Learning How To Recognize Faces In Heterogeneous Environments

Thèse N° 9366

Présentée le 14 juin 2019

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Tiago DE FREITAS PEREIRA

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. H. Bourlard, Dr S. Marcel, directeurs de thèse
Prof. M. Nixon, rapporteur
Prof. J. Fierrez, rapporteur
Prof. J.-Ph. Thiran, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

You'll never walk alone
— Richard Rodgers

Acknowledgements

This Ph.D. thesis is a result of years of work and it was supported by a lot of people.

First, I would like to thank Dr. Sébastien Marcel for opening the doors of Idiap and let me carry on my doctoral studies here and, most import, for his supervision. I am also grateful to my thesis director Prof. Hervé Bourlard, as well as the other members of my jury Prof. Mark Nixon, Prof. Julian Fierrez, Prof. Pascal Frossard and Prof. Jean-Philippe Thiran, for doing me the honor to supervise my oral exam.

Work at Idiap and EPFL is smooth thanks to the awesome administrative support from Nadine, Sylvie, Vanessa and Valérie, and the technical support of Bastien, Cédric, Frank, Louis-Marie, Norbert, Laurent and Vincent.

I also would like to thank my office mates that helped my a lot in the course of this thesis: Laurent, Ivana, Elie, Manuel, Pavel, Amir, Sulshil, Hannah, Vedrana, Guillaume, Olegs, Michael, Anjith, Zohreh, Pranay, Cijo, Gulcan, Pierre-Edouard, Nikos, Phil and David. Thanks to André for the great insights in the course of this journey as well as Ana and David for making my stay here in Switzerland very fun.

Many thanks to the guys from Samsung Research America for the opportunity to carry on my internship in the Mountain View campus and get to know the Silicon Valley. Specially, I would like to thank Bobi, Phillip, Sergi and Abhijit.

I would like to thank Prof. Sandra Aluísio from University of São Paulo and Prof. José Mario De Martino from University of Campinas for opening the doors of the university for me and let me start to think about research.

I'm also very thankful to the guys from CPqD for the friendship and the opportunity to start to investigate what machine learning is. I had a lot of help and everyone was very generous in key moments of my life. Specially, I would like to thank Norberto, Claudinei, Eliana, Henrique, Baldin, Flavio, Ricardo, Mario, Marcus, Dudu, Diego, Bruno, Vanessa, Amanda, Robson, Nagle, Paula and Stuchi.

Many thanks to my friends from Switzerland for the hospitality, friendship and generosity. Specially, I would like to thank Tiziana, Eduardo, Jolanta, Anne, Michael, Aurélie, Yoan and Audrey.

Thanks to the team from L'hôpital du Valais for fixing my legs twice in the course of this thesis. Special thanks to Jonatan, Yanik and Dr. Cédric Perez.

I also would like to thank my friend Laura for the patience and support in this final sprint.

I'm also very thankful to my friends from Brazil that, apart from the distance, were very close in several instances of my life. Specially, I would like to thank Carolina, Bruno, Shimizu, Bruna, Bozoh, Camila, Marcus, Boulos, Moara, dona Lourdes, Scarton, Daniel and Larissa.

Special thanks must go to my parents Sueli and Tote and my brother Rodrigo and his girlfriend Thais for the love, support and to give me the freedom to pursue my dreams. I also would like to thank my grandfather Elias. His way of be is an example of strength and perseverance that I will take for life.

Martigny, 11 Decembre 2018 T.F. P.

Abstract

Face recognition is a mature field in biometrics in which several systems have been proposed over the last three decades. Such systems are extremely reliable under controlled recording conditions and it has been deployed in the field in critical tasks, such as in border control and in less critical ones, such as to unlock mobile phones. However, the lack of cooperation from the subject and variations on the pose, occlusion and illumination are still open problems and significantly affect error rates. Another challenge that arose recently in face recognition research is the ability of matching faces from different image domains. Use cases encompass the matching between Visual Light images (VIS) with Near infra-red images (NIR), Visual Light images (VIS) with Thermograms or Depth maps. This match can occur even in situations where no real face exists, such as matching using sketches. This task is so called **Heterogeneous Face Recognition**. The key difficulty in the comparison of faces in heterogeneous conditions is that images from the same subject may differ in appearance due to changes in image domain.

In this thesis we address this problem of Heterogeneous Face Recognition (HFR). Our contributions are four-fold. First, we analyze the applicability of crafted features used in face recognition in the HFR task. Second, still working with crafted features, we propose that the variability between two image domains can be suppressed with a linear shift in the Gaussian Mixture Model (GMM) mean subspace. That encompasses inter-session variability (ISV) modeling. Third, we propose that high level features of Deep Convolutional Neural Networks trained on Visual Light images are potentially domain independent and can be used to encode faces sensed in different image domains. Fourth, large-scale experiments are conducted on several HFR databases, covering various image domains showing competitive performances. Moreover, the implementation of all the proposed techniques are integrated into a collaborative open source software library called Bob that enforces fair evaluations and encourages reproducible research.

Keywords: Face Recognition, Heterogeneous Face Recognition, Reproducible Research, Domain Adaptation, Gaussian Mixture Modeling, Deep Neural Networks

Résumé

La reconnaissance faciale est un domaine reconnu en biométrie, au sein duquel différents systèmes ont été proposés au cours des trois dernières décennies. De tels systèmes sont extrêmement fiables en conditions d'enregistrement contrôlées et ont été déployés sur le terrain, pour des tâches critiques telles que le contrôle aux frontières, et dans des cas moins critiques, par exemple pour déverrouiller des téléphones mobiles. Cependant, le manque de collaboration du sujet et les variations de la pose, l'occlusion et l'éclairage sont encore des problèmes ouverts qui affectent de manière significative les taux d'erreur. Un autre défi qui a surgi récemment au sein de la recherche en reconnaissance faciale est la capacité de faire correspondre des visages provenant de différents domaines d'image. Les cas d'utilisation englobent la correspondance entre les images Visual Light (VIS) avec les images infrarouge proches (NIR), les images Visual Light (VIS) avec les thermogrammes ou cartes de profondeur (depth maps). Cette correspondance peut se produire même dans des situations où il n'existe aucun visage réel, telle que la correspondance avec des croquis médico-légaux. Cette tâche est appelée Reconnaissance Faciale Hétérogène (HFR). La principale difficulté dans la comparaison de visages en conditions hétérogènes est que les images d'un même sujet puissent avoir une apparence différente en raison des changements de domaine d'image. Dans cette thèse, nous abordons ce problème de Reconnaissance Faciale Hétérogène (HFR). Nos contributions sont au nombre de quatre. Premièrement, nous analysons l'applicabilité des caractéristiques conçues en reconnaissance faciale pour la tâche de HFR. Deuxièmement, toujours en travaillant avec ces caractéristiques, nous proposons que la variabilité entre deux domaines d'image puisse être supprimée par un décalage linéaire dans l'espace formé par le centres d'un Gaussian Mixture Model (GMM) mais également par la modélisation de la variabilité inter-session (ISV). Troisièmement, nous proposons que les caractéristiques de haut niveau d'un Deep Convolutional Neural Network entrainées sur des images Visual Light soient potentiellement indépendantes du domaine et puissent être utilisées pour encoder des visages détectés dans un domaine d'image différent. Quatrièmement, des expériences à grande échelle sont menées sur plusieurs bases de données HFR, couvrant différents domaines d'image montrant des performances compétitives. De plus, toutes les techniques proposées sont intégrée dans une librairie logicielle collaborative opensource appelée Bob qui applique des évaluations non biasées et encourage une recherche reproductible.

Mots-clés : Reconnaissance faciale, Reconnaissance de visage, Reconnaissance Faciale Hétérogène, Recherche Reproductible, Adaptation de domaine, Gaussian Mixture Modeling, Deep Neural Networks

Contents

List of Figures

List of Tables

1 Introduction

Biometrics is the field that addresses the task of identifying human beings by their physical and/or behavioral attributes [Ross et al., 2008]. Along the history, several biometric attributes have being researched, such as face, fingerprint, signature, voice, periocular, gait, DNA, palm veins, hand geometry, iris, ear, among others. Some of them are largely used in the industry, such as fingerprint, face, iris or palm veins and some are still work in progress in research laboratories, such as gait, ear or signature.

Face biometrics, in particular, has existed as a field of research for more than 40 years and its research has been active since the early 1990s. Such biometric trait has some advantages over others. First, it is natural among humans; we do face recognition on a daily basis. Second, it is non intrusive; interaction with special devices is not necessarily a requirement. Finally, it is potentially a good candidate for covert applications.

The current state-of-the-art in automated face recognition consists of systems that work well under relatively constrained conditions. Despite the research efforts over the last years, automated face recognition under unconstrained conditions, where variations on the pose, occlusion, illumination and collaboration of the subjects are not under control, is still a challenge. Among those challenges, one of the most challenging ones is the task of comparison of face images acquired between different image modalities (infrared images, forensic sketches, or thermograms). This field of research is called **Heterogeneous Face Recognition** and their use-cases can increase the robustness of face recognition systems in to more covert scenarios, such as recognition at a distance or at nighttime, or even in situations where no real face exists (forensic sketch recognition).

This thesis is a step towards the development of more robust systems for Heterogeneous Face Recognition (HFR).

1.1 Background and Motivations

Due to the maturity of face recognition research, numerous applications have appeared in the last few years. In the list below we highlight some of them:

1. **Physical and Logical access control**: Face recognition has been widely deployed in border control in the so called *e-gates*. During the 2008 summer olympic games in Beijing, a face recognition system was deployed into the entrance security checks for the opening and closing ceremonies [Jain and Li, 2011]. For several years Lenovo[1] allows users to unlock their laptops using face recognition technology. The same trend was followed by Apple that recently allowed users to unlock and authorize some transactions in their phones using face recognition[2].

2. **Surveillance and Law enforcement**: The large amount of closed-circuit television (CCTV) systems deployed has led to a huge amount of information to be stored and processed. This is of particular interest in law enforcement, since face recognition technology can be employed to reduce the quantity of information to be processed manually, while criminal or terrorism investigations are performed. Several police departments around the world use software to compose sketches in eye witnesses cases, such as Evofit (https://evofit.co.uk/), Identikit (http://identikit.net/) and Faces (https://facialcomposites.com/) and the match of those composite sketches with large mugshot and legacy datasets raised the attention of the research community [Klare et al., 2011; Han et al., 2013].

3. **Data Management and entertainment**: Face identification has been widely used to automatically tag photos and/or video content. Companies such as Google, Microsoft, Facebook or Apple are already providing this feature in their image organizers and image viewer softwares to assist users in the task of organizing visual content and mitigate manual labor. Face identification is also applied in content personalization. For instance, game consoles such as XBox and PlayStation 4 allow users to log in to their online game platforms using face recognition.

The aforementioned applications can be reduced and formalized in three different tasks. (i) - The first one is called **verification**, in which a person claims a particular identity, and the system has to verify this claim given a biometric trait as input. The cardinality of this task is 1:1. (ii) - The second task is called **closed-set identification**, in which the system has to identify a person from a set N possibilities in a gallery given a biometric trait as input. The cardinality of this task is 1:N. (iii) - The third and the last one is called **open-set identification**, in which the system has to identify a person from a set N possibilities if and only if the comparison score between the input biometric trait and the set of N elements in the gallery is higher than

[1] https://www.lenovo.com

[2] https://support.apple.com/en-us/HT208109

a decision threshold τ. The cardinality of this task is also 1:N. These distinctions are depicted in Figure 1.1.

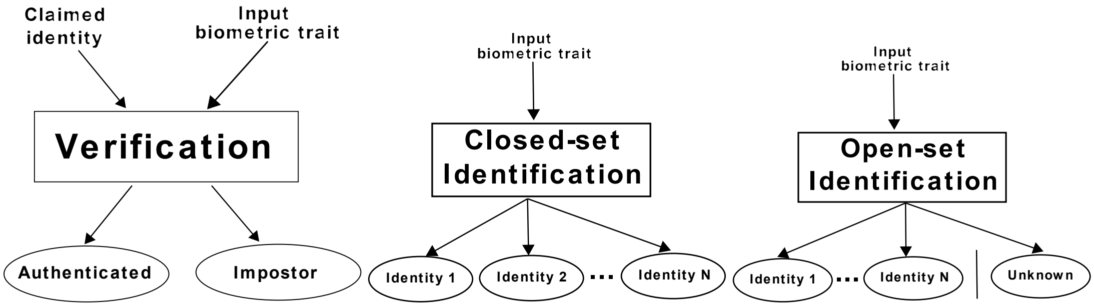


Figure 1.1 – Face recognition: Verification, Closed-set Identification and Open-set Identification tasks

The ability to recognize faces is a natural action performed by humans and make us think that is an easy task to be generalized and statically programmed. In reality, its complexity is so high and with so many degrees of freedom that, so far, we were not able to define a generalized theory that is able to differentiate two random face images in any condition. For this class of tasks, a new field of knowledge emerged as a mix of Computer Science and Statistics called Machine Learning [Samuel, 1959]. Machine learning is a branch of artificial intelligence that considers that a particular task/phenomena can be learnt and generalized from a reduced set of its observations, without being explicitly programmed.

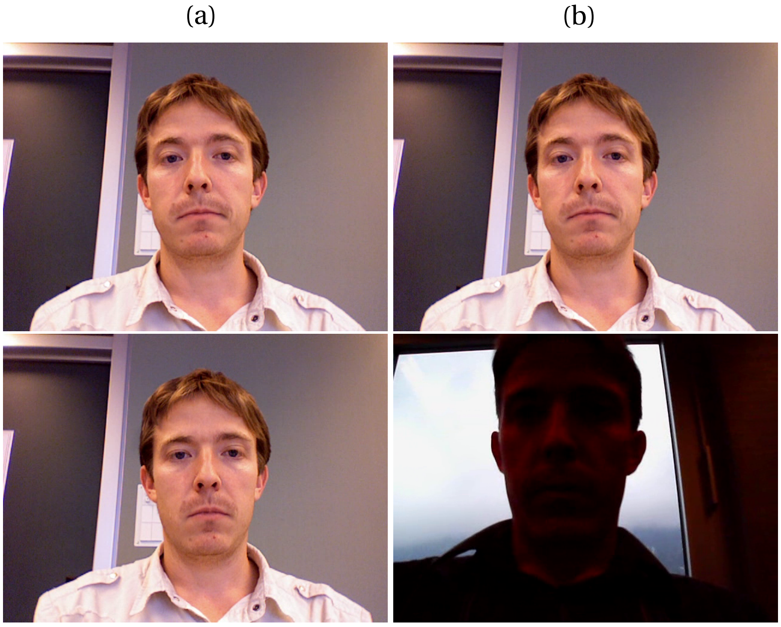


Figure 1.2 – Examples of (a) low within-class variability (b) high within-class variability

As mentioned before, automatic face recognition is practically considered a solved problem for constrained scenarios where variations in illumination, pose, expression and/or collaboration of the subject are not “severe”. Variations in appearance on face images from the same person, due to the mentioned factors, are called within-class variations. These variations can be as not as severe in the comparison between the images in Figure 1.2 (a) or can be very severe as in the comparison between the images in Figure 1.2 (b).

The task of HFR is considered challenging due to its high within-class variability between faces from the same subject but sensed in different image modalities. Example of these types of comparisons are shown in Figure 1.3.

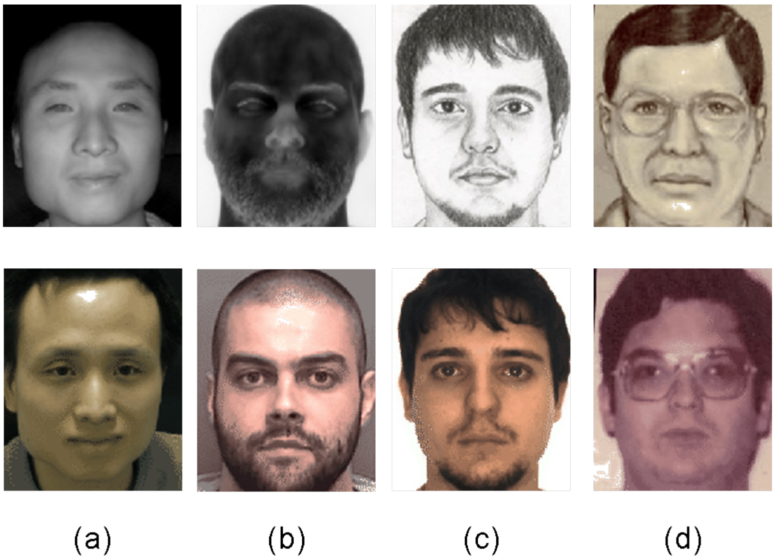


Figure 1.3 – Example images from four different heterogeneous face recognition scenarious (a) NIR (b) Thermal (c) Viewed sketch (d) Forensic sketch.

1.2 Objectives and Contributions

The main objective of this thesis is to investigate methods to handle this high within-class variability between faces sensed in different image modalitites and, in consequence, increase recognition rates.

The major contributions of this thesis are as follows.

1. **Domain Specific Units Framework (DSU)** is proposed. We hypothesize that high level features of Deep Convolutional Neural Networks trained on visual spectra images are potentially domain independent and can be used to encode faces sensed in different image domains. A generic framework for Heterogeneous Face Recognition is proposed by adapting Deep Convolutional Neural Networks low level features and/or their biases only. The adaptation using Domain Specific Units allow the learning of shallow feature detectors specific for each new image domain. Furthermore, it handles its transformation to a generic face space shared between all image domains. **Related papers for this**

contribution: [de Freitas Pereira et al., 2019] and http://vast.uccs.edu/Opensetface/.

2. **Investigation of the face recognition strategies to the HFR task**. We analyze and make public available the effectiveness of some state-of-the-art face recognition systems in the academia and commercial of the shelf (COTS) trained with visual light images only in the *HFR* task. **Related papers for this contribution:** [de Freitas Pereira et al., 2019].

3. **HFR as Gaussian Mixture Model session variability problem** is proposed. We hypothesize that the task of HFR can be approached with a linear shift in the Gaussian Mixture Model (GMM) mean subspace. Such domain shifts can be estimated with inter-session variability (ISV) modeling, joint factor analysis (JFA) and total variability (TV) modeling. **Related papers for this contribution:** [de Freitas Pereira and Marcel, 2015] [de Freitas Pereira and Marcel, 2016] [Sequeira et al., 2017].

4. We successfully **apply the proposed approaches** in several HFR databases covering six pairs of different image modalities and the results in terms of error rates are competitive with respect to the state of the art. Furthermore, this work is made reproducible in the following link [3]. Each one of the techniques applied in this thesis is part of the open source framework for signal processing and machine learning called Bob [4] following the reproducibility methodology defined in [Anjos et al., 2017]. In this methodology, it is emphasized that a reproducible research work should be **repeatable, shareable, extensible, and stable**. **Related papers for this contribution:** [Anjos et al., 2017].

1.3 Thesis Outline

This thesis is composed of 6 chapters.

In this chapter, the motivations, objectives and contributions of this work were briefly summarized.

Chapter 2 gives an overview of related work for the tasks of face and heterogeneous face recognition. In addition, this chapter introduces all the databases used in this work with its corresponding evaluation methodologies, which are used to compare the proposed systems in the experimentation chapters.

Chapter 3 presents how the state of the art face recognition systems developed in the academia and in the industry performs in the Heterogeneous Face Recognition task. Furthermore, a strategy base on Geodesic Flow Kernel using crafted features is introduced for HFR.

Chapter 4, the Gaussian Mixture Model framework for HFR is introduced. Consequently, the session variability modelling techniques that are built on top of this GMM are described for the HFR task. Moreover, experiments and analysis are presented.

[3] http://gitlab.idiap.ch/bob/bob.thesis.tiago

[4] https://www.idiap.ch/software/bob/

Chapter 5 introduces the Domain Specific Units (DSU) framework which is another technique to handle the HFR task. In this framework we hypothesize that high level features of Deep Convolutional Neural Networks trained on Visual Light images are potentially domain independent and can be used to encode faces sensed in different image domains. Moreover, experiments and analysis are presented.

Chapter 6 concludes this thesis by providing a summary of the major contributions and findings. Potential directions for future work are also discussed.

2 Related Work

In machine learning, the task of Face Recognition is phrased as a classification problem under the big umbrella of supervised learning [Bishop, 2006, p.3]. More generally, the classification task can be phrased as an interpolation problem in high-dimensional space. Such task can be described as follows: Given two random variables X and Y, where $X \in \mathbb{R}^d$ (high d-dimensional feature space) with marginal distribution $P(X)$ and a discrete set of labels $Y \in \mathbb{Z}$, the classification task consists in to find a model Θ where the probability of $P(Y|X,\Theta)$ is maximized. For the face recognition task, the variables X and Y are placeholder terms for a face dataset $X = \{x_1, x_2, ..., x_n\}$ and their corresponding set of labels $Y = \{y_1, y_2, ..., y_n\}$.

Along the years, several different strategies were proposed to solve this classification problem. Nevertheless, regardless the implementations, the approaches usually rely on three key components, which are depicted in Figure 2.1.

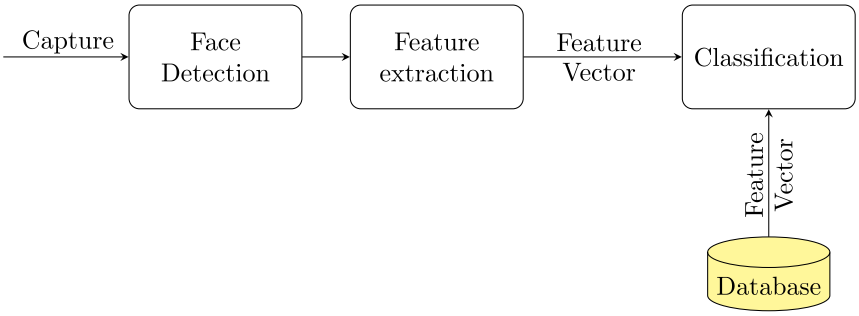


Figure 2.1 – Basic structure of a Face Recognition System

The first component is *Face Detection*. This step has a major impact on the performance of the entire face recognition system. Given either a single image or a video as input, an ideal face detector should be able to identify and locate all present faces regardless of their position, scale, orientation, expression and illumination conditions[Jain and Li, 2011].

The second component is *Feature Extraction*. Given a face image as input, an ideal feature extractor should be able to extract important information of the face which are both: (i) - **robust** against any kind of noise, such as, illumination effects, occlusion, pose variations, image blurring, etc; (ii) - high **discriminative** capability between face images from different identities. To rephrase this, an ideal feature extractor should be able to extract features that have low within-class variability and high between-class variability.

The third step is *Classification*, which is in charge of predict an identity given a feature vector.

In this chapter describes with more detail the efforts made in the literature to approach the second and the third aforementioned items for both Face (Section 2.1) and Heterogeneous Face Recognition (Section 2.2). Emphasizing HFR, in Section 2.3 it is described the databases available to work on the problem. Finally, in Section 2.4 the evaluation methodologies used for this task is introduced.

2.1 Face Recognition

Raw face images are often represented as high dimensional array of pixels of size m-by-n. Hence, face images can be seen as a vector embedded in a $\mathbb{R}^{m \times n}$ space. Due to well known significant statistical redundancies (correlations) that such images contains, it is common to represented them in lower dimension manifolds [Ruderman and Bialek, 1994]. In the last decades we have witnessed numerous scientific publications that explore this direction and applied algebraic, signal processing and statistical tools for extraction and analysis of the underlying manifold. In face analysis this manifold has a special name and it is called **face space** [Jain and Li, 2011].

In this section it is briefly described in roughly chronological order the approaches designed along the years to build this face space.

2.1.1 EigenFaces

Turk and Pentland [1991] proposed the first feature-based automatic face recognition system in the beginning 1990s based on Principal Component Analysis. Principal Component Analysis (PCA) is a dimensionality reduction technique that uses an orthogonal transformation to convert a set of correlated variables into a set of values of linearly uncorrelated variables called principal components. This basis transformation is built in such a way that the vector direction of the first principal component has the largest possible variance, the second principal component has the second largest possible variance and so on. This idea is illustrated in Figure 2.2 (a) where, in $\mathbb{R}^2$ space, the new basis is defined and in Figure 2.2 (b) the first component of this new basis is preserved rather than a second and it's used to do the projection in $\mathbb{R}^1$.

In short, PCA tries to create a projection matrix Θ where the L2 reconstruction (Equation 2.1) is minimized.

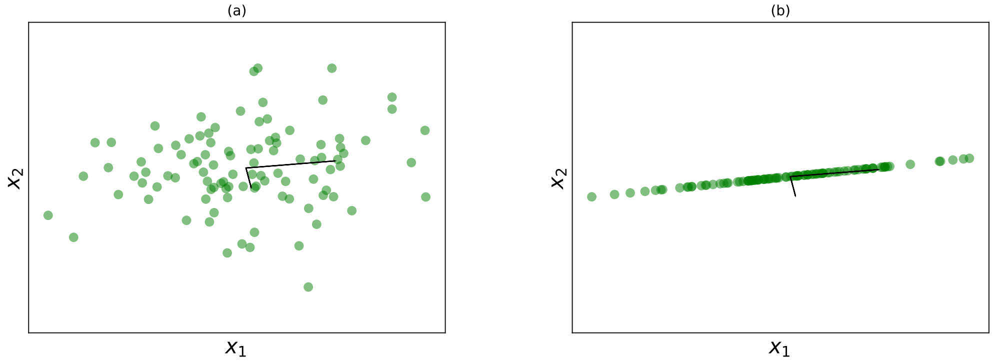


Figure 2.2 – Principal Component Analysis (a) Definition of the new basis (b) The projection in $\mathbb{R}^1$

$$\epsilon(x) = ||x - \sum_{i=0}^{k} (\Theta_i^T x)\Theta_i|| \tag{2.1}$$

There are several ways to achieve that. One of them is via the eigen decomposition of the covariance matrix. Given a set of samples $X = \{x_1, x_2, ..., x_n\}$ where $x \in \mathbb{R}^d$, this can be calculated following the steps below:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \text{ Mean of the dataset} \tag{2.2a}$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\top \text{ Compute the covariance} \tag{2.2b}$$

$$\text{Compute eigenvectors } \mathbf{U} = [\mathbf{u}_1 ... \mathbf{u}_d] \text{ of } \Sigma \text{ where} \tag{2.2c}$$

$$(\Sigma - e_j \mathbf{I})\mathbf{u}_j = 0, \tag{2.2d}$$

where e_j is the corresponding eigenvalues and $j = 1..d$.

Another way to compute this face space is via singular value decomposition (SVD) of X:

$$U, V = svd(X), \tag{2.3}$$

where the eigenvectors is given by U and the eigenvalues is given by $diag(V)$.

The Eigenfaces pipeline can be explained as the following. At **training time** (offline), this face space Θ is estimated given a face dataset $X = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^d$. At **enrollment time**, given one enrollment face image $x_e \in \mathbb{R}^d$, its projection is computed as $x'_e = \Theta^T x_e$. At **scoring time**, given one probe face image $x_p \in \mathbb{R}^d$ its projection is computed as $x'_p = \Theta^T x_p$. To compare x'_e and x'_p any distance measure can be used. Traditionally the L2 norm is employed, but other metrics are very popular too, such as the Mahalanobis distance or the cosine similarity.

2.1.2 Fisher Linear Discriminant; "Fisherfaces"

The face space Θ trained via Principal Component Analysis, although it uncorrelates the image input space, does not approach the desired requirements of low within-class and high between-class variability. In unconstrained scenarios, part of the variability in the face appearance is due to severe variations in pose, illumination, expression, etc; and the PCA face space, possibly retains most of these variations. Belhumeur et al. [1996] propose to solve this problem with an application of Fisher's linear discriminant (FLD)[Fisher, 1936]. Named as "Fisherfaces", FLD selects a Θ which maximizes the ratio:

$$\frac{\Theta^T S_b \Theta}{\Theta^T S_w \Theta} \tag{2.4}$$

where

$$S_b = \sum_{i=0}^{m} N_i (x_i - \mu)(x_i - \mu)^T \tag{2.5}$$

is the between scatter matrix, and

$$S_w = \sum_{i=0}^{m} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T \tag{2.6}$$

is the within scatter matrix.

This hypothesis explicitly finds a linear face space Θ where the within-class variability is minimized while the between class variability is maximized. Furthermore, it also performs dimensionality reduction.

Figure 2.3 shows how the illumination effects are retained using PCA and how it is suppressed using FLD.

The Fisherfaces pipeline can be explained as the following. At **training time** (offline), this face space Θ is estimated given a face dataset $X = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^d$. At **enrollment time**, given one enrollment face image $x_e \in \mathbb{R}^d$ its projection is computed as $x'_e = \Theta^T x_e$.

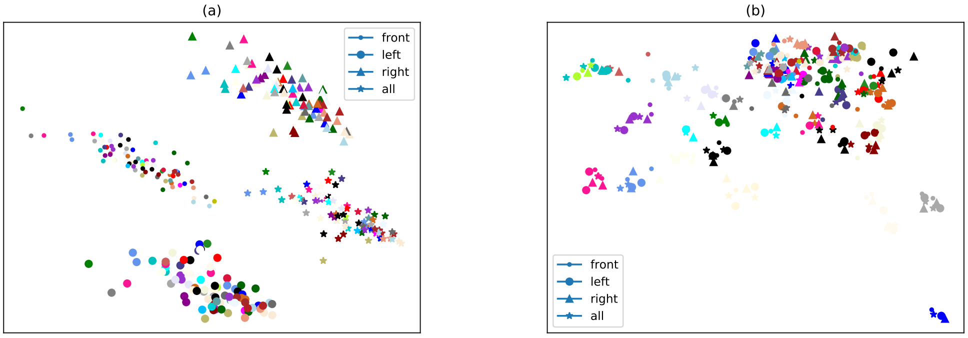


Figure 2.3 – First two principal components using PCA vs FLD under four different sources of illumination. Each color represents one of the 50 identities of the ARFACE database and each shape is one illumination condition (a) PCA face space (b) FLD face space

At **scoring time**, given one probe face image $x_p \in \mathbb{R}^d$ its projection is computed as $x'_p = \Theta^T x_p$. To compare x'_e and x'_p any distance measure can be used. Traditionally the L2 norm is used, but other metrics are very popular too, such as the Mahalanobis distance or the cosine similarity.

2.1.3 Local Binary Patterns histograms

The aforementioned sections presented strategies to model this **face space** using two different statistical hypotheses on top of the image space directly. Along the years, researchers also tried to craft their own set of features based on other assumptions.

The Local Binary Pattern (LBP) operator was originally designed for texture description [Ojala et al., 1996]. This operator is computed in a pixel level basis using a $N \times N$ kernel, thresholding the surroundings of each pixel with the central pixel value and considering the result as a binary value. The decimal form of the LBP code is expressed as:

$$LBP(x_c, y_c) = \sum_{i=0}^{N-1} f(I_i - I_c)2^i, \tag{2.7}$$

where i_c corresponds to the gray intensity of the center pixel (x_c, y_c), N is the number of sampling points, i_n is the gray intensity of the n-th surrounding pixel and $f(x)$ is defined as follows:

$$f(x) = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases} \tag{2.8}$$

Figure 2.4 shows how a face image in encoded in terms of their LBP decimals.

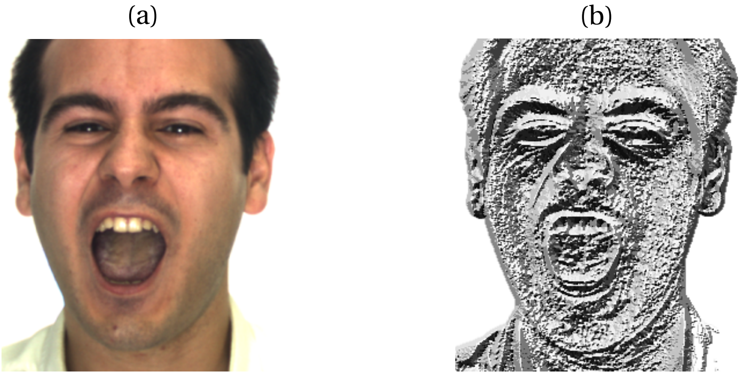
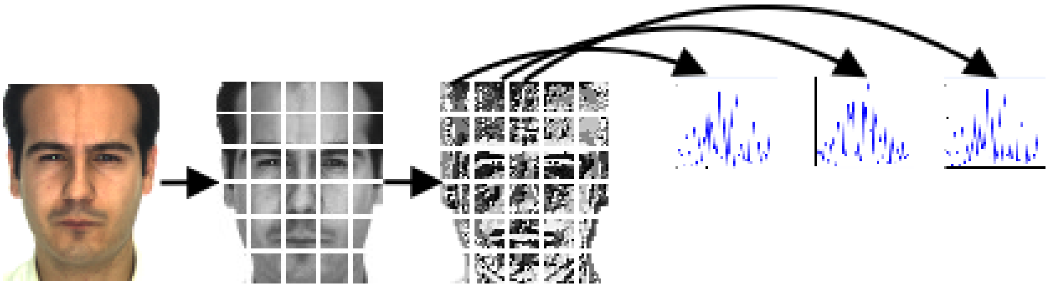


Figure 2.4 – Local Binary Pattern operator (a) Original image (b) LBP processed image

Ahonen et al. [2004] proposed a face recognition system by histograming the LBP output. This method is non parametric, hence, there is nothing to be done at **training time**. The technique first applies LBP encoding to each pixel of the face image and then divides the encoded face image into a set of windows. Histograms are then obtained from each region and concatenated to form a single feature vector. This is done at **enrollment** and **scoring** time. In Figure 2.5 it is possible to observe the application of this operator.



;

Figure 2.5 – Local Binary Pattern histograms

Several metrics were developed to compare two LBP histograms. The most traditional one is the chi-square distance (χ^2). Given two LBP histograms X^e and X^p (for enrollment and for probing) the χ^2 is defined as follows:

$$\chi^2(X^e, X^p) = \sum_{i,j} w_j \frac{(X^e_{i,j} - X^p_{i,j})^2}{X^e_{i,j} + X^p_{i,j}}. \tag{2.9}$$

Furthermore, several classification strategies were proposed using LBPs as front end, such as Rodriguez and Marcel [2006a] with Gaussian Mixture Model and Pereira et al. [2012] with Support Vector Machines.

Several different types of operators were built on top of LBPs. A good survey of all of them can be found in [Pietikäinen et al., 2011].

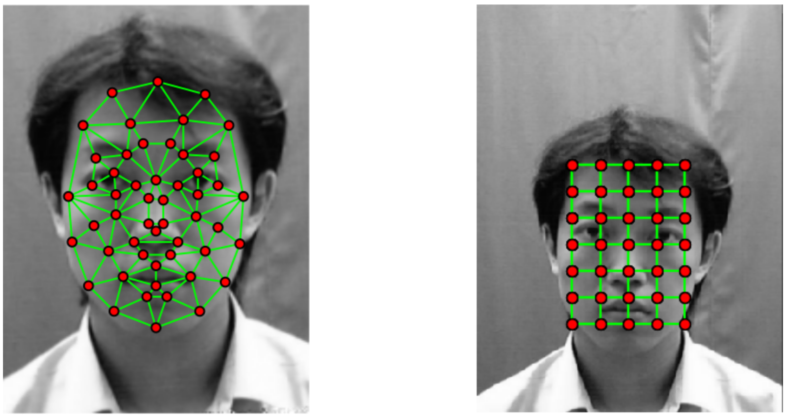
2.1.4 Gabor Wavelets

There is a class of face recognition algorithms that rely on Gabor features. Such features are found to model the (retinal) image processing in the primary visual cortex of mamal brains [Daugman, 1985].

A Gabor wavelet [Würtz, 1995] defined as:

$$\psi_{\vec{k}_j}(\vec{x}) = \frac{\vec{k}_j^2}{\sigma^2} e^{-\frac{\vec{k}_j^2 \vec{x}^2}{2\sigma^2}} \left[e^{i\vec{k}_j^{\mathrm{T}}\vec{x}} - e^{-\frac{\sigma^2}{2}} \right] \tag{2.10}$$

is an image filter that consists of a planar complex wave $e^{i\vec{k}_j^T\vec{x}}$ that is confined by an enveloping Gaussian and normalized to be mean free [Günther et al., 2017]. A Gabor wavelet is parametrized by the width σ of the Gaussian, its spatial orientation φ and the frequency k [Günther et al., 2017]. Commonly, a family of 40 Gabor [Shen and Bai, 2006] wavelets are used to extract the features by discretizing the frequencies and orientations. Complex valued Gabor features are extracted by convoluting the input image with each one of the 40 Gabor wavelets. Traditionally, only the absolute parts of these complex valued features are taken into account [Günther et al., 2017].



(a) EBGM

(b) Grid Graph

Figure 2.6 – Different ways to organize Gabor Jets. Extracted from [Günther, 2011, p.68]

Based on Gabor wavelet responses, several algorithms were proposed. The most well-known example is the elastic bunch graph matching (EBGM) that was proposed in the late 1990s [Wiskott et al., 1997]. The EBGM algorithm for face recognition is non parametric; hence, there is nothing to be computed at **training time**. Landmarks are detect and Gabor wavelet responses are computed in those detected regions of the face (see Figure 2.6a). All the Gabor wavelet responses computed in a particular region of the face are concatenated. The outcome of this concatenation is called Gabor Jet. Commonly the Gabor Jet is a result of the concatenation of the absolute values a_i and phases ϕ_i. The Gabor Jets can also be computed in a grid graph (see Figure 2.6b). Günther et al. [2017] indicated that grid graphs on average perform better than EBGM graphs. At **enrollment** time, Gabor Jets are basically stored. Finally, at **scoring** time, a comparison between stored and the probed Gabor Jet is carried out.

Given a stored Gabor jet $\mathscr{J}$ and the probed Gabor jet $\mathscr{J}'$, with their corresponding absolute values a and phases ϕ, several metrics to compare them was proposed such as:

Scalar product:

$$S(\mathscr{J},\mathscr{J}') = \sum_i a_i \cdot a'_i \tag{2.11}$$

Camberra:

$$S(\mathscr{J},\mathscr{J}') = \sum_i \frac{a_i - a'_i}{a_i + a'_i} \tag{2.12}$$

Absolute Phase:

$$S(\mathscr{J},\mathscr{J}') = \sum_i a_i \cdot a'_i cos(\phi_i - \phi'_i) \tag{2.13}$$

Zhang et al. [2005] proposed the combination between Gabor responses and LBPs. The technique called Local Gabor Binary Pattern Histogram Sequences (LGBPHS), applies Gabor wavelets at multiple scales and orientations to obtain several sub-images. These sub-images are then encoded using a standard MLBP operator and these local Gabor binary maps are then divided into non-overlapping regions. Then, a histogram is computed on each region. This approach is also non parametric; hence, nothing is done at **training** time. At **enrollment** time, such histogram are stored. At **scoring** time, a comparison between stored and the probed histograms is carried out using 2.9.

2.1.5 Deep Convolutional Neural Networks

Deep Convolutional Neural Networks have shown to be very powerful machine learning tool as they can be trained to learn complex non-linear mappings from high-dimensional data. But before its introduction, a more simple statistical model which is an elementary building block of those complex models shall be introduced: linear regression. Given a set of N input-output pairs $X = \{(x_1 ... x_n)\}$ and $Y = \{(y_1 ... y_n)\}$, in linear regression, it is hypothesized that exists a linear function mapping each $X \in \mathbb{R}^d$ to $Y \in \mathbb{R}$. Such model in this case is a linear transformation of the inputs: $f(x) = W^{\intercal}X + \beta$, where W is a $1 \times d$ matrix and $\beta \in \mathbb{R}$ is a bias term. Different values for W and β define different linear transformations and in general the goal is to find the parameters that minimizes some particular loss function $\mathscr{L}$. For instance, such loss can be the mean square error defined as: $\mathscr{L}(W,\beta) = ||Y - (W^{\intercal}X + \beta)||_2^2$. This is a convex function and its global minima can be found using different methods, such as via closed-form. One of the most popular and scalable ones is the so called gradient descend which is depicted by the Algorithm 1.

```
Data: $X$, $Y$, $it$,$\lambda$
Result: $W$, $\beta$
$W$ = random(dimension(X)) ;                         // Random initialization
$\beta = 0$ ;                                        // Usually initialized by 0
for i=0 to it do
    for j=0 to size(X) do
        $\frac{\partial \mathscr{L}}{\partial W,\beta} = y[j] - x[j]W + \beta$ ;      // Gradient
        $W = W + \lambda \frac{\partial \mathscr{L}}{\partial W}$;
        $\beta = \beta + \lambda \frac{\partial \mathscr{L}}{\partial \beta}$;
    end
end
```

Algorithm 1: Gradient descent training, where X is a $m \times d$ matrix, Y is a $m \times 1$ matrix, it is the number of iterations of the algorithm and λ is the learning rate

In most of the cases, specially in real world scenario, the relation between X and Y is not linear and a non-linear basis function $g(x)$ that maps X to Y has to be defined [Bishop, 2006, p.137]. Hence, the same linear regression can be performed between the pair $X = \{(g(x_1)...g(x_n)\}$ and $Y = \{(y_1...y_n)\}$. These basis functions can be polynomials, logistic functions, ReLU[1], etc. This basic building block is often called Perceptron [Haykin, 2009, p.48] and its graphical representation is depicted in Figure 2.7.

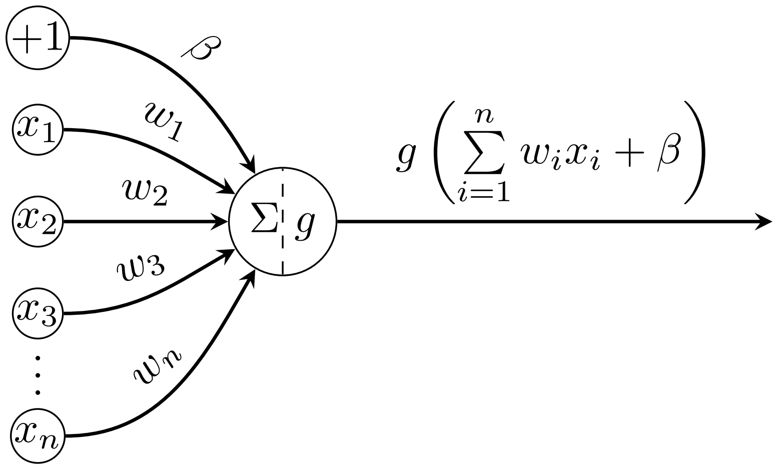


Figure 2.7 – Classical perceptron representation

The foundation of deep neural networks can be defined by a set those perceptrons stacked "vertically", making W a $n \times d$ matrix. For historical reasons, this n is coined the number of

[1] $g(x) = max(0,x)$

neurons. Furthermore, those perceptrons can also be stacked “horizontally”, hence, the non-linear outputs from $l_1 = g(W_1^\intercal X + \beta_1)$ can be provided as input to another set of non-linear operations (called hidden layer) $l_2 = g(W_2^\intercal l_1 + \beta_2)$ and finally this l_2 can be forwarded to our regressed output $o = g(W_3^\intercal l_2 + \beta_3)$. In this example, W_1, W_2 and W_3 is $n_1 \times d$, $n_2 \times n_1$ and $n_2 \times 1$ matrices respectively. This mechanism of stacking those perceptrons is a very powerful tool to solve very complex non-linear mappings and it is called Multi-Layer Perceptron (MLP) [Haykin, 2009, p.122]. Its classical graphic representation is depicted in Figure 2.8. The process

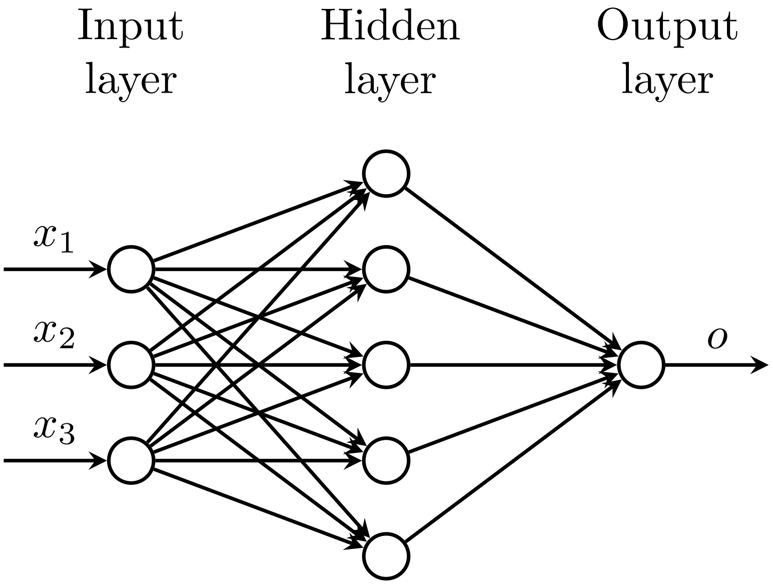


Figure 2.8 – Classical MLP representation with three inputs and one hidden layer

to learn all the possible values for W_1, W_2 and W_3 for this **non-convex function** is similar to the one defined for linear regression. The gradient of a particular loss (e.g mean square error) with respect to each $W_{[1..3]}$ and $\beta_{[1..3]}$ ($\frac{\partial \mathscr{L}}{\partial W_{[1..3]}, \beta_{[1..3]}}$) has to be propagated to all $W_{[1..3]}$ and $\beta_{[1..3]}$. This is carried out by an algorithm called Back Propagation [Haykin, 2009, p.153].

MLPs can also be used for classification. One way to approach such task is by adding as much as output peceptrons as the number of classes and make $Y \in \mathbb{Z}_2^c$, where c is the number of classes. Figure 2.9 presents an example of MLP for a two class problem.

