable over time and sensitive to variations (luminosity, pose, noise, etc.), the comparison step of the system will not output a binary value for each comparison, but a numerical score. For each presented probe face picture and identity pair, the system returns either a similarity score (higher score means same identity), or a distance score (higher score means different identity). From now on, all scores will be assumed to be similarity scores, unless specified.

The score computation can be represented as:

$$s(e, p) \tag{2.1}$$

where e is the enrolled representation and p is the probe representation.

The evaluation comparisons will generate a score for each probe-reference pair (see Figure 2.2).

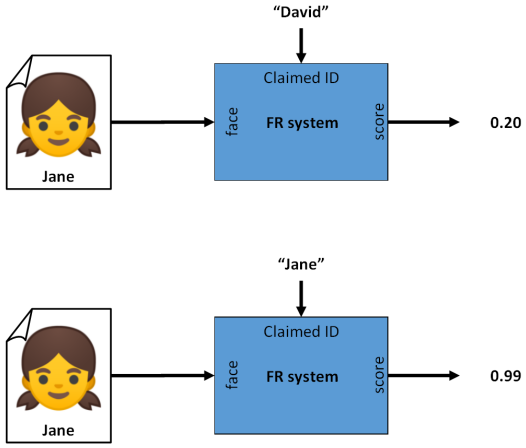


Figure 2.2 – Two comparisons, one impostor (wrong identity, top) and one genuine (bottom).

Probe identity	Reference identity	score
Jane	David	0.20
Jane	Mary	0.45
Jane	Jane	0.99
David	David	0.85
...	...	...

Table 2.1 – Example of scores resulting from an evaluation.

2.3 Threshold and decision

The comparisons give a series of scores that are labeled with the identities of the probe and reference (see Table 2.1). The system must then be designed to decide if a score represents a genuine match or an impostor one. This decision is done with a threshold value T, estimating all scores over it to be genuine matches, and all the scores under that value to be impostors matches:

$$\begin{cases} \text{predict a Match (Genuine)}, & \text{if } s(e,p) \geq T \\ \text{predict a Non Match (Impostor)}, & \text{if } s(e,p) < T \end{cases} \tag{2.2}$$

		Real Condition	
		Same Identity	Different Identity
Prediction	Same Identity	True Positive	False Match, Type I error
	Different Identity	False Non Match, Type II error	True Negative

Table 2.2 – Representation of all cases of a biometric recognition problem.

At the decision stage, four outcomes are possible:

- **True Positive (TP)**: The identity of the probe and reference is the same (Condition Positive), and the system estimated it was a match (Prediction Positive).
- **True Negative (TN)**: The identity of the probe and reference is different (Condition Negative), and the system estimated it was not a match (Prediction Negative).
- **False Positive (FP)**: The identity of the probe and reference is different (Condition Negative), and the system estimated wrongly it was a match (Prediction Positive).
- **False Negative (FN)**: The identity of the probe and reference is the same (Condition Positive), and the system estimated wrongly it was not a match (Prediction Negative).

After the scoring and decision over a full dataset, the number of each outcome can be used to evaluate the performance of the system.

In a perfect system, the distribution of impostors scores would be completely separated from the distribution of genuine scores (see Figure 2.3). But most of the time, these distributions overlap, making the decision harder, leading to compromises to be made. The choice of the threshold value will affect how much of each type of error the system makes.

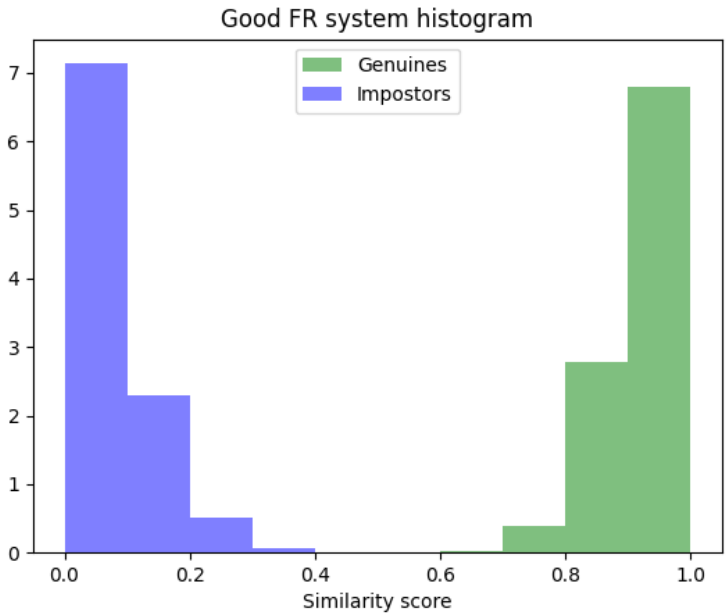


Figure 2.3 – Scores distributions of a good face recognition system. All the genuine match scores are close to 1, and the impostor pairs scores are close to 0. A threshold value of 0.5 will give a perfect separation with no error.

2.4 Errors

A face recognition system can commit the two types of error seen before:

- Missing the correspondence between two faces of the same identity, known as type I error, False Non Match or False Negative.
- Recognizing two different identity as the same, known as type II error, False Match, or False Positive.

Since no system is perfect, the goal is to reduce these errors to a minimum. But reducing one error type means increasing the other, since to do so the threshold value is changed. The goal is to find the threshold value that satisfies a given problem. Some use-cases requiring more security will tend to raise the threshold value, so less type I errors are made, reducing the number of falsely accepted identifications, at the expense of more type II errors being made (see Figure 2.4). Some systems may focus on the user-friendliness and prefer a lower threshold value, allowing more impostors to be allowed but improving the genuine users convenience (the user will be more easily accepted, not needing multiple tries).

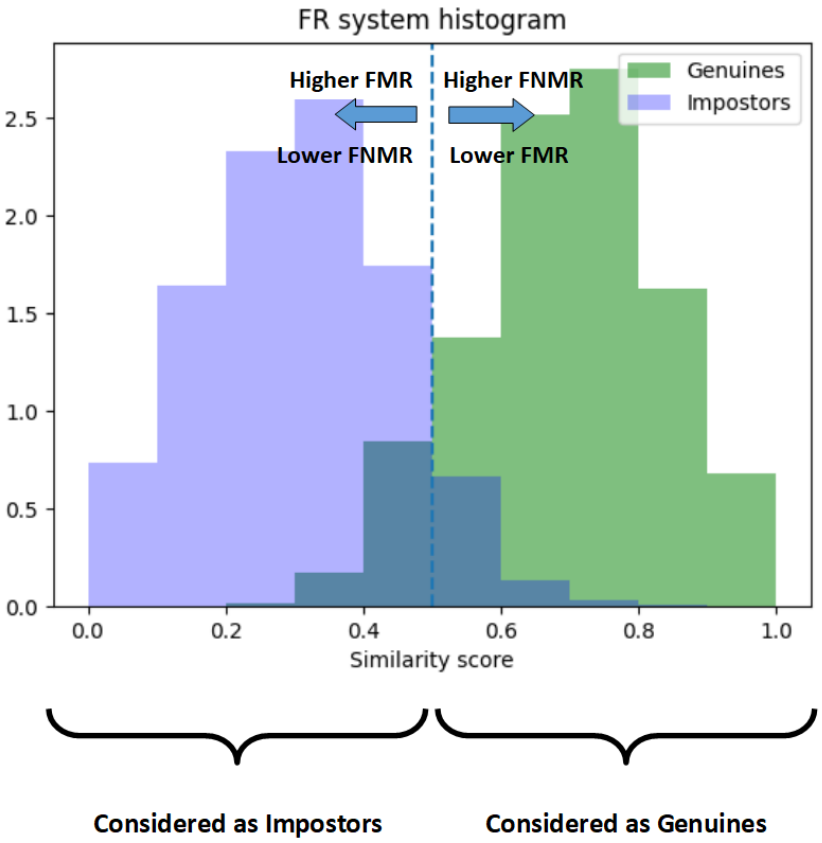


Figure 2.4 – Types of error accentuated by changing the threshold value. It is impossible to choose a threshold value that gives no error in this example. The difference between FMR and FNMR can be minimized (EER), or one type of error can be reduced over the other being increased.

For example, a system used in a border security post that verifies that the identity of a person corresponds to the one in their passport should have a low threshold value, to prevent impostors to cross with stolen papers, maybe requiring an image capture step to be more constrained (fixed position, no smiling, etc). On the opposite, a smartphone or computer unlock system would tend to raise its threshold value to accommodate the users: if any noise or light fluctuation makes the recognition of the registered user fail, requiring a few new take at every use (and it could be multiple times a day), the user would not be satisfied.

2.5 Evaluation data

To evaluate a system for face recognition, a face dataset is used that will provide a set of enrollment pictures as well as a set of probing pictures. Each probe is repeatedly

fed to the system with every identity of the enrolled subjects. This generates a score for every reference-probe pair (see Table 2.1). A good system would give high scores for pairs whose identity is the same for the reference and the probe, and low scores for pairs of different identities.

2.5.1 Evaluation sets

To better represent a real-case scenario, the evaluation of a system should be done with two different sets of data. A first set of identities (development set, *dev*) is used to define a threshold value. This would represent the threshold value set in the final implementation in a real-case scenario. The second set of identities (evaluation set, *eval*), with different subjects than those in the development set, will serve to evaluate the performance of the system given the threshold value defined previously.

2.6 Evaluation metrics and graphs

Metrics are defined to compare different systems. They are generated using the scores and labels of the evaluations done with the data from a dataset, and allow to compare the performance of a system on a dataset. The True Positives, True Negatives, False Positives, and False Negatives can be retrieved from these scores and labels, and a given threshold value.

2.6.1 FMR, FNMR and EER

The quantity of error of each type a system commits is represented by the False Match Rate (FMR) for type I error and False Non Match Rate (FNMR) for type II error. Let FP be the number of false matches and TN be the number of true non-matches, the FMR is defined as:

$$\text{FMR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.3}$$

And with FN being the number of false non-matches and TP the number of True matches, the FNMR is defined as:

$$\text{FNMR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{2.4}$$

Rising the threshold value results in higher FMR and lower FNMR. Inversely, lowering the threshold value increases the FNMR and lowers the FMR (see Figure 2.4). The

error rate made by the system where the threshold value is selected to equalize the FMR and FNMR is named Equal Error Rate (EER). The following results were obtained by using a threshold value at EER on the development set.