For image classification, the selection of features (number of layers and number of neurons) for training a MLP is often empirical and data dependent. A possible solution to approach this issue would be to use directly the raw data and let the MLP training algorithm (Back Propagation) find the best feature extractors by adjusting $W_{[1..n]}$ and $\beta_{[1..n]}$. The problem with this approach is that the dimensionality of the input data is often high (specially for image recognition), hence the number of free parameters (number of connections) is large, since each hidden unit is fully connected. Depending of the amount of data available for training, the neural network tends to overfit.

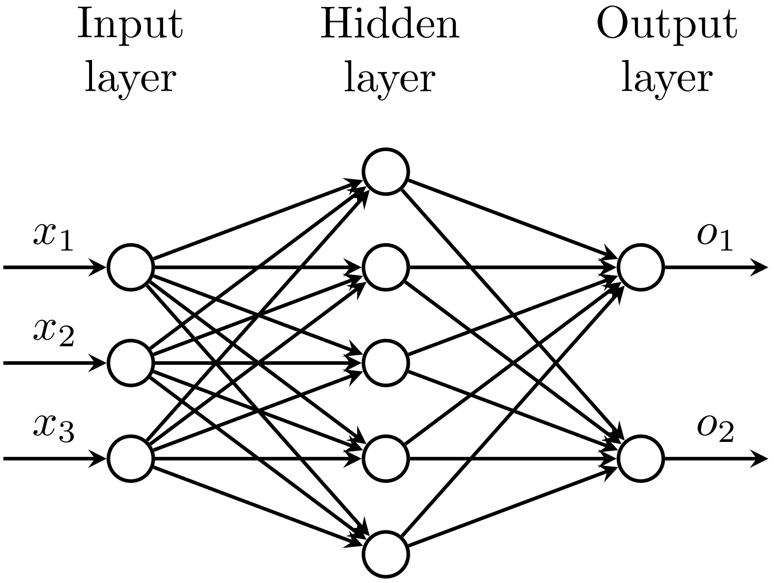


Figure 2.9 – Classical MLP representation for two class classification task with three inputs and one hidden layer

A Convolutional Neural Network (CNN) [LeCun et al., 1998] is an approach that tries to alleviate the aforementioned problem. Base perceptrons are replaced by a **local** linear transformation called convolution that is discretely defined for 1d signals as:

$$w * X = \sum_{i=k/2}^{i=d} \sum_{j=-k/2}^{j=k/2} w[j]X[i-j], \tag{2.14}$$

where w is the convolutional operator also called kernel or filter of dimension k and X is a 1d signal of dimension d. This transformation is highly used in image processing since it preserves spatial information of an input image. The same non-linearity hypothesis can be hypothesized for this operation, hence, non-linear convolutions can be defined as $g(w * X)$. Furthermore, bias terms can be added to this operation $g(w * X) + \beta$. These local linear transformations introduces a weight sharing in the neural networks that reduces drastically the number of free parameters that needs to be learnt, reducing the capacity of the network and improving its generalization capability. In Deep Convolutional Neural Networks, the convolutions are often followed by pooling layers. The purpose of such operation is to locally sub-sample the input signal by some statistical function. Figure 2.10 presents an example of pooling. In most practical cases in image recognition, the operator max is used and such operation is called MaxPooling. Such operations can be stacked as in the MLP and the process of learning w is the same as for MLPs (via Back Propagation).

The success of Deep Convolutional Neural Networks (DCNN) in computer vision research, the availability of several frameworks to instrument such networks and the possibility to work

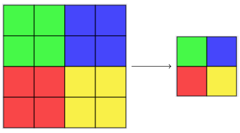


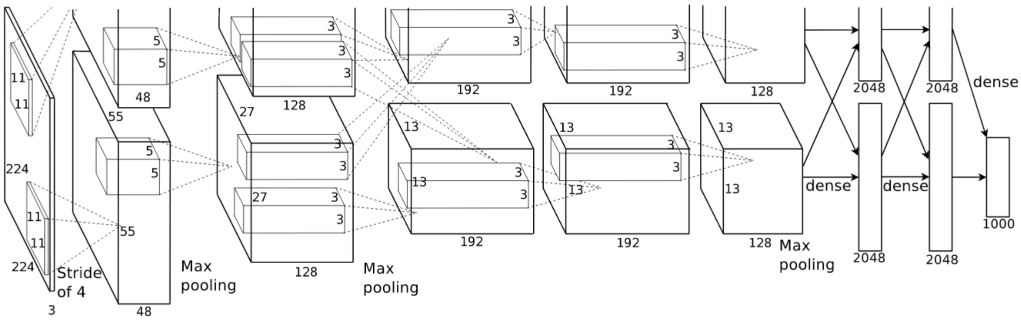
Figure 2.10 – Example of pooling a 2d input signal by patches of 2 × 2

with massive amounts of labeled data (CASIA WebFace [Yi et al., 2014], MS-Celeb [Guo et al., 2016] and Megaface [Kemelmacher-Shlizerman et al., 2016]) made face recognition error rates decrease steadily.

Despite the lack of deep understanding on why such neural networks work well and have good generalization capabilities in several different pattern recognition tasks [Mallat, 2016], practical heuristics were developed in the last five/six years to regularize the training and they are responsible for its success in practice. In the next subsections we would like to highlight some that, in our experience, have direct impact in decreasing face recognition error rates.

Alexnet

Krizhevsky et al. [2012] released in 2012 the AlexNet DCNN. Such work put together seminal elements that are standard until today in any pattern recognition task that relies on DCNN, including face recognition. Its architecture is depicted in Figure 2.11.



;

Figure 2.11 – Alexnet architecture [Krizhevsky et al., 2012]

Three seminal contributions worth mentioning in this work. **First**, it is about the depth of the DCNN. This network scale up the insights from LeNet[LeCun et al., 1998] and implemented a much deeper neural network composed by five convolutional layers and three fully connected layers. It was also roughly demonstrated that, in the case of object detection, depth matters. The **second** contribution was the usage of ReLU as activation function. In their work, the training was 6 times faster than the $tanh$ function. The **third** contribution was the usage of dropout [Hinton et al., 2012] as one of the regularization strategies. They idea of dropout is to

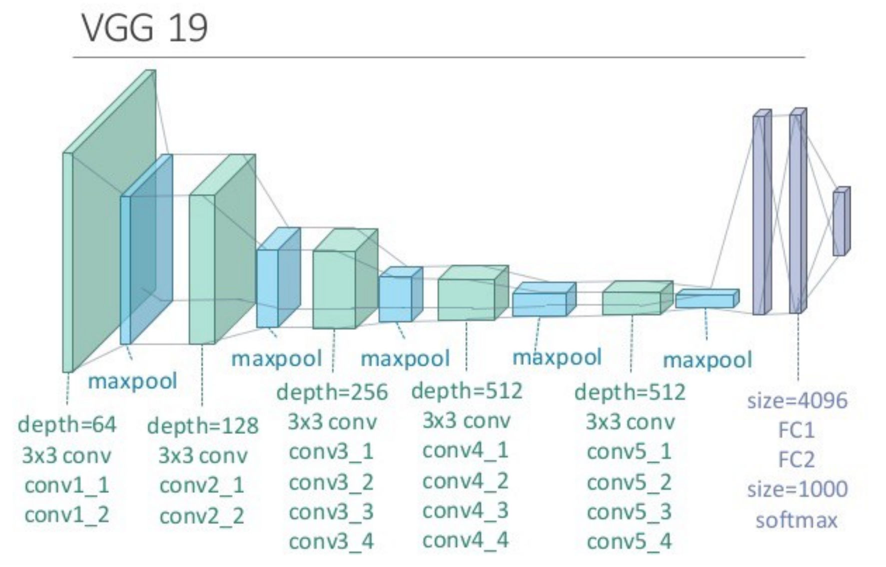


Figure 2.12 – VGG19 architecture. Image extracted from[Simonyan and Zisserman, 2014]

randomly drop connections during the training stage. This can be seen as an approximation of bagging [Bishop, 2006, p.653].

VGG networks

The VGG networks [Simonyan and Zisserman, 2014] were the first to use small kernels in each convolutional layer (3 × 3) and push forward even more the limits of depth in deep neural networks.

Its main contribution was the usage of small convolutional kernels chained in a long sequence of convolutions (even longer than Alexnet). Followed by sub-samplings (pooling), this architecture was able to detect image symmetries in larger areas of image that was thought possible only via larger kernels (5 × 5, 9 × 9 or 11 × 11) like in Alexnet or LeNet.

Figure 2.12 presents the schematic of one of the proposed VGG architectures.

Batch normalization

Introduced by Ioffe and Szegedy [2015], batch normalization consists in shifting (usually zero-mean) and scaling (normally one standard deviation) the output signal of each layer for each mini-batch.

Making this normalization part of the architecture allows the DCNN practitioners to be more "aggressive" with the learning rates and speeding up the convergence with larger architectures.

Inception modules

Szegedy et al. [2015] introduced the Inception modules. Those modules are composed by parallel combination of different convolutional kernels (1 × 1, 3 × 3, and 5 × 5 normally) as can be seen in Figure 2.13. This contribution allowed a dramatic reduction of free parameters to be learnt, increasing the recognition accuracies and generalization for several computer vision tasks.

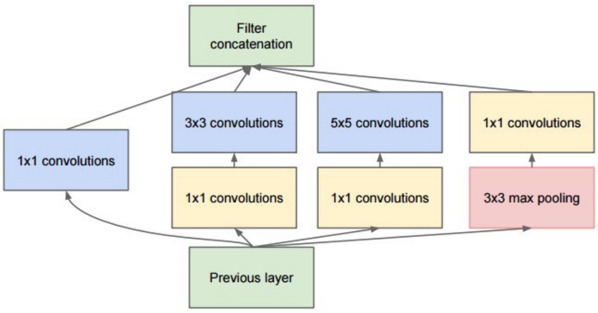


Figure 2.13 – One inception module composed by four parallel modules extracted from[Szegedy et al., 2015]

Residual Connections

As mentioned in the last subsections, practical evidences in several areas of computer vision have shown that depth of a DCNN seems to be a crucial factor in terms for accurate learning. One of the main obstacles to explore depth in DCNNs is the well known gradient vanishing/-exploding [Glorot and Bengio, 2010] problem. He et al. [2016] approached this issue bypassing the output of one intermediate layer and concatenating as the input of one of the layers ahead (two or three layers) as we can see in Figure 2.14. Such approach allowed the training of CNNs larger than 1000 layers[He et al., 2016].

A common way to approach the FR task using DCNNs is to, at **training time**, train it for a particular face dataset (n-class classification task). Then, it is hypothesized the feature detectors learnt for this particular classification task are generic and discriminative enough to be applied to other set of identities unseen by this training procedure. This can be carried out by taking the trained the DCNN and "drop" its outputs and make one of the hidden layers as the new output. Hence, this output can be used as a feature and be directly compared using an arbitrary metric, such as L2 norm, cosine similarity, Mahalanobis, etc. This feature is often called **embedding**. Figure 2.15 presents a simple example on how this embedding generator

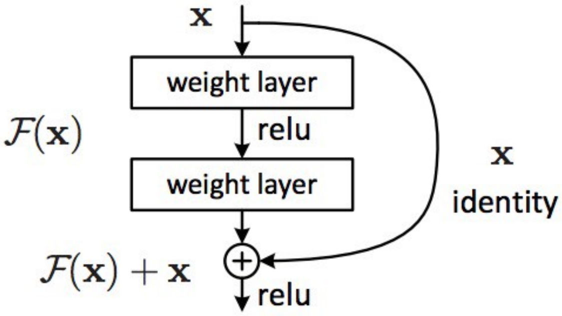

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.14 – One residual connection extracted from[He et al., 2016]

is created by dropping the classification output of DCNN.

2.2 Heterogeneous Face Recognition

In the beginning of this chapter a formalization of supervised learning was presented. We adapted the aforementioned formalization for the task of Heterogeneous Face Recognition and it is defined as the following. Let's assume now that we have two domains $\mathscr{D}^s = \{X^s, P(X^s)\}$ and $\mathscr{D}^t = \{X^t, P(X^t)\}$ called respectively **source domain** and **target domain** with both sharing the same set of labels Y. Hence, the goal of Heterogeneous Face Recognition task is to find a Θ, where $P(Y|X^s, \Theta) = P(Y|X^t, \Theta)$.

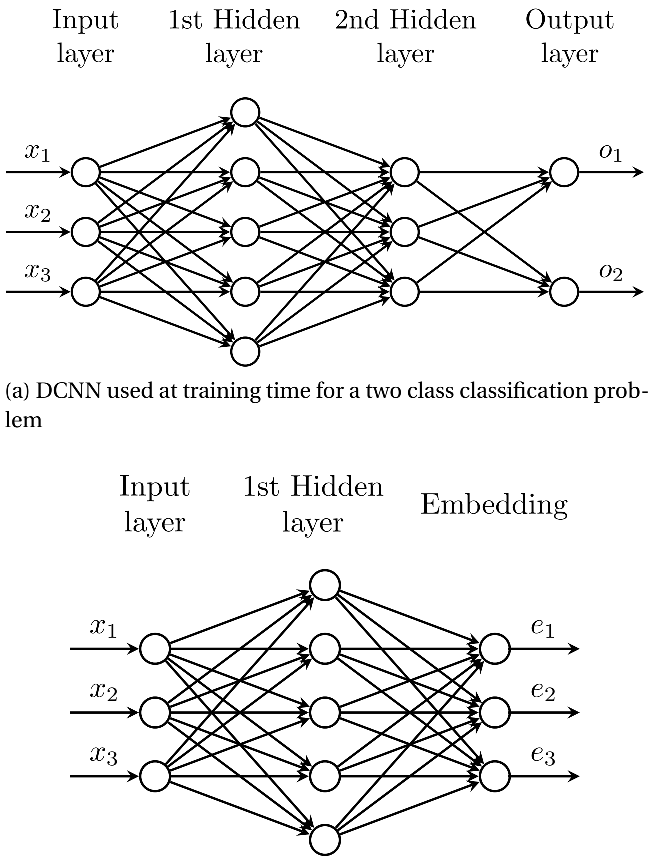
Several assumptions to model Θ were proposed during the last years and we can organize them in three main categories, whose details are described in the following three subsections.

2.2.1 Synthesis methods

In these methods a synthetic version of $\mathscr{D}^s$ is generated from $\mathscr{D}^t$. Once a synthetic version from $\mathscr{D}^t$ is generated, the matching can be done with regular face recognition approaches.

In [Wang and Tang, 2009], the authors proposed a patch based synthesis method that synthesizes VIS images to sketches. Thereafter, synthesized sketches are feed into regular face recognition systems, such as Eigenfaces, Fisherfaces, dual space LDA. At training time, a Markov Random Field generative model, pairing patch nodes (pixel level) from source and target domains, is build in such a way that the probability of a set patches from the source domain given one patch from the target domain is maximized. Although there is no source code officially available for this work, a matlab implementation can be found in[2]. This algorithm provides very appealing reconstructions using the images from the CUHK-CUFS (see Section 2.3.2), where the sketches are very reliable with respect to their corresponding

[2] https://github.com/ClaireXie/face2sketch



(a) DCNN used at training time for a two class classification problem

(b) DCNN used at enrollment and scoring time where the outputs are "dropped"

Figure 2.15 – DCNN - Example of embedding extraction

photographs. Even minimum details of shape and direction of the hair are preserved as we can observe in Figure 2.16. However, using less reliable hand drawn sketches databases, such as the CUHK-CUFSF (see Section 2.3.2) or other image modalities the reconstructions are very poor as we can see in Figure 2.17.

A slightly modification of the aforementioned approach was presented in [Peng et al., 2017]. Differently from [Wang and Tang, 2009], the authors replaced the patches by superpixels [Achanta et al., 2012] as we can observe in the Figure 2.18. An average rank one recognition rate of 99% and 72% was reported in CUHK-CUFS and CUHK-CUFSF databases respectively.

Focusing in thermal images, Zhang et al. [2017] proposed a method based on Generative Adversarial Networks (GANs) in order to generate thermogram images from visual spectra images for further identification using the Pola Thermal dataset [Hu et al., 2016] (see Section 2.3.3). The identification is carried out using the Visual Geometry Group (VGG) network embeddings that are freely available[3]. Such synthesized images are feed into this DCNN and

[3] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

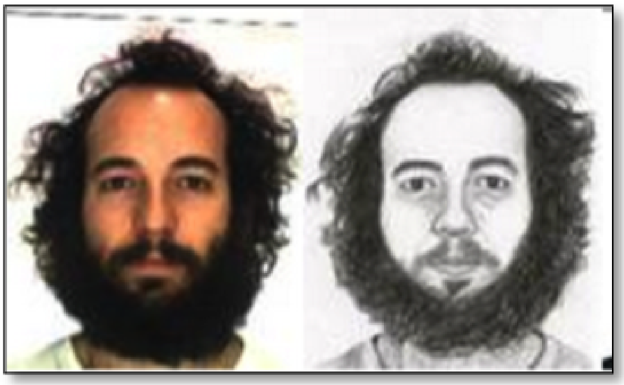
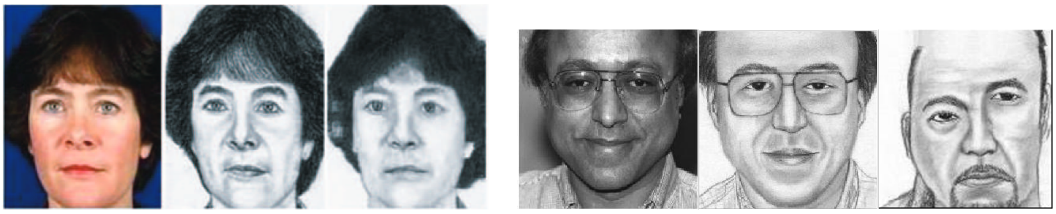


Figure 2.16 – Realism of CUHK-CUFS database. Small details such as, the direction of the hair and beard shape are the very similar



(a) Example from CUHK-CUFS database

(b) Example from CUHK-CUFSF database

Figure 2.17 – Synthesized images generated with the method proposed by Wang and Tang [2009]. Presented in the following order: Original photo, original sketch and synthesized sketch

compared. Using those embeddings, the authors published an Equal Error Rate (EER) of 25.17% using the VIS-to-ThermalPolarized procotols and an EER of 27.34% using the VIS-to-Thermal

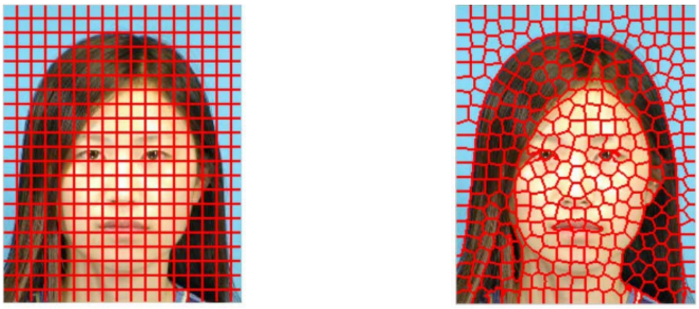
Similarly, Zhang et al. [2018] also proposed a strategy based on GANs for the exact same task (VIS-to-Thermal). With slightly changes in the loss they presented a rank one recognition rate of 19.9% using the private dataset that covers the VIS-to-Thermal problem (with 29 pairs of images to train the GAN from scratch).

2.2.2 Crafted features-based methods

In these methods raw face images from both domains ($\mathscr{D}^s$ and $\mathscr{D}^t$) are encoded with descriptors that are invariant between them.

Liao et al. [2009] proposed a very simple method for the task of VIS to NIR recognition, where both modalities are normalized using difference of gaussian filter as we can see in Figure 2.19.

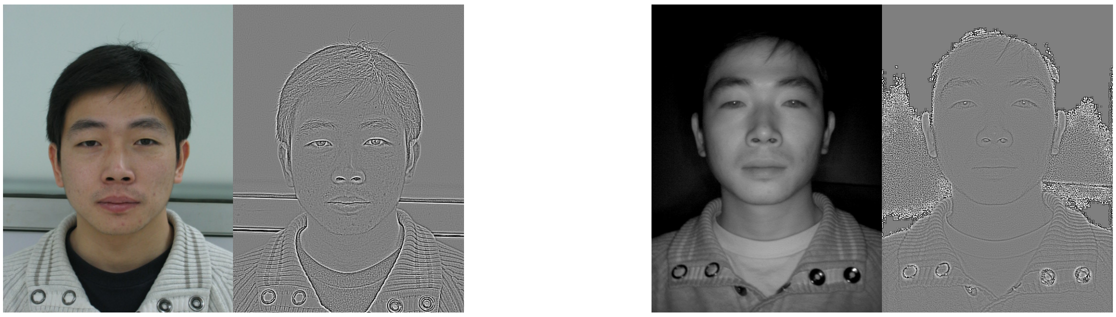
As feature descriptor, MutiScale Local Binary Patterns (MLBP)[Pietikäinen et al., 2011] (with



(a) Patch segmentation

(b) Super pixels segmentation

Figure 2.18 – Different procedures to segment parts of the face experimented by Wang and Tang [2009] and Peng et al. [2017] (images extracted from [Peng et al., 2017])



(a) RGB image (VIS)

(b) RGB image filtered

(c) NIR image

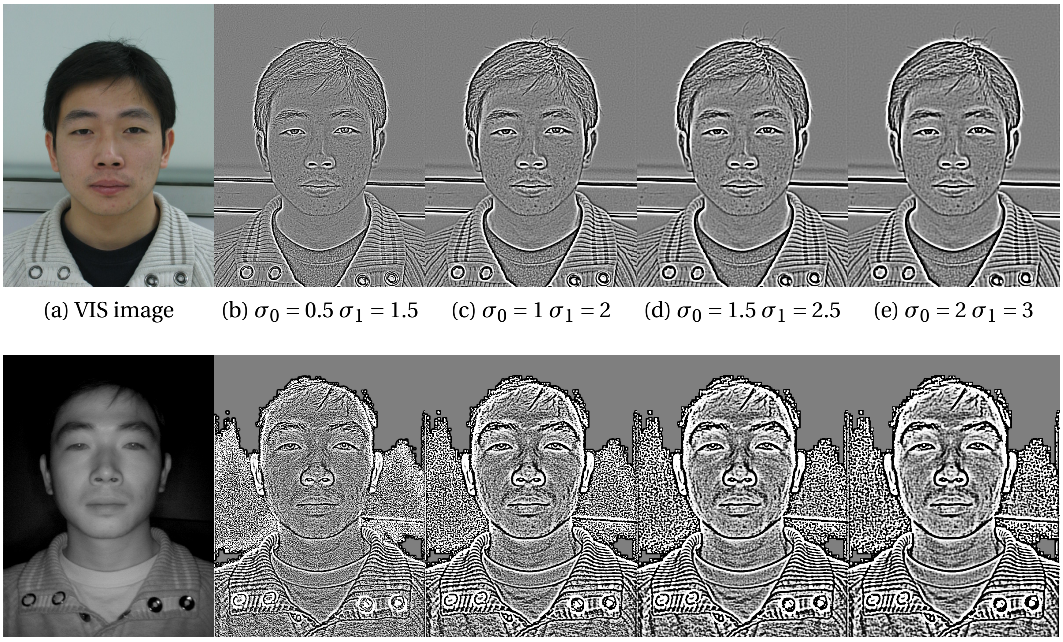
(d) NIR image filtered

Figure 2.19 – VIS and NIR images processed with Difference of Gaussians filter. Images taken from the CASIA NIR-VIS 2.0 database (see 2.3.1)

different radii) is used. Pairs of images VIS and NIR, processed with MLBP histograms, are used to train *FLD* system (see Section 2.1). A verification rate of 67.5% was reported under a false acceptance rate of 0.1% on the CASIA-HFB [Liao et al., 2009] database.

Liu et al. [2012] hypothesized that independent features between VIS and NIR are embedded in a particular range of frequency bands. To approach that the authors searched a particular range of scales of MultiScale Difference-of-Gaussian filter. This search can be seen in Figure 2.20. The authors used two different types of feature descriptor on top of this multiscaled processed images. The first one is the Histogram of Oriented Gradients (HOG) and the Scale-invariant feature (SIFT) descriptor is extracted. The Gentle Boost is used as a classifier [Bishop, 2006, p.657]. A rank one recognition rate of 98.51% was reported in the CASIA HFB database.

In a similar direction, Klare and Jain [2013] proposed an approach where face images from both domains are normalized using three different image processing filters (Difference-of-Gaussians, Center-Surround Divisive Normalization [Meyers and Wolf, 2008] and Gaussian Filter). Afterwards, two different feature local descriptors are extracted from patches of the



(a) VIS image (b) $\sigma_0 = 0.5$ $\sigma_1 = 1.5$ (c) $\sigma_0 = 1$ $\sigma_1 = 2$ (d) $\sigma_0 = 1.5$ $\sigma_1 = 2.5$ (e) $\sigma_0 = 2$ $\sigma_1 = 3$

(f) NIR image (g) $\sigma_0 = 0.5$ $\sigma_1 = 1.5$ (h) $\sigma_0 = 1$ $\sigma_1 = 1.5$ (i) $\sigma_0 = 1.5$ $\sigma_1 = 2.5$ (j) $\sigma_0 = 2$ $\sigma_1 = 3$

Figure 2.20 – Difference-of-Gaussians filter under different scales with VIS Images and NIR images. Images taken from the CASIA NIR-VIS 2.0 database (see 2.3.1)

image. The first one is the MutiScale Local Binary Patterns (with $r = \{1, 3, 5, 7\}$) and the second one is SIFT features. This very dense preprocessing and feature extraction mechanism is summarized in Figure 2.21.

Let I_A and I_B represent a pair of images from two modalities (A and B) and let $f_n | n = 1..6$ be the function that preprocess/feature extract I using one of the six combinations described in figure 2.21. At **training time**, a vector ϕ made of the combination of the cosine similarities between images from the same image modalities is built. Given the cosine similarity k and two images from the same modality:

$$k(f_n(I_i), f_n(I_j)) = \frac{f_n(I_i) \cdot f_n(I_j)}{||f_n(I_i)||.||f_n(I_j)||} \tag{2.15}$$

the vector ϕ is defined for the image modality A:

$$\phi_A = [k(f(I_{A_i}), f(I_{A_j})), ..., k(f(I_{A_i}), f(I_{A_j}))]. \tag{2.16}$$

Similarly for the image modality B:

$$\phi_B = [k(f(I_{B_i}), f(I_{B_j})), ..., k(f(I_{B_i}), f(I_{B_j}))]. \tag{2.17}$$

A matrix X is made of the concatenation ϕ_A and ϕ_B from the training set and the FLD (see

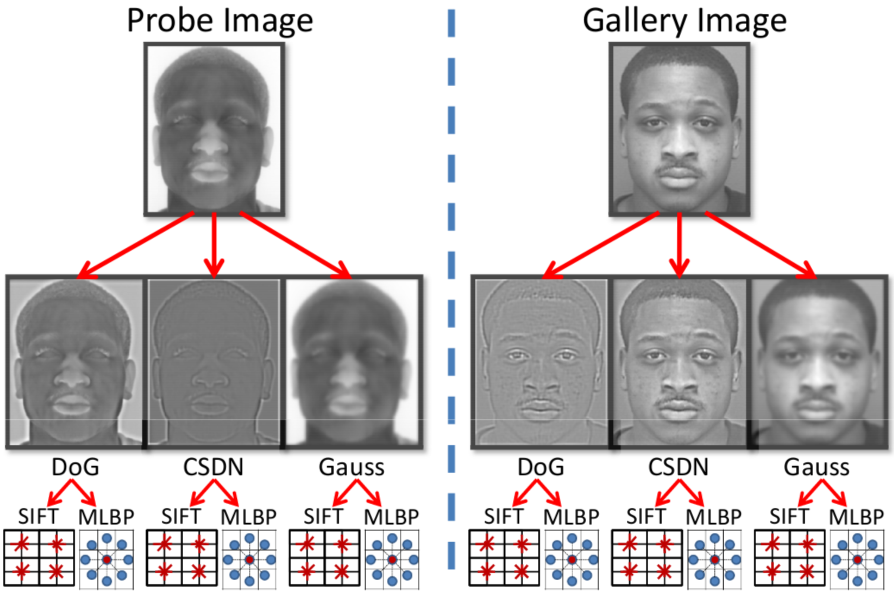


Figure 2.21 – Image processing and feature extraction mechanism proposed by [Klare and Jain, 2013], the probe and gallery images are thermal and VIS images respectively. Note that for one image, six different combinations of pre-processing/features are extracted.

Section 2.1) is estimated. At **scoring time**, ϕ_A and ϕ_B are estimated from a pair of samples and projected on the trained FLD. The cosine similarity is used as a metric. This approach, called prototype random subspace (P-RS) is tested on four different heterogeneous scenarios: NIR to VIS, thermal images to VIS, VIS to viewed sketch and forensic sketch to VIS. For the VIS to sketch, results were reported using the CUHK-CUFS database with a rank one recognition rate of 99%. As VIS to NIR reference, the CASIA HFB was used and a rank one recognition rate of 98% was reported. Experimental results using thermal to VIS and the Forensic-sketch to VIS database were reported in private databases.

With a very complex narrative around the Law of Universal Gravitation, Roy and Bhattacharjee [2016] proposed an illumination invariant filter called Local-Gravity-Face (LG-Face), whose implementation and final appearance is very similar to Local Binary Patterns as we can see in the Figure 2.22.

Images preprocessed using the LG-Face filter are directly compared using L1 norm. Experiments carried out with CUHK-CUFS database and the CASIA HFB showed a rank one recognition rate of 99.96% and 99.78% respectively.

2.2.3 Feature learning based methods

Feature learning based methods, as the name suggests, proposes to learn from data feature detectors that are domain invariant. Hence, in this hypothetical representation, images from

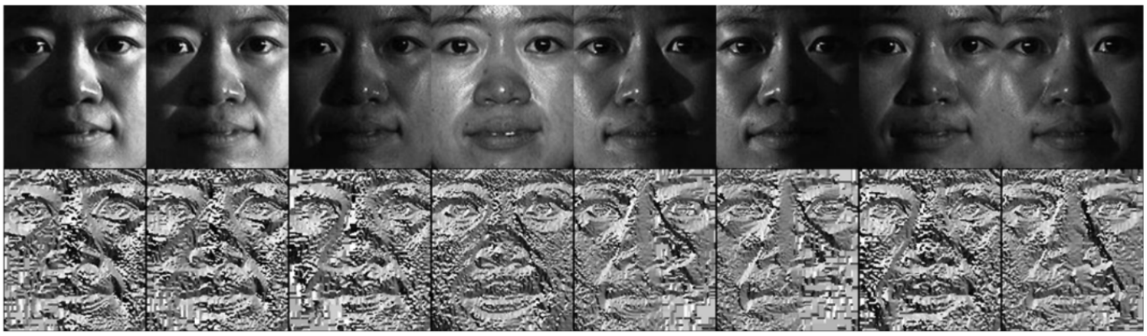


Figure 2.22 – Application of LG-Face under different illumination conditions

different image modalities can be directly compared.

In Jin et al. [2015] the authors proposed a feature learning approach whose goal is to find a pair of convolutional filters α where the LBP processed image difference between images from the same person, but different modalities are the minimum. Experiments carried out with the CASIA NIR-VIS 2 (see Section 2.3.1) showed an average rank one recognition rate of 86.2%. With the CUHK-CUFSF (VIS-to-Sketches) they presented an average rank one recognition rate of 81.3%.

Lu et al. [2018] propose a method that, on top of LBPs, learn simultaneously a code-book D and a feature map W between two image modalities. The optimization function is crafted in a such way that the modality gap between two image domains is explicitly minimized simultaneously with within-class variability while the between class variability is maximized. Experiments carried out using the CASIA NIR-VIS 2.0 dataset showed an average rank one recognition rate of 86.9%.

Based on DCNNs to model the joint mapping between $\mathscr{D}^s$ and $\mathscr{D}^t$, He et al. [2017] proposes a framework for VIS to NIR face matching where the low level feature detectors are learnt with VIS images only. The high level feature detectors are jointly learnt with VIS and NIR images and it is divided in: NIR layers, VIS layers and NIR-VIS shared layers (which are domain invariant). One embedding for each image modality is generated and they are compared at test time as we can observe in Figure 2.23. Experiments carried out using the CASIA NIR-VIS 2.0 dataset showed an average rank one recognition rate of 95.82%.

An extension of this work is presented in He et al. [2018], where the Wasserstein distance between the NIR and VIS signal distributions is incremented to overall loss function. Experiments with CASIA NIR-VIS 2.0 dataset showed an average rank one recognition rate of 98.7%.

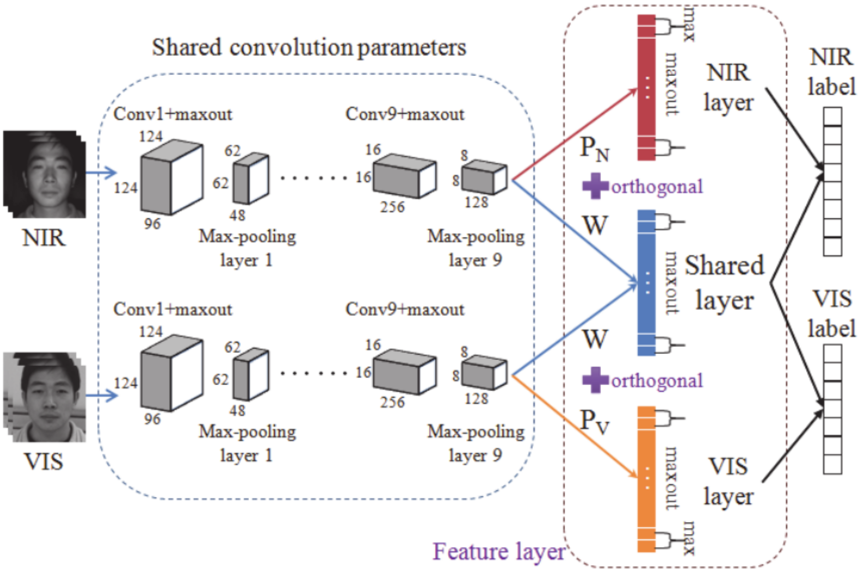


Figure 2.23 – DCNN architecture proposed by [He et al., 2018]

2.3 Heterogeneous Face Recognition Databases

Several databases were built along the years to support Heterogeneous Face Recognition research. This work reports experimental results and analysis under seven different image databases publicly available covering 4 different pairs of image domains. The next subsections describe each one and their respective evaluation protocols.

2.3.1 Visible Light to Near Infrared

As discussed in 2.1, most face recognition systems are based on images captured in the visible light range (VIS) of the electromagnetic spectrum (380 to 750nm).

The infrared spectrum (IR) can be further divided into several spectral bands and the boundaries between them can vary depending, basically, on the field involved (e.g., optical radiation, astrophysics, or sensor technology[Miller, 1994]). It comprises of the reflected IR (active) and the thermal IR (passive) bands. The active band (750 to 2500nm) is divided into the NIR (near infrared) and the SWIR (shortwave infrared) spectrum (100 to 250nm) [Bourlai et al., 2010]. An schematic of the wave lengths segmentation can be found in Figure 2.24.

This subsection presents the datasets used in this work covering the VIS to NIR scenario.

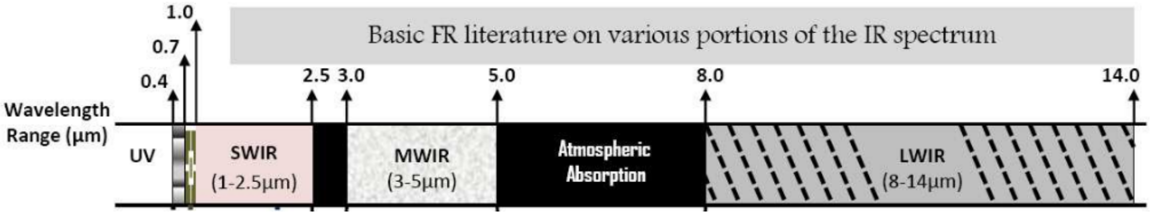


Figure 2.24 – Wave lengths schematic. Extracted from Bourlai et al. [2010]

CASIA NIR-VIS 2.0 Face Database (CASIA)

CASIA NIR-VIS 2.0 database [Li et al., 2013] offers pairs of mugshot images and their corresponding NIR photos. No information about the camera used in this work is provided. The images of this database were collected in four recording sessions: 2007 spring, 2009 summer, 2009 fall and 2010 summer, in which the first session is identical to the CASIA HFB database Li et al. [2009].

It contains pairs of images from 715 subjects. There are from one to twenty two VIS and from five to fifty NIR face images per subject, in a total of ≈ 21797 samples. Furthermore, the annotations of the position of the eyes are also distributed with the images. Figure 2.25 presents some samples of that database.

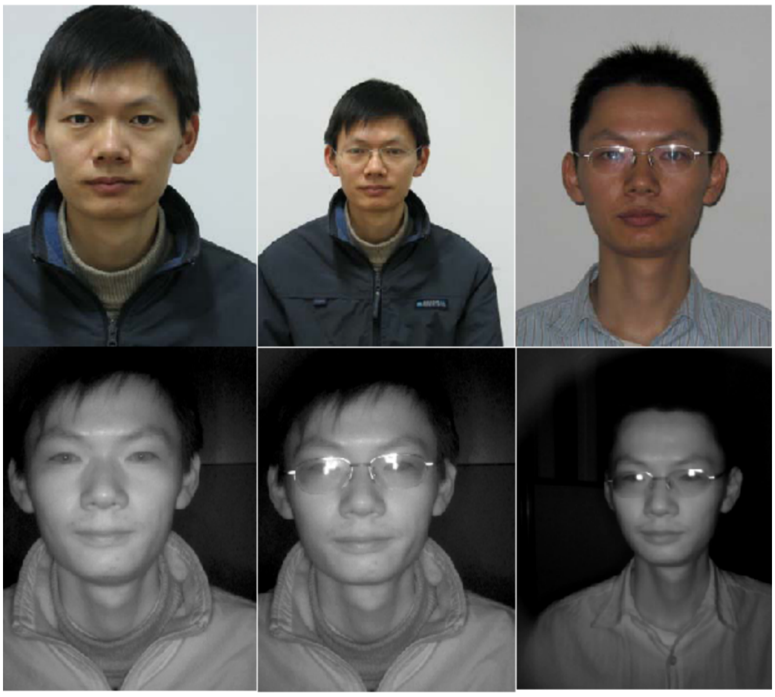


Figure 2.25 – Samples from CASIA NIR VIS 2.0 Database. Extracted from [Li et al., 2013].

This database has a well defined protocol and it is publicly available for download. It consists of ten fold cross closed-set identification protocols. Each fold is split in a training set containing 357 subjects and a test set containing 358 subjects. For reproducibility purposes, this evaluation protocols is published in a python package format[4]. Hence, future researchers will be able to reproduce exactly the same tests with the same identities in each fold. The average rank one recognition rate in the evaluation set (called view 2) is used as evaluation metric.

Near-Infrared and Visible-Light (NIVL) Dataset

Collected between the course of two semesters (fall 2011 and spring 2012) by the University of Notre Dame, the NIVL database [Bernhard et al., 2015] was collected with the objective to analyse the HFR error rates using COTS systems under different pre-processing algorithms. The VIS images were collected using a Nikon D90 camera. The Nikon D90 uses a 23.6 × 15.8 mm CMOS sensor and the resulting images have a 4288 × 2848 resolution. The images were acquired using automatic exposure and automatic focus settings. All images were acquired under normal indoor lighting at about a 5-foot standoff with frontal pose and a neutral facial expression.

The NIR images were acquired using a Honeywell CFAIRS system. CFAIRS uses a modified Canon EOS 50D camera with a 22.3 × 14.9 CMOS sensor. The resulting images have a resolution of 4770 × 3177. All images were acquired under normal indoor lighting with frontal pose and neutral facial expression. NIR images were acquired at both a 5ft and 7ft standoff.

The dataset contains a total of 574 subjects with 2,341 VIS and 22,264 NIR images. A total of 402 subjects had both VIS and NIR images acquired during at least one session during both the fall and spring semesters.

As mentioned before, this dataset was designed and released with the intention of evaluate the error rates of COTS systems in the VIS-NIR task under different image processing algorithms. Since there is no need to train background models for commercial matchers, the original database evaluation protocol does not have a training set. Hence, for this work, we designed a 5-fold cross-validation closed-set identification strategy, where the 574 subjects were split in 344 identities for training and 230 identities for test. The average rank one recognition rate in the test set is used as evaluation metric.

This evaluation protocol is equally available for download in a python package[5]. The database authors don't provide any face annotation with the images. However, annotations were **manually generated during the course of this work** and they are available for download in the aforementioned python package. Figure 2.26 presents some samples of that database.

[4] https://pypi.python.org/pypi/bob.db.cbsr_nir_vis_2

[5] https://pypi.python.org/pypi/bob.db.nivl

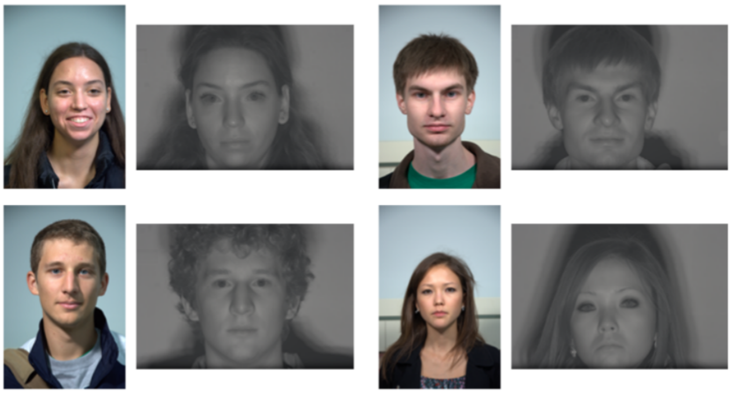


Figure 2.26 – Samples from NIVL Database. Extracted from [Bernhard et al., 2015].

Long Distance Heterogeneous Face Database

Long Distance Heterogeneous Face Database (LDHF-DB)[Kang et al., 2014] was built to address the VIS to NIR HFR task concomitantly with the task of recognition at distance. To address that, data from 100 identities (70 males and 30 females) were collected in both VIS and NIR (at nighttime) in different standoffs: 1m, 60m, 100m and 150m. For each subject, over the course of one month, one image was captured at each distance in daytime and nighttime. Hence, there are in total eight images for each subject, as shown in Figure 2.27.

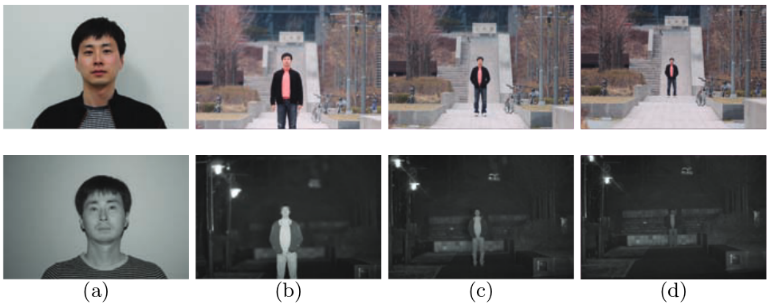


Figure 2.27 – Samples from LDHF-DB Database collect at . (a) 1m (b) 60m (c) 100m (d) 150m. Extracted from [Kang et al., 2014]

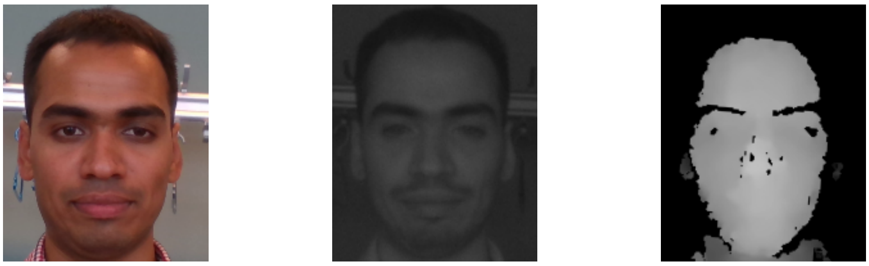
The short distance (1m) VIS images were collected under a fluorescent light by using the DSLR camera with the Canon F1.8 lens; and the NIR images were collected using the modified DSLR camera and NIR illuminator with twenty four infrared LEDs. Long distance (over 60m) VIS images were collected during the daytime using a telephoto lens coupled with a DSLR camera; and NIR images were collected using the DSLR camera with NIR light provided by RayMax300 illuminator[Kang et al., 2014]. All images of a subject are frontal faces without glasses, and collected in a single sitting.

Although this dataset has a well defined 10-fold cross-validation protocol (closed-set identification test), the distribution of the identities were not made publicly available. Each fold is split in to a training set containing 50 subjects and test set containing 50 subjects. VIS Images at 1m standoff are used at enrollment time. At scoring time, NIR images standoffs at 1m, 60m, 100m, 150m are used at probes. For reproducibility purposes, this evaluation protocols is published in a python package format[6]. Hence, future researchers will be able to reproduce exactly the same tests with the same identities in each fold.

The database authors don't provide any face annotation with the images. However, annotations were **manually generated during the course of this work** and they are available for download in the aforementioned python package.

FARGO database

The FARGO database has been recorded across a time period of 5 months on three different sites and differently from other databases, it is focused on the **Face Verification task**. The total of 75 subjects have been recorded, among which 20 are female and 55 males. At the time of recording, most of the subjects were aged between 20 and 30 years old - the exact age is available as metadata. The recordings have been made using an Intel®RealSense™SR-300 device, allowing to capture classical VIS images, NIR images and depth maps video sequences at the same time. Exemplar images derived from each stream are shown in Figure 2.28.

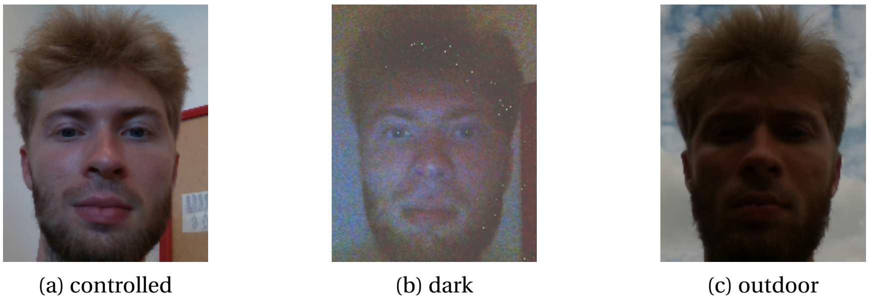


(a) RGB (b) NIR (c) depth

Figure 2.28 – Example of images retrieved from the different streams of the camera.

Each subject was recorded during three sessions. The first session took place in an indoor environment with controlled lighting, ensuring the face to be well lit. Also, subjects were asked to bind their hair or remove hats to ensure complete visibility of the face (this has not been asked in other sessions). The second session has been recorded in a very dark room, and the third one has been recorded outdoor, and hence contains arbitrary illumination conditions. In each session and for each subject, four video sequences were recorded: two where the device was mounted as a webcam on a laptop, and two where the device was mimicking the frontal camera of a mobile phone. This was done in order to simulate a typical case of remote authentication.

[6] https://pypi.python.org/pypi/bob.db.ldhf



(a) controlled

(b) dark

(c) outdoor

Figure 2.29 – Example of images acquired in each session.

During each recording, the subject has been asked to remain still for the first five seconds, and then to move his head to the left, to the right, to the top and to the bottom, while still looking at the device. This has been done for two reasons: the movements in yaw will allow to address the challenge of face recognition across pose and the movements in pitch are trying to mimic the typical pose variations one can observe when using a front-facing smartphone camera. Also, subjects wearing glasses were asked to remove them for at least one recording in each session.

For all recorded face video sequences, 13 specific frames have been manually annotated. Roughly, these frames correspond to a frontal view of the face, to the extreme positions attained when the subject moves her/his head (left, right, top and bottom), plus two frames in between the extreme position and the frontal view. Selected frames have been annotated with 16 keypoints corresponding to salient facial features depicted on Figure 2.30.

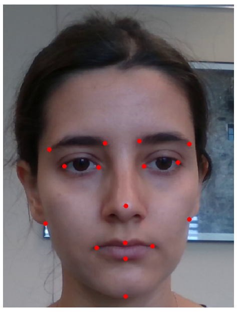


Figure 2.30 – The 16 annotated fiducial points.

To address the HFR verification problem, two major protocols were designed. The first one addresses the task of matching VIS to NIR images and the second one addresses the task of matching VIS images to depth maps. Furthermore, we have created three sub-protocols that address each lighting condition and they are described in the Table 2.1.

These evaluation protocols for face verification are equally available for download[7] with their

[7] https://pypi.python.org/pypi/bob.db.fargo

Table 2.1 – Summary of the different protocols for heterogeneous face recognition: c stands for controlled, **d** for dark and **o** for outdoor.

	Training	**Dev**		**Eval**	
		Enroll	Probe	Enroll	Probe
MC	RGB+NIR+c	RGB+c	c	c	c
UD	RGB+NIR+c	RGB+c	**NIR+d**	RGB+c	**Depth+d**
UO	RGB+NIR+c	RGB+c	**NIR+o**	RGB+c	**Depth+o**
MC	RGB+Depth+c	RGB+c	c	c	c
UD	RGB+Depth+c	RGB+c	**Depth+d**	RGB+c	**Depth+d**
UO	RGB+Depth+c	RGB+c	**Depth+o**	RGB+c	**Depth+o**

corresponding annotations.

2.3.2 Visible Light to Sketches

There are basically three different kinds of sketches used by law enforcement and HFR communities: forensic, composite and viewed sketches. **Forensic sketches** are hand made by highly trained forensics artists working with eye-witnesses that provide verbal descriptions of a subject (usually after crimes). **Composite sketches** are the new trend in law enforcement, since it doesn't require specialized artists to make them (a well trained operator is sufficient for the task of supporting the witness and make the sketch). This one is computed generated using specialized software. Options on the market are: Identi-Kit[8], Faces[9] and Evofit[10]. The last type of sketches are called **viewed sketches**; which is made by an artist looking at the corresponding target photograph. Recognizing people using this type of sketches as input is a hypothetical problem, but it is anyway investigated in the literature, specially in the HFR area, since it's possible to generate database with substantial amount of subjects and there are less legal issues to deal with. In the next subsections we present the sketch databases used in this work.

CUHK Face Sketch Database (CUFS)

CUHK Face Sketch Database (CUFS) is composed by **viewed sketches**. The viewed sketches are made by an artist looking to the corresponding photograph of a subject. It includes 188 faces from the Chinese University of Hong Kong (CUHK) student database, 123 faces from the AR database[Martinez, 1998] and 295 faces from the XM2VTS database[Messer et al., 2003]. Figure 2.31 presents some samples of that database.

There are 606 face images in total. For each face image there is a sketch drawn by an artist

[8] http://identikit.net/
[9] http://www.iqbiometrix.com/products_faces_40.html
[10] https://evofit.co.uk/

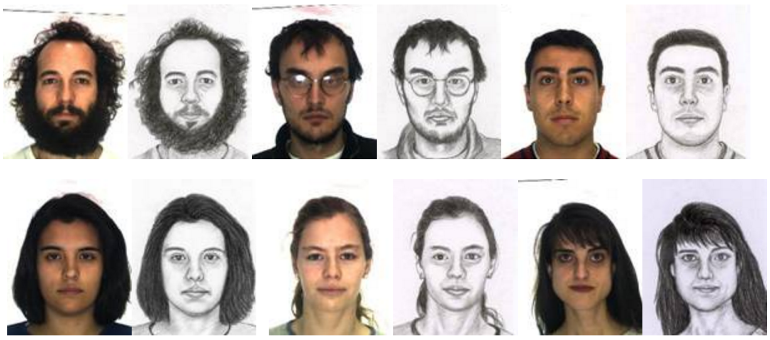


Figure 2.31 – Samples from CUHK CUFS Database. Extracted from [Bernhard et al., 2015].

based on a photo taken in a frontal pose, under normal lighting condition and with a neutral expression.

Unfortunately there is no defined evaluation protocol established for this database. Each work that uses this database implements a different way to report results. In Wang and Tang [2008] the 606 identities were split in three sets (153 identities for training, 153 for development, 300 for evaluation). The rank one recognition rate in the evaluation set is used as performance measure. Unfortunately the file names for each set were not distributed. In Klare and Jain [2013] the authors created a protocol based on a 5-fold cross validation, splitting the 606 identities in two sets with 404 identities for training and 202 for testing. The average rank one recognition rate is used as performance measure. The authors from [Bhatt et al., 2012] evaluated the error rates using only the pairs VIS-Sketch corresponding to the CUHK Student Database and AR Face Database and in [Bhatt et al., 2010] the authors used only the pairs corresponding to the CUHK Student Database. In [Jin et al., 2015] the authors created a protocol based on a 10-fold cross validation splitting the 606 identities in two sets with 306 identities for training and 300 for testing. Also the average rank one recognition error rate in the test is used to report the results. Finally in [Roy and Bhattacharjee, 2016], since the method does not requires a background model, the whole 606 identities were used for evaluation and also to tune the hype-parameters (via grid search). Fine tuning and testing using the same cohort is not a good practice in machine learning and the results presented, in terms of error rates, are possibly biased.

For comparison reasons, we will follow the same strategy as in [Klare and Jain, 2013] and do 5 fold cross-validation splitting the 606 identities in two sets with 404 identities for training and 202 for testing and use the average rank one recognition rate, in the evaluation set as a metric. For reproducibility purposes, this evaluation protocol is published in a python package format[11]. Hence, future researchers will be able to reproduce exactly the same tests with the same identities in each fold.

[11] https://pypi.python.org/pypi/bob.db.cuhk_cufs

CUHK Face Sketch FERET Database (CUFSF)

The CUHK Face Sketch FERET Database (CUFSF) [Zhang et al., 2011] comprises of **viewed sketches**. It includes 1,194 face images from the FERET database[Phillips et al., 1996] and their respectively sketch drawn by an artist. There isn't an evaluation protocol established for this database. Each evaluation using this database implements a different way to report the results in terms of recognition rates. In [Zhang et al., 2011] the authors split the 1,194 identities in two sets with 500 identities for training and 694 for testing. Unfortunately the file names for each set was not distributed. The Verification Rate (VR) considering a False Acceptance Rate (FAR) of 0.1% is used as a performance measure. In [Lei et al., 2012] the authors split the 1,194 identities in two sets with 700 identities for training and 494 for testing. The rank one recognition rate is used as performance measure. Figure 2.32 presents some samples of that database.

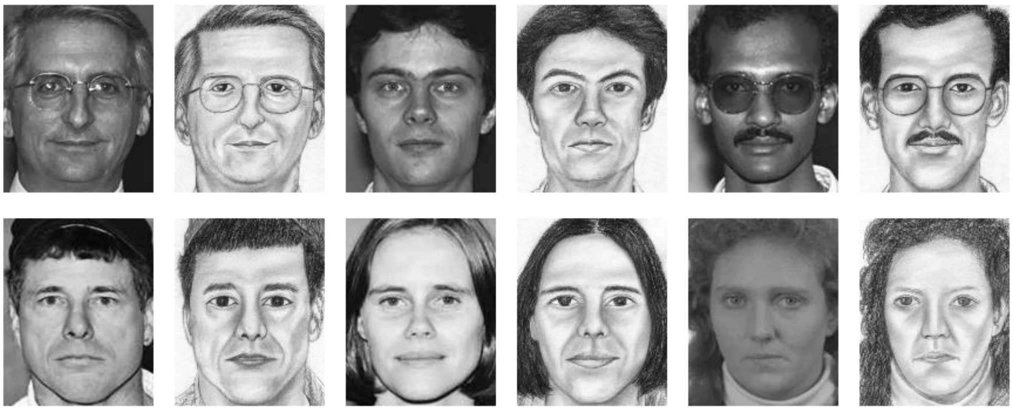


Figure 2.32 – Samples from CUHK CUFSF Database. Extracted from [Zhang et al., 2011].

For comparison reasons, we will follow the same strategy as in [Lei et al., 2012] and do 5 fold cross-validation splitting the 1,194 identities in two sets with 700 identities for training and 494 for testing and use the average rank one recognition rate, in the evaluation set, as a metric. This evaluation protocol is also available for download[12]. The database authors don't provide any face annotation with the VIS images. However, annotations were **manually generated during the course of this work** and they are available for download in the aforementioned python package.

2.3.3 Visible Light to Thermograms

Polarimetric and Thermal Database (Pola Thermal)

Collected by the U.S. Army Research Laboratory (ARL), the Polarimetric Thermal Face Database (first of this kind), contains polarimetric LWIR (long-wave infrared) imagery and simultane-

[12] https://pypi.python.org/pypi/bob.db.cuhk_cufsf

ously acquired visible spectrum imagery from a set of 60 distinct subjects [Hu et al., 2016].

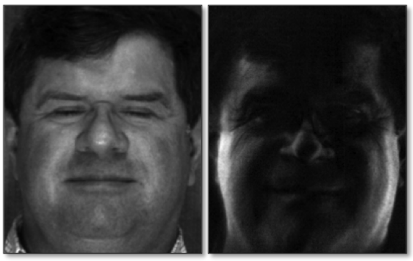


Figure 2.33 – Samples from Pola Thermal Database

For the data collection, each subject was asked to sit in a chair and remove the glasses, if any. A floor lamp with a compact fluorescent light bulb rated at 1550 lumens was placed 2m in front of the chair to illuminate the scene for the visible cameras and a uniform background was placed approximately 0.1m behind the chair. Data was collected at three distances: Range 1 (2.5m), Range 2 (5m), and Range 3 (7.5m). At each range, a baseline condition is first acquired where the subject is asked to maintain a neutral expression looking at the polarimetric thermal imager. A second condition, which is referred as the "expressions" condition, was collected where the subject is asked to count out loud numerically from one upwards. Counting orally results in a continuous range of motions of the mouth, and to some extent, the eyes, which can be recorded to produce variations in the facial imagery. For each acquisition, 500 frames are recorded with the polarimeter (duration of 8.33 s at 60 fps), while 300 frames are recorded with each visible spectrum camera (duration of 10s at 30 fps). Two types of thermal images are provided in this database, the first one is the Conventional Thermal and the Polarimetric Thermal. As opposed to the original protocol, that proposes a 100-fold cross-validation evaluation, we applied a 5-fold cross validation evaluation protocol where the 60 clients are split in 25 identities for training and 35 identities for testing. The average rank one recognition rate in the test set is used as evaluation metric. The protocol called "overall", which probes data from the 3 ranges, is used in this work. This evaluation protocol is also available for download[13].

Table 2.2 summarizes relevant features of all mentioned databases.

2.4 Evaluation Metrics

The evaluation protocols proposed to measure error rates in the databases mentioned in Section 2.3 share the same methodology, exception to FARGO database The majority of them approach the HFR task as closed-set identification problem (see section 1.1). The FARGO database approach the HFR task as a verification problem. Hence, for comparison reasons, in this work, FARGO is approached as verification task and the remaining databases are

[13] https://pypi.python.org/pypi/bob.db.pola_thermal

Table 2.2 – Summary of all database characteristics

Database name	# Identities	Annotations?	Public Protocol?
VIS/Sketch			
CUHK-CUFS	606	✓	✗
CUHK-CUFSF	1,194	✗	✗
VIS/NIR			
CASIA	715	✓	✓
LDHF	100	✗	✗
NIVL	574	✗	✗
FARGO	75	✓	✓
VIS/Thermal			
Thermal	60	✓	✗
PolaThermal	60	✓	✗

approached as closed-set identification task. This subsection describes the evaluation metrics used for each one of the task.

2.4.1 Closed-set identification

In the closed-set identification task, every probe sample is compared with all the class-specific models stored within the system. The decision-making process then consists of returning the set of n-classes (models) that are similar to the one of the probe sample. In practice, this is achieved by returning the n largest scores sorted. The identification of a probe sample is correct when its class belongs to the returned set of n classes. If the model corresponding to the probe sample gives the r^{th} largest score, the rank of this probe sample is said to be equal to r.

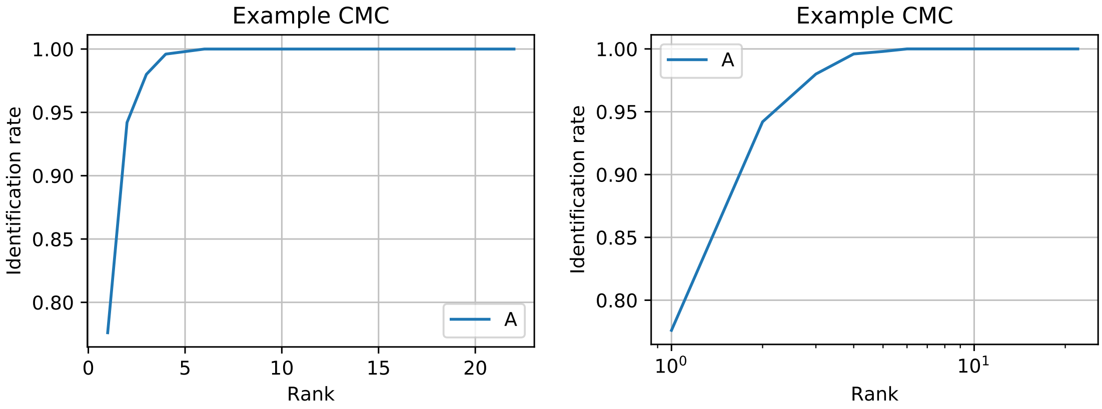
The closed-set identification performance of a system can be represented using a cumulative match characteristics (CMC) curve. For each value r , the CMC curve displays how many probe samples have a rank r or lower, normalized by the total number of probe samples. When $r = 1$, the corresponding measure is known as the recognition rate (RR).

An example of this curve is presented in Figure 2.34.

2.4.2 Verification

In a verification task, the decision-making process consists of comparing a given score $s = P(x|\Theta)$ with a particular threshold θ, where x is the input sample and Θ is the model that corresponds to the claimed identity. In case $s \geq \theta$, it is assumed that the input sample corresponds to the claimed identity. If not, the assumption is False.

The verification task can produce two different types of errors. The first one is the False Match (FM), if the verification system has wrongly accepted a zero effort impostor. The second one is the False Non Match (FNM) if a true claimant (also called genuine) has been rejected. Splitting



(a) Linear Scale

(b) Semi logarithm scale

Figure 2.34 – Cumulative Match Characteristics (CMC) curve under different scales in the x-axis of an arbitrary biometric system

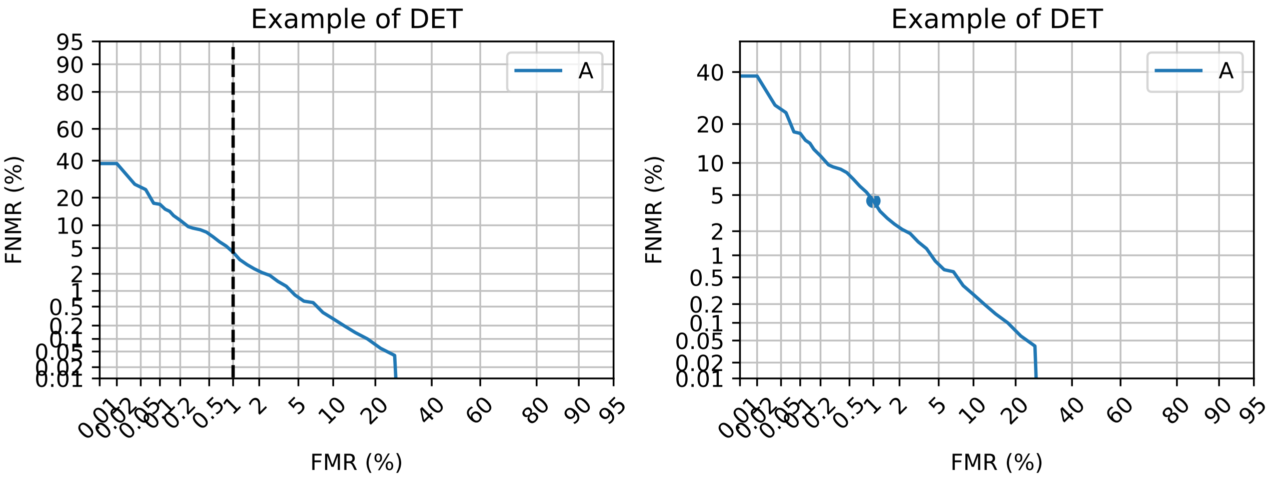
the scores into true claimant scores and zeroth effort impostor scores the False Match Rate (FMR) and the False Non Match Rate (FNMR) can be defined as follows:

$$FMR(\theta) = \frac{FM}{\#\text{ zero effort impostors}} \tag{2.18}$$

$$FNMR(\theta) = \frac{FNM}{\#\text{ genuines}} \tag{2.19}$$

A limitation when reporting FMR and $FNMR$ values for a particular threshold θ is that they describe the performance for one specific operational point. Furthermore, the FMR and the FNMR are correlated. Depending of the value of $theta$, increasing FMR reduces FNMR, and vice versa. To observe this trade-off between those two possible errors under different values for θ, the Detection Error Tradeoff (DET) curve is introduced. In the DET curve FMR vs FNMR are plotted under different values of θ in a bi-logarithm plot as can be observed in Figure 2.35.

In this work θ is estimated using the development set (only for FARGO where verification experiments applies). The value of θ is taken once FMR is at 1% (see dashed line in Figure 2.35 (a)). Then, the FNMR(θ) in the evaluation set is estimated and reported (see the blue dot in Figure 2.35 (b)). In this work such metric is represented as FNMR@FMR=1%(dev).



(a) Development set

(b) Evaluation set

Figure 2.35 – Example of DET curve of an arbitrary biometric system. It is possible to observe an FNMR@FMR=1%(dev) of ≈ 5% in the Evaluation set

3 From Face Recognition to Heterogeneous Face Recognition

In the previous chapter the literature review in Face and Heterogenous Face Recognition was presented.

In the course of the last years Face Recognition researchers investigated ways to find features that are both discriminative and robust against different sources of natural noise, such as, illumination, pose, expression, aging. Noise introduced by these factors introduces covariate shift in the pixel distribution and if this is not taken into account, Face Recognition error rates substantially increase. We can observe this effect in Figure 2.3 (a) where different illumination conditions slightly changed the distribution of the pixels. Advances in terms of algorithms and the volume of data collected to understand these sources of covariate shift made error rates in face recognition decrease steadily.

Along chapter 2.3 it was possible to observe differences in appearance between different image modalities. Intuitively, those differences can be slightly severe, such as VIS to NIR matching (see section 2.3.1) or very severe, such as VIS to Thermal or VIS to Depth (see sections 2.3.1 and 2.3.3). Although there are clear differences in appearance between different image modalities, the assessment on how Face Recognition trained with VIS only perform (in terms of recognition rates) was never carried out. Furthermore, since the current state-of-the-art face recognition approaches were created to handle certain sources of covariate shift, this assessment is something that should be verified.

The goal of this chapter is two fold. First, baselines based on different Face Recognition algorithms trained with only VIS images are established for the HFR task. Second, baselines based on the current state-of-the-art algorithms for HFR are presented and integrated as part of the software package that corresponds to this thesis that allows its reproducibility.

The experiments from this chapter can be regenerated with the software package corresponding to this thesis[1]. More information on how to install this software package, go to the Appendix A.

[1] https://gitlab.idiap.ch/bob/bob.thesis.tiago

In Sections 3.1 and 3.2 presents the Face Recognition and the Heterogeneous Face Recognition baselines used in this work respectively. The hyperparameter selection and implementation specificities are described with details. In the Section 3.3, HFR experiments are introduced. Finally Section 3.4 presents the final discussions of the chapter.

3.1 Face Recognition baselines

In this section Face Recognition systems either based on crafted features or either based on feature learning are presented.

3.1.1 Gabor Graphs

The mechanism around Gabor wavelets was briefly introduced in Section 2.1.4. In this subsection just implementation details are presented.

The approach based on Gabor graphs was introduced by [Günther et al., 2012]. In this recent work Günther et al. [2017] exhaustively fine tuned the wavelet parameters and the dimensions of the face for the VIS face recognition using the BANCA face database [Bailly-Bailliére et al., 2003]. The size of detected faces were set to have a width(w) and height (h) ratio of $w : h = 4 : 5$. Then, h was exhaustively tuned from 20 pixels to 200 pixels. Error rates started to stabilize in a *plateau* with detected face size of 64×80 pixels. Once faces are detected, an alignment is made using manually annotated face landmarks, such that the left eye l_{eye} and the right eye r_{eye} are at $l_{eye} = (\frac{w}{4}, \frac{h}{5})$ and $r_{eye} = (\frac{3w}{4}, \frac{h}{5})$ respectively. For the gabor wavelet parameters the best similarity measure between two Gabor Jets $\mathscr{J}$ and $\mathscr{J}'$ that presented the lowest error rates is the Phase Difference with the Canberra similarity which is defined as:

$$S(\mathscr{J}, \mathscr{J}') = \sum_j \left[\frac{a_j - a'_j}{a_j + a'_j} + \cos(\phi_j - \phi'_j - \vec{k}_j^T \vec{d}) \right]. \tag{3.1}$$

The Gabor Jets are placed in a grid at every six pixels in an image as we can see in Figure 3.1. For the wavelet, the maximum frequency k_{max} was set to $k_{max} = \pi$ with width $\sigma = 2\pi$.

Each baseline is implemented in the thesis software^2 and, once installed, can be triggered with a single command line. To trigger this baseline the following bash command should be typed.

```
$ bob bio htface htface_baseline gabor_graph <database>
```

3.1.2 Local Binary Patterns

The Local Binary Patterns system implemented in this work is an adaptation from [Ahonen et al., 2004]. Faces are detected, cropped and aligned to be with 200×250 pixels. Then, $LBP_{P=8,r=2}$ is computed in the aligned image for further block division of 32×32 pixels with 16 pixels of overlap at each direction. Finally, histogram for each block is computed and then

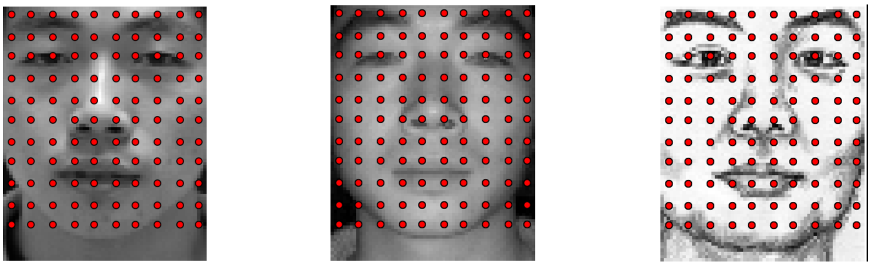

(a) VIS (b) NIR (c) Sketch

Figure 3.1 – Gabor Jets placed in different image modalities

concatenated.

This algorithm is non parametric, hence, no action is made at **training time**. At **enrollment time** the histogram is stored as is. Finally at **scoring time** the chi-square distance (χ^2) between two histograms is computed.

This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline htface_classic_lbp <database>
```

3.1.3 Local Gabor Binary Pattern Histograms

The mechanism around Local Gabor Binary Pattern Histograms [Zhang et al., 2005] was briefly introduced in Section 2.1.4. As before, just implementation details are presented.

In [Günther et al., 2017] the wavelet parameters, the dimensions of the detected face and the LBP parameters were carefully tuned. Hence, we will use these parameters in our work. Faces are detected and cropped in the same way as in 3.1.1 For the wavelets, the maximum frequency k_{max} is set to $k_{max} = \pi$ with width $\sigma = \sqrt{\pi}$. The $LBP_{P=8,r=2}$ with 8 sampling points with radius equals to 2 was selected and the LGBP processed images are split in 4 × 4 blocks with no overlap. Finally, for each block LBP histograms are computed and then concatenated forming a single 1d vector.

This algorithm is non parametric, hence, no action is made at **training time**. At **enrollment time** the computed histogram is stored as is. Finally at **scoring time** the histogram intersection between two histograms is computed.

This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline lgbphs <database>
```

Type	Filter Size/Stride, Pad	Output size
Input		$224 \times 224 \times 3$
Conv1$_{[1-3]}$	$3 \times 3/1,1$	$224 \times 224 \times 64$
Pool1	$2 \times 2/1$	$112 \times 112 \times 64$
Conv2$_{[1-2]}$	$3 \times 3/1,1$	$112 \times 112 \times 128$
Pool2	$2 \times 2/1$	$56 \times 56 \times 128$
Conv3$_{[1-3]}$	$3 \times 3/1,1$	$56 \times 56 \times 256$
Pool3	$2 \times 2/1$	$28 \times 28 \times 256$
Conv4$_{[1-3]}$	$3 \times 3/1,1$	$28 \times 28 \times 512$
Pool4	$2 \times 2/1$	$14 \times 14 \times 512$
Conv5$_{[1-3]}$	$3 \times 3/1,1$	$14 \times 14 \times 512$
Pool5	$2 \times 2/1$	$7 \times 7 \times 512$
fc6	$4,096$	$25,088 \times 4,096$
fc7	$4,096$	$4,096 \times 4,096$

Table 3.1 – The VGG16 architecture

3.1.4 Deep Convolutional Neural Networks

In this section it is described all approaches based on Deep Convolutional Neural Networks. This Section encompasses models either publicly available on the internet or trained in the context of this thesis.

VGG

Details about VGG networks was already introduced in 2.1.5. In this section just implementation details are discussed.

Parkhi et al. [2015] introduced a methodology for large scale data collection using web crawling and, with this data collected, a model called VGG16, whose description is on Table 3.1, was trained with input signals of size $224 \times 224 \times 3$. Such pre-trained model is available for download in their web page[2] and an it is integrated in this thesis software with the following bash command[2].

```
$ bob bio htface htface_baseline htface_vgg16 <database>
```

Light CNN

Wu et al. [2018] proposed an architecture that has ten times less free parameters than the VGG16-Face and claimed that it is naturally able to handle mislabeled data during its training (very common in datasets mined automatically). This is achieved through the usage of a newly introduced Max-Feature-Map (MFM) activation [3]. The Max-Feature-Map operator consists basically in the computation of the MAX between successives feature maps like in Figure 3.2. The input signal of such network are gray scaled images of 112×112 and its architecture is

[2] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

[3] This was implemented by myself in tensorflow https://github.com/tensorflow/tensorflow/pull/11824 and merged to the master branch

Type	Filter Size/Stride, Pad	Output size
Conv1	5 × 5/1,2	128 × 128 × 96
MFM1		128 × 128 × 48
Pool1	2 × 2/2	64 × 64 × 48
Conv2a	1 × 1/1	64 × 64 × 96
MFM2a		64 × 64 × 48
Conv2	3 × 3/1,1	64 × 64 × 192
MFM2		64 × 64 × 96
Pool2	2 × 2/2	32 × 32 × 96
Conv3a	1 × 1/1	32 × 32 × 192
MFM3a		32 × 32 × 96
Conv3	3 × 3/1,1	32 × 32 × 384
MFM3		32 × 32 × 192
Pool3	2 × 2/2	16 × 16 × 192
Conv4a	1 × 1/1	16 × 16 × 384
MFM4a		16 × 16 × 192
Conv4	3 × 3/1,1	16 × 16 × 256
MFM4		16 × 16 × 128
Conv5a	1 × 1/1	16 × 16 × 256
MFM5a		16 × 16 × 128
Conv5	3 × 3/1,1	16 × 16 × 256
MFM5		16 × 16 × 128
Pool4	2 × 2/2	8 × 8 × 128
fc1		512
MFM_fc1		256

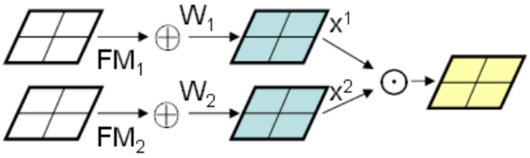
Table 3.2 – The Light CNN architecture

described in Table 3.2.

Although a version of such DCNN was trained in the context of this work, thanks to my contribution to the tensorflow stack, in this chapter, its pre-trained version provided by Wu et al. [2018] is used[4].

This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline htface_lightcnn <database>
```



;

Figure 3.2 – Max-Feature-Map (MFM) activate, where $h(x) = max(x^1, x^2)$

Inception Resnet v1 and Inception Resnet v2

The Inception Resnet v1 and Inception Resnet v2 are implemented and trained in the context of this thesis. Those are the closest open-source implementation of the model proposed in

[4] https://drive.google.com/file/d/0ByNaVHFekDPRMGlLWVBhbkVGVm8/view

[Schroff et al., 2015], where neither training data or source code were made available and it is inspired by the implementation of Szegedy et al. [2017]. An schematic of both architectures can be seen in Figure 3.11 (end of the chapter).

Each one of these DCNNs are trained using VIS images gray scaled and RGB. Hence, four different DCNNs are trained. The number of possible permutations of the hyperparameters to train such DCNNs can take is substantially big. For instance, the batch size, optimizer, regularization parameters, drop out, learning rate strategy, parameters of the convolution, parameters of the pooling, inception layers setup, number of residual connections, number of data augmentation parameters, loss function and many other things. In this thesis it is not hypothesize anything with respect to that, instead, the **same** recipes used in Szegedy et al. [2017] are followed, which presents very high recognition rates in LFW dataset.

In this work the MS-Celeb dataset is used. Such dataset contains a substantial amount of mislabeling. Hence, in the context of this thesis, this dataset was pruned in a semi-automatic manner and the result of this pruning is published here [5]. Faces are detected, cropped, aligned and stored using the MTCNN face detector [Zhang et al., 2016]. This face detector is also integrated in this software thesis[6]. The outcome of this pruning resulted in a dataset of 8M samples with $87,662$ identities.

The RMSProp optimizer is used as a solver[7] with mini-batches of 90 samples. The learning rate is kept to 0.1 for 65 epochs. Then, it is decreased to 0.01 for 15 epochs and finally decreased once more to 0.001 until the end of the training. In total all the DCNNs are trained for 250 epochs.

The embeddings of these four DCNNs are 128d and to train them they were fed into a hot-encoded fully connected layer with $87,662$ outputs. The weight sum between the center and cross entropy loss proposed by Wen et al. [2016] (see Equation (5) in the paper) is used as loss function.

The assessment on how those DCNNs perform in different large scale VIS image databases can be found in the Appendix B.

This baseline is integrated in the thesis software and can be triggered with the following command:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_rgb <database>
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray <database>
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_rgb <database>
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_gray <database>
```

[5] http://gitlab.idiap.ch/tiago.pereira/bob.db.msceleb

[6] https://gitlab.idiap.ch/bob/bob.ip.mtcnn

[7] tensorflow.org/api_docs/python/tf/train/RMSPropOptimizer

DCNN for Face Recognition

The FR task using the systems described in Sections 3.1.4, 3.1.4 and 3.1.4 is approached using their embeddings as described in chapter 2.1.5. Those DCNNs are already trained, hence, no action is made at **training time**. At **enrollment time** the embeddings are stored as is. Finally at **scoring time**, the cosine similarity is applied as score. Given two arbitrary embeddings x_e and x_p, such metric is defined as the following:

$$s(x_e, x_p) = \frac{x_e \cdot x_p}{||x_e||||x_p||} \tag{3.2}$$

3.2 Heterogeneous Face Recognition baselines

In this Section it is described the baselines that are implemented in this thesis. The baselines either consists in source code that is integrated in the software thesis or that is implemented by extracting the informations on the corresponding papers.

3.2.1 Heterogeneous face recognition from local structures of normalized appearance

This section describes the details of version of the work proposed by Liao et al. [2009]. Focused in the task of VIS to NIR, the authors hypothesized that differences between VIS and NIR modalities can be suppressed using Difference-of-Gaussian (DoG). The DoG filter consists in the subtraction of two Gaussian convolved images. Given a 1d signal I the DoG output can be defined as:

$$DoG_{\sigma_1,\sigma_2}(x) = I * \frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x^2)/2\sigma_1^2} - I * \frac{1}{\sigma_2\sqrt{2\pi}} e^{-(x^2)/2\sigma_2^2}, \tag{3.3}$$

where $\sigma_{1,2}$ are the standard deviation of each Gaussian. The outcome of this normalization can be seen in Figure 2.19. For this work $\sigma_1 = 1$ and $\sigma_2 = 2$.

In this work, images are cropped to 120 × 120 and Local Binary Patterns with 8 sampling points and radius equals to 2, $LBP_{8,2}$ is used as feature descriptor, hence the same pattern is set. The way that block division was made is not described in the paper. Hence, a fine tuning using the CASIA-NIR-VIS 2.0 database is carried out varying the block size from 8 × 8 to 64 × 64 pixels. A good trade-off between error rate and dimensionality of the feature vector was found with blocks with 32 pixels. The classification is carried out using FLD (see Section 2.1.2).

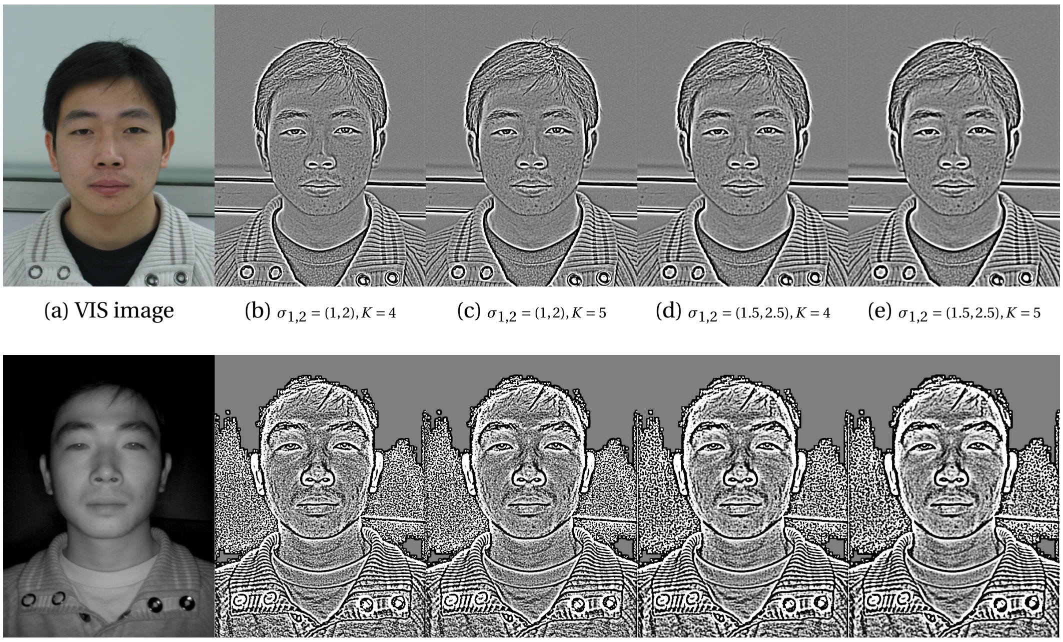
This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline htface_mlbphs <database>
```

3.2.2 Heterogeneous face image matching using multi-scale features

In this section it is described the details of our version of the work proposed by Liu et al. [2012], which is also focused on the task of VIS to NIR HFR. In this work the authors hypothesized that independent features between VIS and NIR are embedded in a particular range of frequency bands and this can be approached also via Difference-of-Gaussian filter. There is no information about the number of DoG filters used and how each convolutional filter is set. In this thesis it is selected a range of 3 different values for the pair $\sigma_{1,2}$, respectively $\sigma_1 = [1, 1.5, 2]$ and $\sigma_2 = [2, 2.5, 3]$ and two values for the kernel size (patched of 3×3 and 4×4).

In Figure 3.3 it is possible to observe the selected setup in the different image modalities.



(a) VIS image (b) $\sigma_{1,2} = (1,2), K = 4$ (c) $\sigma_{1,2} = (1,2), K = 5$ (d) $\sigma_{1,2} = (1.5,2.5), K = 4$ (e) $\sigma_{1,2} = (1.5,2.5), K = 5$

(f) NIR image (g) $\sigma_{1,2} = (1,2), K = 4$ (h) $\sigma_{1,2} = (1,2), K = 5$ (i) $\sigma_{1,2} = (1.5,2.5), K = 4$ (j) $\sigma_{1,2} = (1.5,2.5), K = 5$

Figure 3.3 – Difference-of-Gaussians filter crafted under different values for $\sigma_{1,2}$ and different kernel scales K

In this thesis, images are cropped to 120 × 120 pixels and a combination of HOG and MLBP features are used. A combination of PCA and FLD is used in the classification stage.

This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline htface_multiscale_features <database>
```

3.2.3 Geodesic Flow Kernel

The Geodesic Flow Kernel (GFK) proposed by Gong et al. [2012] explicitly models the source and target domain in individual d-dimensional linear subspaces and then embeds them onto a Grassmann manifold. A Grassmann manifold $G(d,D)$ is the collection of all d-dimensional subspaces of the feature vector space $\mathbb{R}^D$. Given two arbitrary linear subspaces $P_s, P_t \in \mathbb{R}^{D\times d}$ (which are data points into a Grassmann manifold), the GFK approach explicitly construct an infinite-dimensional feature space $\phi(t)$ that maps those two subspaces. Features from both image modalities are then projected into these subspaces forming a feature vector of infinite dimensions:

$$z^{\infty} = \phi(t)^{\intercal} x : t \in \left[0,1\right], \tag{3.4}$$

where $\phi(0) = P_s$ and $\phi(1) = P_t$. For other values of t:

$$\phi(t) = P_s U_1 \Gamma(t) - R_s U_2 \Sigma(t), \tag{3.5}$$

where $R_s \in \mathbb{R}^{D\times(D-d)}$ denotes the orthogonal complement to P_s with $R_s^{\intercal} P_s = 0$ (a.k.a null space), $U_1 \in \mathbb{R}^{d\times d}$ and $U_2 \in \mathbb{R}^{(D-d)\times d}$ are orthonormal matrices that are given by the following pair of SVDs:

$$P_s^{\intercal} P_t = U_1 \Gamma V^{\intercal}, R_s^{\intercal} P_t = -U_2 \Sigma V^{\intercal} \tag{3.6}$$

Using this new representation forces classifiers to use domain invariant features. Given two samples x_s and x_t from both source and target domains, the infinite-dimensional feature vector is handled conveniently by their inner product that gives rise to a positive semi-definite kernel defined on the original features:

$$x_s^{\infty} \cdot x_t^{\infty} = x_s \int_0^1 \phi(t)\phi(t)^{\intercal} x_t dt = x_s^{\intercal} G x_t \tag{3.7}$$

G can be computed efficiently using generalized singular value decomposition[8].

This strategy was implemented in the context of this thesis and it is the only Python-C++ implementation available[9].

Any type of crafted feature can be used to compose P_s and P_t. In this thesis, the absolute values of Gabor Jets (see section 3.1.1) are used, which was the same strategy implemented in [Sequeira et al., 2017]. Then P_s and P_t are defined as basis of PCA (see chapter 2.1.1) linear subspace.

At **training time**, P_s, P_t and G are estimated. At **enrollment time**, Gabor jets are computed and stored as is. Finally, at **scoring time**, the absolute values of a pair of Gabor Jets ($\mathscr{J}$ and

[8] https://www.idiap.ch/software/bob/docs/bob/bob.math/stable/py_api.html#bob.math.gsvd

[9] https://www.idiap.ch/software/bob/docs/bob/bob.learn.linear/stable/py_api.html#bob.learn.linear.GFKTrainer

$\mathscr{J}'$) are compared via kernalized dot product as the following:

$$S(\mathscr{J},\mathscr{J}') = \frac{\sum_{n=1}^{N} \mathscr{J}_n \cdot G \cdot \mathscr{J}'_n}{N}. \tag{3.8}$$

This baseline can be triggered with the following bash command[2].

```
$ bob bio htface htface_baseline htface_gfkgabor <database>
```

3.3 Experiments and Analysis

In this section recognition rates assessment of FR Baselines and HFR Baselines, either implemented in the context of this work or directly depicted in publications, are presented. To make easier the interpretation of the recognition rates, all the tables in this section (Tables 3.3, 3.4, 3.5 and 3.7) are split in three parts. **FR Baselines** corresponds to all FR baselines described in the Section 3.1. **Reproducible Baselines** corresponds to all HFR baselines described in the Section 3.2 and it was implemented or integrated in the context of this work. Finally, **Non Reproducible Baselines** corresponds to HFR baselines whose source code was not made publicly available and its average rank one recognition rate was picked directly from its corresponding publication.

3.3.1 Visible Light to Sketches

In this subsection it is described experiments with two sketch databases: CUHK-CUFS and CUHK-CUFSF. Table 3.3 presents the average rank one recognition rate for each face recognition baseline using those databases.

Sketches are basically composed by shapes and, because of that, have lots of high frequency components. Moreover, all the texture from one sketch comes from the texture either from paper where the sketch was drawn which is the case for the CUHK-CUFS and CUHK-CUFSF databases. Hence, it is reasonable to hypothesize that all the tested **FR Baselines** are not suitable for VIS-Sketch task.

Experiments carried out with CUHK-CUFS database demonstrates that the aforementioned hypothesis can't be confirmed for the FR Baselines, which present an average rank one recognition rates way above a hypothetical random classifier. For instance, the FR systems based on Gabor Graph and LGBPHS, present the highest average rank one recognition rates, respectively 81.29% and 92.97%. Those baselines present higher recognition rates than two Reproducible HFR baselines; MLBP baseline, introduced by [Liao et al., 2009], presents an average rank one recognition rate of 62.27% and the MultiScale features introduced by [Liu et al., 2012] presents an average rank one recognition rate of 64.16%. The GFK presents an average rank one recognition rate of 93.27%. Finally, the FR baselines based on DCNNs vary from ≈ 70% to ≈ 80% and the best one is **Incep. Resnet v2** with 80.29%.

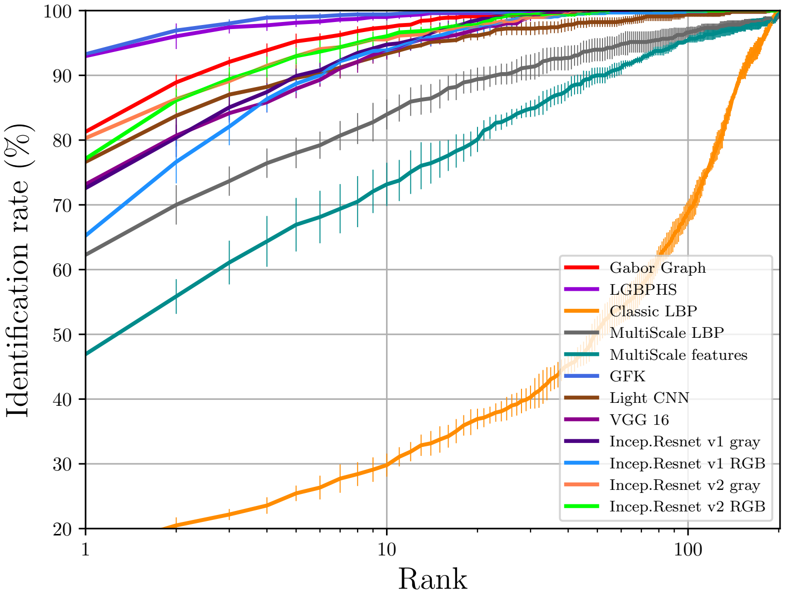


Figure 3.4 – CUHK-CUFS Baselines - Average CMC curves (with error bars)

Figure 3.4 presents the average CMC for all the baselines that was possible to be executed with their respective standard deviations. It is possible to observe that the system based on LGBHS presents an average rank 10 recognition rate above 98%, which is surprising since this system has no knowledge about how to represent sketches. Same trends can be observed for other FR systems, where their average rank 10 recognition rate are also increased. However, those baselines are not better the state-of-the-art published by Klare and Jain [2013] which presents an average rank one recognition rate of 99%.