2.6.2 HTER

The HTER (Half Total Error Rate) is a metric that can be used to measure the detection performance. It is defined as:

$$\text{HTER} = \frac{\text{FMR} + \text{FNMR}}{2} \tag{2.5}$$

2.6.3 ROC and DET

The Receiver Operating Characteristic (ROC), and Detection Error Tradeoff (DET), are two curves representing the evolution of the error types for different threshold values. They represent the performance of a system on a graph, with the FMR on the x-axis, and the True Match Rate on the y-axis for the ROC or FNMR on the y-axis for the DET. The axis can use a logarithmic scale to improve the difference between the curves of well performing systems. The Figure 2.5 and Figure 2.6 show examples of ROC and DET curves. An ideal system would be represented by a point on the coordinate $(0;1)$ for a ROC curve, or a point in $(0;0)$ for a DET curve.

The Area Under the Curve (AUC) can summarize the performance of a system using the ROC or DET curve. A perfect system will have an AUC of 1 for a ROC curve, or an AUC of 0 for the DET curve. The AUC for a ROC curve will be in the range 0.5 (worst) to 1 (perfect recognition), as the worst possible case (random guesses, distributions overlapping completely) will result in a linear curve between FMR and 1-FNMR. AUC below 0.5 indicates that the system is inverting the match and non-match labels. This could be an indication of an implementation error.

2.7 Evaluated systems

Multiple face recognition systems were evaluated, from the older methods using applied linear algebra to the modern Convolutional Neural Networks and commercial systems.

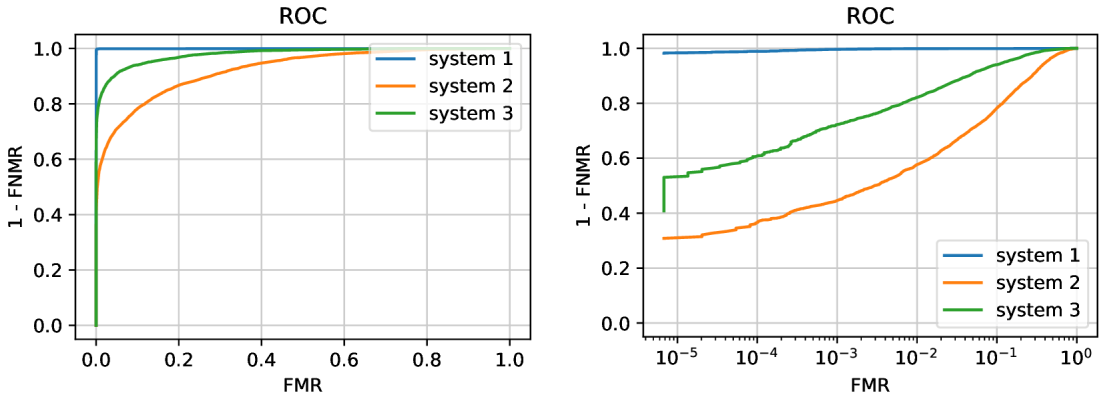


Figure 2.5 – Example of ROC curves comparing three systems. The two graph represent the same data, but the right one uses a logarithmic scale on the X axis. We can deduce that *system 1* performs the best (almost perfect) followed by *system 3*, and *system 2* is the worst of this set.

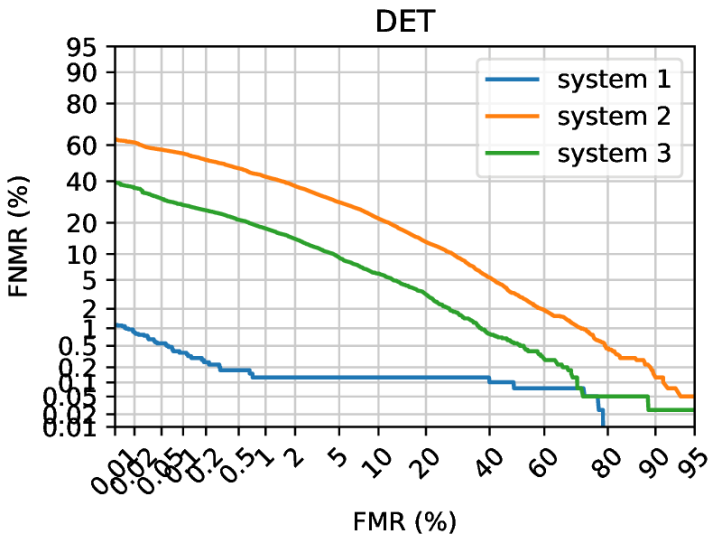


Figure 2.6 – Example of DET curve comparing two systems. Of the three evaluated systems, *system 1* performs the best, followed by *system 3* and *system 2*.

2.7.1 Eigenface

Eigenface is a system using eigenfaces for face recognition based on Principal Component Analysis (PCA). It extracts eigenvectors from face images, retaining a set of main features serving to recognize the identity of a face. (Turk and Pentland [1991])

This algorithm works best on aligned faces that are upright and are described by a set of two-dimensional features.

2.7.2 Gabor-Graph

The Gabor Graph method (Günther et al. [2012] and Zhang et al. [2005]) uses a series of Gabor jets applied a different points of a face to identify a person. A Gabor jet is a collection of response of Gabor wavelets at one point of the image.

2.7.3 ISV

The ISV algorithm (for Inter-Session Variability, Wallace et al. [2011]) is inspired form GMM (Gaussian Mixture Model, McCool and Marcel [2009b]) extensively used in speaker recognition. It focuses on eliminating the impact of inter-session variation (lighting, background, pose or expression changes) on face recognition.

2.7.4 Facenet and Inception ResNet v2

Facenet (Schroff et al. [2015]) [1] and Inception ResNet v2 (Szegedy et al. [2016]) are two architecture of Convolutional Neural Network.

Facenet is based on the Inception ResNet v1 architecture and was trained for face recognition.

The Inception ResNet v2 is an image classification architecture, and the model used here was trained for face recognition at the Idiap Research Institute, on the MS-Celeb dataset (Guo et al. [2016]).

[1] model taken from https://github.com/davidsandberg/facenet

2.7.5 RankOne

ROC by RankOne [2] is a commercial system featuring small times of enrollment and identity comparison, a small template size (enrollment features), and a compatibility with a wide variety of development systems (Android, Linux, Windows, MacOS/iOS).

2.8 Evaluation datasets

These systems were tested on different datasets that feature different aspects of face recognition.

2.8.1 AT&T (ATNT or ORL face database)

The AT&T dataset [3] (formerly the ORL face database) was created by the AT&T laboratories of Cambridge. This small set of black and white face pictures is not recommended to be used for evaluating a system, but its size allows to rapidly test an implementation. Most systems will have no problem to reach an accuracy of 1 on this set.

It is composed of 400 pictures from 40 subjects (10 images for each subject). The pictures are fairly constrained (Only frontal pose) and taken close to the subject with a uniform dark background. However, for some subjects, the images were taken under different lighting condition, and with different facial expressions. The images are in black and white and have a resolution of 92x112. Figure 2.7 shows a part of this dataset.

2.8.2 The MOBIO dataset

The mobile biometry dataset (MOBIO, McCool and Marcel [2009a]) focuses on bimodal recognition (audio and video) on mobile environments. A subset of face images (frames extracted from the video part of the dataset) was used for the evaluation.

The set was captured in six sessions and in six different places around the world. The images subset contains over 32400 pictures from 160 subjects. Each picture is in color and has a resolution of 640x480. Figure 2.8 presents a few pictures of different subjects over multiple sessions.

The mobile focus of this dataset makes it difficult as the lighting and background can

[2] https://www.rankone.io/
[3] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Figure 2.7 – Part of the AT&T dataset. Credit to the AT&T Laboratories Cambridge.

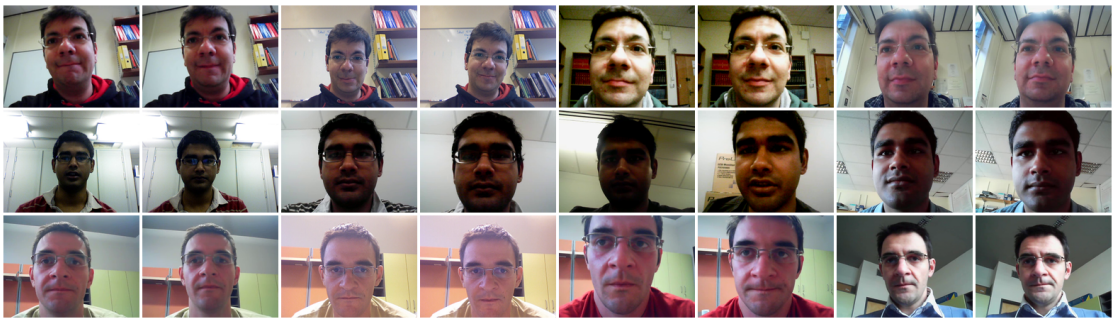


Figure 2.8 – Samples from the MOBIO dataset. Each row belongs to one subject in four different sessions.

change between sessions, as well as the device used to capture.

2.8.3 LFW (Labeled Faces in the Wild)

LFW (Huang et al. [2007]) is a set of face pictures gathered from the internet, designed for unconstrained face recognition. This dataset contains more than 13000 images labeled with the name of the person in the picture. 5749 subjects are represented, with 1680 of them having multiple pictures.

Each picture of this dataset has a resolution of 250x250 in color. As seen in Figure 2.9, there can be multiple faces from different persons in one picture, the background and lighting can change tremendously, and the age of the subject can vary, making this

Figure 2.9 – Samples from the LFW dataset.

dataset fairly challenging.

2.9 Results

Here are the main results of the described systems evaluated on the listed datasets. The appendix A contains all the metrics and results of each system on each dataset.

2.9.1 Evaluation on the AT&T dataset

The AT&T dataset offering very little challenge, most of the recent algorithm achieved a perfect score while evaluating on it. Figure 2.10 shows the comparison between the ROC of each tested system.

RankOne, Inception ResNet v2 and Facenet all achieve a perfect score with no miss. ISV did one False Match. And the Eigenface method had more trouble recognizing faces since not all of them were facing perfectly straight to the camera.

Since the database does not contains a lot of subjects, no distinction was made between development and evaluation sets. All the identities were used in one group.