Experiments carried out with CUHK-CUFSF shows a different reality if compared with CUHK-CUFS. It is possible to observe that the best FR Baseline for CUHK-CUFS (LGBPHS) presents an average rank one recognition rate of 25.38% on CUHK-CUFS. The best DCNN FR Baselines is the VGG 16 with an average rank one recognition rate of 32.99%. Other DCNN FR Baselines present similar performance using the same figure of merit. Among the Reproducible HFR baselines, the best one is GFK that presents an average rank one recognition rate of 41.01%. It is possible to observe, that despite the fact such DCNNs don't have any prior knowledge about the target modality ($\mathscr{D}^t$), the feature detectors of such models are still able to detect discriminant features in of all them (above a hypothetical random classifier). As before, those recognition rates are lower than the state-of-the-art recognition rates for the CUHK-CUFSF database (which consider a joint modeling of both $\mathscr{D}^s$ and $\mathscr{D}^t$). The state-of-the art in this database is the one implemented by Galea [2018]. The DEEPs, system, which is based on DCNNs, presents an average rank one recognition rate of 82.92%.

Table 3.3 – VIS to Sketches - Average rank one recognition rate under different Face Recognition CNN systems.

#	FR Algorithm	CUHK-CUFS	CUHK-CUFSF
FR Baselines			
1	Gabor-Graph	81.29%(2.4)	19.39%(1.0)
2	LGBPHS	92.97%(2.2)	25.38%(1.5)
3	LBP	16.33%(1.9)	6.23%(1.8)
4	Light CNN	76.63%(2.9)	25.87%(1.5)
5	VGG 16	73.17%(1.6)	32.99%(1.1)
6	Incep. Res. v1 - gray scaled	72.57%(3.7)	24.49%(0.5)
7	Incep. Res. v1 - RGB	65.24%(4.7)	20.93%(1.2)
8	Incep. Res. v2 - gray scaled	80.29%(1.5)	29.51%(0.7)
9	Incep. Res. v2 - RGB	77.13%(3.2)	31.05%(1.4)
Reproducible Baselines			
10	MLBP [Liao et al., 2009]	62.27%(3.8)	9.11%(1.7)
11	MultiScale feat. [Liu et al., 2012]	64.16%(2.5)	6.76%(0.7)
12	GFK [Gong et al., 2012; Sequeira et al., 2017]	93.27%(1.4)	41.01%(1.8)
Non Reproducible Baselines			
13	P-RS as in [Klare and Jain, 2013]	99%(n/a)	-
14	TP-LBP [Wolf et al., 2008]	-	59.7%(n/a)
15	CDFL Jin et al. [2015]	-	81.3%(n/a)
16	DEEPS [Galea, 2018]	-	82.92%(1.3)
17	LGMS [Galea, 2018]	-	78.19%(0.5)
18	Face VACS in [Klare and Jain, 2013]	89%(n/a)	

Figure 3.5 presents the average CMC for all baselines that was possible to execute with their respective standard deviations. It is possible to observe that for our best tested system (VGG 16), average rank 10 is ≈ 65% and average rank 100 is ≈ 90% which is still lower than the state-of-the-art for this database (using rank one as a reference).

Considering the FR Baselines, there is a big gap, in terms of average rank one recognition rate, between CUHK-CUFS and CUHK-CUFSF. This could be explained by the realism and lack of distortions of the CUHK-CUFS sketches. With respect to shape, the pairs photos-sketches from this dataset are quite realistic as it can be observed in the Figure 3.6. Details such as expression, proportion of the face and volume of the hair are presented in both image domains. This realism is not presented in the CUHK-CUFSF database and the FR Baselines can't model such within class variability properly.

3.3.2 Visible Light to Near Infrared

This subsection describes experiments on four different image databases: CASIA, NIVL, FARGO and LDHF (see section 2.3.1). Table 3.4 presents the average rank one recognition rate for each face recognition baseline which uses this benchmark as a reference.

Experiments using hand-crafted features presented the lowest recognition rates. For instance,

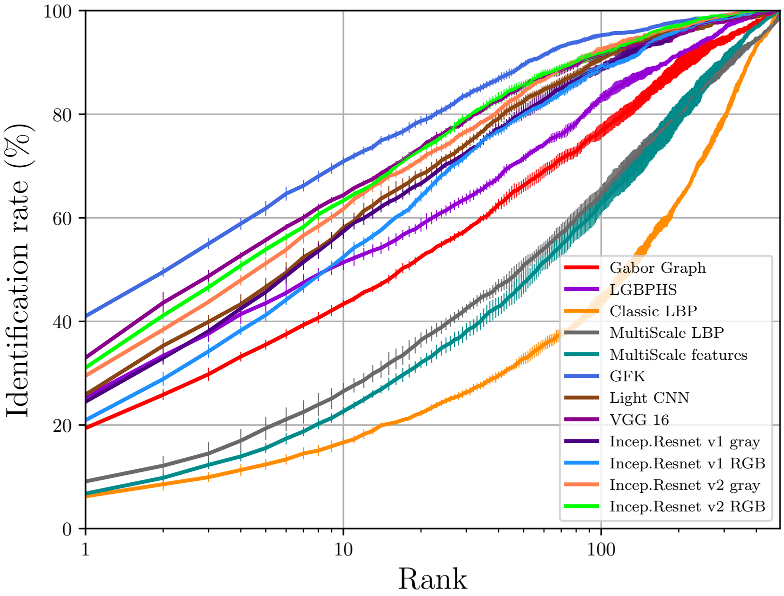


Figure 3.5 – CUHK-CUFSF Baselines - Average CMC curves (with error bars)

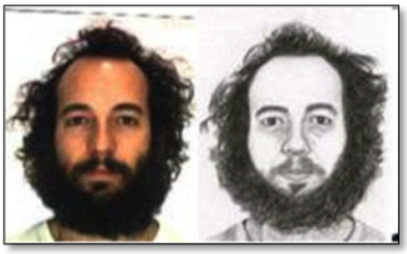


Figure 3.6 – Realism of CUHK-CUFS database

experiments using the FR systems based on Gabor wavelets, such as Gabor Graph and LGBPHS, present an average rank one recognition rate of 16.41% and 30.98% using the NIVL dataset. Using CASIA dataset as a reference, the Gabor Graph and LGBPHS FR systems present an average rank one recognition rate of 21.49% and 22.24% respectively; for the LDHF it is achieved 21.8%(1.4) and 34.9% respectively. The FR system based on Local Binary Patterns presents an average rank one recognition rate of 3.37%, 14.56% and 13.4% for the databases CASIA, NIVL and LDHF respectively.

Among the Reproducible HFR Baselines, the MultiScale features from [Liu et al., 2012] presents higher rank one recognition rate for the CASIA and LDHF databases with respectively 70.33% and 26.6%. For the NIVL, the MLBP proposed by [Liao et al., 2009] presents 90.34% using the same figure of merit.

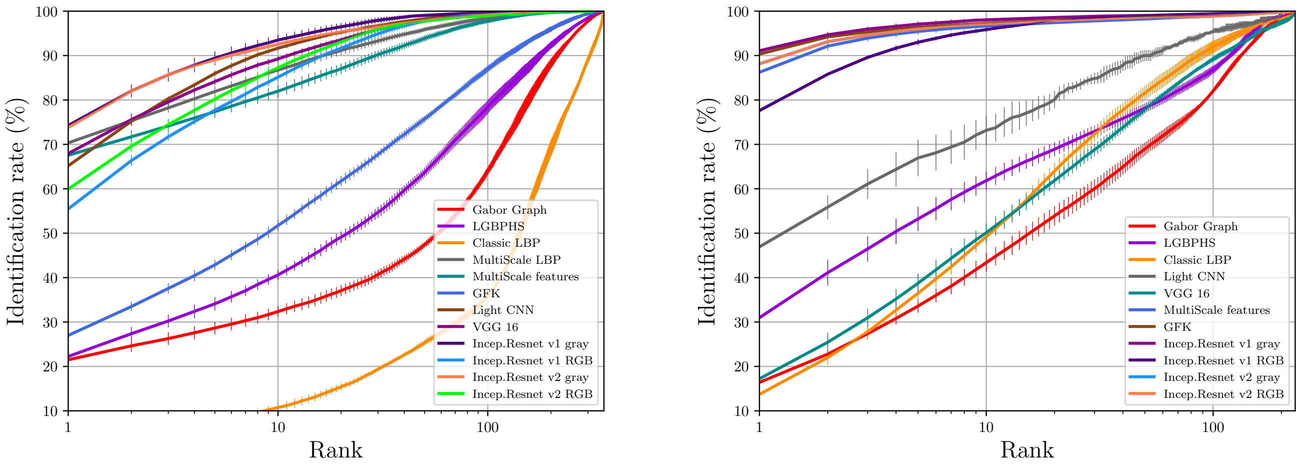
FR systems based on DCNNs present the highest average rank one recognition rates. Surprisingly, for some DCNNs, such benchmarks are better than some marked as **Non Reproducible Baselines** (see Table 3.4). The Light CNN presents an average rank one recognition rate of 65.17%, 86.24% and 41.7% for the databases CASIA, NIVL and LDHF respectively. The VGG16 follows the same trend with 67.92%, 90.34% and 70.4% for the same databases respectively. The recent Inception Resnet DCNNs present the highest average rank one recognition rates. For the CASIA database the best systems are the Incep. Res. v1 and Incep. Res. v2 using RGB inputs with 74.25% and 73.80% respectively. For the NIVL database the best systems are the Incep. Res. v2 using RGB and gray scaled images as input with 91.09% and 88.14% respectively. Finally for the LDHF the best systems is the Incep. Res. v2 - RGB with 53.8%.

Table 3.4 – VIS to NIR - Average rank one recognition rate under different Face Recognition systems

#	**FR Algorithm**	**CASIA**	**NIVL**	**LDHF**
FR Baselines				
1	Gabor-Graph	21.49%(1.1)	16.41%(0.9)	21.8%(1.4)
2	LGBPHS	22.24%(1.6)	30.98%(3.3)	34.9%(1.7)
3	LBP	3.68% (0.6)	13.72%(1.5)	13.40%(2.1)
4	Light CNN	65.17%(0.6)	86.24%(1.4)	41.7%(3.3)
5	VGG 16	67.92%(1.4)	90.34%(1.3)	70.4%(2.3)
6	Incep. Res. v1 - gray	74.25%(1.3)	91.09%(0.3)	51.5%(1.2)
7	Incep. Res. v1 - RGB	55.46%(1.4)	77.61%(0.8)	45.1%(1.5)
8	Incep. Res. v2 - gray	73.80%(1.2)	88.14%(0.6)	45.2%(0.9)
9	Incep. Res. v2 - RGB	60.01%(1.7)	86.06%(0.7)	53.8%(0.9)
Reproducible Baselines				
10	MultiScale feat. [Liu et al., 2012]	70.33%(1.2)	85.35%(1.1)	26.6%(2.4)
11	MLBP [Liao et al., 2009]	67.54%(1.7)	90.34%(1.3)	22.1%(2.9)
12	GFK [Gong et al., 2012; Sequeira et al., 2017]	26.98%(0.9)	63.08%(2.2)	29.9%(4.4)
Non Reproducible Baselines				
13	IDR in [He et al., 2017]	95.82%(0.7)	-	-
14	CDL in [Wu et al., 2017]	98.62%(0.2)	-	-
15	WCNN in [He et al., 2018]	98.70%(0.3)	-	-
16	DSIFT in [Dhamecha et al., 2014] (Table II)	73.28%(1.1)	-	-
17	FaceVACS in [Dhamecha et al., 2014](Table I)	58.56%(1.2)	-	-
18	Gabor+RBM [Jin et al., 2015] (Table I)	86.1% (0.1)	-	-
19	PCA+SYM+HCA [Li et al., 2013]	23.7% (1.9)	-	-
20	CDFL [Jin et al., 2015](Table I)	71.5% (1.4)	-	-
21	TRIVET in [Liu et al., 2016]	95.74%(0.5)	-	-

Figure 3.7 presents the average CMC curves for the CASIA and NIVL databases. The observation of this benchmark corroborates with the observations made in the Table 3.4, that DCNN baselines presents the highest recognition rates in these tests, even for rank equals to 10 and 100. For the NIVL specially the Incep. Res v2 gray achieves an average rank 10 recognition rate of 100%. In this dataset VIS and NIR images are both high resolution and close-ups as is can be observed in Figure 3.8 and this possibly is playing an important role in the recognition rates. For the CASIA, however, such benchmarks are not better than the ones published and

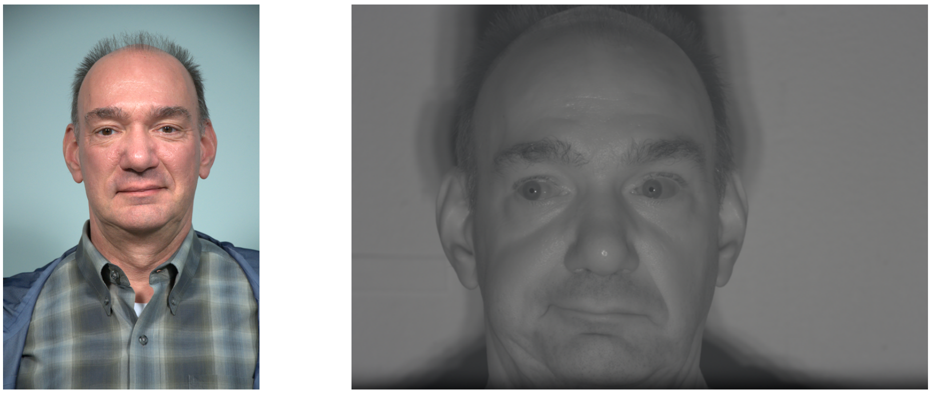
considered the state-of-the-art. For instance the recent WCNN proposed by He et al. [2018] presents an average rank one recognition rate of 98.70%.



(a) Casia

(b) NIVL

Figure 3.7 – VIS to NIR Baselines - Average CMC curves (with error bars)



(a) VIS

(b) NIR

Figure 3.8 – VIS and NIR images from NIVL dataset

The dataset LDHF was designed to approach the problem of surveillance in nighttime. Thus, Kang et al. [2014] collected NIR images in four different distances in indoor and outdoor set ups. Indoor acquisitions were taken from 1m; the outdoors were taken at nighttime from 60m, 100m and 150m (see 2.3.1 for more details). Table 3.4 presents the average rank one recognition rates summarized; to assess the recognition rate under different standoffs, Table 3.5 presents them for each distance in isolation. From this table it is possible to observe the same trends as before; FR systems based on DCNNs presents the highest recognition rates. Moreover, recognition rates steadily decreases once probe images are taken from further distances. For instance, using **Incep. Res. v1 - gray** as reference, the average rank one recognition rate varies from 94.8% to 4.8% with probe images taken from 1m and to 150m respectively.

From Figure 2.27 it is possible to observe severe differences in resolution between 1m and 150m standoffs. As a matter of fact, using the MTCNN[10] face detector the average detected faces from 1m stand-off is 738 × 897 pixels and from 150m stand-off is 60 × 60 pixels. It is possible to suggest that the up-scaling distortions are affecting the effectiveness of the FR Baselines. The same trend is observed for the CASIA dataset where the stand-offs are more unconstrained.

Table 3.5 – LDHF average rank one recognition rates under different standoffs

#	FR Algorithm	1m	60m	100m	150m
FR Baselines					
1	Gabor-Graph	54.80%(3.7)	15.6(1.497)	15.2(3.487)	1.6(1.96)
2	LGBPHS	72.4%(4.3)	32.0%(2.9)	26.0%(3.6)	9.2%(3.2)
3	LBP	34.0%(3.3)	7.2%(2.0)	7.6%(3.2)	4.8%(1.6)
4	Light CNN	77.2%(4.5)	54.4%(6.1)	30.4%(6.4)	4.8%(1.0)
5	VGG 16	98.8%(1.6)	91.2%(2.0)	67.6%(5.5)	24.0%(3.3)
6	Incep. Res. v1 - gray	94.8%(2.0)	78.0%(4.4)	28.4%(1.5)	4.8%(1.6)
7	Incep. Res. v1 - RGB	82.4%(2.6)	60.8%(6.5)	30.4%(3.4)	6.8%(2.4)
8	Incep. Res. v2 - gray	92.8%(2.7)	75.6%(2.9)	9.6%(1.5)	2.8%(1.6)
9	Incep. Res. v2 - RGB	90.4%(1.5)	75.2%(2.7)	41.2%(3.0)	8.4%(1.5)
Reproducible Baselines					
10	MLBP [Liao et al., 2009]	67.2%(7.0)	23.2%(3.0)	10.0%(2.8)	6.0%(1.789)
11	Multiscale Feat. [Liu et al., 2012]	74.4%(3.4)	43.2%(3.7)	22.0%(4.5)	14.8%(3.0)
12	GFK [Gong et al., 2012; Sequeira et al., 2017]	73.6%(4.3)	31.2%(7.2)	12.0%(2.8)	2.8%(3.0)

FARGO database was designed to assess the task of heterogeneous face verification (see Figure 1.1) under different illumination conditions (controlled, dark and outdoor). Hence, a specific set of protocols to assess verification recognition rates were designed and specific set of metrics were defined. In this work we reproduce the same metrics used in [Heusch et al., 2019], where error rates are assessed using Detection Error Trade-off (DET) curves. As scalar reference, it is used the False Non Match Rate (FNMR) at **False Match Rate (FMR) of 1%** (see chapter 2.4).

Table 5.8 presents the FNMR@FMR=1%(dev) for all FR Baselines and Reproducible Baselines. The same trend observed for the other three databases can be observed in this database. Under controlled conditions (mc), whose setup is similar to the NIVL dataset and LDHF dataset (1m stand-off), the DCNNs perform better than the FR Baselines based on crafted features. FR baselines based on Gabor Wavelets, such as Gabor Graphs and LGBPHS, presents very high error rates; FNMR of 57.20% and 45.80% in the evaluation set respectively. For the DCNN baselines, the best ones are the ones based on Incep. Res. v1 and Incep. Res. v2, both using gray scaled images which achieved an FNMR of 2.80% and 4.40%. Light CNN and VGG

[10] http://gitlab.idiap.ch/bob/bob.ip.mtcnn

16 achieve both 26.60% and 12.40% respectively. The reproducible baselines, surprisingly presents higher error rates than the DCNN ones. MLPB and MultiScaled features present a FNMR of 81.40% and 88.60%.

Table 3.6 – Fargo database - FNMR@FMR=1%(dev) taken from the development set

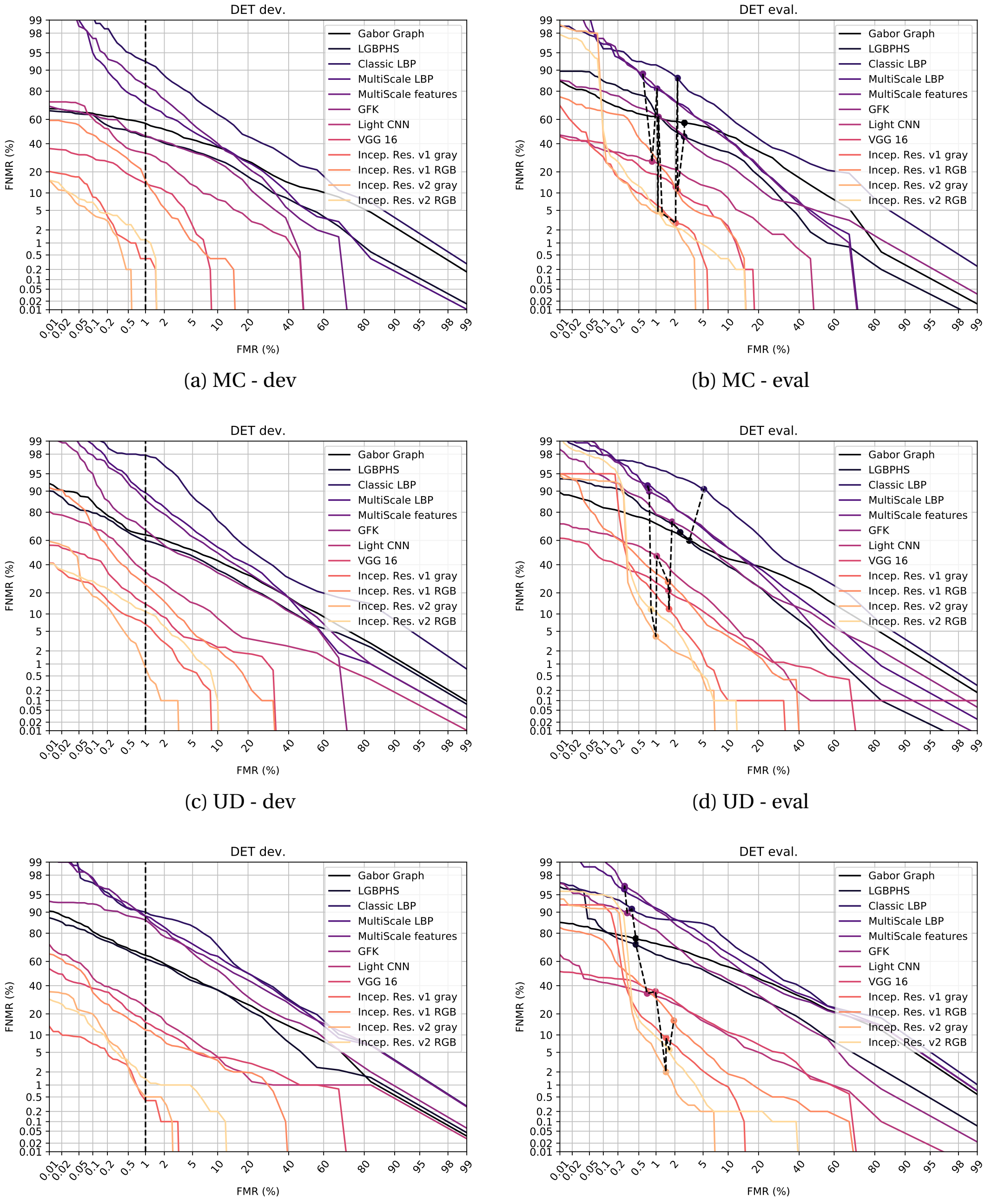
#	FR Algorithm	mc		ud		uo	
		dev	eval	dev	eval	dev	eval
FR Baselines							
1	Gabor-Graph	56.80	57.20	64.40	59.90	64.80	76.80
2	LGBPHS	45.80	45.80	59.80	66.40	62.00	72.80
3	LBP	92.80	86.80	97.90	90.70	90.00	91.10
4	Light CNN	32.60	26.60	34.30	47.10	24.00	33.90
5	VGG 16	14.00	12.40	14.10	21.10	15.40	35.40
6	Incep. Res. v1 - gray scaled	0.40	2.80	6.70	11.90	0.40	9.00
7	Incep. Res. v1 - RGB	15.40	10.80	25.10	27.00	11.90	16.30
8	Incep. Res. v2 - gray scaled	0.00	4.40	0.80	4.00	0.50	2.00
9	Incep. Res. v2 - RGB	1.20	4.80	10.90	11.80	1.40	5.60
Reproducible Baselines							
10	MultiScale feat. [Liu et al., 2012]	83.40	88.60	86.30	89.90	88.40	96.60
11	MLBP [Liao et al., 2009]	71.80	81.40	89.40	91.90	88.50	96.10
12	GFK [Gong et al., 2012; Sequeira et al., 2017]	46.20	62.00	68.00	74.00	86.80	89.70

For the dark acquisitions protocol (ud), the same trends are observed, with the DCNN presenting the lowest error rates. FR baselines based on Gabor Wavelets, such as Gabor Graphs and LGBPHS, presents very high error rates; FNMR of 59.90% and 66.40% in the evaluation set. For the DCNN baselines, the best ones are the ones based on Incep. Res. v1 and Incep. Res. v2, both using gray scaled images which achieves an FNMR of 11.90% and 4.00%. Light CNN and VGG 16 achieve both 47.10% and 21.10% respectively. Finally, MLPBs and MultiScaled features (Reproducible HFR Baselines) presents FNMR of 89.90% and 91.90%. The GFK HFR Baselines presents an average rank one recognition rate of 74.00%.

In the outside acquisitions protocol (uo), the same trends are observed, with the DCNN presenting the lowest error rates. FR baselines based on Gabor Wavelets, such as Gabor Graphs and LGBPHS, presented very high error rate; FNMR of 76.80% and 72.80% in the evaluation set respectively. For the DCNN baselines, the best ones are the ones based on Incep. Res. v1 and Incep. Res. v2. Both gray scaled DCNNs achieve an FNMR of 9.80% and 2.00% respectively. Light CNN and VGG 16 achieved both 33.90% and 35.40% respectively. MLPBs and MutiScale features presented FNMR of 96.60% and 96.10%. The GFK HFR Baselines presents an average rank one recognition rate of 89.70%. Figure 3.9 presents the DET plots in the development set and evaluation set for all the three illumination conditions.

3.3.3 Visible Light to Thermograms

In this subsection it is described experiments using two different databases, both subsets of the Pola Thermal database (see section 2.3.3). Table 3.7 presents the average rank one recognition rate for each face recognition baseline which uses this benchmark as a reference.



(a) MC - dev

(b) MC - eval

(c) UD - dev

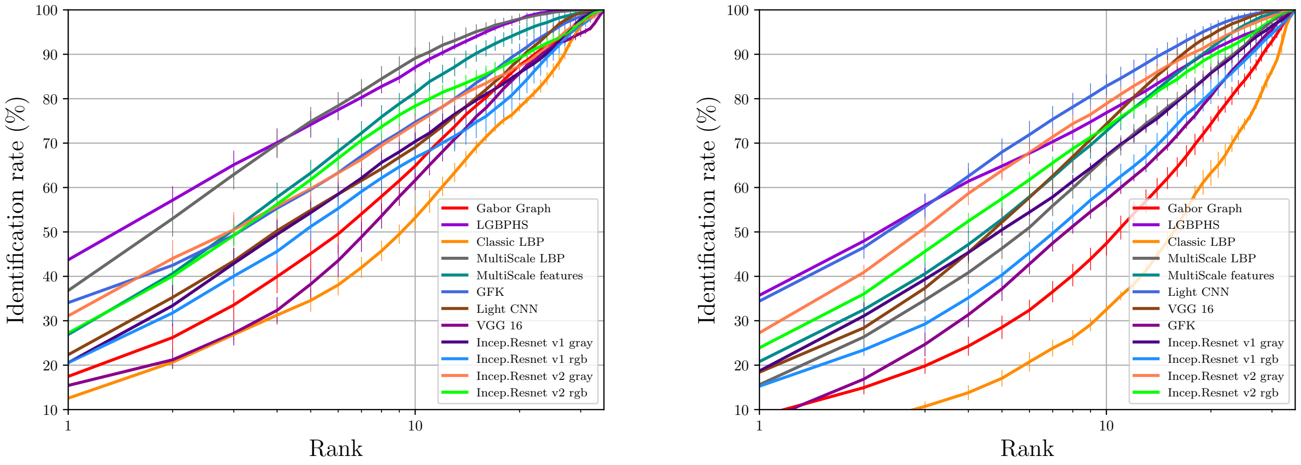
(d) UD - eval

(e) UO - dev

(f) UO - eval

Figure 3.9 – DET curves for the FARGO database verification experiments under the three illumination conditions MC (controlled), UD (dark) and UO (outdoor). The column on the left presents DET curves for the development set and the columns on the right presents DET curves for the evaluation set.

If compared with other image modalities, a different trend can be observed in the two experimented databases. The DCNN feature detectors don't present the highest recognition rates.



(a) Thermal

(b) Pola Thermal

Figure 3.10 – VIS to Thermogram Baselines - Average CMC curves (with error bars)

For the **Thermal** dataset, experiments using the FR systems based on Gabor wavelets, such as Gabor Graph and LGBPHS, presented an average rank one recognition rate of 17.46% and 43.71%. On the other hand the best DCNN baseline, **Incep. Res. v2 - RGB** presents an average rank one recognition rate of 31.09%. Among the reproducible baselines, the **MLBP** presents the highest recognition rates with 36.80%.

Same trend can be observed for the Pola Thermal dataset, experiments using the FR systems based on Gabor wavelets, Gabor Graph and LGBPHS, presented an average rank one recognition rate of 8.46% and 35.73%. The best DCNN baseline is again the Inception Resnet v2, but with gray scaled inputs. Such DCNN presented an average rank one recognition rate of 27.29%. Among the Reproducible HFR Baselines, the **MultiScaled features** presented the highest recognition rates with 20.81%.

All the presented FR baselines presented way lower recognition rates than the state-of-the art Figure 3.10 shows the CMC curve for all the FR and Reproducible HFR baselines. It's possible to observe that even for rank 10 our FR baselines are not able to achieve the same recognition rate as in the rank one or the Paper HFR baselines.

3.4 Discussion

In this chapter it was assessed the effectiveness of different FR systems (FR Baselines) in several databases split in three different image modalities, each one with its idiosyncrasies and some trends could be observed. In general, it was possible to observe that in all image modalitites the FR systems presented recognition rates higher than an hypothetical random

Table 3.7 – VIS to Thermograms - Average rank one recognition rate under different Face Recognition systems.

#	FR Algorithm	Thermal	Pola Thermal
FR Baselines			
1	Gabor-Graph	17.46%(1.9)	8.46%(1.1)
2	LGBPHS	43.71%(3.7)	35.73%(1.8)
3	LBP	12.56%(1.6)	3.64%(1.0)
4	Light CNN	22.35%(3.6)	18.42%(1.7)
5	VGG 16	15.42%(2.6)	7.12%(1.8)
6	Incep. Res. v1 - gray scaled	20.55%(4.2)	18.69%(2.1)
7	Incep. Res. v1 - RGB	20.55%(2.0)	15.26%(1.2)
8	Incep. Res. v2 - gray scaled	31.09%(4.1)	27.29%(0.8)
9	Incep. Res. v2 - RGB	27.21%(1.4)	23.91%(1.2)
Reproducible Baselines			
10	MLBP [Liao et al., 2009]	36.80%(3.5)	15.61%(2.9)
11	Multiscale Feat. [Liu et al., 2012]	26.89%(3.5)	20.81(3.4)
12	GFK [Gong et al., 2012; Sequeira et al., 2017]	34.07%(2.9)	26.17%(2.5)
Non Reproducible Baselines			
13	PLS [Hu et al., 2016]	53.05% (n/a)	58.67% (n/a)
14	DPM [Hu et al., 2016]	75.31% (n/a)	80.54% (n/a)
15	CpNN [Hu et al., 2016]	78.72% (n/a)	82.90% (n/a)

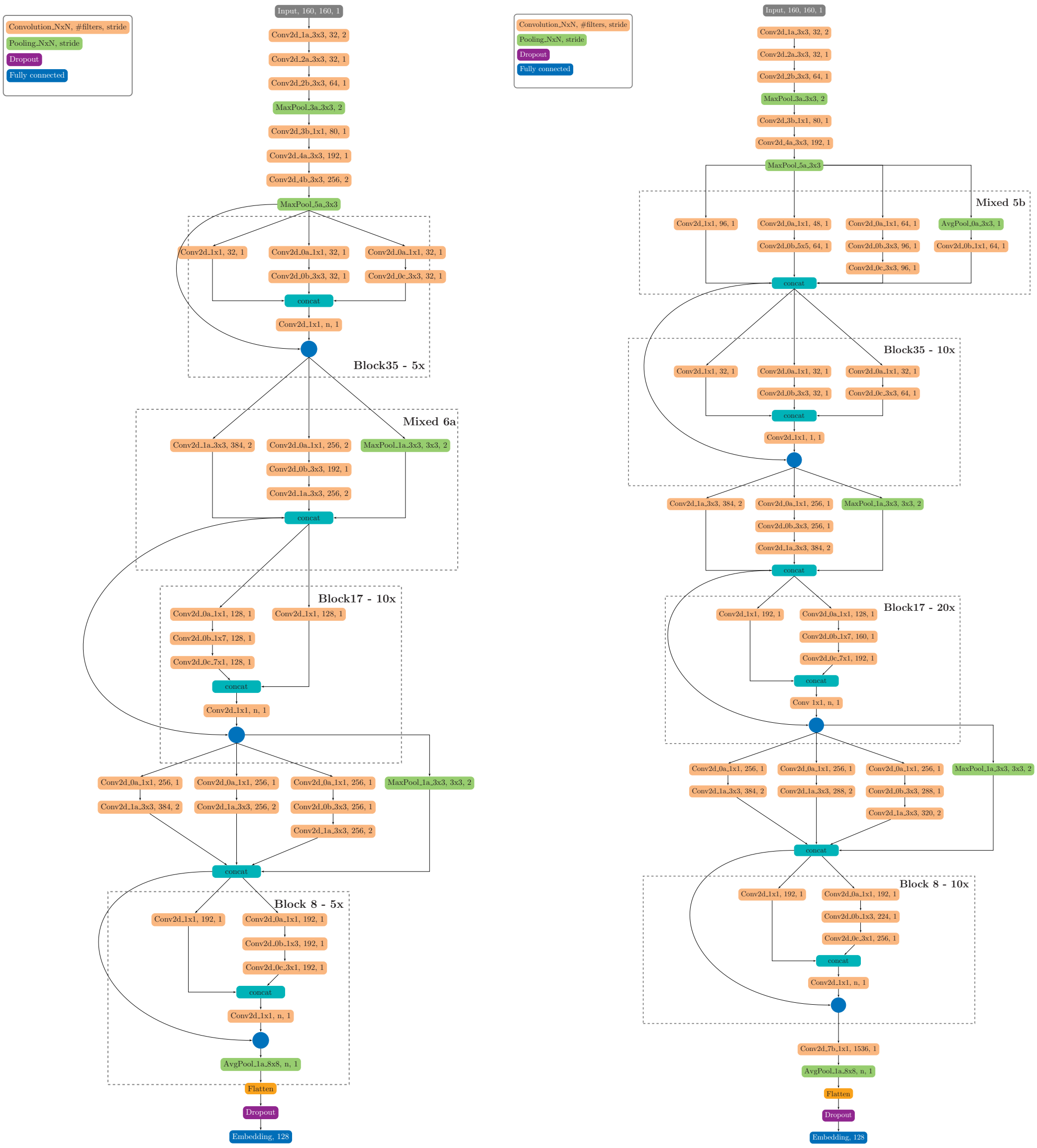
classifier, despite the fact those models don't have any prior knowledge about a target modality ($\mathscr{D}^t$). In special, it is worth noting the recognition rates of the FR Baselines based on DCNNs; although DCNNs have a high capability to overfit into the training data (VIS images in this case), such DCNNs are still able to detect discriminant features between different domains. Possible regularities between them can be suggested.

HFR recognition rates using sketches degrades once its shape get very degraded. For instance, a very simple and non parametric system based on Gabor Wavelets (LBPHS) was able to achieve an average rank one recognition rate of 92.97% using the CUHK-CUFS, which is closer to the current state-of-the-art (P-RS in [Klare and Jain, 2013]). Once shapes are distorted, recognition rates drops drastically. The best FR Baseline for CUHK-CUFSF (Incep. Res. v2 RGB) achieved an average recognition rate of 31.05%.

HFR recognition rates using NIR images as probes, presented the highest recognition rates. It was possible to observe that once images are taken in close up, very high recognition rates are achieved. In this scenario, the FR Baselines based on DCNN achieved the highest recognition rates, sometimes higher than some Non Reproducible and Reproducible HFR Baselines. For instance it was possible to achieve an average rank one recognition rate of 91.09% (Incep. Res. v1 - gray) using the NIVL dataset. Using the subset 1m from LDHF database it was possible to achieve 98.8% using the same figure of merit. Same trend observed in the FARGO database using the controled subset (mc) with an FNMR of 4.40% (FMR@1%). This is particularly surprising and in the best of our knowledge, these observations were never made in the literature. However, once differences on pose (CASIA), distance (LDHF) and different illumination conditions (FARGO) are into play, recognition rates drops, although those DCNNs have samples containing such source of variability in the VIS domain. Average rank one recognition rate dropped to 24% using pictures of probes at 150m in the LDHF database.

The most challenging task seems to be the VIS-Thermal domain. Among the FR Baselines, the ones based on Gabor Wavelets presented the highest rank one recognition rates. For instance, LGBPHS presented 35.73% and 43.71% for Thermal and Pola Thermal databases respectively. Among the DCNNs, the best one the is once more the Incep. Res. v2. Its gray level version presented an average rank one recognition rate of 31.10% using the Thermal database and 27.29% using the Pola Thermal version of the database.

In this chapter it was also presented some baselines that will guide this work (Reproducible HFR Baselines). Such baselines, MLBP from [Liao et al., 2009] and Multi Scale features from [Liu et al., 2012] were introduced for the VIS to NIR task. However, in this work, it is extended to other image modalities. Surprisingly, for the VIS to NIR task, once those baselines are tested to VIS to NIR databases, where variations, such as unconstrained illumination (FARGO), unconstrained pose and expression (CASIA) and different stand-offs (LDHF) are presented, recognition rates decreases. For the task of VIS to Thermograms, such baselines presented higher recognition rates than the DCNN FR Baselines.



(a) Inception Resnet v1

(b) Inception Resnet v2

Figure 3.11 – Inception Resnet architectures. Implementation inspired by Szegedy et al. [2017]

4 Heterogeneous Face Recognition as a Session Variability Problem

In Chapter 2 several sources of session variability for FR was introduced, such as variations on pose, illumination, expression and aging. HFR introduces another source of variability which is the image modality.

In this chapter the task of HFR is modeled using crafted features and Gaussian Mixture Models (GMM). In the last few years, several strategies to improve robustness against different sources of variability of recognition systems based on GMMs were proposed [Vogt et al., 2005; Kenny et al., 2007; Dehak et al., 2011]. Mostly applied for speaker recognition systems, such frameworks are able to suppress variations in different channels of audio data using the same type of crafted features (Mel-Frequency Cepstrum Coefficients(MFCC)). In this chapter Inter-Session Variabiliry (ISV) modeling is investigated. ISV aim to explicitly model and suppress within-class variation in a low-dimensional subspace using Gaussian Mixture Models as a basis.

4.1 Gaussian Mixture Models

A GMM consists of a probabilistic model for density estimation. It is hypothesized that observed data is generated from a mixture of a finite number of Gaussian distributions. More formally, a GMM is composed by a weighted sum of C multivariate gaussian components [Bishop, 2006, p.430]

$$p(x|\Theta_{gmm}) = \sum_{c=1}^{C} w_c \mathcal{N}(x; \mu_c, \Sigma_c), \tag{4.1}$$

where $\Theta_{gmm} = \{w_c, \mu_c, \sigma_c\}_{\{c=1...C\}}$ are the weights, means and the covariances of the model. Moreover, w_c must satisfy these two constrains $0 \leq w_c \leq 1$ and $\sum_{c=1}^{C} w_c = 1$.

Biometric recognition using GMM consists in to estimate one GMM per identity at enrollment time. Then, given a sample x the scoring function is given by $P(x|\Theta_{identity})$.

One of the challenges in biometric recognition in general is that very often the number of samples for enrollment is limited. For instance, in face recognition it can be one sample only. Several methods and different hypotheses are proposed in the literature to estimate Θ_{gmm} when the number of samples are limited. For both face [Cardinaux et al., 2006] and speaker recognition [Reynolds et al., 2000] a very effective method is to first estimate a subject independent GMM, as a prior, and then from this prior, adapt to a particular identity at enrollment time. In biometric recognition such prior is called **Universal Background Model (UBM)** [Reynolds et al., 2000]. Several strategies were proposed in the literature to estimate the parameters of such GMM [McLachlan and Basford, 1988; McLachlan and Peel, 2000]; in this chapter it is focused on the ones used in this thesis.

Maximum Likelihood Estimator

The Maximum Likelihood Estimator (MLE) is one of the most popular strategies to estimate the GMM parameters[Bishop, 2006, p.435] and the UBM[Reynolds et al., 2000]. In statistics, maximum likelihood estimator (MLE) is a method of estimating the parameters of a statistical model given observations by finding the Θ that maximizes $P(X|\Theta_{gmm})|X \in \{x_1,...,x_n\}$. No closed form solution exists for maximizing this function. However, this optimization can be carried out by the Expectation-Maximization (EM) algorithm [Dempster et al., 1977].

The MLE estimation of the GMM parameters using EM begins with an initial estimation Θ^0. In practice this initialization is carried out by using a clustering algorithm, such as k-means [Reynolds et al., 2000]. Then, EM alternates between the following expectation (E) and Maximization (M) steps. During the E-step the probabilities of the training samples are evaluated and accumulated using the current Θ. During the M-step the parameters of Θ are updated using the accumulated probabilities computed during the E-Step. These steps are repeated for certain number of iterations or until some convergence criteria is fulfilled. Algorithm 2 illustrates how Θ_{gmm} is estimated using MLE.

Maximum a posteriori Estimator (MAP)

In biometrics, the Maximum a posteriori estimator for GMM is applied once a class specific GMM needs to be derived from an UBM. As mentioned before, this is very suitable at **enrollment time** when the number of samples are limited.

As for MLE, no closed form solution exists for maximizing $P(X|\Theta_{\text{identity}})|X \in \{x_1,...,x_n\}$. Hence, its estimation is carried out via EM similarly to MLE. Once Θ_{ubm} has been trained (usually via MLE), a class specific GMM is derived by adapting the parameters w_c, μ_c, Σ_c for a particular subject. This is described in the Algorithm 3.

Practical evidences shows that the adaptation only of the means ($\mu_{c,\text{map}}$ in Algorithm 3) is effective for both face and speaker recognition [Reynolds et al., 2000; Cardinaux et al., 2006; McCool and Marcel, 2009; McCool et al., 2013]. Hence, in this work MAP adaptation refers directly to **mean-only adaptation**.

```
Data: $\Theta^0 = \{w_c^0, \mu_c^0, \Sigma_c^0\}_{c=1...C}, X = \{x_1, x_2, ..., x_n\}$
Result: $\Theta = \{w_c, \mu_c, \Sigma_c\}$
while convergence do
    #E-Step;
    for i=0 to size(X) do
        $n_c = 0$;
        $f_c = 0$;
        $s_c = 0$;
        #Computing posterior for each sample;
        $r_c(x_i) = \frac{w_c \mathcal{N}[x_i|\mu_c,\Sigma_c]}{\sum_{c=1}^{C} w_c \mathcal{N}[x_i|\mu_c,\Sigma_c]}$ ;        // Computing responsibilities
        #Accumulating statistics;
        $n_c = n_c + r_c(x_i)$ ;                  // 0th order stats
        $f_c = f_c + n_c \cdot x_i$ ;             // 1st order stats
        $s_c = s_c + n_c \cdot (x_i \cdot x_i)$ ; // 2nd order stats
    end
    #M-Step;
    $w_c = \frac{n_c}{size(X)}$ ;                 // New weights
    $\mu_c = \frac{f_c}{n_c}$ ;                   // New means
    $\Sigma_c = \frac{s_c}{n_c}$ ;                // New variances
end
```

Algorithm 2: MLE Algorithm to estimate GMM parameters

A convenient way to write MAP adaptation is by using the GMM mean-supervector notation [Vogt and Sridharan, 2008; McCool et al., 2013]. The GMM mean-supervector notation consists of taking means of the GMM and create a single vector to represent them. Follow below an example on how to represent an UBM with this mean-supervector notation:

$$m_{\text{ubm}} = \left[\mu_1, \mu_2, ..., \mu_C\right] \tag{4.2}$$

Then, the mean-MAP adaptation can be represented as:

$$m_{\text{map}} = m_{\text{ubm}} + d, \tag{4.3}$$

where m_{map} is the class specific model and d is the class specific offset from the UBM (m_{ubm})[Vogt and Sridharan, 2008] defined as:

$$d_{\text{map}} = Dz_{\text{map}}. \tag{4.4}$$

Here D is a diagonal matrix of size $(C\text{dim}_x \times C\text{dim}_x)$ where $I = \tau D^{\intercal}\Sigma^{-1}D$. Σ is a block diagonal covariance of the UBM and z is the latent variable of the client offset which is assumed to be normally distributed, $z \sim \mathcal{N}(0, I)$. Since the MAP adaptations applies only for the GMM means, it is possible to write 4.3 as:

$$\Theta_{\text{map}} = \Theta_{\text{ubm}} + d \tag{4.5}$$

One possible way of computing scores using Θ_{map} is to directly compute $P(x|\Theta_{\text{map}})$. In practical applications to have zero centered scores is suitable. One way to achieve that is via the Log-Likelihood Ratio (LLR) between Θ_{map} and Θ_{ubm}. This is computed as the following:

$$\text{score} = ln(P(X|\Theta_{\text{map}}) - ln(P(X|\Theta_{\text{ubm}}). \tag{4.6}$$

Here, given an arbitrary GMM Θ_{gmm} and a sequence of samples $X = \{x_1, x_2 ... x_n\} \in \mathbb{R}^D$ the $ln(P(X|\Theta_{gmm})$ is defined as:

$$ln(P(X|\Theta_{gmm}) = \sum_{i=0}^{N} ln(P(x_i|\Theta_{gmm})). \tag{4.7}$$

```
Data: $\Theta_{\text{ubm}} = \{w_{c;\text{ubm}}, \mu_{c;\text{ubm}}, \Sigma_{c;\text{ubm}}\}_{c=1...C}, X = \{x_1, x_2, ..., x_n\}, R \in \mathbb{R}$
Result: $\Theta_{\text{map}} = \{w_{c;\text{map}}, \mu_{c;\text{map}}, \Sigma_{c;\text{map}}\}$
#E-Step;
for i=0 to size(X) do
    $n_c = 0$;
    $f_c = 0$;
    $s_c = 0$;
    #Computing posterior for each sample;
    $r_c(x_i) = \frac{w_{c;\text{ubm}}\mathcal{N}[x_i|\mu_{c;\text{ubm}},\Sigma_{c;\text{ubm}}]}{\sum_{c=1}^{C} w_{c;\text{ubm}}\mathcal{N}[x_i|\mu_{c;\text{ubm}},\Sigma_{c;\text{ubm}}]}$ ;          // Computing responsibilities
    #Accumulating statistics;
    $n_c = n_c + r_c(x_i)$ ;                                              // 0th order stats
    $f_c = f_c + n_c \cdot x_i$ ;                                          // 1st order stats
    $s_c = s_c + n_c \cdot (x_i \cdot x_i)$ ;                              // 2nd order stats
end
#M-Step;
$\alpha_c = \frac{n_c}{n_c+R}$;                                          // Adjusting adaptation factor
$w_{c;\text{map}} = \frac{\alpha_c n_c}{size(X)} + (1-\alpha_c) w_{c;\text{ubm}}$ ;                          // New weights
$u_{c;\text{map}} = (\alpha_c f_c) + (1-\alpha_c)\mu_{c;\text{ubm}}$ ;                                 // New means
$\Sigma_{c;\text{map}} = (\alpha_c s_c) + (1-\alpha_c)(\Sigma_{c;\text{ubm}} + (\mu_{c;\text{ubm}})^2 - (\mu_{c;\text{map}})^2)$ ;       // New Covariance
```

Algorithm 3: MAP Algorithm to estimate class specific GMM parameters

4.2 Intersession Variability Modeling

Built on top of GMMs, Intersession Variability Modeling (ISV) [Vogt and Sridharan, 2008] proposes to explicitly model the with class variability and compensate them during enrollment and test time. The ISV approach hypothesizes that within-class variability is embedded in a linear subspace of the GMM mean super-vector space, which is defined as:

$$u = Uw, \tag{4.8}$$

where U is the low-dimensional subspace of size $(C\text{dim}_x, D_U)$ that contains all possible within-class variations and w is a latent session variable, which is assumed to be normally distributed $w \sim \mathcal{N}(0, I)$. Like in the MAP adaptation, this modeling also has the class specific offset d defined as:

$$d = Dz. \tag{4.9}$$

This class specific latent variable z is not the same as the one defined by the MAP adaptation. Here, z is jointly estimated with w. This estimation is explained further.

To summarize, ISV hypothesizes that a given mean-supervector μ can be decomposed as an UBM offset of a session factor and a client specific factor as the following:

$$\mu = \Theta_{ubm} + Uw + Dz. \tag{4.10}$$

Hence, a class specific model, free of session variability is defined as:

$$\Theta_{\text{isv}} = \Theta_{\text{ubm}} + Dz. \tag{4.11}$$

The **scoring** is defined as the LLR between the client-specific model and the UBM. Given an arbitrary enrolled model Θ_{isv}, a UBM Θ_{ubm} and a sequence of samples $X = \{x_1, x_2 ... x_n\} \in \mathbb{R}^D$ the LLR is defined as:

$$\text{score} = \sum_{i=1}^{N} \left[ln\Big(\frac{p(x_i|\Theta_{isv} + Uw)}{p(x_i|\Theta_{ubm} + Uw} \Big) \right] \tag{4.12}$$

For a given gaussian component c, a set of identities I and a set of input samples from each identity J, the subspace U is estimated by solving the following system of equations.

$$U_c \Big(\sum_{i=0}^{I} \sum_{j=0}^{J} n_{i,j;c} E[w_{i,j} w_{i,j}^\intercal] \Big) = \sum_{i=0}^{I} \Big(\sum_{j=0}^{J} f_{i,j;c} - n_{i,j;c}(-D_c z_i) E[w_{i,j}]^\intercal \Big), \tag{4.13}$$

where $n_{i,j;c}$ and $f_{i,j;c}$ are the 0th and 1st order statistics of a MAP adapted GMM (see Algorithm 3). D_c is the client specific offset defined as:

$$D_c \Big(\sum_{i=0}^{I} n_{i,j;c} E[z_i z_i^\intercal] \Big) = \sum_{i=0}^{I} \Big(\sum_{j=0}^{J} \Big[f_{i,j;c} - n_{i,j;c}(-U_c w_{i,j}) \Big] \Big) E[z_i]^\intercal, \tag{4.14}$$

$E[z_i z_i^\intercal]$ is computed as:

$$E[z_i z_i^\intercal] = \Big(I + \Sigma^- 1 n_i \Big)^{-1} + E[z_i] E[z_i]^\intercal, \tag{4.15}$$

where $E[z_i]$ is defined as:

$$E[z_i] = \left(I + D^{\intercal}\Sigma^{-1} n_i D\right)^{-1} D^{\intercal}\Sigma^{-1}\left[f_i - n_i - \sum_{j=1}^{J} n_{i,j} U w i, j\right] \tag{4.16}$$

Finally, $E[w_{i,j} w_{i,j}^{\intercal}]$ is computed as:

$$E[w_{i,j} w_{i,j}^{\intercal}] = \left(I + U^{\intercal}\Sigma^{-1} n_{i,j} U\right)^{-1} + E[w_{i,j}]E[w_{i,j}]^{\intercal}, \tag{4.17}$$

where $E[w_{i,j}]$ is computed:

$$E[w_{i,j}] = \left(I + U^{\intercal}\Sigma^{-1} n_{i,j} U\right)^{-1} U^{\intercal}\Sigma^{-1}\left[f_{i,j} - n_{i,j} D z_i\right]. \tag{4.18}$$

4.3 InterSession Variability modeling for Heterogeneous Face Recognition

This section is defined by the following hypothesis:

Hypothesis 4.1 *Given $X_s = \{x_{s1}, x_{s2}, ..., x_{sn}\}$ and $X_t = \{x_{t1}, x_{t2}, ..., x_{tn}\}$ being a set of crafted features from $\mathscr{D}^s$ and $\mathscr{D}^t$, respectively, with their corresponding shared set of labels $Y = \{y_1, y_2, ..., y_n\}$ and Θ being an arbitrary GMM, possible within-class variations from different image modalities can be suppressed in the GMM mean-supervector space using InterSession Variability modeling.*

In this section ISV is formulated for HFR task as the following. A given mean-supervector μ can be decomposed as an UBM offset of a session factor and a client specific factor as the following:

$$\mu = \Theta_{\mathscr{D}_s\mathscr{D}_t} + U_{\mathscr{D}_s\mathscr{D}_t} w + D_{\mathscr{D}_s\mathscr{D}_t} z, \tag{4.19}$$

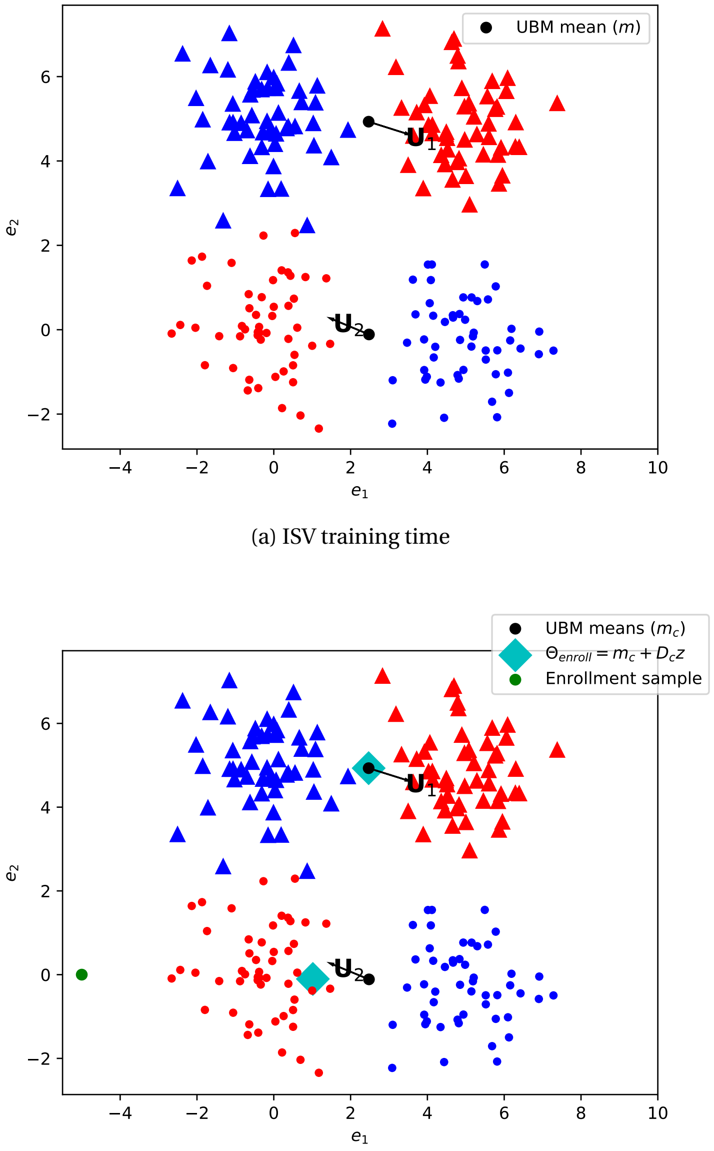
where $\Theta_{\mathscr{D}_s\mathscr{D}_t}$ is a UBM jointly estimated from samples two image modalities $\mathscr{D}_s$ and $\mathscr{D}_t$ using MLE. $U_{\mathscr{D}_s\mathscr{D}_t}$ is the subspace that contains all possible session effects that image modalities may introduce to crafted features, w is its associated latent session variable, while $D_{\mathscr{D}_s\mathscr{D}_t} z$ represents the client offset (modality free offset). Both $\Theta_{\mathscr{D}_s\mathscr{D}_t}$ and $U_{\mathscr{D}_s\mathscr{D}_t}$ are estimated at **training time** using algorithm 2 and equation 4.13 respectively.

At **enrolment time**, given $X_s = \{x_{s1}, x_{s2}, ..., x_{sn}\} \in \mathscr{D}^s$ the GMM free of modality variability is obtained by estimating:

$$\Theta_{\text{enroll}} = \Theta_{\mathscr{D}_s\mathscr{D}_t} + D_{\mathscr{D}_s\mathscr{D}_t} z. \tag{4.20}$$

Finally, at **scoring time** given a set of samples $X_t = \{x_{t1}, x_{t2}, ..., x_{tj}\} \in \mathscr{D}^t$ the LLR score is computed as the following:

$$\text{score} = \sum_{j=1}^{J} \left[ln \Big(\frac{p(x_{t;j} | \Theta_{\text{enroll}} + U_{\mathscr{D}^s\mathscr{D}^t} w_{t;j})}{p(x_{t;j} | \Theta_{\mathscr{D}^s\mathscr{D}^t} + U_{\mathscr{D}^s\mathscr{D}^t} w_{t;j})} \Big) \right]. \tag{4.21}$$

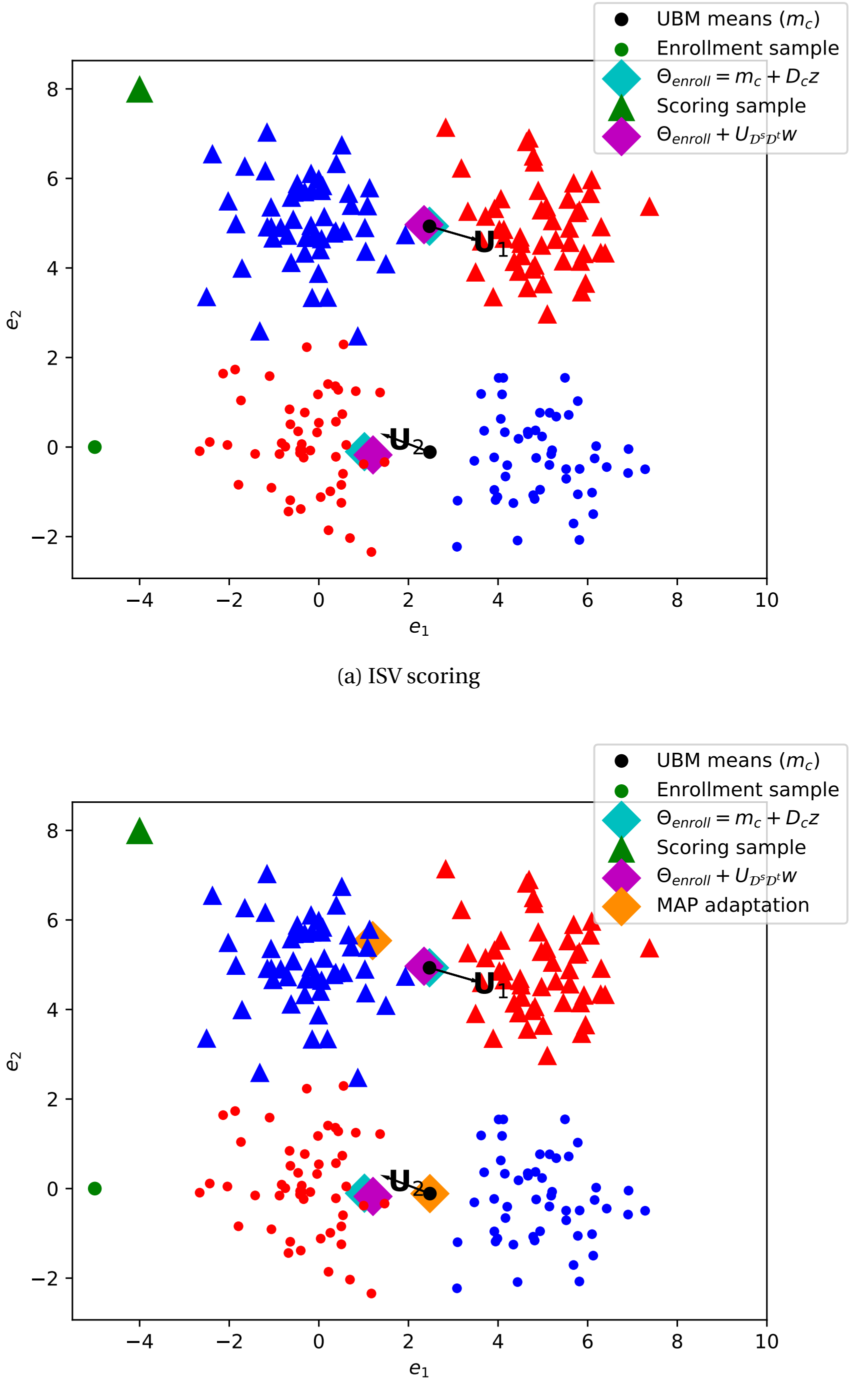


(a) ISV training time

(b) ISV Enrollment time

Figure 4.1 – ISV Intuition (a) Estimation of m and U (background model) (b) Enrollment considering the session varibility using one sample

Figures 4.1 and 4.2 presents an intuition on how ISV models heterogeneous data in a toy heterogeneous dataset. Let's assume that the data points in the Figure 4.1 (a) are a training



(a) ISV scoring

(b) MAP scoring

Figure 4.2 – ISV Intuition (a) Scoring using ISV (b) Scoring using MAP adaptation

set. This training set is composed by samples from 2 identities represented by the colors red and blue. The dots in the figure are samples from $\mathscr{D}_s$ and the triangles are samples from

$\mathscr{D}_t$. In Figure 4.1 (a) it is possible to observe the UBM means estimated with two gaussian components trained with the MLE estimator (see Algorithm 2). Once this UBM is trained, the U subspace is then estimated (see Equation 4.13). The direction of session variations with respect to each gaussian component can be seem in Figure 4.1(a) with the black arrows (U_1 and U_2). To be able to plot them in 2d, the rank of U is set to one. Those are the main variables estimated at **training time**.

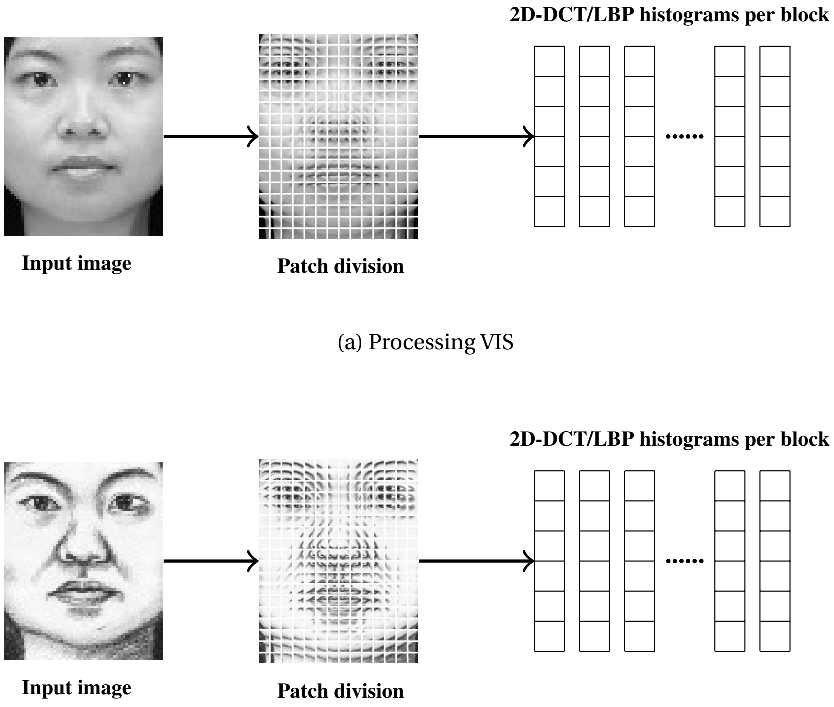
Figure 4.1(b) demonstrates the **enrollment** process. Let's consider that the green dot in the Figure 4.1 (b) is one data sample of an unknown identity from $\mathscr{D}_s$. Then, the enrollment is carried out using Equation 4.20. The output mean super-vector from this enrollment process can also be decomposed in terms of each Gaussian component. This is represented by the cyan diamonds in Figure 4.1 (b).

In Figure 4.2 demonstrates the scoring process. Let's consider that the green triangle in Figure 4.2 (a) is one data sample of the same unknown identity, but now from $\mathscr{D}_t$. The magenta diamonds represents the mean super-vector decomposition with respect to each Gaussian component by doing $\Theta_{\text{enroll}} + U_{\mathscr{D}^s\mathscr{D}^t} w_{t;j}$ (see Equation 4.21). It is possible to observe that the magenta diamonds are almost overlapped with the cyan diamonds. This is an indicator of a high LLR using Equation 4.21.

For the sake of comparison, figure 4.2 (b) illustrates the MAP client adaptation using the same sample (green triangle) as input. The mean-supervector decomposition using MAP adaptation (see Equation 3) is illustrated with the orange diamonds. MAP doesn't consider possible session effects (within-class variations) in its modeling, hence, their estimated means are severely shifted with respect to the cyan diamonds (the reference used during at enrollment time). This is an indicator of a low LLR using the Equation 4.6.

4.4 Implementation details

In this thesis two types of crafted features are evaluated as input to this framework. The first one is the LBP histograms (see chapter 2.1.3). The Local Binary Patterns system implemented in this work is an adaptation from [Rodriguez and Marcel, 2006b]. First, faces are detected, cropped and aligned to be with 64 × 80 pixels. Then, $LBP_{P=8,r=2}$ is computed in the aligned image for further patch division of 32 × 32 pixels with 31 pixels of overlap at each direction. Differently from chapter 2.1.3, those patches are not concatenated in one single vector, but treated independently. Hence, $P(X|\Theta_{isv})$ is a result of the accumulation of the LLR scores for each patch. The second type of crafted feature is the DCT coefficients. Each cropped and geometric normalized face image from each modality is sampled in patches of 12 × 12 pixels moving the sampled window in one pixel (11 pixels of overlap). Then each patch is mean and variance normalized and the first 45 DCT coefficients are extracted. The first coefficient (DC component) is discarded resulting in a feature vector of 44 elements per patch. This setup is an adaptation from [McCool et al., 2013]. Each sampled patch is considered as an independent observation. A schematic of such patch division is illustrated in Figure 4.3.


2D-DCT/LBP histograms per block
Input image
Patch division
......
2D-DCT/LBP histograms per block
Input image
Patch division
......

(a) Processing VIS

(b) Processing Sketch

Figure 4.3 – Feature extraction of the proposed approach

The most relevant hyper-parameters for ISV are the number of Gaussian components of the UBM and the rank of U. For both databases we will tune first the number of Gaussian components keeping the rank of $U = 50$. Then, the rank of U is fine tuned for some databases.

4.5 Experiments and Analysis

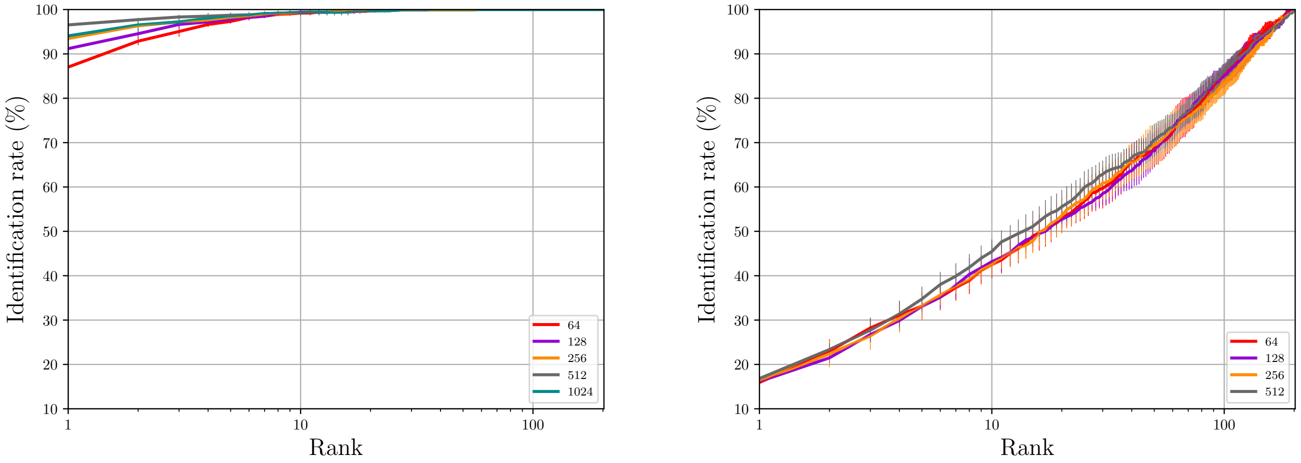
In this section the experiments assessing the session variability hypothesis is presented. To make it easier the interpretation of the recognition rates, all the tables in this section (4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8) are split in three parts. **FR Baselines** corresponds to all FR baselines described in the Section 3.1. **Reproducible Baselines** corresponds to all HFR baselines described in the Section 3.2 and it was implemented or integrated in the context of this work. Finally, **Non Reproducible Baselines** corresponds to HFR baselines whose source code was not made publicly available and its average rank one recognition rate was picked directly from its corresponding publication.

4.5.1 Visible Light to Sketches

In this subsection it is described experiments with two sketch databases: CUHK-CUFS and CUHK-CUFSF.

CUHK-CUFS

Figure 4.4 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 87%. This figure of merit is increased to ≈ 91% with 128 gaussians and to ≈ 93% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 96.53%. With 1024 gaussians the average rank on recognition rate is decreased to ≈ 94%. Figure 4.4 (b) presents the CMC curves varying the number of Gaussians using LBP histograms as input. Using LBPs as crafted features it is possible to observe that the average rank one recognition rates stabilizes in ≈ 16% while the number of gaussians varies from 64 to 512 gaussians. Hence, the same trends observed with DCT coefficients can't be observed with LBP histograms.



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.4 – CUFS - Average CMC curves (with error bars) using DCT coefficients and LBP histograms varying the number of gaussians from Θ_{ubm}

Experiments in Figure 4.4 (a) are conducted with the rank of U set to 50 and it is possible to observe that the highest rank one recognition rate is observed with 512 gaussians. Figure 4.5 presents the same experiment, but varying the rank of U from 10 to 200 while the number of gaussians is set to 512. Using rank equals to 10 it is achieved an average rank one recognition rate of ≈ 93%. This figure of merit is increased to 96.53% with rank equals to 50 and then decreases to ≈ 95% for ranks equals to 100 and 160 respectively. In this database the highest average rank-one recognition rate is achieved with rank equals to 50. This value presents a good trade-off between complexity and accuracy. Hence, this value is kept for the next experiments using this image modality.

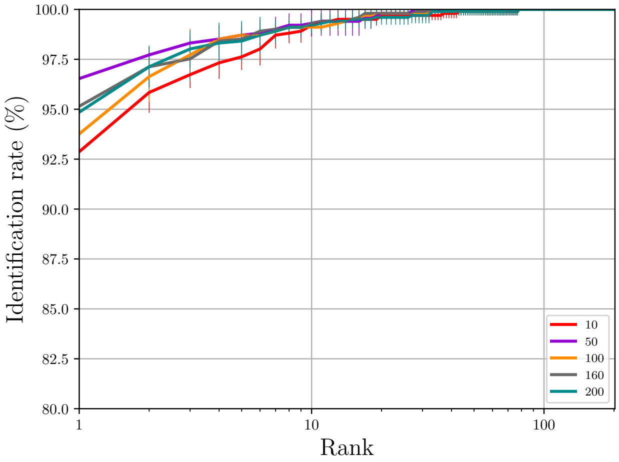


Figure 4.5 – CUFS - Average CMC curves (with error bars) using DCT coefficients varying the rank of U

Table 4.1 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 96.53%, which is lower than P-RS (Non Reproducible baselines). However, this approach presents higher recognition rate than the Reproducible and the FR Baselines. For instance, a variation of GFK presents 93.27% and LGBPHS presents 92.97% using the same figure of merit.

With this set of experiments it was possible to observe highest recognition rates using **DCT coefficients**. Using these coefficients it was possible to confirm Hypothesis 4.1.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 cuhk-cufs
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see^2.

CUHK-CUFSF

Figure 4.6 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 36%. This figure of merit is increased to ≈ 44% with 128 gaussians and to ≈ 54% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 55.58%.

Table 4.1 – CUHK-CUFS - Average rank one recognition rate under different feature setups for ISV

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	72.57%(3.7)
2	Incep. Res. v2 - gray scaled	80.29%(1.5)
3	Gabor-Graph	81.29%(2.4)
4	LGBPHS	92.97%(2.2)
Reproducible Baselines		
5	MLBP [Liao et al., 2009]	62.27%(3.8)
6	MultiScale feat. [Liu et al., 2012]	64.16%(2.5)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	93.27%(1.4)
Non Reproducible Baselines		
8	P-RS as in [Klare and Jain, 2013]	**99%(n/a)**
9	Face VACS in [Klare and Jain, 2013]	89%(n/a)
ISV		
10	DCT - ISV 512 Gaussians	**96.53%(0.8)**
11	LBP - ISV 256 Gaussians	16.83%(1.2)

Figure 4.6 (b) presents the CMC curves varying the number of Gaussians using LBP histograms as input. Using LBPs as crafted features it is possible to observe that the average rank one recognition rates stabilizes in ≈ 5% while the number of gaussians varies from 64 to 512 gaussians. Hence, the same trends observed with DCT coefficients can't be observed with LBP histograms.

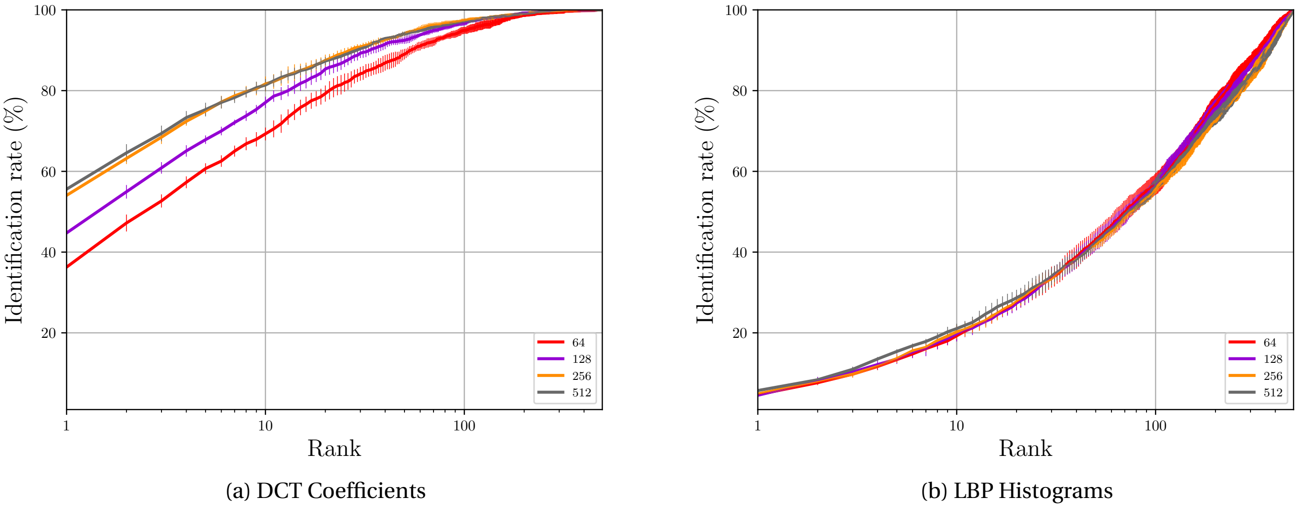
Table 4.2 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 55.58%, which is lower than most of the Non Reproducible baselines. For instance, the DEEPS system [Galea, 2018] presents an average rank one recognition rate of 82.92%. However, this approach presents higher recognition rate than the Reproducible and the FR Baselines. For instance, a variation of GFK presents 41.01% and Incep. Res. v2 presents 29.51% using the same figure of merit.

With this set of experiments it was possible to observe highest recognition rates using **DCT coefficients**. These are the same trends observed previously. Using these coefficients it was possible to confirm Hypothesis 4.1 although the recognition rates are lower than the state-of-the-art.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 cuhk-cufsf
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.6 – CUFSF - Average CMC curves (with error bars) using DCT coefficients and LBP histograms varying the number of gaussians from Θ_{ubm}

check how to train other setups see[2].

4.5.2 Visible Light to Near Infrared

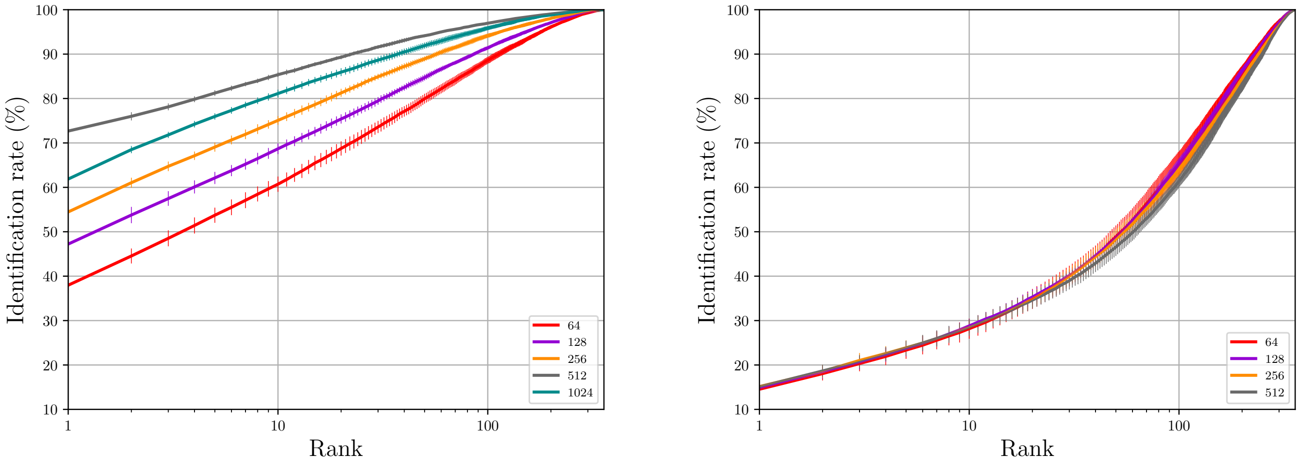
In this subsection it is described experiments with four NIR databases: CASIA, NIVL, LDHF and FARGO.

CASIA

Figure 4.7 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 38%. This figure of merit is increased to ≈ 47% with 128 gaussians and to ≈ 54% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 72.67%. With 1024 gaussians the average rank on recognition rate is decreased to ≈ 62%. Those experiments are conducted with the rank of U set to 50 and it is possible to observe that the highest rank one recognition rate is observed with 512 gaussians. Figure 4.8 presents the same experiment, but varying the rank of U from 10 to 200 while the number of gaussians is set to 512. Using rank equals to 10 it is achieved an average rank one recognition rate of ≈ 39%. This figure of merit is increased to 72.67% with rank equals to 50 and then decreases to ≈ 71%, ≈ 68% and ≈ 58% for rank equals to 100, 160 and 200 respectively. The highest average rank-one recognition rate is achieved with rank equals to 50. Hence, this value is kept for the next experiments using this image modality (while varying the number of gaussians). This value presents a good trade-off between complexity and accuracy. Moreover, since no

Table 4.2 – CUHK-CUFSF - Average rank one recognition rate under different feature setups for ISV

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	24.49%(0.5)
2	Incep. Res. v2 - gray scaled	29.51%(0.7)
3	Gabor-Graph	19.39%(1.0)
4	LGBPHS	25.38%(1.5)
Reproducible Baselines		
5	MLBP in [Liao et al., 2009]	9.11%(1.7)
6	MultiScale feat. in [Liu et al., 2012]	6.76%(0.7)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	41.01%(1.8)
Non Reproducible Baselines		
8	TP-LBP [Wolf et al., 2008]	59.7%(not available)
9	CDFL [Jin et al., 2015]	81.3%(not available)
10	DEEPS [Galea, 2018]	**82.92%(1.25)**
11	LGMS [Galea, 2018]	78.19%(0.52)
ISV		
11	DCT - ISV 512 Gaussians	**55.58%(1.2)**
12	LBP - ISV 256 Gaussians	5.71%(0.6)



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.7 – CASIA - Average CMC curves (with error bars) using DCT coefficients and LBP histograms varying the number of gaussians from Θ_{ubm}

improvements (in terms of error rates) could be observed beyond 512 gaussians, in the next experiments this fine tuning is carried out until 512 gaussians.

Figure 4.4 (b) presents the CMC curves varying the number of Gaussians using LBP histograms

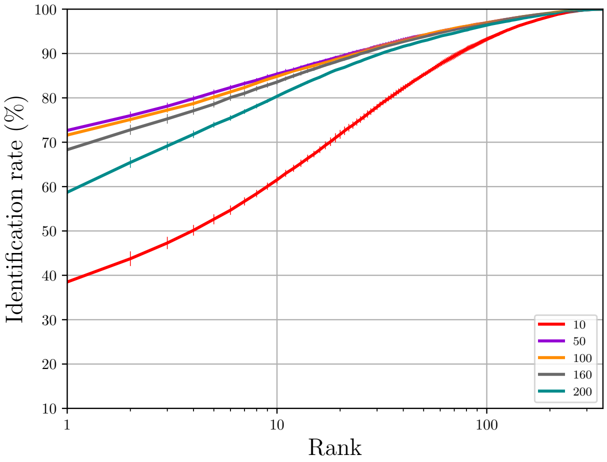


Figure 4.8 – CASIA - Average CMC curves (with error bars) using DCT coefficients varying the rank of U

as input. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 14%. This figure of merit is increased to ≈ 15% with 128 gaussians and to 15.15% with 256 gaussians. With 512, the average rank one recognition rate decreases to 15.00%. Hence, the same trends observed with DCT coefficients can't be observed with LBP histograms.

Table 4.3 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 72.67%, which is higher than all Reproducible baselines. For instance, the MLBP strategy proposed by Liao et al. [2009] achieved an average rank one recognition rate of 70.33%. Although this could confirm Hypothesis 4.1, the average rank one recognition rate is lower than some FR Baselines that doesn't rely on NIR data in its training. For instance, the DCNNs Incep. Res. v1 gray and Incep. Res. v2 gray achieved and average rank one recognition rate of 74.25% and 73.80% respectively. The proposed approach with ISV presents an average rank one recognition rate ≈ 26% lower than the state-of-the-art approaches. The Non Reproducible baselines CDL [Wu et al., 2017] and WCCN [He et al., 2018] presents respectively 98.62% and 98.70%.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 casia
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].

Table 4.3 – CASIA - Average rank one recognition rate under different Face Recognition systems

#	**FR Algorithm**	**Average rank one rec. rate**
FR Baselines		
1	Incep. Res. v1 - gray	74.25%(1.3)
2	Incep. Res. v2 - gray	73.80%(1.2)
3	Gabor-Graph	21.49%(1.1)
4	LGBPHS	22.24%(1.6)
Reproducible Baselines		
5	MLBP in [Liao et al., 2009]	70.33%(1.2)
6	Multiscale Feat. in [Liu et al., 2012]	67.54%(1.7)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	26.98%(0.9)
Non Reproducible Baselines		
8	IDR in [He et al., 2017]	95.82%(0.7)
9	CDL in [Wu et al., 2017]	**98.62%(0.2)**
10	WCNN in [He et al., 2018]	**98.70%(0.3)**
11	TRIVET in [Liu et al., 2016]	95.74%(0.5)
ISV		
12	DCT - ISV 512 Gaussians	**72.67%(1.0)**
13	LBP - ISV 256 Gaussians	15.15%(1.5)

NIVL

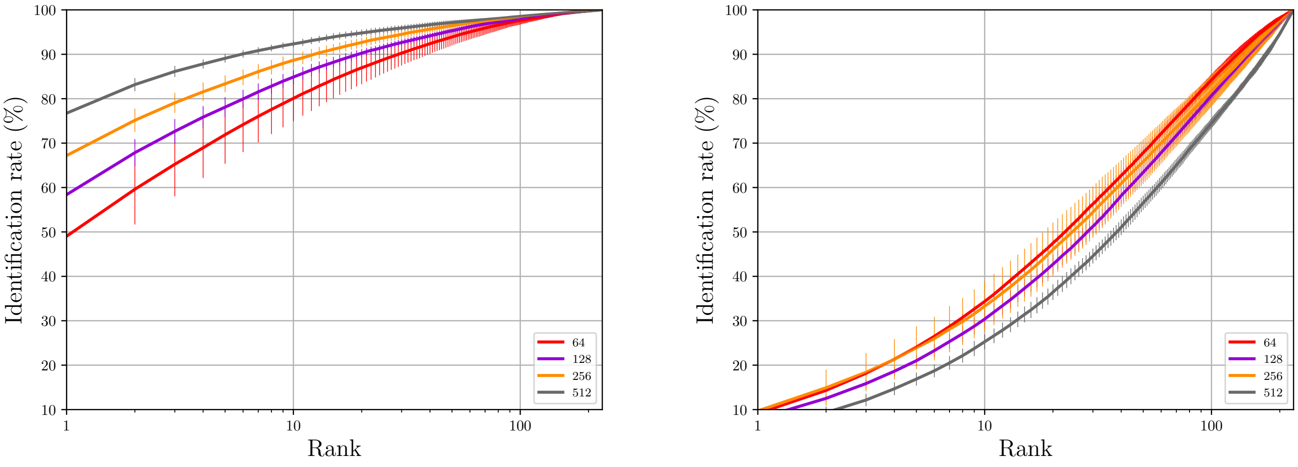
Figure 4.9 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 49%. This figure of merit is increased to ≈ 58% with 128 gaussians and to ≈ 67% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 76.73%. The same trends are not followed with LBP histograms as can be observed in Figure 4.9 (b). Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 9%. This figure of merit is decreased to ≈ 7% with 128 gaussians and increased to 9.70% with 256 gaussians. Experiments with 512 gaussians achieved and average rank one recognition rate of 5.6%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline isv_g512_u50 nivl
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].

Table 4.4 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 76.73%, which is lower than all Reproducible baselines. For instance, the MLBP strategy proposed by Liao et al.



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.9 – NIVL - Average CMC curves (with error bars) using DCT coefficients and LBP histograms varying the number of gaussians from Θ_{ubm}

Table 4.4 – NIVL - Average rank one recognition rate under different Face Recognition systems

#	FR Algorithm	Average Rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray	**91.09%(0.3)**
2	Incep. Res. v2 - gray	88.14%(0.6)
3	Gabor-Graph	16.41%(0.9)
4	LGBPHS	30.98%(3.3)
Reproducible Baselines		
5	MLBP [Liao et al., 2009]	85.35%(1.1)
6	Multiscale Feat. [Liu et al., 2012]	90.34%(1.3)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	63.08%(2.2)
ISV		
8	DCT - ISV 512 Gaussians	**76.73%(2.0)**
9	LBP - ISV 256 Gaussians	9.70%(3.4)

[2009] achieved an average rank one recognition rate of 85.35%. Although this could confirm Hypothesis 4.1, the average rank one recognition rate is lower than some FR Baselines that doesn't rely on NIR data in its training. For instance, the DCNNs Incep. Res. v1 gray and Incep. Res. v2 gray achieved and average rank one recognition rate of 91.09% and 88.14% respectively.

LDHF

Table 4.5 presents the average rank one recognition rates using **DCT coefficients** as input. Analysing the **1m** stand-off it is possible to observe an average rank one recognition rate of from 75.2% with 64 gaussians. Using 128 and 256 gaussians this value increases to 84.8% and 96.0% respectively. Finally, for 512 gaussians this values drops to 94.8%. Analysing the **60m** stand-off it is possible to observe an average rank one recognition rate of 30.8% with 64 gaussians. With 128 and 256 gaussians this value increases to 34.4% and 59.2% respectively. Finally, for 512 gaussians this values drops to 51.2%. For **100m** stand-off it is possible to observe an average rank one recognition rate of 11.2% with 64 gaussians. With 128 and 256 gaussians this value increases to 12.8% and 37.2% respectively. Using 512 gaussians this values drops to 27.2%. Finally, for **150m** stand-off it is possible to observe an average rank one recognition rate of 4.0% with 64 gaussians. With 128 gaussians this value increases to 4.4%. This figure of merit has a substantial increase with 256 and 512 gaussians with 14.4% and 13.6% respectively.

Table 4.5 – LDHF - average rank one recognition rates under different ISV setups

#	FR Algorithm	1m	60m	100m	150m
FR Baselines					
1	Incep. Res. v1 - gray	94.8%(2.0)	78.0%(4.4)	28.4%(1.5)	4.8%(1.6)
2	Incep. Res. v2 - gray	92.8%(2.7)	75.6%(2.9)	9.6%(1.5)	2.8%(1.6)
3	Gabor-Graph	54.8%(3.7)	15.6%(1.5)	15.2(3.5)	1.6%(2.0)
4	LGBPHS	72.4%(4.3)	32.0%(2.9)	26.0%(3.6)	9.2%(3.2)
Reproducible Baselines					
5	MLBP in [Liao et al., 2009]	67.2%(7.0)	23.2%(3.0)	10.0%(2.8)	6.0%(1.8)
6	Multiscale Feat. in [Liu et al., 2012]	74.4%(3.4)	43.2%(3.7)	22.0%(4.5)	14.8%(3.0)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	73.6%(4.3)	31.2%(7.2)	12.0%(2.8)	2.8%(3.0)
DCT coefficients					
8	ISV 64 gaussians	75.2%(3.5)	30.8%(3.2)	11.2%(2.7)	4.0%(2.8)
9	ISV 128 gaussians	84.8%(3.5)	34.4%(4.9)	12.8%(2.7)	4.4%(2.6)
11	ISV 256 gaussians	**96.0%(1.3)**	**59.2%(6.0)**	**37.2%(7.4)**	**14.4%(6.6)**
10	ISV 512 gaussians	94.8%(3.5)	51.2%(3.2)	27.2%(2.4)	13.6%(2.0)
LBP Histograms					
12	ISV 64 gaussians	32.8%(3.2)	25.6%(1.5)	22.8%(4.5)	17.6%(5.8)
13	ISV 128 gaussians	28.4%(5.4)	20.8%(2.7)	22.5%(3.5)	15.2%(2.0)
14	ISV 256 gaussians	23.6%(3.0)	22.4%(5.1)	17.2%(4.6)	14.8%(2.4)
15	ISV 512 gaussians	24.8%(5.0)	21.2%(7.2)	16.8%(3.6)	15.2%(3.3)

Table 4.5 presents also the average rank one recognition rates using **LBP histograms** as input. Analysing the **1m** stand-off it is possible to observe an average rank one recognition rate of from 32.8% with 64 gaussians. Using 128 and 256 gaussians this value decreases to 28.4% and 23.6% respectively. Finally, for 512 gaussians this values drops to 24.8%. Analysing the **60m** stand-off it is possible to observe an average rank one recognition rate of 25.6% with 64 gaussians. With 128 and 256 gaussians this value decreases to 20.8% and 22.4% respectively. Finally, for 512 gaussians this values drops to 21.2%. For **100m** stand-off it is possible to observe an average rank one recognition rate of 22.8% with 64 gaussians. With 128 and 256

gaussians this value increases to 22.5% and 17.2% respectively. Using 512 gaussians this values drops to 16.8%. Finally, for **150m** stand-off it is possible to observe an average rank one recognition rate of 17.6% with 64 gaussians. With 128 gaussians this value decreases to 15.2%. This figure of merit decreases 14.8% and 15.2% respectively for 256 and 512 gaussians. Differently from the previous experiment, in this one it is possible to observe, in average, a rank one recognition rate of ≈ 15% with 150m stand-off, with is higher than the one with DCT coefficients.

With this set of experiments it was possible to observe highest recognition rates using **DCT coefficients**. These are the same trends observed previously. Using these coefficients it was possible to confirm Hypothesis 4.1.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 ldhf
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].

FARGO

Table 4.6 presents the FNMR@FMR=1%(dev) using the ISV approach with DCT coefficients and LBP histograms as inputs.