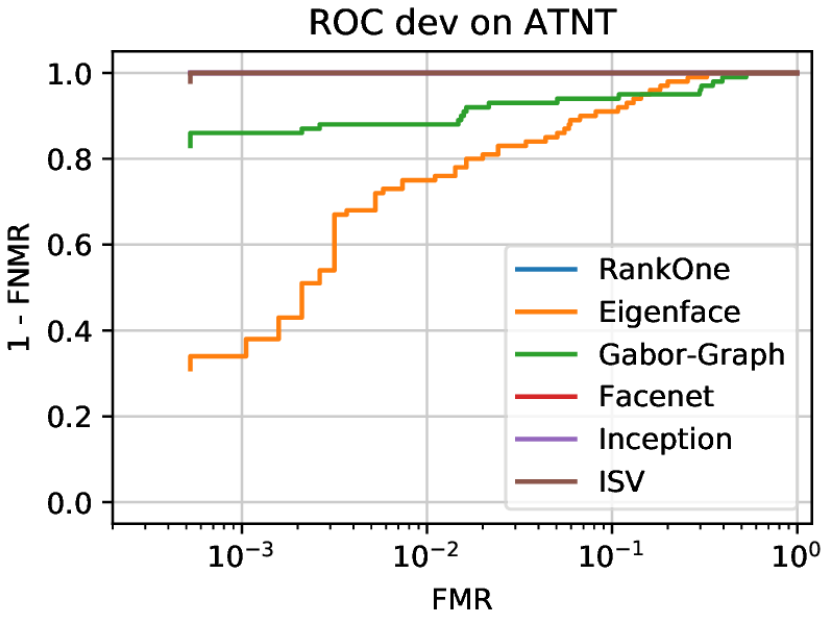


Figure 2.10 – ROC of various face recognition algorithms on the AT&T dataset. All CNN architectures achieve a perfect score on this small dataset.

System	FMR	FNMR	HTER
Eigenface	9.00%	9.00%	9.00%
Facenet	0.00%	0.00%	0.00%
Gabor graph	6.00%	6.00%	6.00%
Inception ResNet v2	0.00%	0.00%	0.00%
ISV	0.05%	0.00%	0.03%
RankOne	0.00%	0.00%	0.00%

Table 2.3 – Metrics of the evaluated systems on AT&T at EER for each system.

The Table 2.3 shows the FMR, FNMR and HTER of each system. The threshold value was chosen at EER on the development set for each system.

Figure 2.11 shows how differently Facenet and Eigenface separate the Impostors from the Genuines. The threshold value is set at the EER point for each system.

2.9.2 Evaluation on the MOBIO dataset

The MOBIO dataset being more challenging than the AT&T dataset, the results allow to work out the best algorithm between the four that tied on the previous dataset (ISV, Inception ResNet v2, Facenet and RankOne).

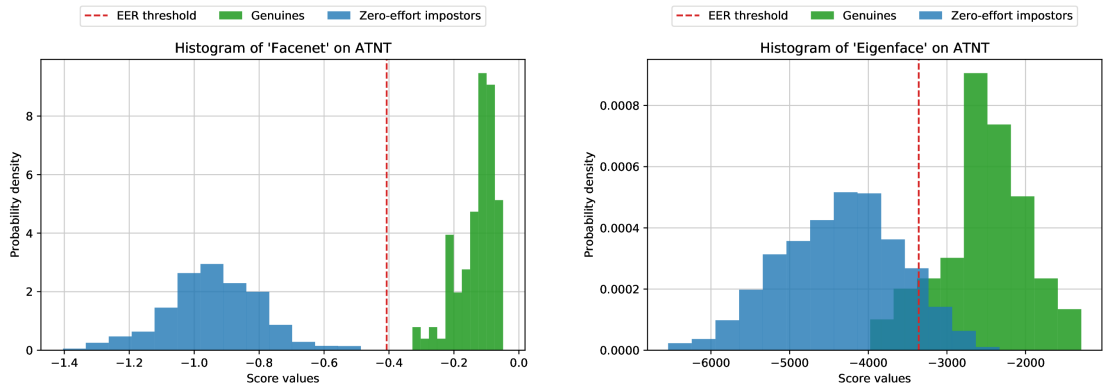


Figure 2.11 – Histogram of scores from Facenet (left) and Eigenface (right) on the AT&T dataset. The scores of Genuines and Impostors are well separated with Facenet, but much less using Eigenface.

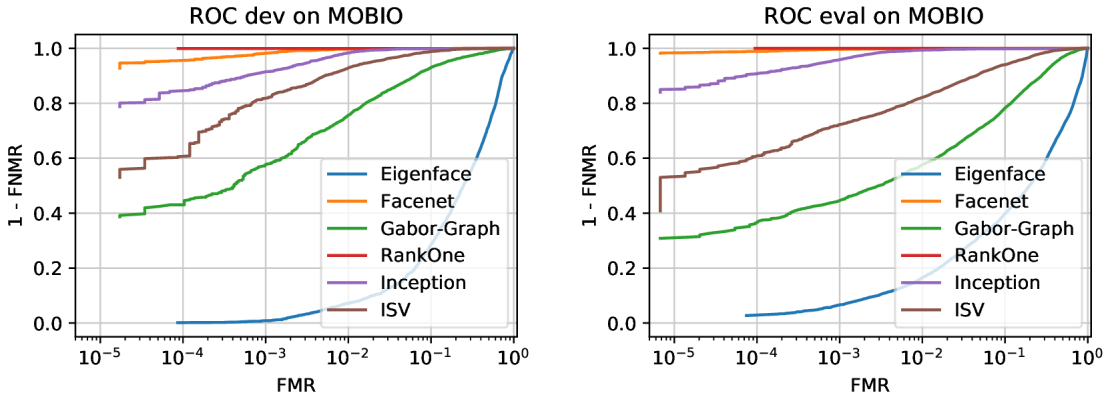


Figure 2.12 – ROC curves of the systems evaluated on the MOBIO dataset.

System	FMR	FNMR	HTER
Eigenface	40.75%	33.76%	37.25%
Facenet	0.14%	0.30%	0.22%
Gabor graph	12.28%	19.45%	15.87%
Inception ResNet v2	1.15%	0.63%	0.89%
ISV	4.21%	10.25%	7.23%
RankOne	0.03%	0.03%	0.03%

Table 2.4 – Metrics of the evaluated systems on MOBIO. The threshold value is placed at EER in the development set, and results are taken from the evaluation set, for each system.

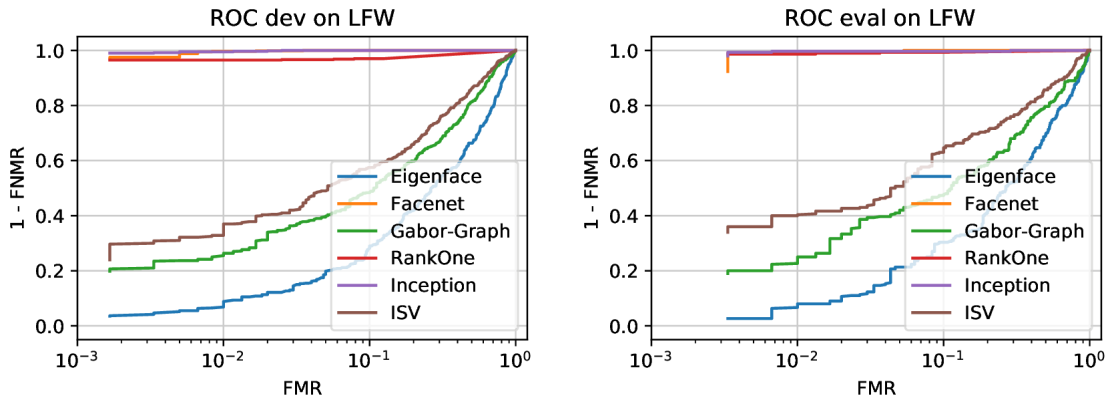


Figure 2.13 – ROC curves of the systems evaluated on the LFW dataset.

Table 2.4 and Figure 2.12 show that the best system on this dataset is RankOne, followed by both CNN-based methods. The Eigenface method has a really hard time on this dataset.

The RankOne algorithm failed to find a face in 0.10% of the images, giving a False Reject Rate of 0.13 instead of 0.03.

2.9.3 Evaluation on the LFW dataset

Figure 2.13 presents that the best system results on the LFW dataset is the Inception ResNet v2, with an HTER of 0.5%.

Table 2.5 show that the ISV algorithm is not doing as well in this dataset as with MOBIO.

System	FMR	FNMR	HTER
Eigenface	41.00%	37.33%	39.17%
Facenet	0.33%	1.67%	1.00%
Gabor graph	33.33%	30.33%	31.83%
Inception ResNet v2	0.33%	0.67%	0.50%
ISV	24.67%	27.00%	25.83%
RankOne	3.67%	0.67%	2.17%

Table 2.5 – Metrics of the evaluated systems on LFW. The threshold value is placed at EER in the development set, and results are taken from the evaluation set, for each system.

2.10 Conclusion

Out of the three database used for evaluation of these systems, the most discriminative was the MOBIO dataset. The differences in poses and the varying parameters (background and lighting) makes it a challenging set.

The best performing algorithms were the Convolutional Neural Networks and the commercial system (probably CNN-based, too). This shows the power of these tools, capable of recognizing a face in multiple situation, ignoring perturbations.

Systems like Eigenface cannot compete with the CNN-based architectures. However, CNN systems require a lot of processing power to run, but more importantly, to train. Training a big Neural Network can take days and a lot of resources.

3 Convolutional Neural Networks training

This chapter focuses on face recognition using Convolutional Neural Networks, and the training of such algorithms.

3.1 Definition

Artificial Neural Networks are a set of algorithms inspired from some parts of a brain and can be trained to recognize patterns. They consist of a set of small elements similar to neurons communicating via signals whose weight of importance can be changed. Such algorithms are "trained" to do a task by feeding them labeled data, and their weights are adjusted so that the output of the network best matches the labels.

Convolutional Neural Networks (CNN) use a set of filters applying a convolution between the input and a kernel. They are very good at handling images and extracting visual features. CNN have been around for a long time (LeCun et al. [1989]), but their use has increased significantly with the need for image recognition and the increase in processing power.

This work will focus on one architecture of CNN in particular, the Inception ResNet v2 (Szegedy et al. [2016]).

The basic idea of the Inception architectures is to have filters with different sizes applied at the same level, instead of in series one after another. This idea emerged due to the fact that a subject in an image will not always have the same scale. Allowing to detect any of the possible scale at each level can be a big improvement. Figure 3.1 presents the naïve version of one Inception module. However, to reduce the computational cost, a 1x1 convolution was added in front of the 3x3 and 5x5 convolutions.