Under the controlled protocol **(mc)**, using **DCT coefficients** presents a FNMR of 46.00% using 64 gaussians. For 128 gaussians such figure of merit is reduced to 44.60% and to 40.00% for 256 gaussians. Finally with 512 gaussians such figure of merit drastically decreases to 29.60%. In the same experiment, using the protocol dark **(ud)** presents a FNMR of 65.40% using 64 gaussians. For 128 gaussians such figure of merit is increased to 67.6% and to 61.2% with 256 gaussians. Finally with 512 gaussians such figure of merit decreases to 56.00%. Experiments using the protocol outside **(uo)** presents a FNMR of 65.60% using 64 gaussians. For 128 gaussians such figure of merit is increased to 65.80% and decreases to 63.10% with 256 gaussians. Finally, with 512 gaussians such figure of merit decreases to 59.90%.

Under the controlled protocol **(mc)**, using **LBP histograms** presents a FNMR of 72.40% using 64 gaussians. For 128 gaussians such figure of merit is reduced to 71.20% and to 72.00% for 256 gaussians. Finally with 512 gaussians such figure of merit increases to 73.20%. In the same experiment, using the protocol dark **(ud)** presents a FNMR of 79.30% using 64 gaussians. For 128 gaussians such figure of merit is decreased to 78.60% and to 75.50% with 256 gaussians. Finally with 512 gaussians such figure of merit increases to 78.90%. Experiments using the protocol outside **(uo)** presents a FNMR of 91.20% using 64 gaussians. For 128 gaussians such figure of merit is decreased to 90.20% and increases to 91.70% with 256 gaussians. Finally, with 512 gaussians such figure of merit increases once more to 91.80%.

Table 4.6 – Fargo database - FNMR@FMR=1%(dev) taken from the development under different ISV setups

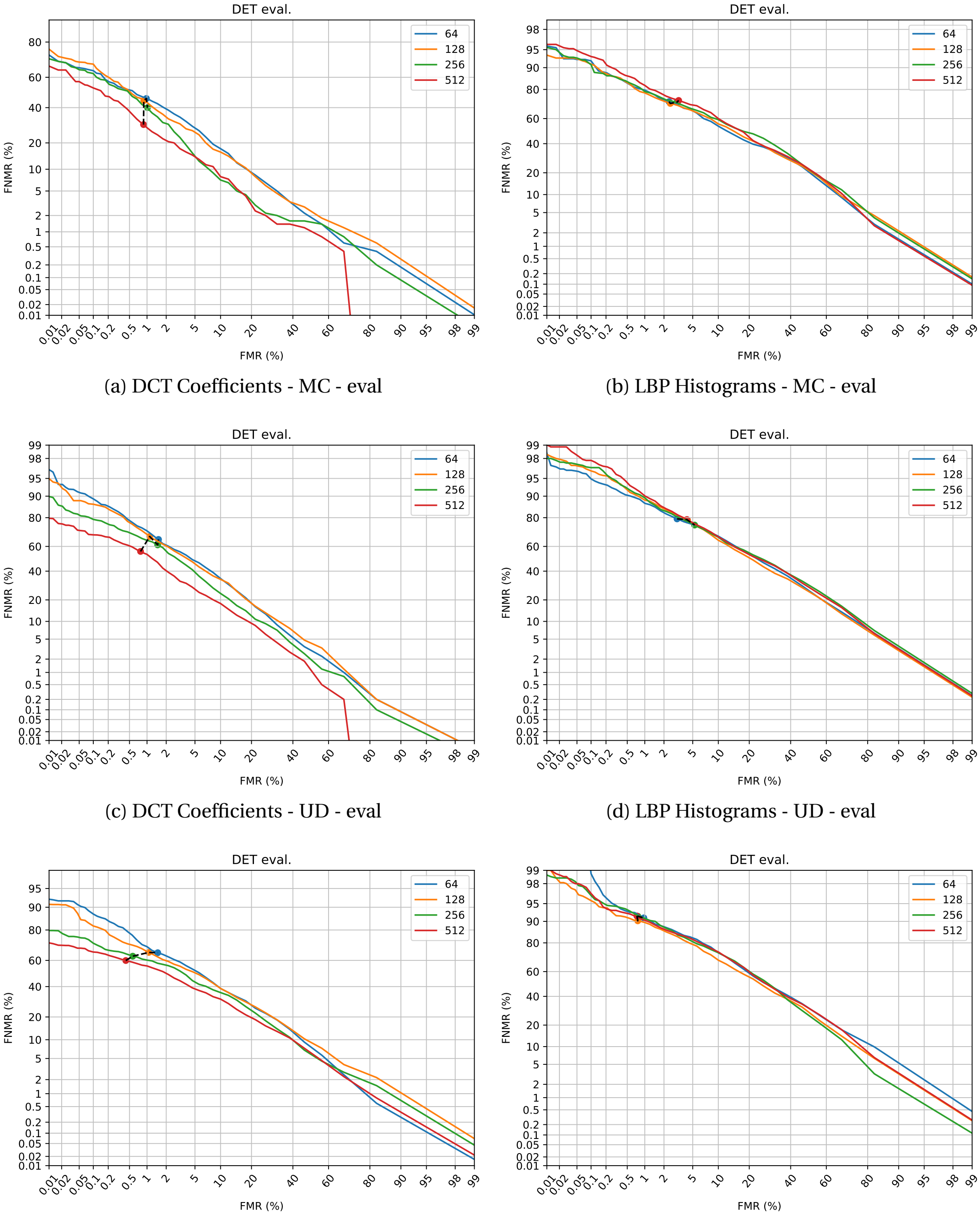
#	FR Algorithm	mc		ud		uo	
		dev	eval	dev	eval	dev	eval
FR Baselines							
1	Incep. Res. v1 - gray scaled	0.40	**2.80**	6.70	11.90	0.40	9.00
2	Incep. Res. v2 - gray scaled	0.00	4.40	0.80	**4.00**	0.50	**2.00**
3	Gabor-Graph	56.80	57.20	64.40	59.90	64.80	76.80
4	LGBPHS	45.80	45.80	59.80	66.40	62.00	72.80
Reproducible Baselines							
5	MultiScale feat. [Liu et al., 2012]	20.80	23.00	26.70	23.70	32.30	42.40
6	MLBP [Liao et al., 2009]	23.80	21.40	29.00	27.30	34.10	51.60
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	16.80	15.60	21.60	19.60	25.30	30.70
DCT coefficients							
8	ISV 64 gaussians	32.80	46.00	63.60	65.40	49.50	65.60
9	ISV 128 gaussians	28.00	44.60	57.80	67.60	42.60	65.80
10	ISV 256 gaussians	27.40	40.00	49.50	61.20	35.00	63.10
11	ISV 512 gaussians	22.60	**29.60**	43.30	**56.00**	30.70	**59.90**
LBP Histograms							
12	ISV 64 gaussians	74.00	72.40	89.90	79.30	91.30	91.20
13	ISV 128 gaussians	74.40	71.20	90.50	78.60	94.50	90.20
14	ISV 256 gaussians	74.00	72.00	92.20	75.50	93.50	91.70
15	ISV 512 gaussians	76.20	73.20	94.30	78.90	94.40	91.80

With these set of experiments it was possible to observe very high FNMR for all conditions using both DCT coefficients and LBP histograms. Compared with Reproducible baselines the system based on GFK [Gong et al., 2012; Sequeira et al., 2017], under the controlled protocol (mc), presents a FMR of 15.60% compared with 29.60% using ISV with DCT coefficients (512 gaussians). This figure of merit decreases even more with FR Baselines based on DCNN. For instance, the Incep. Res. v1 presents an FNMR of 2.80%. It was also possible to observe a severe impact, in terms of FNMR, using the protocol dark and outside (ud and uo). For instance, compared with the Reproducible baselines the system based on GFK [Gong et al., 2012; Sequeira et al., 2017] presents a FNMR of 19.60% and 30.70% respectively compared with 56.00% and 59.90% using ISV with DCT coefficients. As before, this figure of merit decreases steadily using DCNN baselines. The DCNN Incep. Res. v2 presents an FNMR of 4.00% and 2.00% respectively. The same trends can be observed in Figure 4.10 where the DET plots for both input features are presented.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 fargo
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].



(a) DCT Coefficients - MC - eval

(b) LBP Histograms - MC - eval

(c) DCT Coefficients - UD - eval

(d) LBP Histograms - UD - eval

(e) DCT Coefficients - UO - eval

(f) LBP Histograms - UO - eval

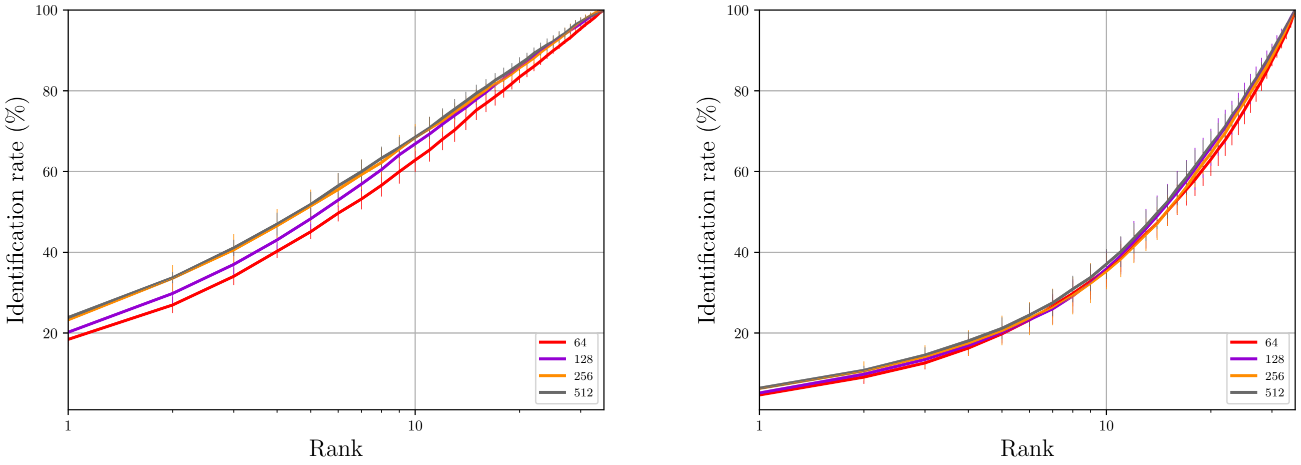
Figure 4.10 – FARGO - DET curves for verification experiments under the three illumination conditions MC (controlled), UD (dark) and UO (outdoor) trained with ISV. The column on the left presents DET curves using DCT coefficients as input and the column on the right presents DET curves using LBP histograms as a basis

4.5.3 Visible Light to Thermograms

In this subsection it is described experiments with two subsets of the Pola Thermal database: Thermal and Pola Thermal.

Thermal

Figure 4.11 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 18%. This figure of merit is increased to ≈ 20% with 128 gaussians and to ≈ 23% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 23.86%. Figure 4.11 (b) presents the CMC curves varying the number of Gaussians using LBP histograms as input. Using LBPs as crafted features it is possible to observe an average rank one recognition rate of ≈ 4% with 64 gaussians. This figure of merit is increased to ≈ 5% with 128 gaussians and it stabilizes in ≈ 6% with 256 and 512 gaussians. For both types of features as input it is possible to observe very low recognition rates.



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.11 – Thermal - Average CMC curves (with error bars) using DCT coefficients and LBP histograms

Table 4.7 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 23.86%, which is lower than all of the Non Reproducible baselines. For instance, the CpNN system [Hu et al., 2016] presents an average rank one recognition rate tree times higher (78.72%). The same trend observed with DPM system [Hu et al., 2016] with an average rank one recognition rate of

75.31%. Furthermore, all Reproducible Baselines presents higher average rank one recognition rate than the best ISV system (with DCT coefficients). A variation of GFK [Gong et al., 2012; Sequeira et al., 2017] presents an average rank one recognition rate of 34.07%. Using the same figure of merit the MLBP [Liao et al., 2009] and Multiscale features [Liu et al., 2012] presents 36.80% and 26.89 respectively. The same trends are followed by the FR Baselines. The LGBPHS system presents an average rank one recognition rate of 43.71% while the Incep. Res. v2 31.09%.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 thermal
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].

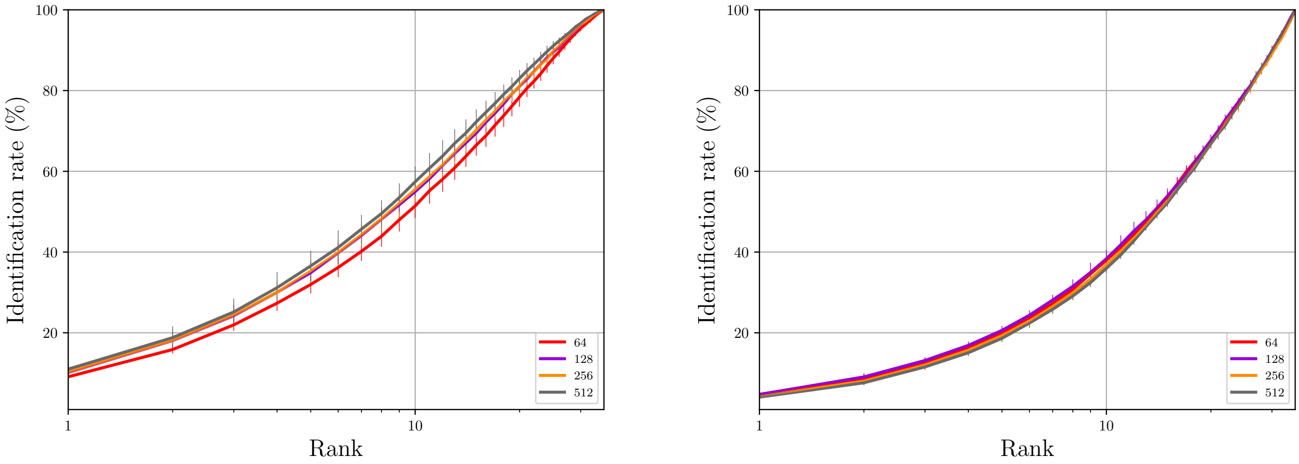
Table 4.7 – Thermal database - Average rank one recognition rate under different feature setups for ISV

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	20.55%(4.2)
2	Incep. Res. v2 - gray scaled	31.09%(4.1)
3	Gabor-Graph	17.46%(1.9)
4	LGBPHS	43.71%(3.7)
Reproducible Baselines		
5	MLBP in [Liao et al., 2009]	36.80%(3.5)
6	Multiscale Feat. in [Liu et al., 2012]	26.89%(3.5)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	34.07%(2.9)
Non Reproducible Baselines		
8	PLS [Hu et al., 2016]	53.05% (n/a)
9	DPM [Hu et al., 2016]	75.31% (n/a)
10	CpNN [Hu et al., 2016]	**78.72% (n/a)**
ISV		
11	DCT - ISV 512 Gaussians	**23.86%(1.3)**
12	LBP - ISV 512 Gaussians	6.35%(0.9)

Pola Thermal

Figure 4.12 (a) presents the CMC curves varying the number of Gaussians using DCT coefficients. Using 64 Gaussians it is possible to achieve an average rank one recognition rate of ≈ 9%. This figure of merit is increased to ≈ 10% with 128 gaussians and to ≈ 10% with 256 gaussians. Experiments with 512 gaussians get its best average rank one recognition rate with 11.0%. Figure 4.11 (b) presents the CMC curves varying the number of Gaussians using LBP histograms as input. Using LBPs as crafted features it is possible to observe an average rank one recognition rate of 4.75% with 64 gaussians. Then, this figure of merit stabilizes to ≈ 4%

with 128, 256 and 512 gaussians. For both types of features as input it is possible to observe very low recognition rates. Those are the same trends observed in the Thermal database.



(a) DCT Coefficients

(b) LBP Histograms

Figure 4.12 – Pola Thermal - Average CMC curves (with error bars) using DCT coefficients and LBP histograms

Table 4.8 shows the average rank one recognition rate comparing the experiments using the two different types of features (the one with the highest recognition rate for each setup) with the FR, Reproducible and the Non Reproducible baselines. The approach based on ISV with DCT coefficients achieved an average rank one recognition rate of 11.0%, which is lower than all of the Non Reproducible baselines. For instance, the CpNN system [Hu et al., 2016] presents an average rank one recognition rate tree times higher (82.90%). The same trend observed with DPM system [Hu et al., 2016] with an average rank one recognition rate of 80.54%. All Reproducible Baselines presents higher average rank one recognition rate than the best ISV system (with DCT coefficients). The GFK system with Gabor Jets [Gong et al., 2012; Sequeira et al., 2017] presents an average rank one recognition rate of 34.43%. Using the same figure of merit the MLBP [Liao et al., 2009] and Multiscale features [Liu et al., 2012] presents 36.80% and 26.89 respectively. The same trends are followed by the FR Baselines. The LGBPHS system presents an average rank one recognition rate of 35.73% while the Incep. Res. v2 27.29%.

Using the thesis software this strategy can be triggered with the following bash command:

```
$ bob bio htface htface_baseline isv_g512_u50 thermal
```

This command lines demonstrates just how to train the ISV setup using DCT coefficients. To check how to train other setups see[2].

Table 4.8 – Pola Thermal database - Average rank one recognition rate under different feature setups for ISV.

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	18.69%(2.1)
2	Incep. Res. v2 - gray scaled	27.29%(0.8)
3	Gabor-Graph	8.46%(1.1)
4	LGBPHS	35.73%(1.8)
Reproducible Baselines		
5	MLBP in [Liao et al., 2009]	15.61%(2.9)
6	Multiscale Feat. in [Liu et al., 2012]	20.81%(3.4)
7	GFK [Gong et al., 2012; Sequeira et al., 2017]	34.43%(2.3)
Non Reproducible Baselines		
8	PLS [Hu et al., 2016]	58.67% (n/a)
9	DPM [Hu et al., 2016]	80.54% (n/a)
10	CpNN [Hu et al., 2016]	**82.90% (n/a)**
ISV		
11	DCT - ISV 512 Gaussians	**11.0%(1.6)**
12	LBP - ISV 128 Gaussians	4.74%(0.6)

4.6 Discussion

It this chapter one hypothesis was drawn. Hypothesis 4.1 argue that given an arbitrary set of crafted features, possible within-class variations from different image modalities can be suppressed in the GMM mean-supervector space using InterSession Variability modeling.

Experiments were carried with two different types of crafted features, DCT coefficients and LBP histograms, and three different images modalities. In Section 4.5 it was possible to observe that experiments with DCT coefficients provided substantially higher recognition rates compared with LBP histograms for all experiments. Recognition rates using ISV with LBP features also presented lower recognition rates compared with other strategies based on LBPs, such as, MLBP from Liao et al. [2009] and Multiscale Feat. from Liu et al. [2012]. Both strategies are patch based and their histograms are concatenated forming one feature vector only per image, preserving possible spacial relations in the face. In the strategy based on ISV, the LBP histograms are not concatenated; the LLR (see Equation 4.12) is accumulated for each image patch independently. With this set of experiments, it is possible to suggest that the spacial ordering is a factor that must be preserved while using LBP features. This effect couldn't be observed using DCT coefficients and the recognition rates were higher. Hence, next paragraphs refers only to experiments using DCT coefficients.

In the **VIS to Sketches** task it was possible to observe best recognition rates using 512 gaussians keeping the rank of U to 50. For instance, experiments with CUHK-CUFS, where the sketches are very reliable, the highest average rank one recognition rate is 96.53%. For CUHK-CUFSF, where the sketch line is not aligned with its corresponding photo, the average rank one recognition is 55.58%.

For the **VIS to NIR** task, more data are available for experimentation and such data was captured under different conditions. Hence, different analysis can be made. Under constrained conditions, where subjects are closer to the camera, with neutral expression and no pose/illumination variations it was possible to observe high recognition rates. For instance, experiments with LDHF, considering 1m stand-off only it was possible to observe an average rank one recognition rate of 96.0% with 256 gaussians. Experiments with NIVL database, the average rank one recognition rate with 512 gaussians is 76.73%. Finally, experiments using the FARGO dataset, considering only the controlled protocol (**mc**) the ISV model with 512 gaussians presented a $FNMR@FMR = 1\%$ of 29.6%.

Under the same task, it was possible to observe severe degradation under more uncontrolled scenarios. For instance, experiments with CASIA database, where NIR face images with several variations in pose and expression are recorded the ISV with 512 gaussians presented an average rank one recognition rate of 72.67%. Experiments using the FARGO dataset, considering the protocol dark (**ud**) it was posssible to achieve a $FNMR@FMR = 1\%$ of 56.00% and considering the protocol outside it was posssible to achieve 59.9% using the same figure of merit.

Experiments using **VIS to Thermal** presented the lowest recognition rates. For instance, using the Thermal database it was possible to achieve an average rank one recognition rate of 23.86% (see Table 4.7). The same trend is followed using the Pola Thermal database where an average rank one recognition rate of 11.00% was achieved (see Table 4.8).

In the next chapter it is considered the learning of features that are specific to one particular image modality instead of relying on crafted ones.

5 Domain Specific Units

Many researchers pointed out that DCNNs progressively compute more powerful feature detectors as depth increases [Mallat, 2016; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014]. Practical evidences of this were extensively discussed in Chapter 2. Yosinski et al. [2014] and Li et al. [2015] empirically demonstrated that feature detectors that are closer to the input signal (called low level features) are base features that resemble Gabor features, color blobs, edge detectors, etc. On the other hand, features that are closer to the end of the DCNN (called high level features) are considered to be more task specific and carry more discriminative power.

In Chapter 3, it was possible to observe that feature detectors from DCNNs trained only with VIS images have some discriminative power over all target domains tested; with VIS to NIR task being the "easiest" ones under certain conditions and the VIS to Thermograms being the most challenging ones. In this Chapter, a strategy that leverages from this prior discriminative power is introduced. Called Domain Specific Units (DSU), such strategy hypothesizes that high level features from a DCNN encode general facial feature detectors that are independent of the image modality. Hence, feature detectors from low level layers can be adapted to better suit a particular image modality. Experiments carried out under different image modalities shows that some image modalities can be encoded with less than 1,000 free parameters and have its recognition rate increased.

5.1 Introduction

This section is defined by the following hypothesis:

Hypothesis 5.1 *Given $X_s = \{x_{s1}, x_{s2}, ..., x_{sn}\}$ and $X_t = \{x_{t1}, x_{t2}, ..., x_{tn}\}$ being a set of samples from $\mathscr{D}^s$ and $\mathscr{D}^t$, respectively, with their corresponding shared set of labels $Y = \{y_1, y_2, ..., y_n\}$ and Θ being all set of DCNN feature detectors from $\mathscr{D}^s$ (already learnt), there are two consecutive subsets: one that is domain* ***dependent****, θ_t, and one that is domain* ***independent****, θ_s, where $P(Y|X_s,\Theta) = P(Y|X_t,[\theta_s,\theta_t])$. Such θ_t, that can be learnt via back-propagation, is so called* ***Domain Specific Units****.*

A possible assumption one can make is that θ_t is part of the set of low level features, directly connected to the input signal. In this chapter this assumption is extensively tested and has practical advantages. First, low level features are less dense than high level ones, since for most DCNN architectures they are composed by convolutional filters. This may reduce the number of hyperparameters that needed to be learnt. Second it is possible to make all image modalities share the same face space, which is particularly interesting for future deployment and further classification.

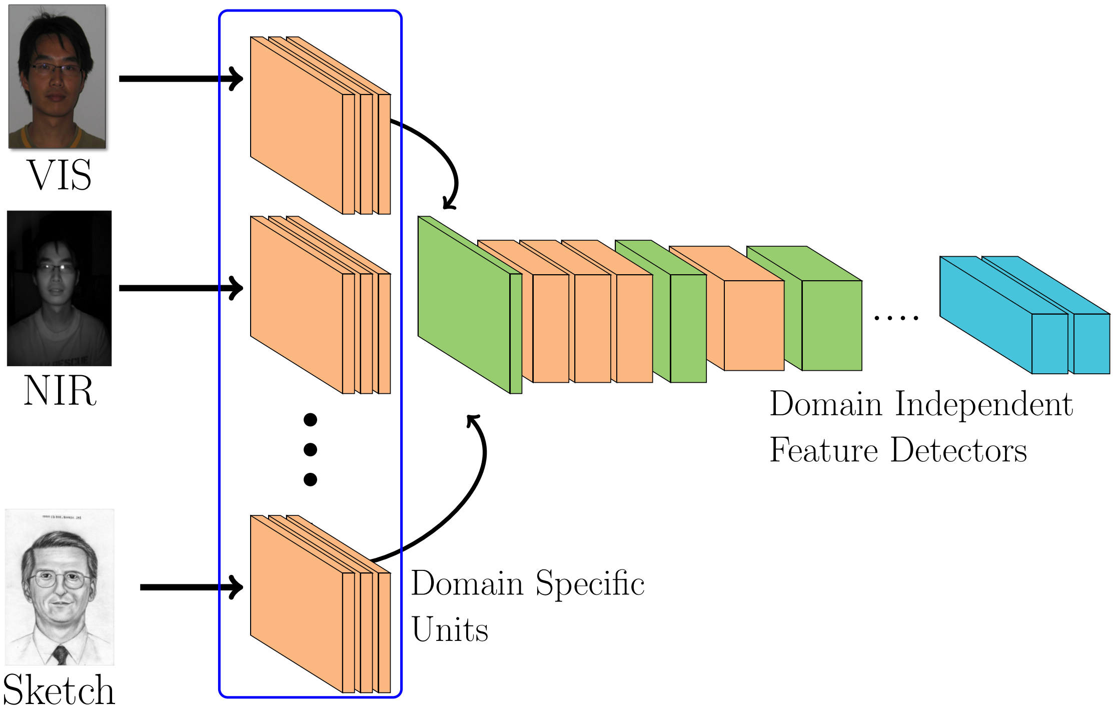


Figure 5.1 – Domain Specific Units - General Schematic

Figure 5.1 presents a general schematic of the proposed approach where each image domain has its own specific set of feature detectors (low level features) and further share the same face space (high level features). Such face space is previously estimated using VIS images only.

In this approach, the free parameters from each target domain (θ_t) are jointly estimated with VIS images (source domain). To jointly train such DSUs, two different strategies are proposed and they are described as follows.

Siamese Networks

The first strategy is based on **Siamese Networks [Chopra et al., 2005]** and it is depicted in Figure 5.2. During the forward pass, Figure 5.2 (a), a pair of face images, one from each image modality is forwarded to the DCNN. Those pair of images can either be from the same identity or not. VIS images (x_s) are forwarded using the DCNN pre-trained for FR (the one at the top in Figure 5.2 (a)); and images from the target domain (x_t) are first forwarded to its domain specific set of feature detectors and then amended to the DCNN trained for VIS images (where the hypothesized domain independent features are). During the backward pass, Figure 5.2 (b), errors are backpropagated only for θ_t. With such structure only a small subset of feature detectors are learnt, reducing the capacity of the joint model. The loss $\mathscr{L}$ is defined as [Chopra et al., 2005]:

$$\mathscr{L}(\Theta) = 0.5\Big[(1-Y)D(x_s,x_t) + Y\max(0, m - D(x_s,x_t))\Big], \tag{5.1}$$

where m is the contrastive margin, Y is the label (1 when x_s and x_t belong to the same subject and 0 otherwise) and D is defined as:

$$D(x_s,x_t) = ||\phi(x_s) - \phi(x_t)||_2^2, \tag{5.2}$$

where ϕ are the embeddings from the jointly trained DCNN.

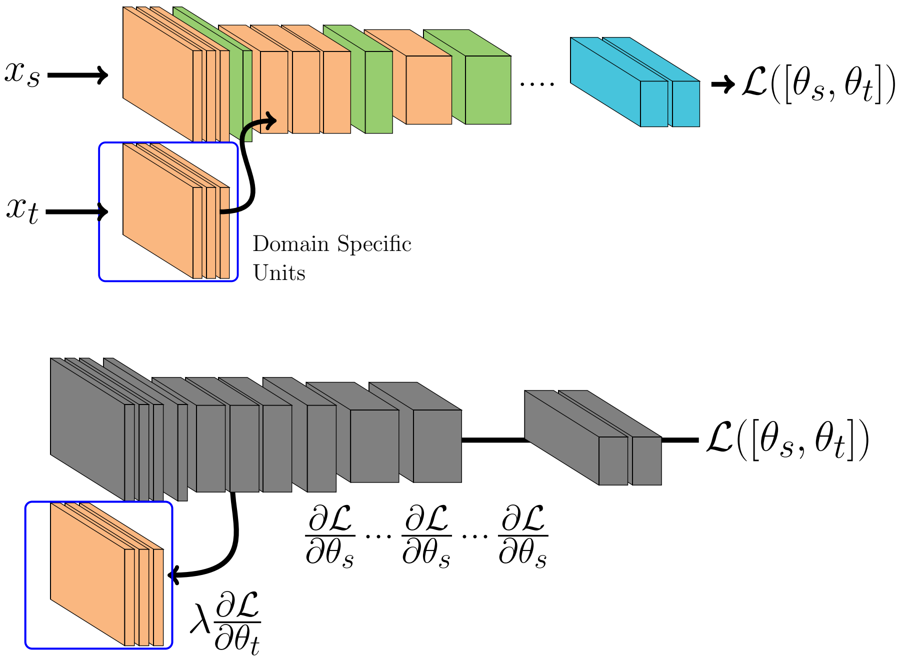

x_s
x_t
$\mathcal{L}([\theta_s, \theta_t])$
Domain Specific
Units
....
$\mathcal{L}([\theta_s, \theta_t])$
$\frac{\partial \mathcal{L}}{\partial \theta_s} \cdots \frac{\partial \mathcal{L}}{\partial \theta_s} \cdots \frac{\partial \mathcal{L}}{\partial \theta_s}$
$\lambda \frac{\partial \mathcal{L}}{\partial \theta_t}$

Figure 5.2 – Domain Specific Units learnt with Siamese Neural Networks given a pair of samples x_s and x_t from source and target domain respectively. (a) Forward pass behaviour (b) Backward pass behaviour

Triplet Networks

The second strategy is based on **Triplet Networks [Schroff et al., 2015]** and it is depicted in Figure 5.3. During the forward pass, Figure 5.3 (a), a triplet of face images are forwarded to the network. x_s^a corresponds to VIS images inputs; x_t^p and x_t^n corresponds to face images sensed in the target domain in such a way that x_s^a and x_t^p are from the same identity and x_s^a and x_t^n are from different identities. The training procedure is similar as with Siamese Networks. VIS images (x_s^a) are forwarded using the DCNN pre-trained for FR (the one at the top in Figure 5.3 (a)); face images from the target domain (x_t^p and x_t^n) are forwarded first to its domain specific set of feature detectors and then amended to the DCNN trained for VIS images (where the hypothesized domain independent features are). During the backward pass, Figure 5.3 (b), errors are back-propagated only for θ^t, that is shared between the inputs x_t^p and x_t^n. With such structure only a small subset of features are learnt, reducing the capacity of the model. The loss $\mathscr{L}$ is defined as:

$$\mathscr{L}(\theta) = ||\phi(x_s^a) - \phi(x_t^p)||_2^2 - ||\phi(x_s^a) - \phi(x_t^n)||_2^2 + \lambda, \tag{5.3}$$

where λ is the triplet margin and ϕ are the embeddings from the DCNN.

During a DCNN training, two types of free parameters are updated (see Chapter 2). The first one corresponds to the feature detectors variables, such as convolutional/deconvolutional filters or the weights of linear combinations. The second are the biases terms added to those operations. With these basic operations (feature detectors and biases), a secondary hypothesis is derived and it is the following.

Hypothesis 5.2 *Face recognition DCNNs automatically craft feature detectors that are both robust against different sources of noise and discriminative. Since the target structure that those feature detectors model is shared among domains (they are face images), θ_t might be embedded in the subset of biases (β) of those detectors.*

To approach Hypothesis 5.2 during the DSU training, the gradients from θ_t corresponding to all structural operations (convolutions, deconvolutions, linear combinations) are discarded. Hence, only the gradients corresponding to the biases are considered.

Algorithm 4 presents a generic pseudo-code of the training procedure that is independent of training method and DCNN architecture. It is worth noting that only the Domain Specific Units (θ_t) are updated.

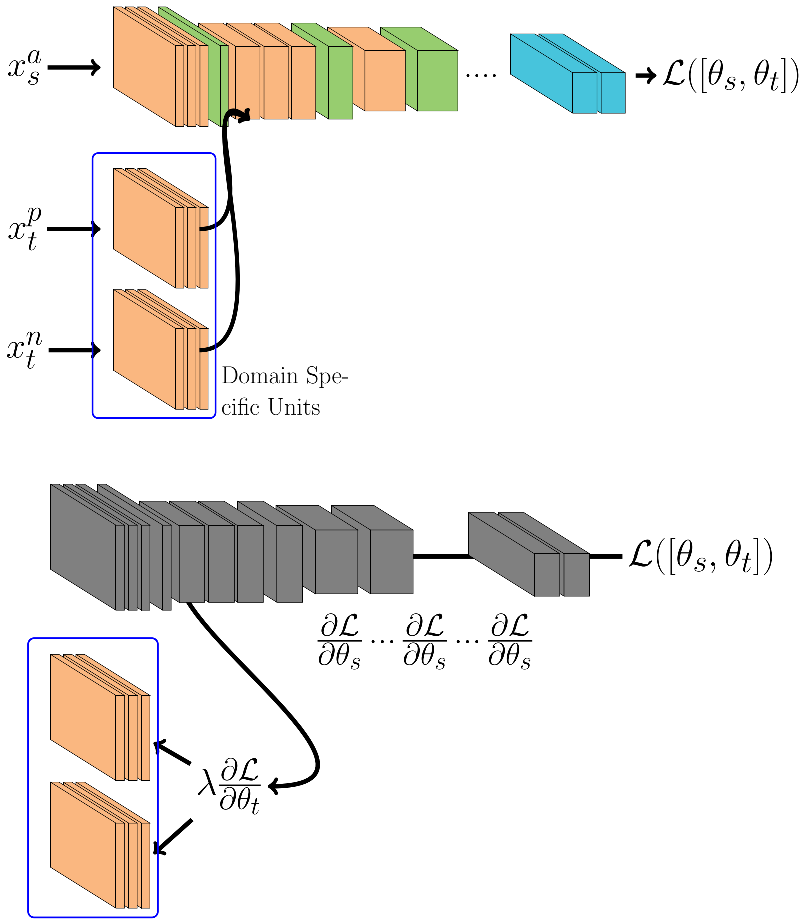

x_s^a
$\mathcal{L}([\theta_s, \theta_t])$
x_t^p
x_t^n
Domain Specific Units
$\mathcal{L}([\theta_s, \theta_t])$
$\frac{\partial \mathcal{L}}{\partial \theta_s} \cdots \frac{\partial \mathcal{L}}{\partial \theta_s} \cdots \frac{\partial \mathcal{L}}{\partial \theta_s}$
$\lambda \frac{\partial \mathcal{L}}{\partial \theta_t}$

Figure 5.3 – Domain Specific Units learnt with Triplet Neural Networks given a triplet of samples: x_s^a from $\mathscr{D}_s$, and x_t^p and x_t^n from $\mathscr{D}_t$. (a) Forward pass behaviour (b) Backward pass behaviour

5.2 Implementation details

It was possible to observe in Chapter 3 that, among the DCNNs tested, the ones based on Inception Resnet presented the highest recognition rates overall. Hence, experiments are carried out with Incep. Res. v1 and Incep. Res. v2 architectures both in gray scaled versions. Such DCNNs were previously trained with a pruned version of the MSCeleb and presented an average FNMR of 99% on LFW dataset. and in the IJB-C unconstrained protocol. Appendix B presents the implementation details of such DCNN.

The goal of DSU is to find the set of low level feature detectors, θ_t, that maximizes recognition rates for each image domain. To find such set, both DCNNs are exhaustively adapted increasing the adaptation depth at every test using either Siamese or Triplet Networks as training strategy. Five possible θ_t sets are analysed and they are called $\theta_{t[1-1]}$, $\theta_{t[1-2]}$, $\theta_{t[1-4]}$, $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$. Table 5.1 presents the variables that are adapted for each one of the tested

```
Data: $\Theta_s$, $\mathscr{L}$, $n\_layers$
Result: $\theta_t$
$\theta_t = \Theta_s[1 : n\_layers]$ ;                          // Domain Spec.  Units
$\theta_s = \Theta_s[n\_layers :]$ ;                            // Domain Indep.  Units
while has_data do
    batch = get_batch();
    $[\frac{\partial \mathscr{L}}{\partial \theta_s}, \frac{\partial \mathscr{L}}{\partial \theta_t}]$ = forward_backward(batch, $\theta_s$, $\theta_t$, $\mathscr{L}$);
    $\theta_t[\beta] = \theta_t[\beta] + \lambda \frac{\partial \mathscr{L}}{\partial \theta_t}[\beta]$;
    if adapt_kernels then
        $\theta_t[W] = \theta_t[W] + \lambda \frac{\partial \mathscr{L}}{\partial \theta_t}[W]$
    end
end
```

Algorithm 4: Training strategy given a pretrained DCNN Θ_s, loss function $\mathscr{L}$ and the number of layers to be adapted n_layers. θ_t is split between the convolutional kernels W and the biases β

architectures. Those names match the ones presented in Figure 3.11. It is worth noting that all operations listed in this table are **convolutional** operations.

One characteristic of both DCNNs is that once a signal is forwarded through one operation, this signal is batch normalized (see Section 2.1.5). For convolutions, such batch normalization step is defined, for each layer i, as the following:

$$h(x) = \beta_i + \frac{g(W_i * x) + \mu_i}{\sigma_i}, \tag{5.4}$$

where β is the batch normalization offset (role of the biases), W is the convolutional kernel, g is the non-linear function applied to the convolution (ReLU activation), μ is the accumulated mean of the batch and σ is the accumulated standard deviation of the batch. In the Equation 5.4, two variables are updated via backpropagation, the values of the kernel (W) and the offset (β).

To address the hypotheses 5.1 and 5.2 two groups of experiments are carried out. Each one is conducted using the two architectures (Incep. Res. v1 and Incep. Res. v2) and the two training mechanisms (Siamese and Triplet). The first one addresses more specifically Hypothesis 5.2 and it tests if DSUs are embedded in biases only. For this one, only the corresponding βs are updated during the DSU training. The second group assess if the feature detectors are also domain specific. To address that both, W and β, are updated during the DSU training. To train such DSUs, the same procedure adopted for training the prior DCNN is adopted. The RMSProp optimizer is used as a solver[1] with mini-batches of 90 samples. The learning rate is kept to 0.1 for 65 epochs. Then it was decreased to 0.01 for 15 epochs and finally decreased

[1] tensorflow.org/api_docs/python/tf/train/RMSPropOptimizer

Layers considered as θ_t	**Incep. Res. v1**	**Incep. Res. v2**
$\theta_{t[1-1]}$	Conv2d_1a_3x3	Conv2d_1a_3x3
$\theta_{t[1-2]}$	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1
$\theta_{t[1-4]}$	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3, Conv2d_4b_3x3	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3
$\theta_{t[1-5]}$	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3, Conv2d_4b_3x3, Block35	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3, Mixed_5b
$\theta_{t[1-6]}$	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3, Conv2d_4b_3x3, block35, Mixed_6a	Conv2d_1a_3x3, Conv2d_2a_3x3, Conv2d_2b_3x3, Conv2d_3b_1x1, Conv2d_4a_3x3, Mixed_5b, block35

Table 5.1 – List of variables adapted for each one the tested architectures

once more to 0.001 until the end of the training. In total all the DCNNs were trained for 250 epochs. This procedure is carried out at **training time**. At **enrollment time**, VIS images ($\mathscr{D}_s$) are forwarded to the VIS specific DCNN and then the embeddings are stored as is. Finally at **scoring time**, images from the target domain ($\mathscr{D}_t$) are forwarded first to its domain specific set of feature detectors (θ_t) and then to the domain independent set of feature detectors (θ_s). Those embeddings are directly compared with the enrolled ones using cosine similarity defined in equation 5.5.

$$d(\phi(x_s),\phi(x_t)) = \frac{\phi(x_s)\cdot\phi(x_t)}{||\phi(x_s)||||\phi(x_t)||} \tag{5.5}$$

5.3 Experiments and Analysis

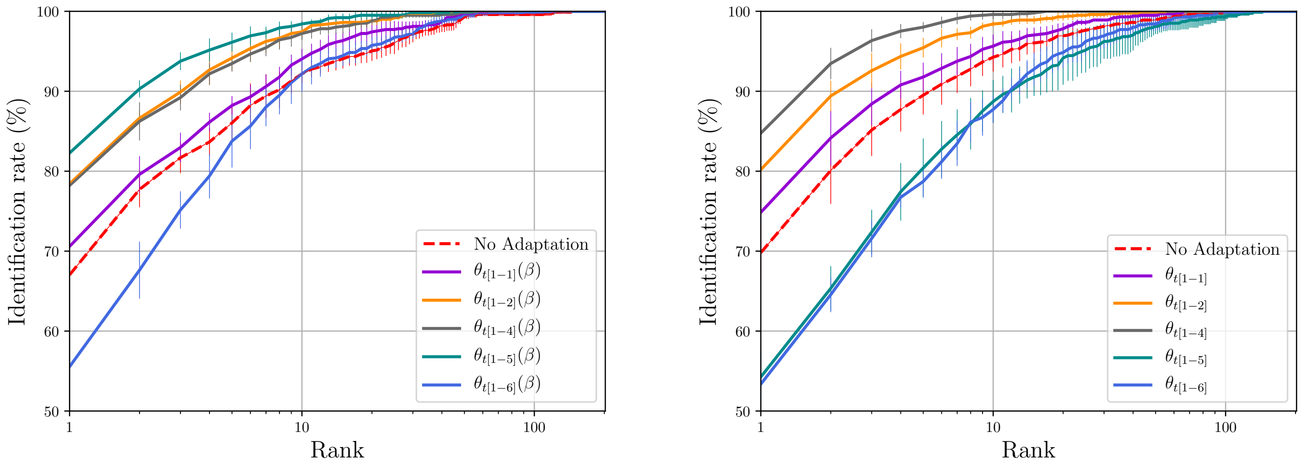
In this section the experiments assessing the two hypotheses using two different DCNNs and two different training mechanisms are presented. To make it easier the interpretation of the recognition rates, all the tables in this section (Tables 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11) are split in three parts. **FR Baselines** corresponds to all FR baselines described in the Section 3.1. **Reproducible Baselines** corresponds to all HFR baselines described in the Section 3.2 and it was implemented or integrated in the context of this work. Furthermore, the best recognition rates reported in Chapter 4 are also reported. Finally, **Non Reproducible Baselines** corresponds to HFR baselines whose source code was not made publicly available and its average rank one recognition rate was cherry picked directly from its corresponding publication.

5.3.1 Visible Light to Sketches

In this subsection it is described experiments with two sketch databases: CUHK-CUFS and CUHK-CUFSF.

CUHK-CUFS

Figure 5.4 (a) presents the CMC curves with adaptation of the biases only for the **Incep. Res v2 using the Siamese Networks**. Such DCNN, with no adaptation, has an average rank one recognition rate of 67.03%. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to get this benchmark improved to $\approx$ 70%. The biases adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improves the average rank one recognition rate to $\approx$ 78% for both. Experiments with $\theta_{t[1-5]}$ get its best average rank one recognition rate with 82.2%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to $\approx$ 55%. A possible overfitting can be suggested for $\theta_{t[1-6]}$. Figure 5.5 shows the plot of the average rank one recognition rates and the number of parameters learnt as a function of $\theta_{t[1-n]}$ for the Siamese Networks using the Incep. Res. v2 as a basis. It is possible to observe a drop, in terms of average rank one recognition rate, from $\theta_{t[1-5]}$ to $\theta_{t[1-6]}$ when the number of parameters learnt drastically grows (from 928 to 3328). Due to this increasing, a possible overfitting can be suggested for $\theta_{t[1-6]}$. Figure 5.6, shows the training loss ($\mathscr{L}$) for the first fold of the $\theta_{t[1-6]}$ training. It is possible to observe that $\mathscr{L}$ quickly converges and stabilizes to 0.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.4 – CUFS - Average CMC curves (with error bars) for the adaptation of biases only

As in the other chapters this strategy is implemented in the thesis software and can be triggered with the following bash commands:

```
 $ bob bio htface htface_baseline
```

```
   htface_idiap_msceleb_inception_v2_centerloss_gray cuhk-cufs --preprocess-
   training-data # generating prior
$ bob bio htface htface_train_dsu
   siamese_inceptionv2_first_layer_betas_nonshared_batch_norm cuhk-cufs  #
   Training DSU
$ bob bio htface htface_baseline
   siamese_inceptionv2_first_layer_betas_nonshared_batch_norm cuhk-cufs
```

These command lines demonstrate just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

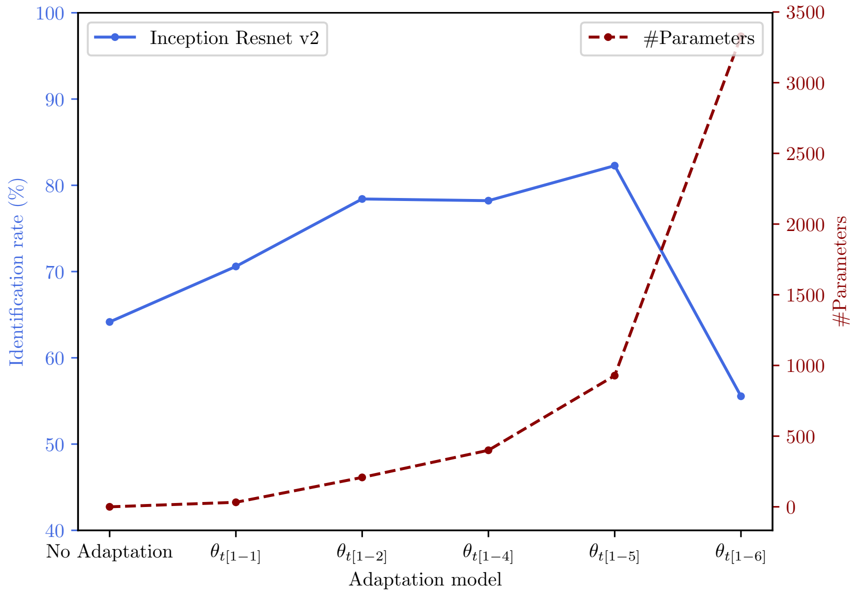


Figure 5.5 – Average rank one recognition rate vs number of parameters learnt

The same trend can be observed for **Incep. Res. v1 using Siamese Networks** (see Figure 5.4 (b)). The average recognition rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 69.8%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}$ improves the average rank one recognition rate to ≈ 74%. For $\theta_{t[1-2]}$ it was achieved ≈ 80%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 84.7%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to ≈ 54% and ≈ 53%, respectively. In this case, the number of parameters learnt drastically grows from 656 ($\theta_{t[1-4]}(\beta)$) to 1,616 ($\theta_{t[1-5]}(\beta)$) and 2,640 ($\theta_{t[1-6]}(\beta)$). Due to this increasing, the same overfitting hypothesis can be suggested for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$.

Using the thesis software this strategy can be triggered with the following bash commands:

[2] https://gitlab.idiap.ch/bob/bob.thesis.tiago

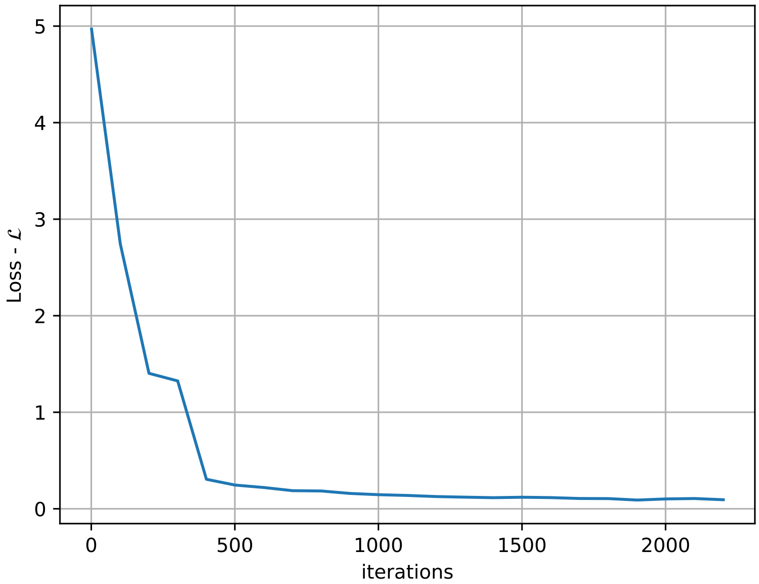


Figure 5.6 – CUHK-CUFS - Training loss for $\theta_{t[1-6]}$ using Siamese Networks. Check points at every 100 steps.

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray cuhk-cufs --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm cuhk-cufs  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm cuhk-cufs
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

The same trends are observed using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rates are improved to ≈ 75% and ≈ 72% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 80% and ≈ 78% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rates are improved to ≈ 80% and ≈ 79% respectively. Using Incep. Res. v1 the average rank one recognition rate drops to ≈ 39% and ≈ 24% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate improved to ≈ 83% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 59% for $\theta_{t[1-6]}$.

With this set of experiments it was possible to observe that the adaptation of batch normalization offsets (βs) improved recognition rates. This confirms both Hypotheses, that there are DSUs and such DSUs are embedded in the biases (β). To investigate if there are domain specific feature detectors, in the next set of experiments the same experimental procedure is performed, but instead of adapting only β, it is adapted β and W (Equation 5.4).

Figure 5.7 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 67.03%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}(\beta + W)$ in the plots) it is possible to improve this benchmark to ≈ 74%. The adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improves the average rank one recognition rates to ≈ 87% and ≈ 89% respectively. Experiments with $\theta_{t[1-5]}$ get its best average rank one recognition rate with 97.7%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 60%. The same aforementioned overfitting can be suggested for $\theta_{t[1-6]}$.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_gray cuhk-cufs --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv2_first_layer_nonshared_batch_norm cuhk-cufs  # Training
  DSU
$ bob bio htface htface_baseline
  siamese_inceptionv2_first_layer_nonshared_batch_norm cuhk-cufs
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

The same trend can be observed for **Incep. Res. v1 using the Siamese Networks** (see Figure 5.7 (b)). The average recognition rate increases once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 69.8%. The adaptation of β and W for $\theta_{t[1-1]}$ leads to an average rank one recognition rate of ≈ 76%. For $\theta_{t[1-2]}$ it is achieved ≈ 89%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 90.7%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 56% and ≈ 44%, respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

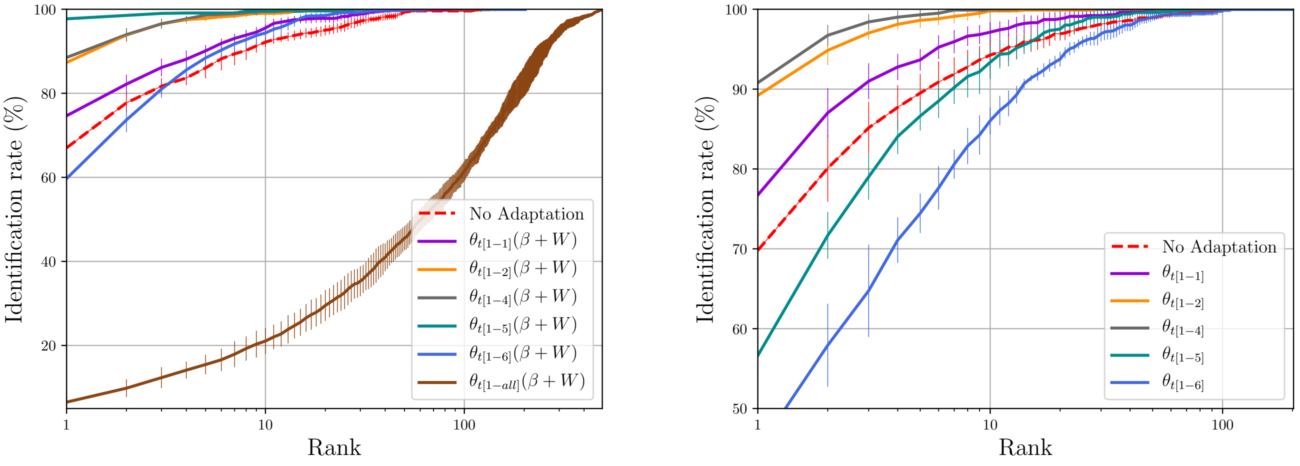
```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray cuhk-cufs --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_nonshared_batch_norm cuhk-cufs  # Training
  DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_nonshared_batch_norm cuhk-cufs
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

As before, with the Siamese Networks, the same trends are observed using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate improves to ≈ 73% and ≈ 75% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 77% and ≈ 78% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rates are improved to ≈ 80% and ≈ 81% respectively. Using Incep. Res. v1 the average rank one recognition rates drop to ≈ 51% and ≈ 46% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate improved to 81.5% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 51% for $\theta_{t[1-6]}$.

With these set of experiments it was possible to observe that, despite the adaptation of only the $\beta's$ increase the recognition rates, the joint adaptation of β and W increases even more such figure of merit. It is possible to suggest that there are domain specific feature detectors, therefore confirming once more Hypothesis 5.1.

From the experiments above, the best average rank one recognition rate is achieved with Incep. Res v2 trained using Siamese Networks. The model $\theta_{t[1-5]}$ achieved an average recognition rate of 97.72%(1.0).



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.7 – CUFS - Average CMC curves (with error bars) for the adaptation of kernel and biases

Table 5.2 shows the average rank one recognition rate comparing different configurations of the DSU approach (the one with the highest recognition rate for each setup) with the Reproducible and the Non Reproducible baselines.

Comparing the DSU approach with P-RS, in terms of average rank one recognition rate, the difference is ≈ 1%, which represents ≈ 2 miss classifications. The HFR approach implemented in P-RS is composed by a score a fusion of 180 different face recognition systems (6 systems with 30 bags each; see Chapter 2). Compared with the DSU approach, which is composed by only one system instead of 180 complex systems (several bags, different types of feature, different image processing algorithms), the difference of 2 miss classifications is marginal. The DSU approach presents slightly higher recognition rates than the Reproducible Baselines. For instance, the approach based on ISV presents an average rank one recognition rate of 96.53% and a variation of GFK presents 93.27%.

Table 5.2 – CUHK-CUFS - Average rank one recognition rate under different DSU training.

#	FR Algorithm	Average rank one rec. rate (std. dev.)
FR Baselines		
1	Incep. Res. v1 - gray scaled	72.57%(3.7)
2	Incep. Res. v2 - gray scaled	80.29%(1.5)
Reproducible Baselines		
3	MLBP [Liao et al., 2009]	62.27%(3.8)
4	MultiScale feat. [Liu et al., 2012]	64.16%(2.5)
5	GFK [Gong et al., 2012; Sequeira et al., 2017]	93.27%(1.4)
6	ISV (see Table 4.1)	96.53% (0.8)
Non Reproducible Baselines		
7	P-RS as in [Klare and Jain, 2013]	**99%(n/a)**
8	Face VACS in [Klare and Jain, 2013]	89%(n/a)
DSU Adapt β		
9	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	84.7% (3.6)
10	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	82.2% (1.7)
11	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	80.5% (2.9)
12	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	82.9% (2.3)
DSU Adapt $\beta + W$		
13	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	90.7% (1.6)
14	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	**97.7% (1.0)**
15	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	81.6% (2.4)
16	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	81.5% (2.9)

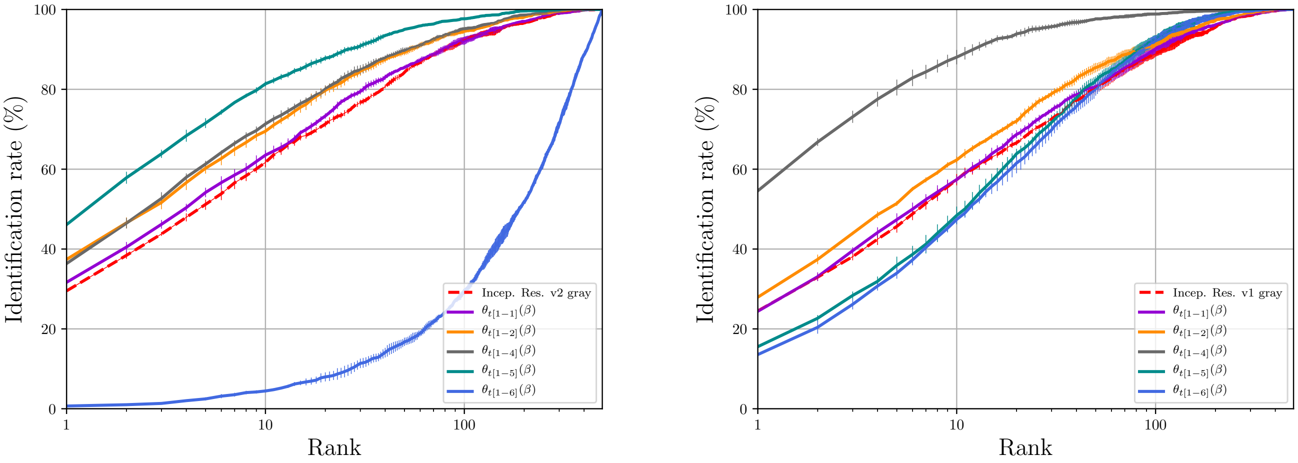
CUHK-CUFSF

Figure 5.8 (a) presents the CMC curves with adaptation of the biases only for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 29.51%. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to get this benchmark improved to $\approx$ 32%. The biases adaptation for $\theta_{t[1-2]}$ improved this figure of merit to $\approx$ 37%. Adapting $\theta_{t[1-4]}$ it is improved to $\approx$ 58% (its best). Adapting $\theta_{t[1-5]}$ such figure of merit decreases to 46% For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to $\approx$ 1%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_gray cuhk-cufsf --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv2_first_layer_betas_nonshared_batch_norm cuhk-cufsf  #
  Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv2_first_layer_betas_nonshared_batch_norm cuhk-cufsf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.8 – CUFSF - Average CMC curves (with error bars) for the adaptation of biases only

It is possible to observe the same trends for **Incep. Res. v1 using Siamese Networks** (see 5.8 (b)). The average recognition rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 24.49%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}(\beta)$ leads to an average rank one recognition rate of ≈ 25%. For $\theta_{t[1-2]}$ such figure of merit is increased to ≈ 28%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 54.57%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drop drastically to ≈ 16% and ≈ 14%, respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray cuhk-cufsf --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_betas_nonshared_batch_norm cuhk-cufsf  #
  Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_betas_nonshared_batch_norm cuhk-cufsf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

Training with **Triplet Networks** the same trends are observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rates improved to ≈ 24% and ≈ 35% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 34% and ≈ 35% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rate improves to ≈ 41% and ≈ 44%. Using Incep. Res. v1 the average rank one recognition rates drops to ≈ 12% and ≈ 7% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate is ≈ 44% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 27% for $\theta_{t[1-6]}$.

The same trends observed in the previous experiments are observed in this database. The adaptation of the batch normalization biases (β) only do improve the recognition rates confirming Hypothesis 5.2. In the next set of experiments it is investigated if there are domain specific feature detectors by adapting β and W (Equation 5.4).

Figure 5.9 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 29.51%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}(\beta + W)$ in the plots) it is possible to get this benchmark improved to ≈ 36%. The adaptation for $\theta_{t[1-2]}(\beta + W)$ improves the average rank one recognition rate to ≈ 61%. The best average rank one recognition rate is achieved with $\theta_{t[1-4]}(\beta + W)$ with 85.05%. With $\theta_{t[1-5]}$ the average rank one recognition rate decreases to 58.18%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 2%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray cuhk-cufsf --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_nonshared_batch_norm cuhk-cufsf  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_nonshared_batch_norm cuhk-cufsf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

The same trends can be observed for **Incep. Res. v1 training with Siamese Networks** (see Figure 5.9 (b)). The average recognition rate increases once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 24.49%. The adaptation of β and W for $\theta_{t[1-1]}$ and $\theta_{t[1-2]}$ leads to an average rank one recognition rates of ≈ 27% and ≈ 54% respectively. With $\theta_{t[1-4]}$ the average rank one recognition rate is increased to 84.45% (its best). Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 22% and ≈ 15%, respectively.

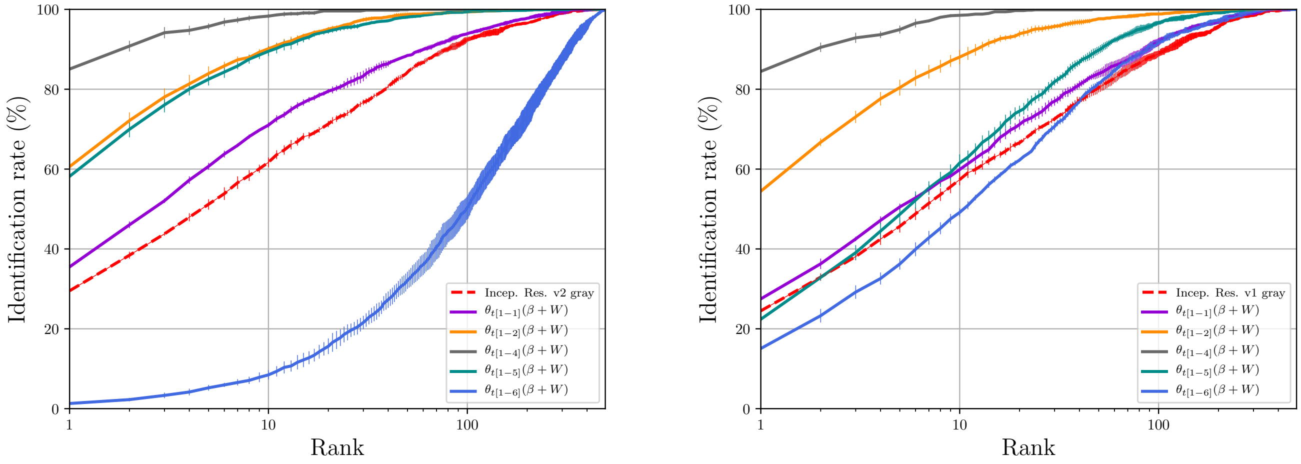
Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray cuhk-cufsf --preprocess-
```

```
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_nonshared_batch_norm cuhk-cufsf  # Training
  DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_nonshared_batch_norm cuhk-cufsf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

As before, with Siamese Networks, the same trends are observed using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate improves to ≈ 26% and ≈ 38% respectively. For $\theta_{t[1-2]}$ such benchmark is improved to ≈ 38% and ≈ 44% respectively. Using Incep. Res. v1 the average rank one recognition rate is improved to ≈ 53% for $\theta_{t[1-4]}$ and drastically drops to ≈ 41% and ≈ 20% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively. For Incep. Res. v2 the average rank one recognition rates are improved to 61.9% and ≈ 46% for $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ respectively and it drastically drops to ≈ 31% for $\theta_{t[1-6]}$.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.9 – CUFS - Average CMC curves (with error bars) for the adaptation of kernel and biases

With this set of experiments it was possible to observe that, despite the adaptation of only the βs increase the recognition rates, confirming Hypotheses 5.2, the joint adaptation of β and W increase even more such figure of merit. It is possible to suggest that there are domain specific feature detectors and such feature detectors need to be taken into account for the HFR task.

Table 5.3 shows the average rank one recognition rate comparing different configurations of DSU approach jointly with the **FR baselines**, **Reproducible baselines** and the **Paper baselines**. The best setup found is the Incep. Res. v2 trained with Siamese Networks (model $\theta_{t[1-5]}(\beta +$

Table 5.3 – CUHK-CUFSF - Average rank one recognition rate under different DSU training.

#	FR Algorithm	Average rank one rec. rate (std. dev.)
FR Baselines		
1	Incep. Res. v1 - gray scaled	24.49%(0.5)
2	Incep. Res. v2 - gray scaled	29.51%(0.7)
Reproducible Baselines		
3	MLBP in [Liao et al., 2009]	9.11%(1.7)
4	MultiScale feat. in [Liu et al., 2012]	6.76%(0.7)
5	GFK [Gong et al., 2012; Sequeira et al., 2017]	41.01%(1.8)
6	ISV (see Table 4.2)	55.59%(1.2)
Non Reproducible Baselines		
7	TP-LBP [Wolf et al., 2008]	59.7%(not available)
8	CDFL [Jin et al., 2015]	81.3%(not available)
9	DEEPS [Galea, 2018]	**82.92%(1.25)**
10	LGMS [Galea, 2018]	78.19%(0.52)
DSU β		
11	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	81.88%(2.9)
12	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	42.3% (1.5)
13	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	41.21%(1.2)
14	Trip. Incep. Res. v2 $\theta_{t[1-4]}$	44.61%(2.9)
DSU β + W		
15	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	84.45%(3.4)
16	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	**85.05%(2.1)**
17	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	53.59%(8.6)
18	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	61.90%(0.8)

W)). Such model achieved an average rank one recognition rate of 85.05% with 2.1 of standard deviation. It is possible to observe that this model presents higher rank one recognition rate compared with Galea [2018]. With respect to the **Reproducible Baselines**, the DSU strategy performs substantially better.

5.3.2 Visible Light to NIR

In this subsection it is described experiments with four NIR databases: CASIA, NIVL, LDHF and FARGO.

CASIA

Figure 5.10 (a) presents the CMC curves with adaptation of the biases only for the Incep. Res. v2 using the Siamese Networks. Such DCNN, with no adaptation, presents an average rank one recognition rate of 73.80%. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to improve this benchmark to $\approx$ 77%. The biases adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improve the average rank one recognition rates to $\approx$ 83% and

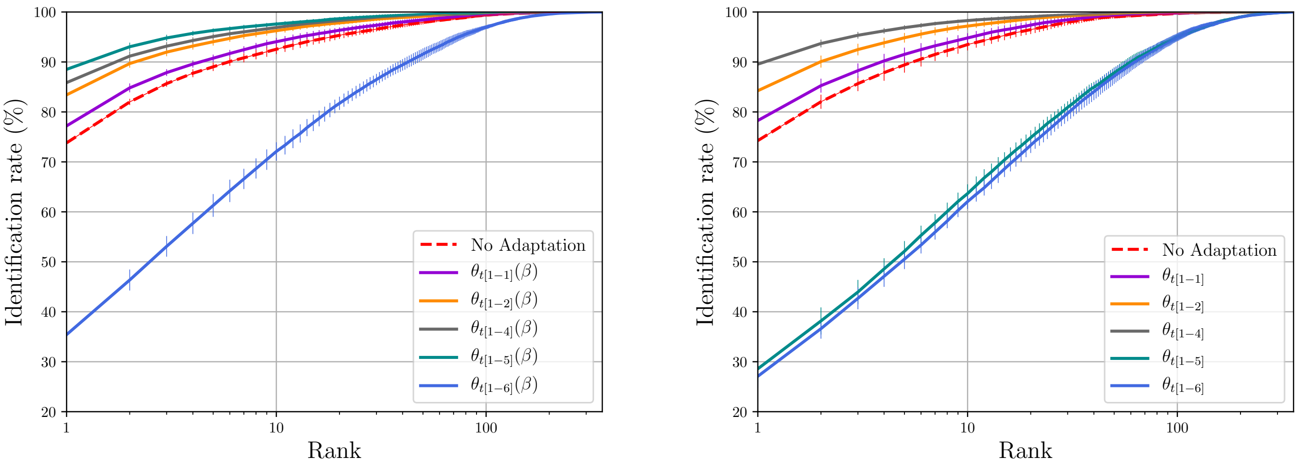
≈ 86% respectively. Experiments with $\theta_{t[1-5]}$ get its best average rank one recognition rate with 88.5%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 35%. The same overfitting hypothesis suggested before can be applied for $\theta_{t[1-6]}$.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_gray casia-nir-vis-2 --
  preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv2_first_layer_betas_nonshared_batch_norm casia-nir-vis-2  #
   Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv2_first_layer_betas_nonshared_batch_norm casia-nir-vis-2
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

The same trends can be observed for **Incep. Res. v1 using Siamese Networks** (see Figure 5.10 (b)). Average recognition rates increase once depth is increased. With no adaptation, such DCNN presents an average rank one recognition rate of 74.25%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}$ leads to an average rank one recognition rate of ≈ 78%. For $\theta_{t[1-2]}$ it is achieved ≈ 84%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 89.5%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 28% and ≈ 27%, respectively.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.10 – CASIA - Average CMC curves (with error bars) for the adaptation of biases only

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray casia-nir-vis-2 --
  preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_betas_nonshared_batch_norm casia-nir-vis-2  #
   Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_betas_nonshared_batch_norm casia-nir-vis-2
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

In this particular experiment, the same trend **couldn't be observed** using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rates drops to ≈ 73% and ≈ 75% respectively. For $\theta_{t[1-2]}$ the it drops to ≈ 70% and ≈ 74% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rates is decreased to ≈ 64% and ≈ 60% respectively. Using Incep. Res. v1 the average rank one recognition rate drop to ≈ 52% and ≈ 68% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate drops to ≈ 57% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 15% for $\theta_{t[1-6]}$.

The same trends observed in the previous experiments are observed in this database. The adaptation of the batch normalization biases (β) only do improve the recognition rates confirming both Hypotheses. In the next set of experiments it is investigated if there are domain specific feature detectors by adapting β and W (Equation 5.4).

Figure 5.11 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 73.8%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}$ in the plots) it is possible to get this benchmark improved to ≈ 80%. The adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improves the average rank one recognition rates to ≈ 91% and ≈ 93% respectively. Experiments with $\theta_{t[1-5]}$ get its best average rank one recognition rate with 96.3%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 49%.

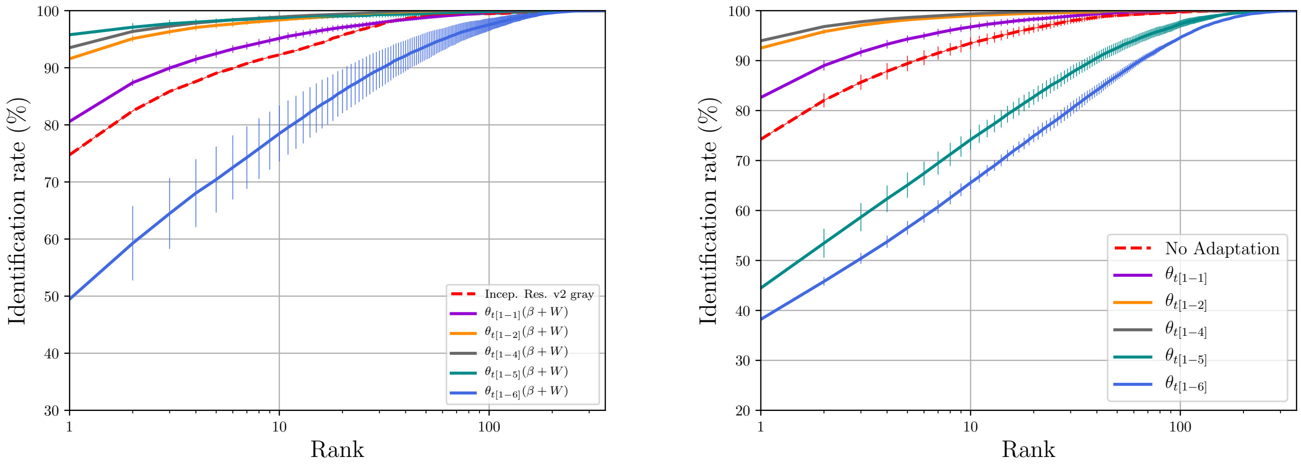
Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v2_centerloss_gray casia-nir-vis-2 --
  preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv2_first_layer_nonshared_batch_norm casia-nir-vis-2  #
  Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv2_first_layer_nonshared_batch_norm casia-nir-vis-2
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

The same trend can be observed for **Incep. Res. v1 trained with Siamese Networks** (see 5.11

(b)). The average recognition rates increases once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 74.25%. The adaptation of β and W for $\theta_{t[1-1]}$ leads to an average rank one recognition rate of ≈ 83%. For $\theta_{t[1-2]}$ it is achieved ≈ 92%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 93.9%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to ≈ 44% and ≈ 38%, respectively.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.11 – CASIA - Average CMC curves (with error bars) for the adaptation of biases and kernels

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray casia-nir-vis-2 --
  preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_nonshared_batch_norm casia-nir-vis-2  #
  Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_nonshared_batch_norm casia-nir-vis-2
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

As before, with Siamese Networks, it is also observed the same trends using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rates are improved to ≈ 75% and ≈ 76% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 76% and ≈ 79% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rate is improved to ≈ 88% and ≈ 89% respectively. Using Incep. Res. v1 the average rank one recognition rate drops to ≈ 50% and ≈ 49% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese Networks). For Incep. Res. v2 the average rank one recognition rate improves to

90.1% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 51% for $\theta_{t[1-6]}$.

With this set of experiments it was possible to observe that, despite the adaptation of only the βs increase the recognition rates, confirming Hypothesis 5.2, the joint adaptation of β and W increase even more such figure of merit, reinforcing both Hypotheses. It is possible to suggest that there are domain specific feature detectors and such feature detectors need to be taken into account for the HFR task.

Table 5.4 – CASIA - Average rank one recognition rate under different Face Recognition systems

#	FR Algorithm	Average rank one rec. rate (std. dev.)
FR Baselines		
1	Incep. Res. v1 - gray	74.25%(1.3)
2	Incep. Res. v2 - gray	73.80%(1.2)
Reproducible Baselines		
3	MLBP in [Liao et al., 2009]	70.33%(1.2)
4	Multiscale Feat. in [Liu et al., 2012]	67.54%(1.7)
5	GFK [Gong et al., 2012; Sequeira et al., 2017]	26.98%(0.9)
6	ISV (see Table 4.3	72.67%(1.8)
Non Reproducible Baselines		
7	IDR in [He et al., 2017]	95.82%(0.7)
8	CDL in [Wu et al., 2017]	**98.62%(0.2)**
9	WCNN in [He et al., 2018]	**98.70%(0.3)**
10	TRIVET in [Liu et al., 2016]	95.74%(0.5)
DSU Adapt β		
11	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	89.5% (1.2)
12	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	88.5% (1.1)
13	Trip. Incep. Res. v1 $\theta_{t[1-2]}$	70.0% (1.6)
14	Trip. Incep. Res. v2 $\theta_{t[1-1]}$	73.8% (2.0)
DSU Adapt $\beta + W$		
15	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	93.9% (0.3)
16	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	**96.3% (0.4)**
17	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	87.7% (1.5)
18	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	90.1% (2.9)

Table 5.4 shows the average rank one recognition rate comparing different configurations of DSU approach jointly with the **FR baselines**, **Reproducible baselines** and **Non Reproducible baselines**. In terms of average rank one recognition rate, different setups of the DSU proposed approach are substantially better than the Reproducible baselines. The best DSU setup (96.3% with the model $\theta_{t[1-5]}(\beta + W)$ trained with Siamese Neural Networks and Incep. Res. v2), presents a slightly better recognition performance compared with the TRIVET system in Liu et al. [2016] (95.74%). However, the systems CDL[Wu et al., 2017] and WCNN [He et al., 2018] present a slight better average rank one recognition rate with 98.76% and 98.70% respectively.

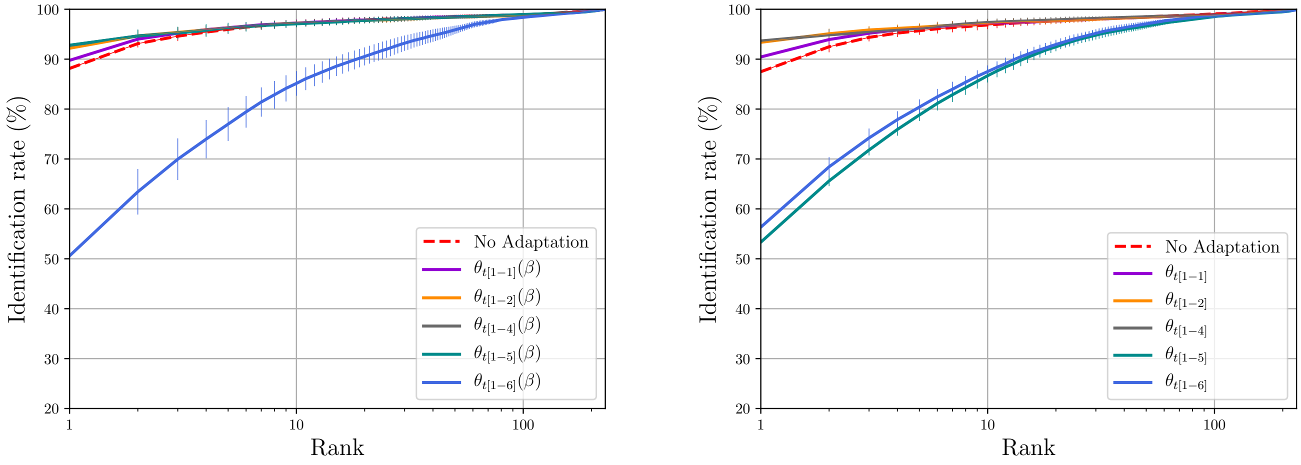
NIVL

Figure 5.12 (a) presents the CMC curves with adaptation of the biases only for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, has an average rank one recognition rate of 88.14%. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to improve this benchmark to $\approx$ 89%. The biases adaptation for $\theta_{t[1-2]}$ improves the average rank one recognition to $\approx$ 92%. Adapting $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ improves this benchmark to 92.7% and 92.8% respectively. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to $\approx$ 51%. The same overfitting hypothesis suggested before can be verified for $\theta_{t[1-6]}$.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray nivl --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm nivl  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm nivl
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.12 – NIVL - Average CMC curves (with error bars) for the adaptation of biases only

The same trend can be observed for **Incep. Res. v1 using the Siamese Networks** (see Figure 5.12 (b)). The average recognition rate increases once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 87.48%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}(\beta)$ leads to an average rank one recognition rate of $\approx$ 90%. For

$\theta_{t[1-2]}$ it is achieved ≈ 93%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 93.4%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to ≈ 53% and ≈ 56%, respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray nivl --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm nivl  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm nivl
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

Training with **Triplet Networks** the same trends are observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate gets improved to ≈ 91% and ≈ 90% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 92% and ≈ 91% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rate for the Incep. Res. v1 decreased to ≈ 83% and improves to ≈ 92% Incep. Res. v2. Using Incep. Res. v1 the average rank one recognition rates drops to ≈ 12% and ≈ 14% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate decreases to ≈ 90% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 30% for $\theta_{t[1-6]}$.

The same trends observed before was observed for this database. The adaptation of the batch normalization offsets (β) only do improve the recognition rates confirming both Hypotheses. In the next set of experiments it is investigated if there are domain specific feature detectors by adapting β and W (Equation 5.4).

Figure 5.13 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, has an average rank one recognition rate of 88.14%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}(\beta + W)$ in the plots) it is possible to get this benchmark improved to ≈ 91%. The adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improves the average rank one recognition rates to ≈ 94% and ≈ 94% respectively. Experiments with $\theta_{t[1-5]}$ get its best average rank one recognition rate with 94.5%. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 59%.

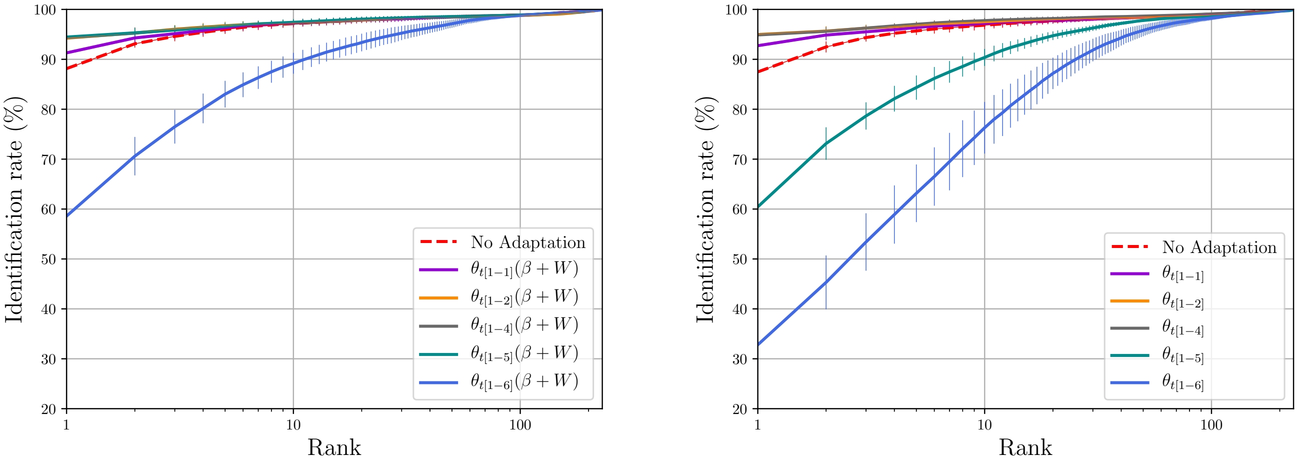
Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray nivl --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_nonshared_batch_norm nivl  # Training DSU
$ bob bio htface htface_baseline
```

```
    siamese_inceptionv2_first_layer_nonshared_batch_norm nivl
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta+W)$. To check how to train other DSUs, check[2].

As before, the same trends are observed for **Incep. Res. v1 trained with Siamese Networks** (see Figure 5.13 (b)). The average recognition rate increases once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 87.48%. The adaptation of β and W for $\theta_{t[1-1]}$ and $\theta_{t[1-2]}$ leads to average rank one recognition rates of 92.7% and 94.8% respectively. The average rank one recognition rate slightly increases to 94.9% for $\theta_{t[1-4]}$ (its best). Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drop drastically to $\approx 60\%$ and $\approx 32\%$, respectively.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.13 – NIVL - Average CMC curves (with error bars) for the adaptation of kernel and biases

Using the thesis software this strategy can be triggered with the following bash commands:

```
  $ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray nivl --preprocess-training-data # generating prior
  $ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_nonshared_batch_norm nivl  # Training DSU
  $ bob bio htface htface_baseline siamese_inceptionv1_first_layer_nonshared_batch_norm nivl
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta+W)$. To check how to train other DSUs, check[2].

As before, with Siamese Networks, the same trends using **Triplet Networks** as training strategy are observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rates gets improved to ≈ 92% and ≈ 90% respectively. For $\theta_{t[1-2]}$ such benchmark stabilizes to ≈ 92% and ≈ 90% respectively. Using Incep. Res. v1 the average rank one recognition rate drops to ≈ 89% for $\theta_{t[1-4]}$ and drastically drops to ≈ 48% and ≈ 45% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively. For Incep. Res. v2 the average rank one recognition rates are improved to ≈ 92% for $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ and it drastically drops to ≈ 54% for $\theta_{t[1-6]}$.

With this set of experiments it was possible to observe that, despite the adaptation of only βs increase the recognition rates, the joint adaptation of β and W **slightly increased** such figure of merit confirming both Hypotheses.

Table 5.5 – NIVL - Average rank one recognition rate under different Face Recognition systems

#	FR Algorithm	Average Rank one rec. rate (std. dev.)
FR Baselines		
1	Incep. Res. v1 - gray	91.09%(0.3)
2	Incep. Res. v2 - gray	88.14%(0.6)
Reproducible Baselines		
3	MLBP in [Liao et al., 2009]	85.35%(1.1)
4	Multiscale Feat. in [Liu et al., 2012]	90.34%(1.3)
5	ISV (see Table 4.4)	76.73%(2.0)
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	63.08%(2.2)
DSU Adapt β		
5	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	93.4%(1.3)
6	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	92.8%(1.2)
7	Trip. Incep. Res. v1 $\theta_{t[1-2]}$	92.0%(0.8)
8	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	91.9%(1.8)
DSU Adapt $\beta + W$		
9	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	**94.9%(1.0)**
10	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	**94.5%(1.2)**
11	Trip. Incep. Res. v1 $\theta_{t[1-1]}$	91.9%(1.6)
12	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	92.2%(1.4)

Table 5.5 shows the average rank one recognition rate comparing different configurations of DSU approach jointly with the **FR baselines**, **Reproducible baselines** and **Non Reproducible baselines**. As mentioned in Chapter 2.3.1, there is no official evaluation protocol for this database. In terms of average rank one recognition rate the DSU approach is slightly better than the Reproducible baselines. The best setup is the model $\theta_{t[1-4]}$ trained with Siamese Neural Networks using the Incep. Res. v1 as a basis and achieved a recognition rate of 94.9%.

LDHF

Table 5.6 presents the average rank one recognition rates with adaptation of the **biases** only for different stand-offs. The same trends observed for the other VIS to NIR databases can be observed for this one, for all base DCNNs (Incep. Res. v1 and Incep. Res. v2) and for all base trainers (Siamese and Triplet Networks).

Analysing the **1m** stand-off it is possible to observe an improvement from 94.8% to 99.6% using the **Incep. Res. v1 and Siamese Networks** for $\theta_{t[1-1]}$. For $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ this value decreases to 97.6% and 98.4% respectively. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 34.0% and 30.4%. Analysing the **60m** stand-off it is possible to observe an impressive improvement from 78.8% to 94.0% using the Incep. Res. v1 and Siamese Networks for $\theta_{t[1-1]}$ and to 94.4% for $\theta_{t[1-2]}$. For $\theta_{t[1-4]}$ this value decreases to 92.7%. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 28.4% and 26.4%. For **100m** stand-off it is possible to observe an improvement from 28.4% to 45.2% using the Incep. Res. v1 and Siamese Networks for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-4]}$ with 68.0%. Finally for **150m** stand-off it is possible to observe an improvement from 4.8% to 19.2% using the Incep. Res. v1 and Siamese Networks for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-4]}$ with 22.8%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray ldhf --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm ldhf  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm ldhf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check^2.

Analysing the **1m** stand-off it is possible to observe an improvement from 92.8% to 97.6% using the **Incep. Res. v2 and Siamese Networks** for $\theta_{t[1-1]}$. For $\theta_{t[1-2]}$, $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ this value increases to 96.4%, 96.0% and 94.8% respectively. Finally, for $\theta_{t[1-6]}$ this values drops to 15.6%. Analysing the **60m** stand-off it is possible to observe an impressive improvement from 75.6% to 87.2% for $\theta_{t[1-1]}$ and to 90.8% for $\theta_{t[1-2]}$. For $\theta_{t[1-4]}$ this value decreases to 83.6%. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 86.0% and 9.2%. For **100m** stand-off it is possible to observe an improvement from 9.6% to 27.6% for $\theta_{t[1-1]}$. The best recognition for this stand-off rate is achieved with $\theta_{t[1-5]}$ with 51.6%. Finally for **150m** stand-off it is possible to observe an improvement from 2.8% to 13.2% for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-5]}$ with 21.2%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray ldhf --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm ldhf  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm ldhf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other

DSUs, check[2].

The same trends are followed by using Triplet Networks as a training method as can be observed in Table 5.6.

With this set of experiments it was possible to observe that the adaptation of batch normalization offsets (βs) improved recognition rates for all stand-offs. This confirms both Hypotheses, that there are DSUs and such DSUs are embedded in the biases (β). To investigate if there are domain specific feature detectors, the next set of experiments the same experimental procedure is performed, but instead of adapting only β, it is adapted β and W (Equation 5.4).

Table 5.6 – LDHF - average rank one recognition rates under different stand-offs **adapting β only**

#	FR Algorithm	1m	60m	100m	150m
FR Baselines					
1	Incep. Res. v1 - gray	94.8%(2.0)	78.0%(4.4)	28.4%(1.5)	4.8%(1.6)
2	Incep. Res. v2 - gray	92.8%(2.7)	75.6%(2.9)	9.6%(1.5)	2.8%(1.6)
Reproducible Baselines					
3	MLBP in [Liao et al., 2009]	67.2%(7.0)	23.2%(3.0)	10.0%(2.8)	6.0%(1.8)
4	Multiscale Feat. in [Liu et al., 2012]	74.4%(3.4)	43.2%(3.7)	22.0%(4.5)	14.8%(3.0)
5	ISV (see Table 4.5)	96.0%(1.3)	59.2%(6.0)	37.2%(7.4)	14.4%(6.6)
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	73.6%(4.3)	31.2%(7.2)	12.0%(2.8)	2.8%(3.0)
Siamese Networks training					
7	Incep. Res. v1 $\theta_{t[1-1]}$	**99.6%(0.8)**	94.0%(3.3)	45.2%(4.1)	19.2%(3.0)
8	Incep. Res. v1 $\theta_{t[1-2]}$	97.6%(0.8)	**94.4%(3.2)**	**68.0%(3.3)**	16.8%(2.7)
9	Incep. Res. v1 $\theta_{t[1-4]}$	98.4%(0.8)	92.8%(3.7)	59.6%(3.4)	**22.8%(2.7)**
10	Incep. Res. v1 $\theta_{t[1-5]}$	34.0%(4.7)	28.4%(4.4)	22.8%(2.0)	13.2%(3.2)
11	Incep. Res. v1 $\theta_{t[1-6]}$	30.4%(2.9)	26.4%(4.1)	21.6%(1.5)	16.8%(3.5)
12	Incep. Res. v2 $\theta_{t[1-1]}$	97.6%(1.5)	87.2%(6.0)	27.6%(3.4)	13.2%(2.4)
13	Incep. Res. v2 $\theta_{t[1-2]}$	96.4%(2.3)	90.8%(2.7)	33.6%(4.8)	14.4%(1.5)
14	Incep. Res. v2 $\theta_{t[1-4]}$	96.0%(1.2)	83.6%(3.8)	39.2%(5.8)	14.8%(4.1)
15	Incep. Res. v2 $\theta_{t[1-5]}$	94.8%(1.6)	86.0%(3.3)	51.6%(6.5)	21.2%(1.0)
16	Incep. Res. v2 $\theta_{t[1-6]}$	15.6%(3.4)	9.2%(1.6)	9.6%(1.4)	10.0%(2.53)
Triplet Networks training					
17	Incep. Res. v1 $\theta_{t[1-1]}$	99.1%(0.4)	94.8%(3.7)	47.2%(2.7)	25.2%(2.7)
18	Incep. Res. v1 $\theta_{t[1-2]}$	98.4%(1.5)	88.0%(1.8)	57.2%(8.6)	21.2%(4.0)
19	Incep. Res. v1 $\theta_{t[1-4]}$	93.6%(3.9)	70.0%(8.8)	37.6%(6.4)	12.0%(2.8)
20	Incep. Res. v1 $\theta_{t[1-5]}$	34.4%(3.4)	30.4%(3.4)	16.4%(4.8)	12.0%(1.8)
21	Incep. Res. v1 $\theta_{t[1-6]}$	33.6%(6.6)	23.6%(4.4)	16.8%(2.7)	14.8%(3.7)
22	Incep. Res. v2 $\theta_{t[1-1]}$	92.4%(7.1)	69.2%(17.2)	29.2%(6.0)	11.2%(2.4)
23	Incep. Res. v2 $\theta_{t[1-2]}$	96.4%(2.6)	80.4%(11.5)	30.4%(7.9)	17.2%(3.0)
24	Incep. Res. v2 $\theta_{t[1-4]}$	94.4%(3.8)	76.0%(10.5)	28.0%(4.7)	19.2%(1.6)
25	Incep. Res. v2 $\theta_{t[1-5]}$	64.0%(8.5)	48.4%(16.5)	28.0%(8.6)	19.2%(2.7)
26	Incep. Res. v2 $\theta_{t[1-6]}$	14.0%(3.3)	16.4%(1.4)	12.4%(2.3)	12.8%(4.3)

Table 5.7 presents the average rank one recognition rates with adaptation of the $W + \beta$ for different stand-offs. The same trends observed for the other VIS to NIR databases can be observed for this one, for all base DCNNs (Incep. Res. v1 and Incep. Res. v2) and for all base trainers (Siamese and Triplet Networks).