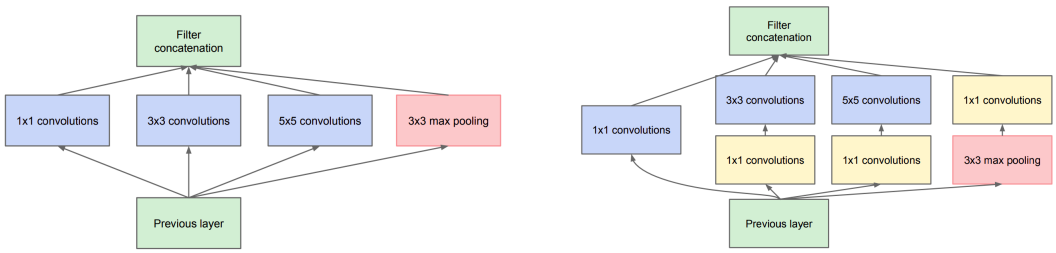


Figure 3.1 – The basic Inception module. On the left, a naïve version, and on the right, the module with dimensionality reduction. source: Szegedy et al. [2015]

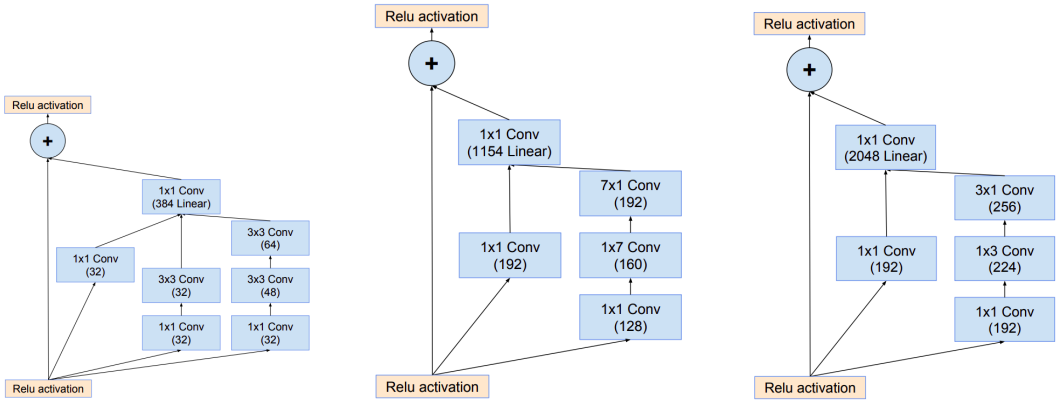


Figure 3.2 – The Inception ResNet modules. Those are the three blocks A,B and C (from left to right), used to build the full Inception ResNet architectures. source: Szegedy et al. [2017]

Later on, the Inception ResNet were developed (Szegedy et al. [2017]). The idea was to allow the input or part of the input to pass through a module (residuals) this was done by using the blocks shown in Figure 3.2.

The reductions blocks (A and B) are used to change the width and height of the grid. They serve to adapt the output of one type of module to the input of the following type (Inception-resnet-A to Inception-resnet-B, and Inception-resnet-B to Inception-resnet-C)

Those blocks are combined to build the architecture like presented in Figure 3.4. This results in the Inception ResNet v2 being composed of 572 layers and more than 55 million parameters.

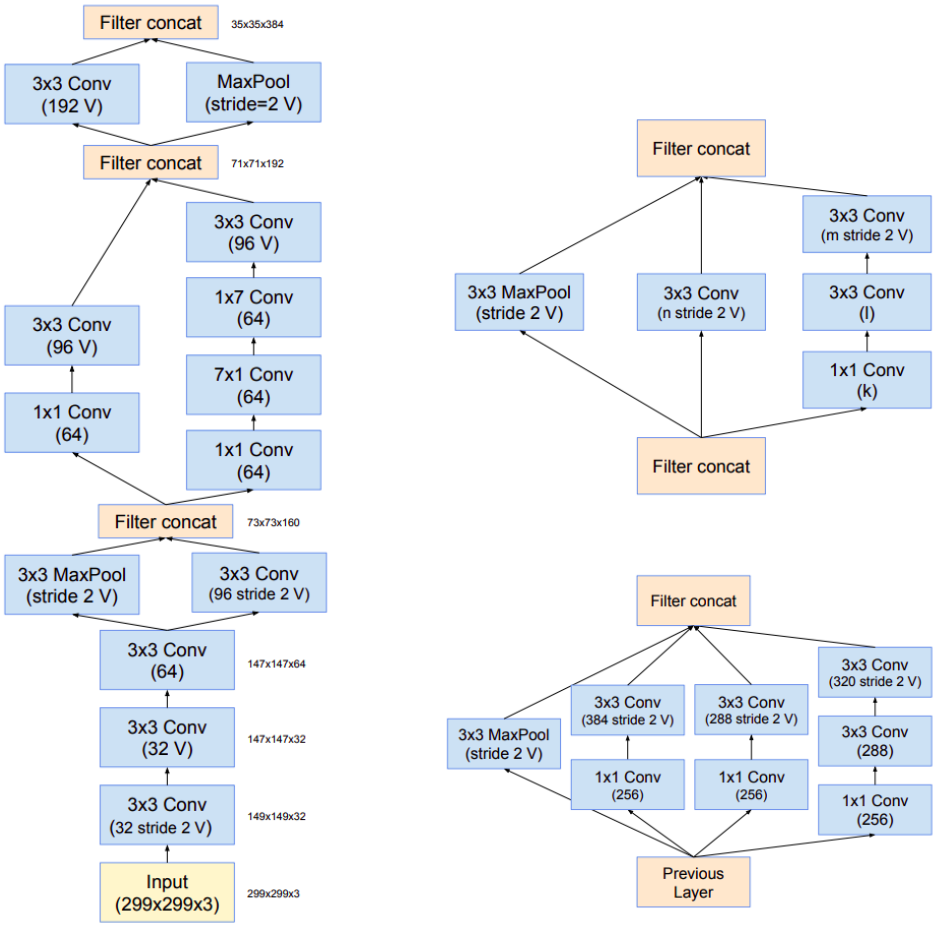


Figure 3.3 – Here are the Inception ResNet Stem (left) and reduction blocs A (upper right) and B (lower right). source: Szegedy et al. [2017]

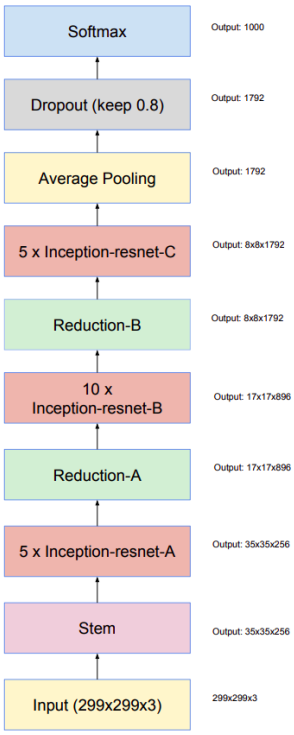


Figure 3.4 – This is the full Inception ResNet v2 architecture, using the blocks previously defined. source: Szegedy et al. [2017]

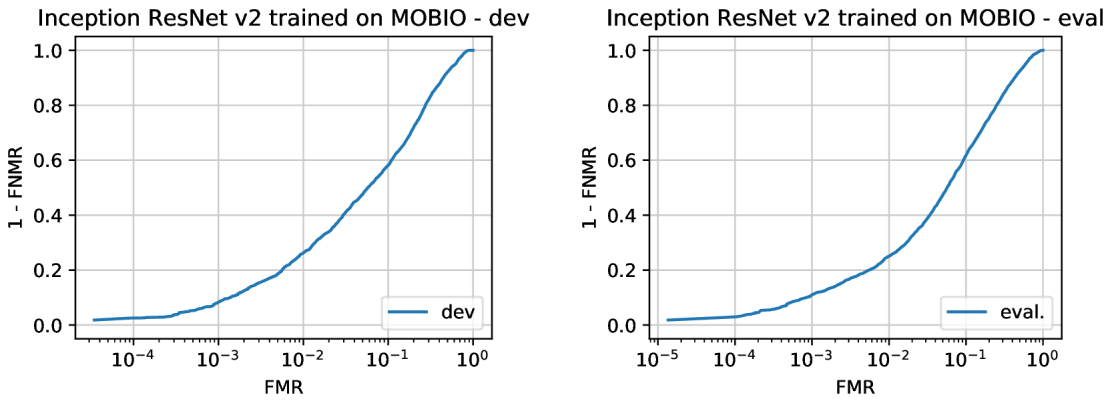


Figure 3.5 – ROC curve of Inception ResNet v2 trained on the MOBIO dataset.

	Development	Evaluation
False Match Rate	24.0% (13892/57960)	22.5% (33170/147630)
False Non Match Rate	24.0% (604/2520)	22.3% (889/3990)
Half Total Error Rate	24.0%	22.4%

Table 3.1 – Metrics of Inception ResNet v2 trained on MOBIO. The threshold value was set for an EER on the development set.

3.2 Training

Inception ResNet v2 is a CNN architecture oriented toward image classification. In order to be able to use it, it needs to be trained for the task we want. In order to recognize identity through faces, it needs to be trained on a face images dataset labeled with identity, so facial attributes can be learned. By feeding the network with labeled faces (with an identity number or a name), the network can distinguish the parts of a face that define the identity of a person and the parts that have to be ignored.

3.2.1 Using the MOBIO train set

First, the MOBIO dataset was used for training. However, since this database contains 32'000 images of 160 subjects, the network started to overfit on this data and performed poorly.

Figure 3.5 and Table 3.1 show the poor performance of the trained dataset on the MOBIO development and evaluation sets.
To circumvent this, the MS-Celeb (Guo et al. [2016]) database was used. Consisting of over 10'000'000 images of 100'000 individuals, a system training on it is less likely to overfit due to a lack of data. After training the dataset on MS-Celeb, the results shown in Figure 3.6 and Table 3.2 were obtained.

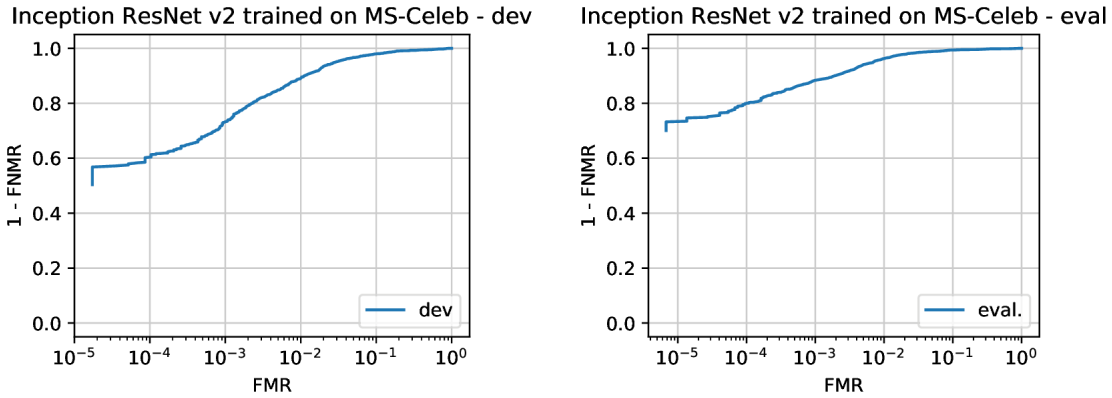


Figure 3.6 – ROC curve of Inception ResNet v2 trained on the MS-Celeb dataset.

	Development	Evaluation
False Match Rate	4.0% (2300/57960)	1.7% (2468/147630)
False Non Match Rate	4.0% (100/2520)	2.4% (96/3990)
Half Total Error Rate	4.0%	2.0%

Table 3.2 – Metrics of Inception ResNet v2 trained on MS-Celeb. The threshold value was set for an EER on the development set.

More training iterations would have been required to reach greater performance. A model previously trained on MS-Celeb at Idiap reaches the results in Figure 3.7 and Table 3.3.

	Development	Evaluation
False Match Rate	0.4% (207/57960)	0.1% (152/147630)
False Non Match Rate	0.4% (9/2520)	0.6% (23/3990)
Half Total Error Rate	0.4%	0.3%

Table 3.3 – Metrics of Inception ResNet v2 trained for a longer time on MS-Celeb. The threshold value was set for an EER on the development set.

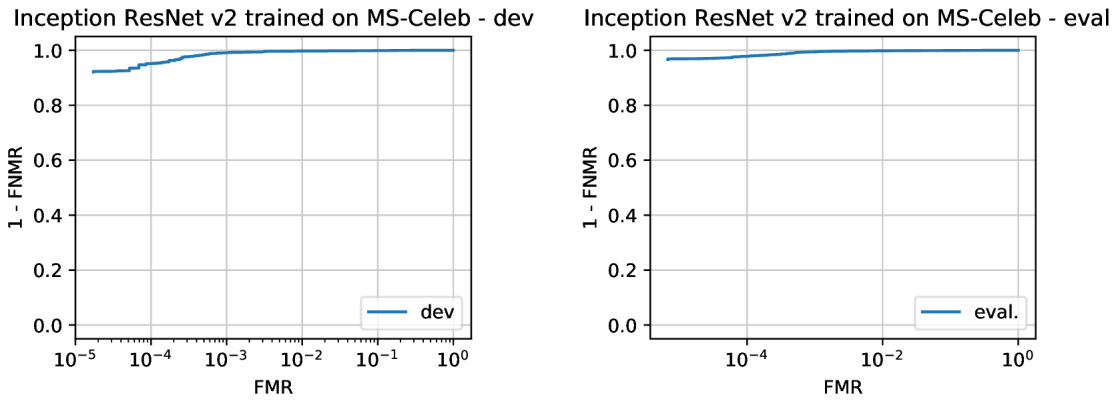


Figure 3.7 – ROC curve of Inception ResNet v2 trained for a longer time on the MS-Celeb dataset.

4 Bias in face recognition systems

In this chapter, the racial bias of different face recognition systems will be analysed, taking care to confirm that the bias is not induced by the evaluation, and ways to fix this bias will be presented.

4.1 Bias

A bias in a Face Recognition system is a performance imbalance between different groups of subjects. These groups can be defined based on various parameters, the most common ones being gender, age, or race.

A bias is present because we only set one threshold value for the whole population, despite the fact that the system gives different distributions for different cohorts (groups of people with a common trait.)

For example, in the Figure 4.1, synthetic scores of a biased system are presented. For each demographic cohort, the Genuines and Impostors scores distributions are presented. We can see that when a single threshold value is set, the system will not perform the same on each demographic.

Looking at the Genuines distributions (blue), if the threshold is raised, the *demographic 1* will have a False Non-Match Rate greater than the other two. This can be seen as the box-plot is shifted compared to the others. This means that a Genuine subjects from *demographic 1* has more chance to be rejected than a subject from *demographic 0* or *demographic 2*.
For the Impostors distributions (red), the system recognizes better Impostors for *demographic 0*, then *demographic 1*, and has more trouble with *demographic 2*. If we lower the threshold value, *demographic 2* will see more Impostors accepted as Genuines than in other cohorts.

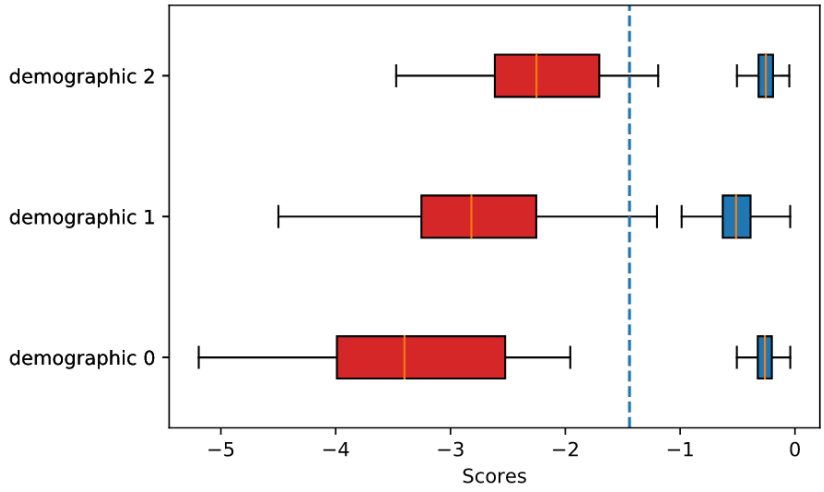


Figure 4.1 – Representation of a biased system's scores distributions (synthetic data). This system is rejecting more Genuines subjects in *demographic 1* than the other cohorts, and accepting more Impostors from *demographic 2*.

4.2 Race

On the problem of estimating a bias, there is also the definition of what kind of bias we want to evaluate. Here we focus on the difference between geographical origin of people.

Two terms exists that define the classes of people: *ethnicity* and *race*. However, those are not clearly defined, and are subject to discussion. If *ethnicity* sounds better and less pejorative than *race*, the term is not exact for our usage. Where *race* is a heritable trait, like the color of the skin or the shape of the face, *ethnicity* is a learnable trait, like religion or a spoken accent. A Face Recognition system will use mainly the physical traits of a person to identify them, and thus, the racial features have more importance in this case. So the term *race* will be used.

4.3 Evaluation of bias

Multiple systems were selected to be evaluated for bias. Focusing mainly on CNN architectures, but also including another non-trained method, based on Gabor-Graph.

An evaluation benchmark was put in place using the new bob pipelines structure[1], benefiting greatly from the computation grid available. This new architecture of bob using Dask[2] allows to run these experiments without a grid setup or on a different computation grid with minimal configuration changes.

4.3.1 Evaluation Datasets

To ensure that the bias really comes from the system and not from the used evaluation dataset, three datasets were chosen to do the racial bias estimation.

Not all datasets have annotations for every type of bias. Generally, one dataset built specifically to analyse a type of bias will lack information in other types. For example, RFW that is built toward racial equalization does not include age annotations, and no efforts were done to distribute equally other covariates than ethnicity.

[1] https://gitlab.idiap.ch/bob/bob.pipelines
[2] https://dask.org/

The RFW (Racial Faces in the Wild) dataset

The RFW dataset (Wang et al. [2018]) consisting of face pictures is build for racial bias analysis. It identifies four different classes: African, Asian, Caucasian and Indian.

For each class, 10'000 images from approximately 3'000 subjects were extracted from the MS-Celeb dataset in a *test* set, allowing the evaluation of racial bias of a system, by inputting each class and analyzing the differences between the results.

Figure 4.2 – Samples of the RFW dataset with 4 instances of one subject from each classes.

MEDS (Multiple Encounter DataSet)

MEDS (or MEDS-II, Founds et al. [2011]) is a test database organized from pictures of now deceased people with prior multiple encounters. It contains over 1'300 pictures from 500 people. The labels contain the age of the person, and its age difference at each encounter. A racial classification is also provided (*White*, *Black* or *Other*). Face landmarks coordinates and pose information (frontal or profile) are also provided.

Figure 4.3 – Samples of the MEDS dataset with 3 instances per subject.

The MORPH database

The MORPH database (Ricanek and Tesafaye [2006]) focuses on longitudinal data. Similarly to MEDS, it consists of face images of people taken over time from a few months, up to twenty years apart. It contains over 400'000 images for 67'600 subjects.

Figure 4.4 – Samples of the MORPH dataset with 3 instances of 4 subjects.

Verification protocols

For this experiment, and for each dataset, the subjects were distributed like so:

- Any subject represented by only one image is part of the *train* set, not used in our experiment;
- Half of the subjects with multiple images (selected randomly) is part of the *development* set;
- The remaining half of subjects with multiple images is part of the *evaluation* set;

For the *development* and *evaluation* sets, one image of each subject is used as an enrollment reference, and the rest of the images is used as probes.

To allow more control and more confident evaluations, three folds were established, with the development set subjects being selected randomly with a different seed for each fold. Those were named *verification_fold1*, *verification_fold2* and *verification_fold3*.

Those definitions were implemented in three Bob packages.

4.3.2 Evaluated Systems

Three face recognition methods were evaluated for racial bias:

- A CNN with an Inception ResNet v2 architecture trained on the MS-Celeb dataset,
- A CNN with the Facenet architecture,
- A system based on Gabor-graph (not dependent on training data).

Each system was evaluated on the three folds of every dataset.

4.4 Results

Figure 4.5 shows the score distributions obtained by applying our Inception ResNet v2 model on one fold of MEDS, and separating the scores by ethnicity (Black and White). The box plots represent the distributions of scores for Genuines (blue) and Impostors (red) for each Black on Black and White on White comparisons. Since no Genuines distribution exists for White probes compared with Black references, or Black probes with White references, these cohorts are not shown.

In this plot, we can observe two kinds of bias. The first one is that for a high threshold value, the White subjects will always have a higher portion of the Genuines considered as Impostors than the Black subjects. Meaning that a Genuine White subject will encounter more errors than a Genuine Black subject. And for a lower threshold value, Black Impostors will be accepted as Genuines more often than White Impostors.

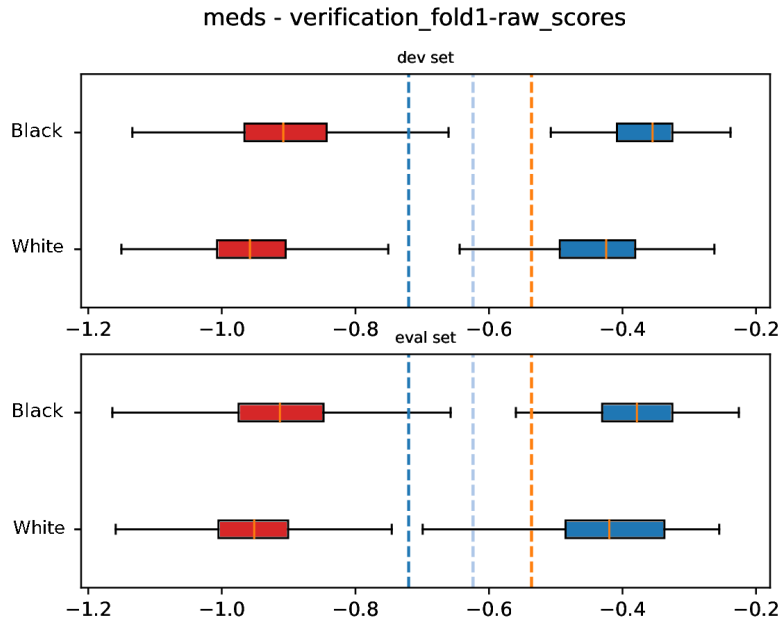


Figure 4.5 – Results of bias evaluation on MEDS.

We also obtain the metrics shown this in Table 4.1:

In the FMR table (a), for three threshold values giving a chosen FMR on the development set, the FMR on the development and evaluation sets of each possible cohort combination of probe and reference is shown. If we consider one threshold value (especially for lower values), we can see that there is a difference between white and black subjects.

In the FNMR table (b), for the three same threshold values (resulting in a given FMR on the development set), the FNMR on the development and evaluation sets of each possible cohort combination of probe and reference is shown. Note that it is impossible to have a true match between subjects of different cohorts (no White subject matches with a Black cohort subject), so there is no False Non-Matches between cohorts. A bigger difference can be noticed between Black and White cohorts than with the FMRs, for each threshold value. So this system is more biased for Genuines than Impostors.

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0071	0.0074	0.0013	0.0005	0.0	0.0
White	Black	0.0	0.0005	0.0	0.0	0.0	0.0
Black	White	0.0007	0.0004	0.0	0.0	0.0	0.0
Black	Black	0.0324	0.0287	0.0026	0.0038	0.0003	0.0

(a) FMR

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0364	0.0	0.0545	0.0976	0.1455	0.1463
Black	Black	0.0	0.0	0.0179	0.0	0.0179	0.0263

(b) FNMR

Table 4.1 – FMR (a) and FNMR (b) for three threshold values and two races measured on MEDS.

4.5 Score normalization

In order to reduce the bias impact on the results, a score normalization can be implemented as post-processing. Multiple variations of normalization exists on top of the Z-normalization and T-normalization. Most of these variation are a different combination of those. Below are presented the ones used in this evaluation.

In every case, a set of subjects (normalization cohort, ϵ) is used to represent the population. This cohort should be selected to best represent the subjects distributions regarding the bias we want to attenuate.

Given e the enrolled reference features, and p the probe features, a score is computed using the following representation:

$$s(e,p) \tag{4.1}$$

4.5.1 Z-norm

For the Z-normalization, the cohort ϵ containing N subjects is used to compute the normalization scores $S_{\text{Z-norm}}$:

$$S_{\text{Z-norm}} = \left\{ s(e,\epsilon_i) \right\}_{i=0}^{N} \tag{4.2}$$

These scores are then used to set the mean of of the final distribution to 0, using the mean of the z-norm scores $\mu(S_{\text{Z-norm}})$, and the standard deviation to 1 using the z-norm scores standard deviation $\sigma(S_{\text{Z-norm}})$:

$$s(e,p)_{\text{Z-norm}} = \frac{s(e,p) - \mu(S_{\text{Z-norm}})}{\sigma(S_{\text{Z-norm}})} \tag{4.3}$$

This resulted in the distributions in Figure 4.7 and the metrics in Table 4.2.

An improvement can be observed compared to the raw scores, as the distributions are more aligned. In the Table 4.2 we can see that the system is less biased as the FMR are closer between classes. There is still a big gap between the classes when looking at the FNMR.

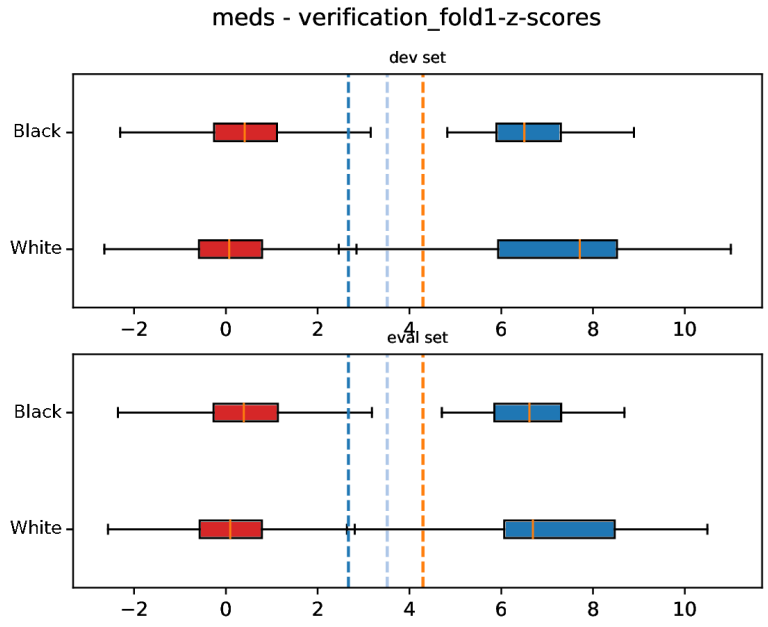


Figure 4.6 – Results of bias evaluation on MEDS normalized with Z-norm.

		dev FMR = 0.01		**dev FMR = 0.001**		**dev FMR = 0.0001**	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0168	0.0156	0.0017	0.0037	0.0003	0.0014
White	Black	0.001	0.0055	0.0	0.0	0.0	0.0
Black	White	0.0013	0.0004	0.0	0.0	0.0	0.0
Black	Black	0.0212	0.0249	0.0023	0.0047	0.0	0.0

(a) FMR

		dev FMR = 0.01		**dev FMR = 0.001**		**dev FMR = 0.0001**	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0182	0.0244	0.0364	0.0244	0.0727	0.0976
Black	Black	0.0179	0.0	0.0179	0.0	0.0	0.0

(b) FNMR

Table 4.2 – FMR (a) and FNMR (b) for MEDS normalized with Z-norm, for various threshold defined for a given FMR on the dev set.

4.5.2 T-norm

The T-normalization is similar to Z-normalization, at the exception that the Impostors scores are normalized. The cohort $S_{\text{T-norm}}$ is defined like this:

$$S_{\text{T-norm}} = \{s(p, \epsilon_i)\}_{i=0}^{N} \tag{4.4}$$

And the T-normalized scores are computed as:

$$s(e,p)_{\text{T-norm}} = \frac{s(e,p) - \mu(S_{\text{T-norm}})}{\sigma(S_{\text{T-norm}})} \tag{4.5}$$

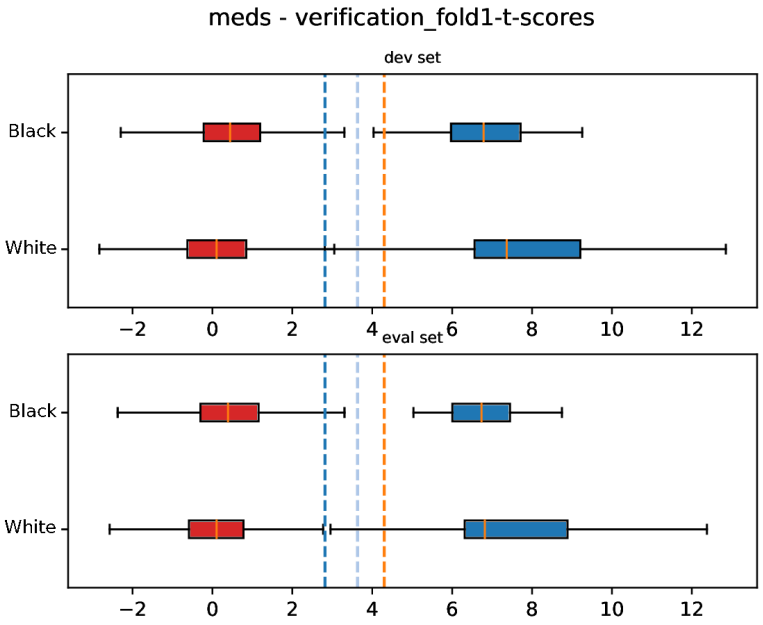


Figure 4.7 – Results of bias evaluation on MEDS normalized with T-norm.

T-normalization is the best performing for this data, as can be seen in the Table 4.3. The FNMR are similar for both classes, and the system performance is not impacted as much as with other normalization types. As long as we keep the threshold value in a reasonable range, the system can be considered fair. However, the bias still exists if the threshold value is set very high or very low.

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0162	0.0166	0.001	0.006	0.0003	0.0014
White	Black	0.0003	0.0	0.0	0.0	0.0	0.0
Black	White	0.0017	0.0004	0.0	0.0	0.0	0.0
Black	Black	0.0221	0.0244	0.003	0.0061	0.0	0.0009

(a) FMR

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0545	0.0	0.0545	0.0488	0.0545	0.0488
Black	Black	0.0	0.0	0.0	0.0	0.0179	0.0

(b) FNMR

Table 4.3 – FMR (a) and FNMR (b) for MEDS normalized with T-norm, for various threshold defined for a given FMR on the dev set.

4.5.3 ZT-norm

ZT-normalization is the combination of the two previous normalization methods in series. Two different cohorts were used to compute each normalization.

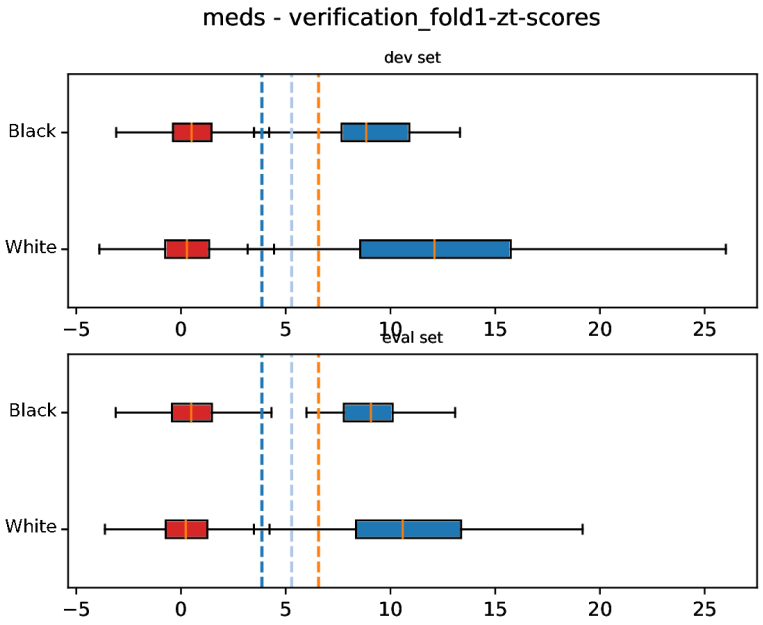


Figure 4.8 – Results of bias evaluation on MEDS normalized with ZT-norm.

We observed that ZT-normalization is not very conclusive for this application. It is impairing the results of the system without really removing the bias.

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0256	0.0202	0.0037	0.0078	0.0003	0.0023
White	Black	0.0006	0.0005	0.0	0.0	0.0	0.0
Black	White	0.001	0.0009	0.0	0.0	0.0	0.0
Black	Black	0.0132	0.0169	0.0003	0.0023	0.0	0.0

(a) FMR

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0364	0.0244	0.0545	0.0488	0.0909	0.0732
Black	Black	0.0179	0.0	0.0179	0.0	0.0536	0.0526

(b) FNMR

Table 4.4 – FMR (a) and FNMR (b) for MEDS normalized with ZT-norm, for various threshold defined for a given FMR on the dev set.

4.5.4 S-norm

S-normalization (symmetric-normalization) computes the average of the normalized scores from Z- and T-normalization:

$$S(e,p)_{\text{S-norm}} = 0.5 * \left(s(e,p)_{\text{Z-norm}} + s(e,p)_{\text{T-norm}}\right) \tag{4.6}$$

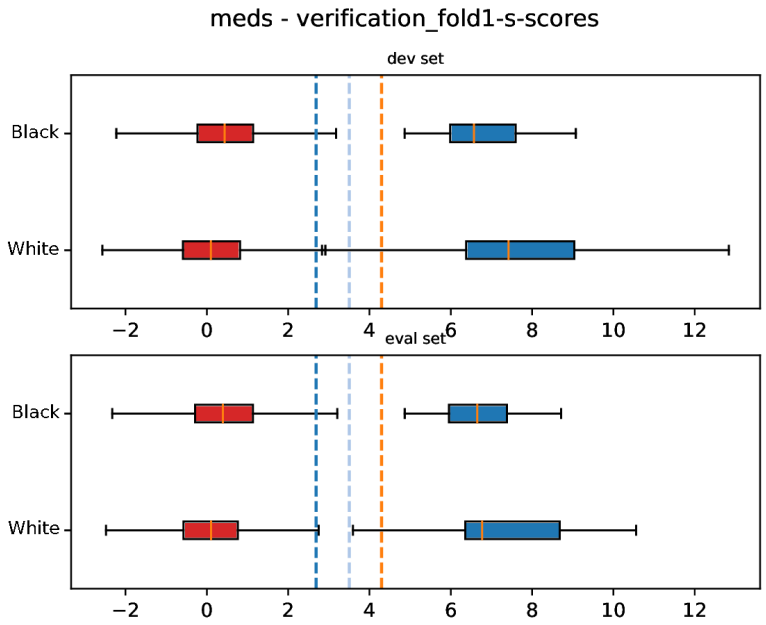


Figure 4.9 – Results of bias evaluation on MEDS normalized with S-norm.

S-normalization is not as bad as ZT-normalization, but is not performing better than T-normalization.

4.5.5 Normalization choice

Over all the datasets and systems, the T-normalization is the one that shows the most effect against the racial bias, without impairing the system recognition performance.

Normalizing the scores distributions did not remove the bias completely on the whole range of possible threshold values. However, for a set of threshold values, we saw that compensating for a bias is possible only by normalization of the scores, and without training the model from scratch with a balanced dataset.

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0152	0.0152	0.0007	0.0041	0.0003	0.0014
White	Black	0.0003	0.0005	0.0	0.0	0.0	0.0
Black	White	0.001	0.0004	0.0	0.0	0.0	0.0
Black	Black	0.0238	0.0254	0.0033	0.0052	0.0	0.0

(a) FNMR

		dev FMR = 0.01		dev FMR = 0.001		dev FMR = 0.0001	
reference	**probe**	**dev**	**eval**	**dev**	**eval**	**dev**	**eval**
White	White	0.0182	0.0	0.0545	0.0244	0.0545	0.0732
Black	Black	0.0	0.0	0.0179	0.0	0.0179	0.0

(b) FMR

Table 4.5 – FMR (a) and FNMR (b) for MEDS normalized with S-norm, for various threshold defined for a given FMR on the dev set.

5 Conclusion

In this dissertation, I first demonstrated face recognition system evaluation and presented the results of different evaluations on three datasets. I demonstrated that the MOBIO dataset is challenging and suited for evaluating face recognition systems. After evaluation of various systems on the dataset, the best performing systems were the ones based on Deep Convolutional Neural Network.

I presented the Inception ResNet v2 architecture and the results of my attempt to train it with the MOBIO dataset, and the more successful training on MS-Celeb.

I then used the bias analysis tool that we developed to estimate the racial bias of diverse face recognition systems. The score normalization was approached to attempt to reduce the impact of the bias. The T-normalization appears to be the best choice to reduce the impact of racial bias on the results in these cases, without impairing the recognition performance of the system. This shows that it is possible to correct a bias in a recognition system by normalizing the scores, and without training a model from scratch.

During this master project, I could get an in-depth knowledge of how facial recognition systems and more generally biometric systems work. I got a sense of the impact of bias in such systems.

5.1 Contributions

During this project, I implemented multiple databases interfaces (MEDS, MORPH, RFW) for the Bob toolbox and the new pipelines interface.

I adapted Bob packages for the new version of the ROC SDK from RankOne.

I also ensured that colleagues research experiments were reproducible by replicating the results they produced.

5.2 Future work

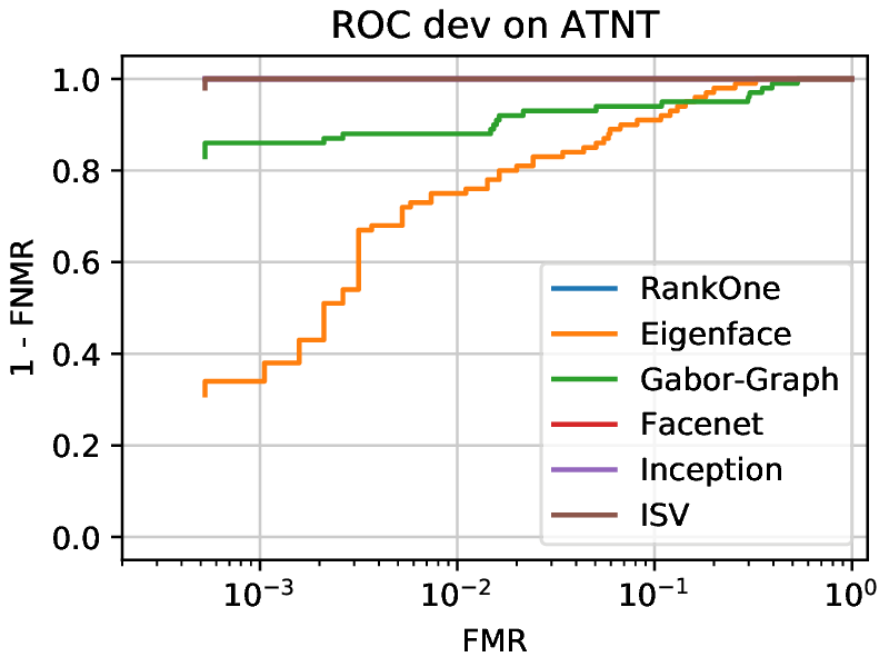
As future work, I can suggest to:

- Investigate the presence of other types of bias (age, gender) on diverse face recognition systems, since the evaluation pipelines are in place. It will not require a lot of efforts to evaluate on other covariates.
- Research the presence of bias in other biometric systems.
- Establish a metric to measure the bias of a system, allowing the comparison of bias between different systems.

A Results of FR systems evaluation

A.1 Evaluation on AT&T

A.1.1 ROC curves



A.1.2 Metrics

Eigenface

[Criterion: EER] Threshold on ‘Eigenface/scores‘: -3.358931e+03

	Full set
Failure to Acquire	0.00%
False Match Rate	9.00% (171/1900)
False Non Match Rate	9.00% (9/100)
False Accept Rate	9.00%
False Reject Rate	9.00%
Half Total Error Rate	9.00%

Facenet

[Criterion: EER] Threshold on ‘Facenet/scores‘: -4.071383e-01

	Full set
Failure to Acquire	0.00%
False Match Rate	0.00% (0/1900)
False Non Match Rate	0.00% (0/100)
False Accept Rate	0.00%
False Reject Rate	0.00%
Half Total Error Rate	0.00%

Gabor Graph

[Criterion: EER] Threshold on ‘Gabor-Graph/scores‘: 5.387814e-01

	Full set
Failure to Acquire	0.00%
False Match Rate	6.00% (114/1900)
False Non Match Rate	6.00% (6/100)
False Accept Rate	6.00%
False Reject Rate	6.00%
Half Total Error Rate	6.00%

Inception ResNet v2

[Criterion: EER] Threshold on 'Inception/scores': -1.886094e-01

	Full set
Failure to Acquire	0.00%
False Match Rate	0.00% (0/1900)
False Non Match Rate	0.00% (0/100)
False Accept Rate	0.00%
False Reject Rate	0.00%
Half Total Error Rate	0.00%

ISV

[Criterion: EER] Threshold on 'ISV/scores': 3.703242e-01

	Full set
Failure to Acquire	0.00%
False Match Rate	0.05% (1/1900)
False Non Match Rate	0.00% (0/100)
False Accept Rate	0.05%
False Reject Rate	0.00%
Half Total Error Rate	0.03%

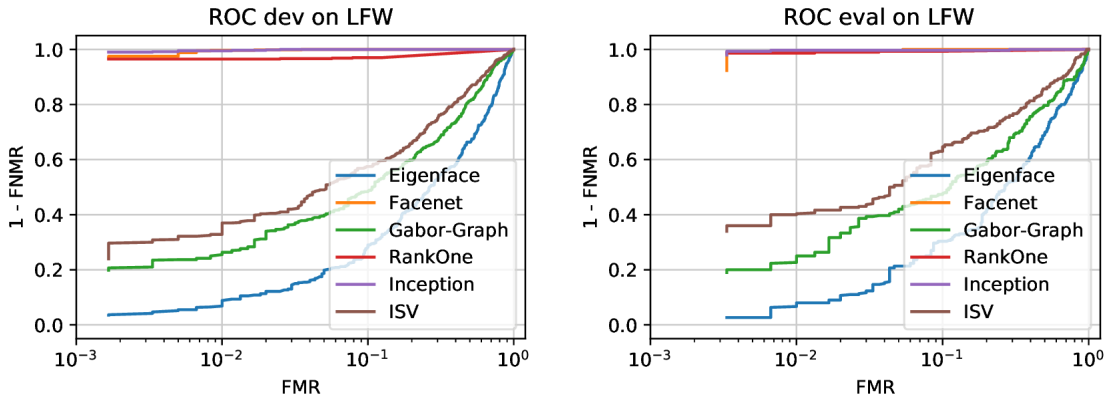
RankOne

[Criterion: EER] Threshold on 'RankOne/scores': 7.912091e-01

	Full set
Failure to Acquire	0.00%
False Match Rate	0.00% (0/1900)
False Non Match Rate	0.00% (0/100)
False Accept Rate	0.00%
False Reject Rate	0.00%
Half Total Error Rate	0.00%

A.2 Evaluation on LFW

A.2.1 ROC curves



A.2.2 Metrics

Eigenface

[Criterion: EER] Threshold on Development set 'Eigenface/scores-dev': $-1.535123e+03$

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	40.83% (245/600)	41.00% (123/300)
False Non Match Rate	40.83% (245/600)	37.33% (112/300)
False Accept Rate	40.83%	41.00%
False Reject Rate	40.83%	37.33%
Half Total Error Rate	40.83%	39.17%

Facenet

[Criterion: EER] Threshold on Development set 'Facenet/scores-dev': $-5.718787e-01$

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	0.67% (4/600)	0.33% (1/300)
False Non Match Rate	0.67% (4/600)	1.67% (5/300)
False Accept Rate	0.67%	0.33%
False Reject Rate	0.67%	1.67%
Half Total Error Rate	0.67%	1.00%

Gabor Graph

[Criterion: EER] Threshold on Development set 'Gabor-Graph/scores-dev': $4.665343e-01$

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	31.50% (189/600)	33.33% (100/300)
False Non Match Rate	31.50% (189/600)	30.33% (91/300)
False Accept Rate	31.50%	33.33%
False Reject Rate	31.50%	30.33%
Half Total Error Rate	31.50%	31.83%

Inception ResNet v2

[Criterion: EER] Threshold on Development set 'Inception/scores-dev': $-2.911757e-01$

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	0.50% (3/600)	0.33% (1/300)
False Non Match Rate	0.50% (3/600)	0.67% (2/300)
False Accept Rate	0.50%	0.33%
False Reject Rate	0.50%	0.67%
Half Total Error Rate	0.50%	0.50%

ISV

[Criterion: EER] Threshold on Development set 'ISV/scores-dev': 3.230776e-02

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	27.00% (162/600)	24.67% (74/300)
False Non Match Rate	27.00% (162/600)	27.00% (81/300)
False Accept Rate	27.00%	24.67%
False Reject Rate	27.00%	27.00%
Half Total Error Rate	27.00%	25.83%

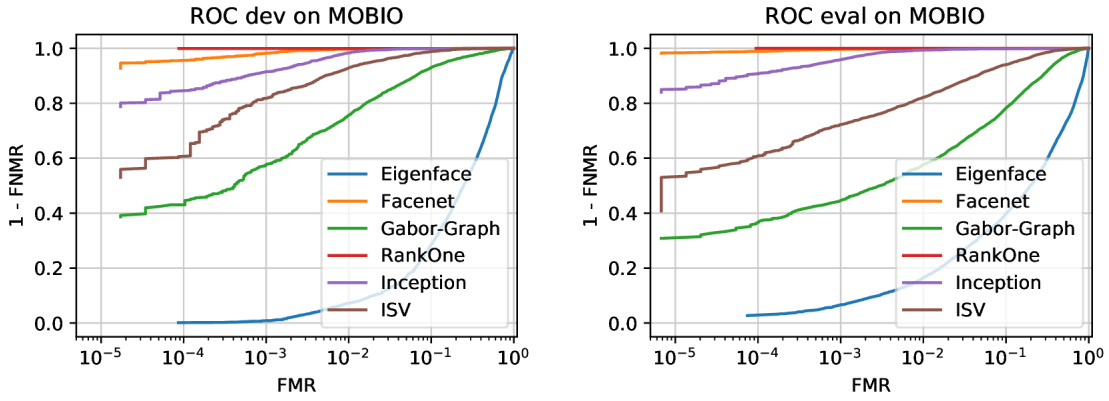
RankOne

[Criterion: EER] Threshold on Development set 'RankOne/scores-dev': $1.505210e-01$

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	3.33% (20/600)	3.67% (11/300)
False Non Match Rate	3.33% (20/600)	0.67% (2/300)
False Accept Rate	3.33%	3.67%
False Reject Rate	3.33%	0.67%
Half Total Error Rate	3.33%	2.17%

A.3 Evaluation on MOBIO

A.3.1 ROC curves



A.3.2 Metrics

Eigenface

[Criterion: EER] Threshold on Development set 'Eigenface/scores-dev': -3.622807e+03

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	37.86% (21942/57960)	40.75% (60159/147630)
False Non Match Rate	37.86% (954/2520)	33.76% (1347/3990)
False Accept Rate	37.86%	40.75%
False Reject Rate	37.86%	33.76%
Half Total Error Rate	37.86%	37.25%

Facenet

[Criterion: EER] Threshold on Development set 'Facenet/scores-dev': -4.221386e-01

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	0.56% (322/57960)	0.14% (207/147630)
False Non Match Rate	0.56% (14/2520)	0.30% (12/3990)
False Accept Rate	0.56%	0.14%
False Reject Rate	0.56%	0.30%
Half Total Error Rate	0.56%	0.22%

Gabor Graph

[Criterion: EER] Threshold on Development set 'Gabor-Graph/scores-dev': 5.568844e-01

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	8.21% (4761/57960)	12.28% (18134/147630)
False Non Match Rate	8.21% (207/2520)	19.45% (776/3990)
False Accept Rate	8.21%	12.28%
False Reject Rate	8.21%	19.45%
Half Total Error Rate	8.21%	15.87%

Inception ResNet v2

[Criterion: EER] Threshold on Development set 'Inception/scores-dev': -4.377619e-01

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	1.23% (713/57960)	1.15% (1702/147630)
False Non Match Rate	1.23% (31/2520)	0.63% (25/3990)
False Accept Rate	1.23%	1.15%
False Reject Rate	1.23%	0.63%
Half Total Error Rate	1.23%	0.89%

ISV

[Criterion: EER] Threshold on Development set 'ISV/scores-dev': 2.777790e-01

	Development	Evaluation
Failure to Acquire	0.00%	0.00%
False Match Rate	3.44% (1994/57960)	4.21% (6216/147630)
False Non Match Rate	3.45% (87/2520)	10.25% (409/3990)
False Accept Rate	3.44%	4.21%
False Reject Rate	3.45%	10.25%
Half Total Error Rate	3.45%	7.23%

RankOne

[Criterion: EER] Threshold on Development set 'RankOne/scores-dev': 6.855241e-01

	Development	Evaluation
Failure to Acquire	1.47%	0.10%
False Match Rate	0.08% (46/57109)	0.03% (46/147482)
False Non Match Rate	0.08% (2/2483)	0.03% (1/3986)
False Accept Rate	0.08%	0.03%
False Reject Rate	1.55%	0.13%
Half Total Error Rate	0.08%	0.03%

Bibliography

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

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

Founds, A. P., Orlans, N., Genevieve, W., and Watson, C. I. (2011). Nist special databse 32-multiple encounter dataset ii (meds-ii). Technical report.

Günther, M., Haufe, D., and Würtz, R. P. (2012). Face recognition with disparity corrected gabor phase differences. In *International Conference on Artificial Neural Networks*, pages 411–418. 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 *European Conference on Computer Vision*. Springer.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

McCool, C. and Marcel, S. (2009a). Mobio database for the icpr 2010 face and speech competition. Technical report, Idiap.

McCool, C. and Marcel, S. (2009b). Parts-based face verification using local frequency bands. In *International Conference on Biometrics*, pages 259–268. Springer.

Phillips, P. J. and O'toole, A. J. (2014). Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32(1):74–85.

Ricanek, K. and Tesafaye, T. (2006). Morph: A longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 341–345. IEEE.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

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

Wallace, R., McLaren, M., McCool, C., and Marcel, S. (2011). Inter-session variability modelling and joint factor analysis for face authentication. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE.

Wang, M., Deng, W., Hu, J., Peng, J., Tao, X., and Huang, Y. (2018). Racial faces in-the-wild: Reducing racial bias by deep unsupervised domain adaptation. *CoRR*, abs/1812.00194.

Zhang, W., Shan, S., Gao, W., Chen, X., and Zhang, H. (2005). Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 786–791. IEEE.