Analysing the **1m** stand-off it is possible to observe an improvement from 94.8% to 100.0% using the **Incep. Res. v1 and Siamese Networks** for $\theta_{t[1-1]}$ (the highest for this experiment). For $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ this value decreases to 98.0% and 98.4% respectively. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 33.2% and 31.1%. Analysing the **60m** stand-off it is possible to observe an impressive improvement from 78.8% to 90.8% for $\theta_{t[1-1]}$ and to 98.0% for $\theta_{t[1-2]}$. For $\theta_{t[1-4]}$ this value decreases to 92.8%. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 28.8% and 24.4%. For **100m** stand-off it is possible to observe an improvement from 28.4% to 51.6% using the Incep. Res. v1 and Siamese Networks for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-4]}$ with 59.6%. Finally for **150m** stand-off it is possible to observe an improvement from 2.8% to 21.6% using the Incep. Res. v1 and Siamese Networks for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-4]}$ with 22.8%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray ldhf --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_nonshared_batch_norm ldhf  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_nonshared_batch_norm ldhf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

Analysing the **1m** stand-off it is possible to observe an improvement from 92.8% to 99.2% using the **Incep. Res. v2 and Siamese Networks** for $\theta_{t[1-1]}$ (the highest for this experiment). For $\theta_{t[1-2]}$, $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ this value decreased to 96.8%, 95.6% and 86.0% respectively. Finally, for $\theta_{t[1-6]}$ this values drops to 13.6%. Analysing the **60m** stand-off it is possible to observe an impressive improvement from 75.6% to 85.2% for $\theta_{t[1-1]}$ and to 84.0% for $\theta_{t[1-2]}$. For $\theta_{t[1-4]}$ this value decreased to 82.0%. Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ this values drops to 78.4% and 12.4% respectively. For **100m** stand-off it is possible to observe an improvement from 9.6% to 40.4% for $\theta_{t[1-1]}$. The best recognition for this stand-off rate is achieved with $\theta_{t[1-5]}$ with 52.8%. Finally for **150m** stand-off it is possible to observe an improvement from 2.8% to 12.8% for $\theta_{t[1-1]}$. The best recognition rate is achieved with $\theta_{t[1-2]}$ with 21.2%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray ldhf --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_nonshared_batch_norm ldhf  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_nonshared_batch_norm ldhf
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

The same trends are observed by using Triplet Networks as a training method as can be observed in Table 5.7.

With this set of experiments it was possible to confirm both hypotheses. Furthermore, it was possible to observe similar recognition rates between β and $\beta + W$ adaptations. This is particularly advantageous from the storage points of view. For instance, in the experiments using Incep. Res. v1 trained with Siamese Networks the model $\theta_{t[1-1]}(\beta)$ presents an average rank one recognition rate of 99.6%. Such model corresponds to the learning of only 32 new free parameters.

Table 5.7 – LDHF - average rank one recognition rates under different stand-offs **adapting** $\beta + W$

#	FR Algorithm	1m	60m	100m	150m
FR Baselines					
1	Incep. Res. v1 - gray	94.8%(2.0)	78.0%(4.4)	28.4%(1.5)	4.8%(1.6)
2	Incep. Res. v2 - gray	92.8%(2.7)	75.6%(2.9)	9.6%(1.5)	2.8%(1.6)
Reproducible Baselines					
3	MLBP in [Liao et al., 2009]	67.2%(7.0)	23.2%(3.0)	10.0%(2.8)	6.0%(1.8)
4	Multiscale Feat. in [Liu et al., 2012]	74.4%(3.4)	43.2%(3.7)	22.0%(4.5)	14.8%(3.0)
5	ISV (see Table 4.5)	96.0%(1.3)	59.2%(6.0)	37.2%(7.4)	14.4%(6.6)
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	73.6%(4.3)	31.2%(7.2)	12.0%(2.8)	2.8%(3.0)
Siamese Networks training					
7	Incep. Res. v1 $\theta_{t[1-1]}$	**100.0%(0.0)**	90.8%(2.4)	51.6%(3.0)	21.6%(1.5)
8	Incep. Res. v1 $\theta_{t[1-2]}$	98.0%(0.1)	**98.0%(0.3)**	56.0%(4.1)	18.8%(2.7)
9	Incep. Res. v1 $\theta_{t[1-4]}$	98.4%(0.8)	92.8%(3.7)	**59.6%(3.4)**	**22.9%(2.0)**
10	Incep. Res. v1 $\theta_{t[1-5]}$	33.2%(7.9)	28.8%(4.8)	22.4%(4.4)	12.4%(2.6)
11	Incep. Res. v1 $\theta_{t[1-6]}$	31.2%(6.5)	24.4%(3.0)	21.2%(2.7)	15.2%(3.2)
12	Incep. Res. v2 $\theta_{t[1-1]}$	99.2%(0.9)	85.2%(5.1)	40.4%(6.0)	12.8%(3.7)
13	Incep. Res. v2 $\theta_{t[1-2]}$	96.8%(1.0)	84.0%(2.5)	50.4%(3.9)	21.2%(5.6)
14	Incep. Res. v2 $\theta_{t[1-4]}$	95.6%(1.5)	82.0%(2.8)	51.6%(4.4)	19.2%(5.8)
15	Incep. Res. v2 $\theta_{t[1-5]}$	86.0%(3.8)	78.4%(3.4)	52.8%(6.8)	21.2%(3.7)
16	Incep. Res. v2 $\theta_{t[1-6]}$	13.6%(1.9)	12.4%(1.5)	14.4%(3.4)	10.8%(1.6)
Triplet Networks training					
17	Incep. Res. v1 $\theta_{t[1-1]}$	99.6%(0.8)	91.2%(5.3)	48.8%(6.9)	22.8%(3.0)
18	Incep. Res. v1 $\theta_{t[1-2]}$	96.0%(2.8)	70.8%(8.6)	42.0%(8.1)	18.0%(5.5)
19	Incep. Res. v1 $\theta_{t[1-4]}$	86.4%(5.6)	70.0%(6.1)	50.0%(4.6)	20.8%(2.0)
20	Incep. Res. v1 $\theta_{t[1-5]}$	38.8%(1.6)	31.2%(4.7)	20.4%(3.4)	14.0%(4.6)
21	Incep. Res. v1 $\theta_{t[1-6]}$	38.8%(2.7)	26.4%(3.4)	19.6%(5.0)	14.8%(1.0)
22	Incep. Res. v2 $\theta_{t[1-1]}$	92.4%(7.1)	69.2%(7.2)	29.2%(6.0)	11.2%(2.4)
23	Incep. Res. v2 $\theta_{t[1-2]}$	42.0%(11.2)	27.6%(5.6)	20.0%(5.2)	14.4%(1.5)
24	Incep. Res. v2 $\theta_{t[1-4]}$	41.6(11.412)	33.2(6.765)	22.8(2.713)	15.6(2.653)
25	Incep. Res. v2 $\theta_{t[1-5]}$	46.8%(6.2)	32.4%(3.4)	24.8%(4.3)	19.2%(4.8)
26	Incep. Res. v2 $\theta_{t[1-6]}$	14.8(3.709)	8.8(2.04)	12.4(1.96)	10.8(2.04)

Compared to all **Reproducible Baselines**, the approach based on DSU presented higher recognition rates for all stand-offs.

It is worth noting that the training set of this database contains VIS and NIR images from 1m stand-off only. Even when samples from 60m, 100m and 150m are not presented, it was possible to observe improvements in the recognition rates in these stand-off by only having a proper DSUs crafted for NIR. Furthermore, different from other databases, for this one the best model was the **Incep. Res. v1** trained with Siamese Networks

FARGO

Table 5.8 presents the FNMR@FMR=1%(dev) with adaptation of the **biases only** using both based architectures (Incep. Res. v1 and Incep. Res. v2) and both training methods (Siamese and Triplet networks). The same trends observed before can be observed in this experiment.

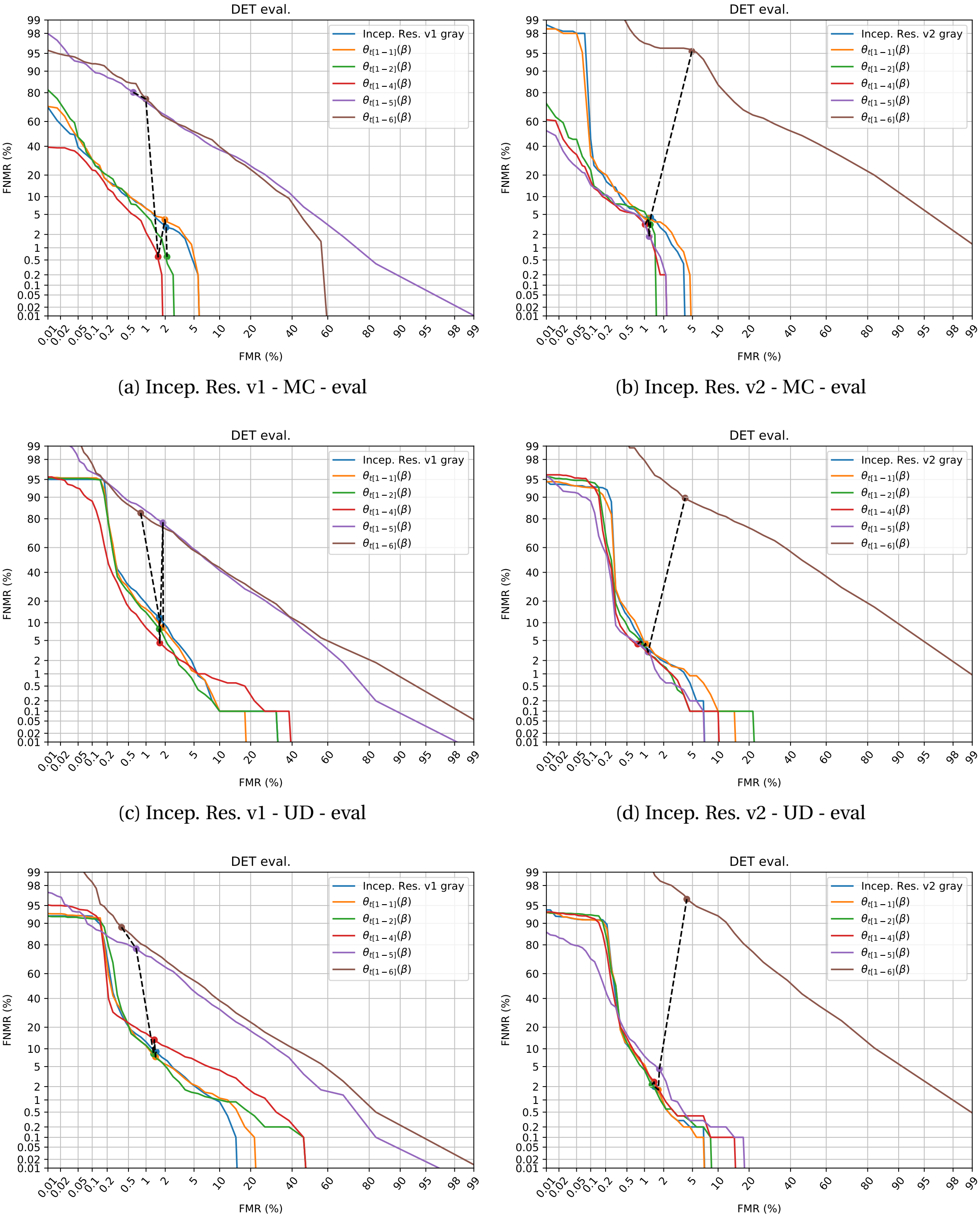
Under the controlled protocol **(mc)**, the adaptation of the biases presented a FNMR decrease in the evaluation set from 4.40% to 4.00% using **Incep. Res. v1 and Siamese Networks** $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is reduced to 0.6% and to 0.6% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 80.20% and 76.20% respectively. In the same experiment, using the protocol dark **(ud)**, the adaptation of the biases presented a FNMR reduction in the evaluation set from 11.90% to 8.40% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is decreased to 7.9% and to 4.5% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 77.60% and 83.10% respectively. Experiments using the protocol outside **(uo)**, the adaptation of the biases presented a FNMR reduction in the evaluation set from 9.00% to 7.40% using **Incep. Res. v1 and Siamese Networks** $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is increased to 8.3% and to 13.6% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 77.70% and 88.50% respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray fargo --preprocess-training-data # generating prior
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray fargo --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm fargo  # Training DSU
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm fargo  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm fargo
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm fargo
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

The same trends can be observed for **Incep. Res. v2 and Siamese Networks**. Under the controlled protocol **(mc)**, the adaptation of the biases presented a FNMR decrease in the evaluation set from 4.40% to 4.20% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is decreased to



(a) Incep. Res. v1 - MC - eval

(b) Incep. Res. v2 - MC - eval

(c) Incep. Res. v1 - UD - eval

(d) Incep. Res. v2 - UD - eval

(e) Incep. Res. v1 - UO - eval

(f) Incep. Res. v2 - UO - eval

Figure 5.14 – FARGO - **Adapting β only** - DET curves for verification experiments under the three illumination conditions MC (controlled), UD (dark) and UO (outdoor) trained with Siamese Networks. The column on the left presents DET curves using Incep. Res. v1 as a basis and the column on the right presents DET curves using Incep. Res. v2 as a basis.

3.20% and to 3.20% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ such figure of merit decreases to 1.8% and drastically increases to 95.40% to $\theta_{t[1-6]}$. In the same experiment, using the protocol dark **(ud)**, the adaptation of the biases presents a FNMR increase in the evaluation set from 4.00% to 4.40% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is increased to 4.8% and increased to 4.30% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ such figure of merit decreases to 3.0% and drastically increases to 89.70% to $\theta_{t[1-6]}$. In experiments using the protocol outside **(uo)**, the adaptation of the biases presented a FNMR decrease in the evaluation set from 2.00% to 1.70% using for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is increased to 2.2% and to 2.5% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 4.5% and 96.20% respectively. Figure 5.14 presents the DET plots for the this evaluation set using Incep. Res. v1 and Incep. Res. v2. The same observation can be made for different operational points.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray fargo --preprocess-training-data # generating prior
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray fargo --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm fargo  # Training DSU
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm fargo  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm fargo
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm fargo
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

The same trends are observed by using Triplet Networks as a training method as can be observed in Table 5.8.

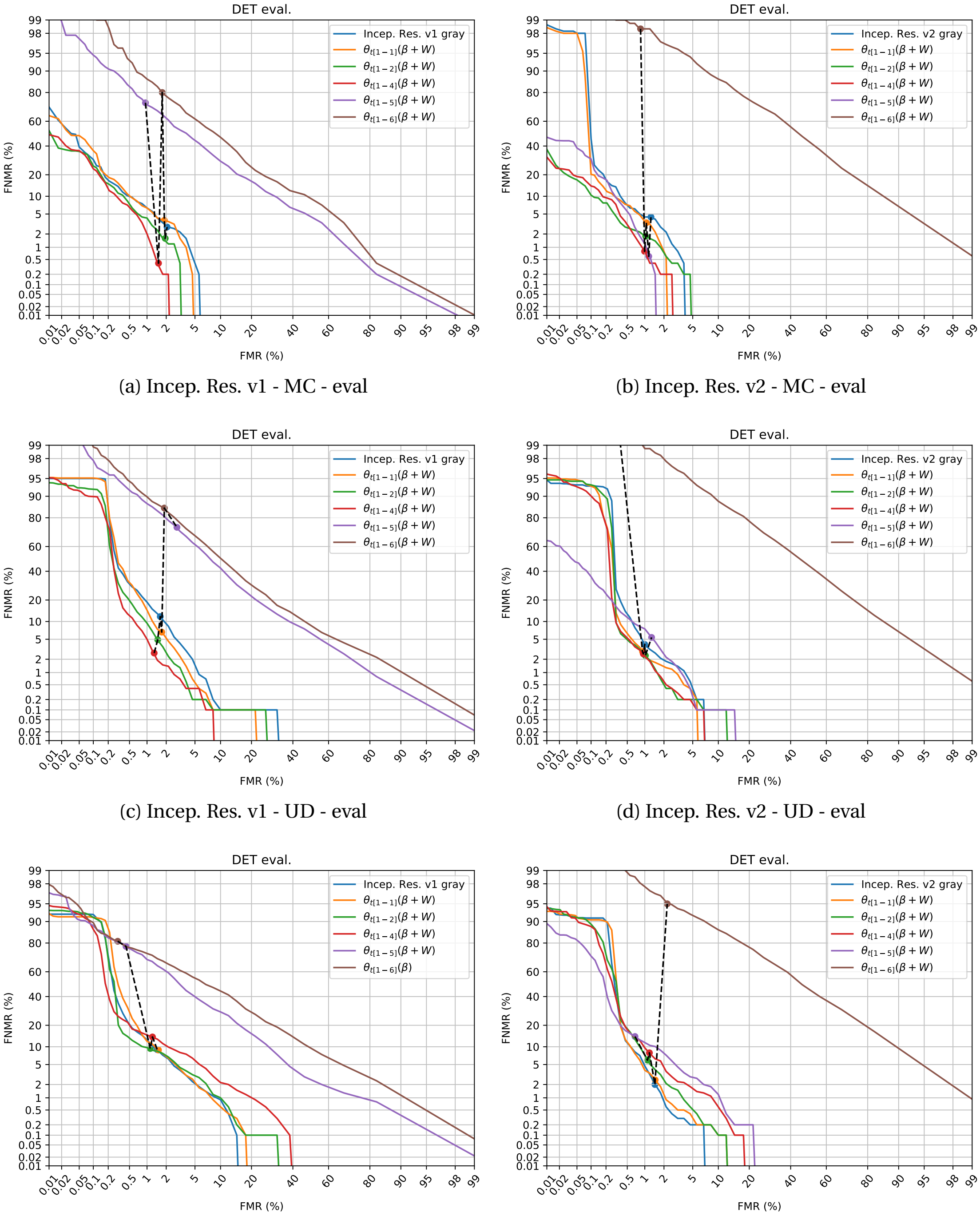
With this set of experiments it was possible to observe that the adaptation of batch normalization offsets (βs) improved recognition rates in all conditions (controlled, dark and outside). This confirms both Hypothesis, that there are DSUs and such DSUs are embedded in the biases (β). To investigate if there are domain specific feature detectors, the next set of experiments the same experimental procedure is performed, but instead of adapting only β, it is adapted β and W (Equation 5.4).

Table 5.9 presents the FNMR@FMR=1%(dev)% with adaptation of the **kernels and the biases** using both based architectures (Incep. Res. v1 and Incep. Res. v2) and both training methods (Siamese and Triplet networks). The same trends observed before can be observed in this experiment.

Table 5.8 – Fargo database - FNMR@FMR=1%(dev) taken from the development set adapting β only

#	FR Algorithm	mc		ud		uo	
		dev	eval	dev	eval	dev	eval
FR Baselines							
1	Incep. Res. v1 - gray scaled	0.40	2.80	6.70	11.90	0.40	9.00
2	Incep. Res. v2 - gray scaled	0.00	4.40	0.80	4.00	0.50	2.00
Reproducible Baselines							
3	MultiScale feat. [Liu et al., 2012]	20.80	23.00	26.70	23.70	32.30	42.40
4	MLBP [Liao et al., 2009]	23.80	21.40	29.00	27.30	34.10	51.60
5	ISV	10.80	8.40	12.00	14.10	8.00	39.50
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	16.80	15.60	21.60	19.60	25.30	30.70
Siamese Networks training							
7	Incep. Res. v1 $\theta_{t[1-1]}$	0.80	4.00	5.50	8.40	0.00	7.40
8	Incep. Res. v1 $\theta_{t[1-2]}$	1.60	**0.60**	4.70	7.90	1.90	8.30
9	Incep. Res. v1 $\theta_{t[1-4]}$	2.80	0.60	1.30	4.50	1.80	13.60
10	Incep. Res. v1 $\theta_{t[1-5]}$	71.20	80.20	74.20	77.60	83.60	77.70
11	Incep. Res. v1 $\theta_{t[1-6]}$	64.20	76.20	78.90	83.10	86.20	88.50
12	Incep. Res. v2 $\theta_{t[1-1]}$	0.00	4.20	2.40	4.40	0.80	**1.70**
13	Incep. Res. v2 $\theta_{t[1-2]}$	0.00	3.20	1.00	4.80	2.00	2.20
14	Incep. Res. v2 $\theta_{t[1-4]}$	0.20	3.20	0.50	**4.30**	0.70	2.50
15	Incep. Res. v2 $\theta_{t[1-5]}$	0.00	1.80	0.60	3.00	1.10	4.50
16	Incep. Res. v2 $\theta_{t[1-6]}$	99.20	95.40	97.30	89.70	99.90	96.20
Triplet Networks training							
17	Incep. Res. v1 $\theta_{t[1-1]}$	0.40	4.00	6.50	9.10	0.00	7.40
18	Incep. Res. v1 $\theta_{t[1-2]}$	1.00	1.00	4.20	8.60	2.00	11.00
19	Incep. Res. v1 $\theta_{t[1-4]}$	4.60	3.20	6.30	12.20	2.20	17.10
20	Incep. Res. v1 $\theta_{t[1-5]}$	71.40	84.00	87.80	78.30	87.60	89.50
21	Incep. Res. v1 $\theta_{t[1-6]}$	89.40	89.20	92.80	96.10	90.00	89.30
22	Incep. Res. v2 $\theta_{t[1-1]}$	0.00	3.80	2.00	5.80	0.90	2.10
23	Incep. Res. v2 $\theta_{t[1-2]}$	0.00	4.80	0.60	4.00	4.40	9.00
24	Incep. Res. v2 $\theta_{t[1-4]}$	0.20	3.80	1.00	5.10	1.40	7.80
25	Incep. Res. v2 $\theta_{t[1-5]}$	0.00	4.20	3.80	7.90	6.60	13.40
26	Incep. Res. v2 $\theta_{t[1-6]}$	98.40	97.40	95.50	99.20	97.80	97.80

Under the controlled protocol **(mc)**, the adaptation of the **kernel and biases** presented a FNMR increase in the evaluation set from 2.80% to 3.80% using **Incep. Res. v1 and Siamese Networks** $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is reduced to 1.6% and to 0.4% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 73.60% and 80.00% respectively. In the same experiment, using the protocol dark **(ud)**, the adaptation of the biases presented a FNMR increase in the evaluation set from 4.00% to 6.70% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is increased to 4.9% and decreased to 2.7% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 74.10% and 85.10% respectively. Experiments using the protocol outside **(uo)**, the adaptation of the biases presented a FNMR reduction in the evaluation set from 9.00% to 8.40% using **Incep. Res. v1 and Siamese Networks** $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is reduced to 9.4% and to 14.00% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 78.00% and 81.00% respectively.



(a) Incep. Res. v1 - MC - eval

(b) Incep. Res. v2 - MC - eval

(c) Incep. Res. v1 - UD - eval

(d) Incep. Res. v2 - UD - eval

(e) Incep. Res. v1 - UO - eval

(f) Incep. Res. v2 - UO - eval

Figure 5.15 – FARGO - **Adapting** $W + \beta$ - DET curves for verification experiments under the three illumination conditions MC (controlled), UD (dark) and UO (outdoor) trained with Siamese Networks. The column on the left presents DET curves using Incep. Res. v1 as a basis and the column on the right presents DET curves using Incep. Res. v2 as a basis

The same trends can be observed for **Incep. Res. v2 and Siamese Networks**. Under the controlled protocol **(mc)**, the adaptation of the biases presented a FNMR decrease in the evaluation set from 4.40% to 3.40% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is reduced to 1.8% and to 0.8% for $\theta_{[1-4]}$. For $\theta_{[1-5]}$ it is reduced to 0.6%. Finally for $\theta_{[1-6]}$ such figure of merit drastically increases to 98.40% respectively. In the same experiment, using the protocol dark **(ud)**, the adaptation of the biases presented a FNMR reduction in the evaluation set from 4.00% to 2.90% for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is reduced to 2.4% and to 2.60% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 5.40% and 100.0% respectively. Experiments using the protocol outside **(uo)**, the adaptation of the **kernel an biases** presented a FNMR increase in the evaluation set from 2.00% to 2.50% using **Incep. Res. v2 and Siamese Networks** for $\theta_{[1-1]}$. For $\theta_{[1-2]}$ such figure of merit is increased to 6.0% and to 8.0% for $\theta_{[1-4]}$. Finally for $\theta_{[1-5]}$ and $\theta_{[1-6]}$ such figure of merit drastically increases to 14.20% and 95.44% respectively. Hence, no improvements are observed in this experiment. Figure 5.15 presents the DET plots for the evaluation set using Inception Res. v1 and Inception Res. v2.

The same trends are observed by using Triplet Networks as a training method as can be observed in Table 5.9.

With these set of experiments it was possible to observe that, despite the adaptation of only the $\beta's$ increase the recognition rates, the joint adaptation of β and W increases even more such figure of merit. It is possible to suggest that there are domain specific feature detectors, therefore confirming once both hypotheses.

Compared to all **Reproducible Baselines**, the approach based on DSU presented higher recognition rates for all conditions. It is worth noting that the training set of this database contains VIS and NIR images from under the controlled environment only. Even if samples from uo and ud are not presented, it was possible to observe improvements in the recognition rates in these conditions by just making the adaptation for the NIR channel.

5.3.3 Visible Light to Thermograms

In this subsection it is described experiments with two subsets of the Pola Thermal database: Thermal and Pola Thermal.

Thermal

Figure 5.16 (a) presents the CMC curves with adaptation of the biases only for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, has an average rank one recognition rate of 31.09 %. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to improve this benchmark to ≈ 33%. The biases adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ achieves an average rank one recognition to ≈ 48% and ≈ 47% respectively. Adapting $\theta_{t[1-5]}$ the average rank one recognition rates increases to ≈ 59%. For

Table 5.9 – Fargo database - FNMR@FMR=1% adapting $W + \beta$

#	FR Algorithm	mc		ud		uo	
		dev	eval	dev	eval	dev	eval
FR Baselines							
1	Incep. Res. v1 - gray scaled	0.40	2.80	6.70	11.90	0.40	9.00
2	Incep. Res. v2 - gray scaled	0.00	4.40	0.80	4.00	0.50	2.00
Reproducible Baselines							
3	MultiScale feat. [Liu et al., 2012]	20.80	23.00	26.70	23.70	32.30	42.40
4	MLBP [Liao et al., 2009]	23.80	21.40	29.00	27.30	34.10	51.60
5	ISV	10.80	8.40	12.00	14.10	8.00	39.50
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	16.80	15.60	21.60	19.60	25.30	30.70
Siamese Networks Trainer							
7	Incep. Res. v1 $\theta_{t[1-1]}$	1.20	3.80	4.00	6.70	0.50	8.40
8	Incep. Res. v1 $\theta_{t[1-2]}$	0.40	1.60	3.30	4.90	2.00	9.40
9	Incep. Res. v1 $\theta_{t[1-4]}$	2.00	**0.40**	0.90	2.70	2.80	14.00
10	Incep. Res. v1 $\theta_{t[1-5]}$	71.20	73.60	79.30	74.10	88.70	78.00
11	Incep. Res. v1 $\theta_{t[1-6]}$	83.00	80.00	82.60	85.10	91.00	81.00
12	Incep. Res. v2 $\theta_{t[1-1]}$	0.00	3.40	1.20	2.90	1.20	**2.50**
13	Incep. Res. v2 $\theta_{t[1-2]}$	0.20	1.80	1.40	**2.40**	3.60	6.00
14	Incep. Res. v2 $\theta_{t[1-4]}$	0.60	0.80	2.20	2.60	4.20	8.00
15	Incep. Res. v2 $\theta_{t[1-5]}$	1.40	0.60	1.00	5.40	5.50	14.20
16	Incep. Res. v2 $\theta_{t[1-6]}$	98.20	98.40	98.50	100.0	97.70	95.00
Triplet Networks Trainer							
17	Incep. Res. v1 $\theta_{t[1-1]}$	0.80	3.80	4.00	8.20	3.90	7.80
18	Incep. Res. v1 $\theta_{t[1-2]}$	6.80	2.60	45.00	33.00	12.90	16.10
19	Incep. Res. v1 $\theta_{t[1-4]}$	17.80	10.00	22.70	20.90	12.30	25.80
20	Incep. Res. v1 $\theta_{t[1-5]}$	89.80	84.00	89.90	85.60	84.80	92.00
21	Incep. Res. v1 $\theta_{t[1-6]}$	85.40	94.40	90.90	91.80	88.90	92.07
22	Incep. Res. v2 $\theta_{t[1-1]}$	0.00	3.20	1.30	7.60	1.80	3.90
23	Incep. Res. v2 $\theta_{t[1-2]}$	82.40	75.00	75.00	60.10	20.60	30.40
24	Incep. Res. v2 $\theta_{t[1-4]}$	35.60	26.80	11.80	20.10	18.60	22.30
25	Incep. Res. v2 $\theta_{t[1-5]}$	8.80	10.60	48.80	31.90	19.30	22.60
26	Incep. Res. v2 $\theta_{t[1-6]}$	95.60	98.20	99.10	98.40	95.80	99.30

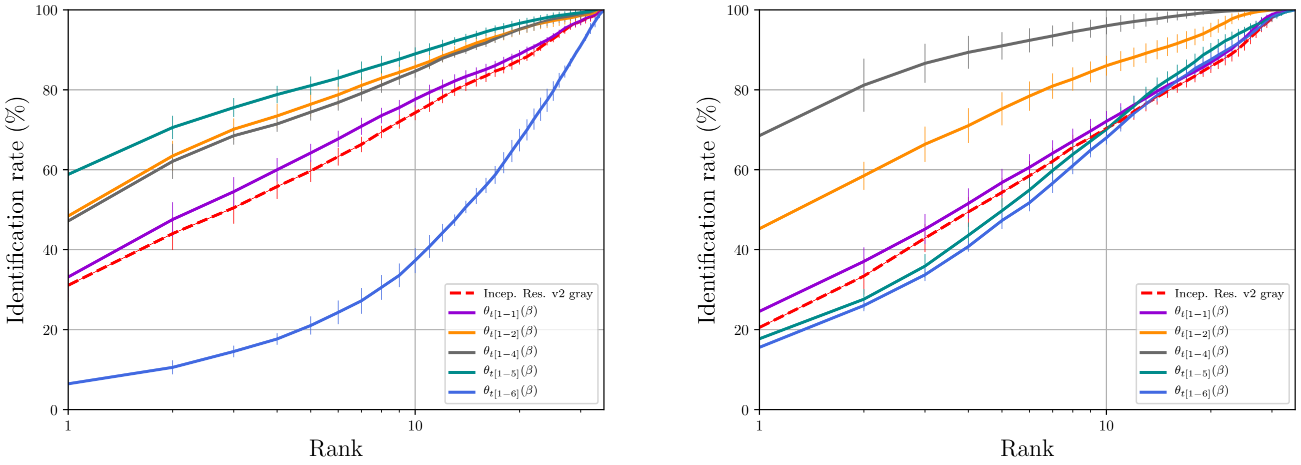
$\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 6%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm thermal  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, see[2].

The same trend can be observed for **Incep. Res. v1** (see Figure 5.16 (b)). The average recogni-



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.16 – Thermal - Average CMC curves (with error bars) for the adaptation of biases

tion rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 20.55%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}(\beta)$ leads to an average rank one recognition rate of ≈ 24%. For $\theta_{t[1-2]}$ it is achieved ≈ 45%. Experiments with $\theta_{t[1-4]}$ get its best average rank one recognition rate with 68.53%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drop drastically to ≈ 18% and ≈ 15%, respectively.

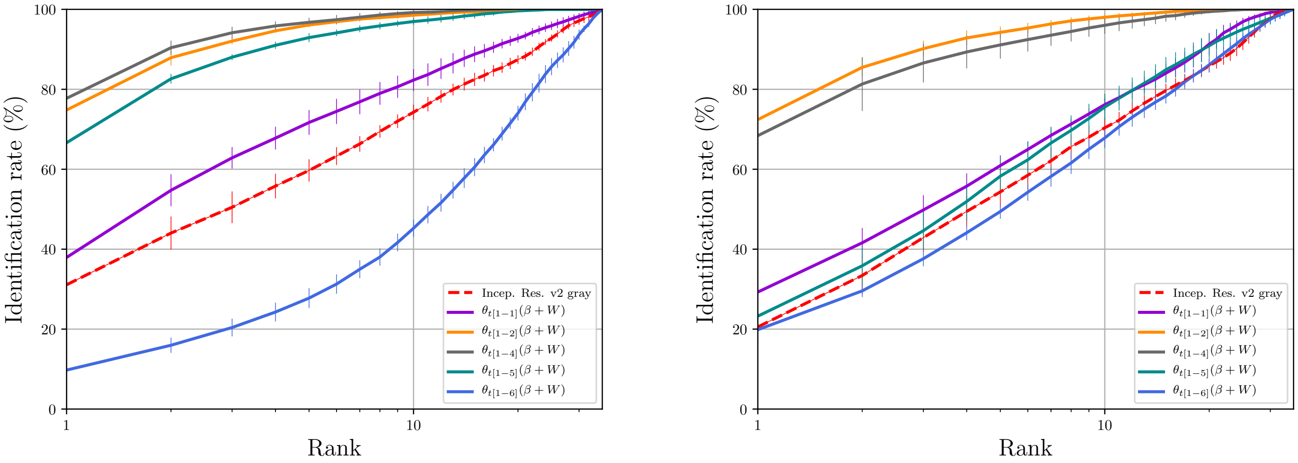
Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm thermal   # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

Training with **Triplet Networks** same trends are observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate gets improved to ≈ 30% and ≈ 42% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 40% and ≈ 48% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rate for the Incep. Res. v1 is improved to ≈ 30% and to ≈ 50% for Incep. Res. v2. Using Incep. Res. v1 the average rank one recognition rates drops to ≈ 15% and ≈ 13% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate increases to ≈ 49% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 5%

for $\theta_{t[1-6]}$.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.17 – Thermal - Average CMC curves (with error bars) for the adaptation of kernel and biases

The same trends observed in the previous subsections were observed in this database. The adaptation of the batch normalization offsets only improve the recognition rates, confirming both hypothesis. In the next set of experiments it is investigated if there are domain specific feature detectors by adapting β and W (Equation 5.4)

Figure 5.17 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 31.09%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}(\beta + W)$ in the plots) it is possible to get this benchmark improved to $\approx 38\%$. The adaptation for $\theta_{t[1-2]}$ improves this benchmark to $\approx 75\%$. For $\theta_{t[1-4]}$ this benchmark is improved to 77.74% (its best). With $\theta_{t[1-5]}$ the average rank one recognition rate drops to $\approx 67\%$. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to $\approx 9\%$.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_nonshared_batch_norm thermal # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_nonshared_batch_norm thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

The same trends can be observed for **Incep. Res. v1 trained with Siamese Networks** (see 5.17

(b)). The average recognition rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 20.55%. The adaptation of β and W for $\theta_{t[1-1]}$ improves this benchmark to ≈ 29%. Adapting $\theta_{t[1-2]}$ this benchmark is improved to 72.41%. With $\theta_{t[1-4]}$ the average rank one recognition rate decreases to ≈ 68% Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to ≈ 23% and ≈ 20%, respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray thermal --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_nonshared_batch_norm thermal # Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_nonshared_batch_norm thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

As before, with Siamese Networks, the same trends using **Triplet Networks** as training strategy is observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate improves to ≈ 30% and ≈ 28% respectively. For $\theta_{t[1-2]}$ such benchmark is improved to ≈ 53% and ≈ 42% respectively. Using Incep. Res. v1 the average rank one recognition rate drops to ≈ 42% for $\theta_{t[1-4]}$ and drastically drops to ≈ 17% and to ≈ 13% for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively. For Incep. Res. v2 the average rank one recognition rates improves to ≈ 48% for $\theta_{t[1-4]}$ and to ≈ 51% for $\theta_{t[1-5]}$ and it drastically drops to ≈ 27% for $\theta_{t[1-6]}$.

With this set of experiments it was possible to observe that, despite the adaptation of only the βs increase the recognition rates, the joint adaptation of β and W increased even more such figure of merit, confirming both hypotheses. It is possible to suggest that there are domain specific feature detectors and such feature detectors need to be taken in to account for the VIS to Thermal task.

Table 3.7 shows the average rank one recognition rate comparing different configurations of DSU approach jointly with the **FR baselines**, **Reproducible baselines** and **Non Reproducible baselines**. In terms of average rank one recognition rate the proposed approach based on DSU presented competitive recognition rates. The best setup is the model $\theta_{t[1-4]}$ trained with Siamese Neural Networks using the Incep. Res. v1 as a basis and achieved a recognition rate of 77.73%. Compared with the Non Reproducible baselines, this is slightly lower than the CpNN system proposed by Hu et al. [2016].

Table 5.10 – Thermal database - Average rank one recognition rate under different Face Recognition systems.

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	20.55%(4.2)
2	Incep. Res. v2 - gray scaled	31.09%(4.1)
Reproducible Baselines		
3	MLBP in [Liao et al., 2009]	36.80%(3.5)
4	Multiscale Feat. in [Liu et al., 2012]	26.89%(3.5)
5	ISV	23.86%(1.3)
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	34.07%(2.9)
Non Reproducible Baselines		
7	PLS [Hu et al., 2016]	53.05% (n/a)
8	DPM [Hu et al., 2016]	75.31% (n/a)
9	CpNN [Hu et al., 2016]	**78.72% (n/a)**
DSU Adapt β		
10	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	68.54% (7.4)
11	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	58.83% (4.0)
12	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	46.24%(6.3)
13	Trip. Incep. Res. v2 $\theta_{t[1-4]}$	50.21%(2.3)
DSU Adapt β+ W		
14	Siam. Incep. Res. v1 $\theta_{t[1-2]}$	72.42% (3.2)
15	Siam. Incep. Res. v2 $\theta_{t[1-5]}$	**77.74% (2.6)**
16	Trip. Incep. Res. v1 $\theta_{t[1-2]}$	52.98%(4.4)
17	Trip. Incep. Res. v2 $\theta_{t[1-4]}$	57.97%(3.1)

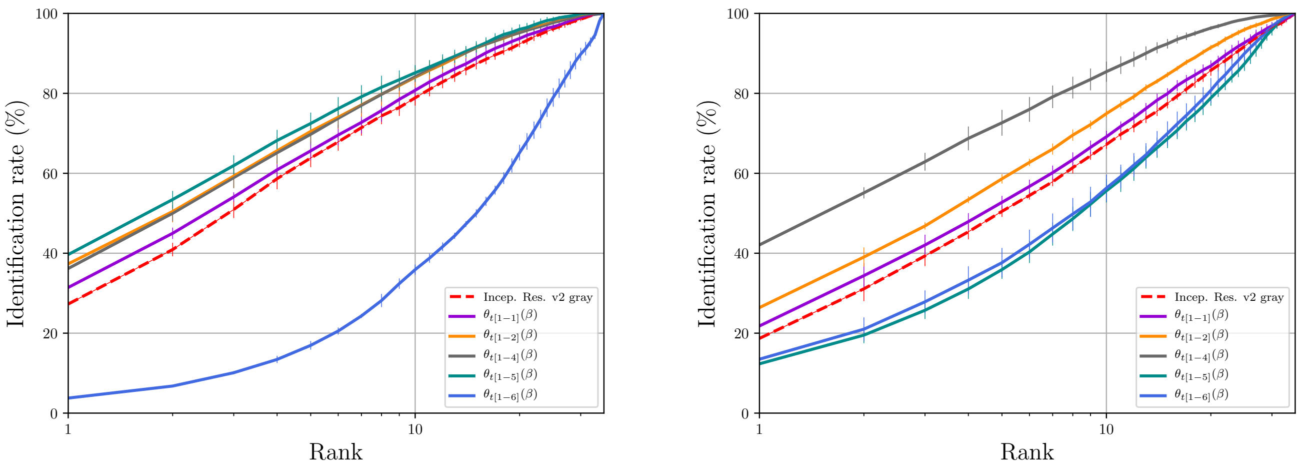
Pola Thermal

Figure 5.18 (a) presents the CMC curves with adaptation of the biases only for the **Incep. Res. v2 using the Siamese Networks**. Such DCNN, with no adaptation, has an average rank one recognition rate of 27.29 %. Adapting only the biases (β in Equation 5.4) of the first layer ($\theta_{t[1-1]}(\beta)$ in the plots) it is possible to improve this benchmark to ≈ 32%. The biases adaptation for $\theta_{t[1-2]}$ achieves an average rank one recognition to ≈ 37%. Adapting $\theta_{t[1-4]}$ and $\theta_{t[1-5]}$ the average rank one recognition rates increases to ≈ 36% and 39.67% respectively. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to ≈ 4%.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray pola_thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_betas_nonshared_batch_norm pola_thermal  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv2_first_layer_betas_nonshared_batch_norm pola_thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.18 – Pola Thermal - Average CMC curves (with error bars) for the adaptation of biases

It is possible to observe the same trends for **Incep. Res. v1 using the Siamese Networks** (see 5.18 (b)). The average recognition rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 18.69%. The adaptation of the biases (β in Equation 5.4) for $\theta_{t[1-1]}(\beta)$ leads to an average rank one recognition rate of ≈ 22%. For $\theta_{t[1-2]}$ it is achieved ≈ 26%. Experiments with $\theta_{t[1-4]}$ gets its best average rank one recognition rate with ≈ 42%. For $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to ≈ 12% and ≈ 13%, respectively.

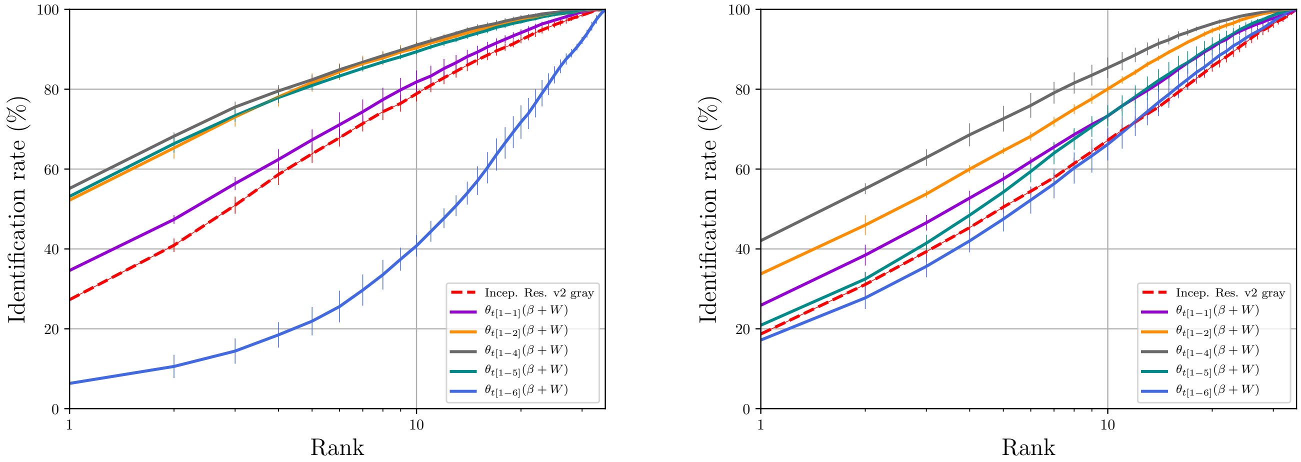
Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v1_centerloss_gray pola_thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv1_first_layer_betas_nonshared_batch_norm pola_thermal  # Training DSU
$ bob bio htface htface_baseline siamese_inceptionv1_first_layer_betas_nonshared_batch_norm pola_thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta)$. To check how to train other DSUs, check[2].

Training with **Triplet Networks** the same trends are observed. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate get improved to ≈ 23% and ≈ 30% respectively. For $\theta_{t[1-2]}$ the improvements are ≈ 26% and ≈ 32% respectively. For $\theta_{t[1-4]}$ the average rank one recognition rate for the Incep. Res. v1 are improved to ≈ 26% and to

$\approx 32\%$ for Incep. Res. v2. Using Incep. Res. v1 the average rank one recognition rates drops to $\approx 12\%$ and $\approx 13\%$ for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively (same trend as Siamese). For Incep. Res. v2 the average rank one recognition rate increases to $\approx 30\%$ for $\theta_{t[1-5]}$ and it drastically drops to $\approx 4\%$ for $\theta_{t[1-6]}$.



(a) Incep. Res. v2

(b) Incep. Res. v1

Figure 5.19 – Pola Thermal - Average CMC curves (with error bars) for the adaptation of kernel and biases

The same trends observed in the previous subsections were observed in this database. The adaptation of the batch normalization offsets only improve the recognition rates confirming both hypotheses. In the next set of experiments it is investigated if there are domain specific feature detectors by adapting β and W (Equation 5.4)

Figure 5.19 (a) presents the CMC curves with adaptation of convolutional kernels and biases for the **Incep. Res. v2 using Siamese Networks**. Such DCNN, with no adaptation, presents an average rank one recognition rate of 27.29%. Adapting both, biases and kernels (β and W in Equation 5.4), of the first layer ($\theta_{t[1-1]}(\beta + W)$ in the plots) it is possible to get this benchmark improved to $\approx 35\%$. The adaptation for $\theta_{t[1-2]}$ and $\theta_{t[1-4]}$ improves the average rank one recognition rate to $\approx 52\%$ and 55.15% (its best) respectively. With $\theta_{t[1-5]}$ the average rank one recognition rate drops to $\approx 53\%$. For $\theta_{t[1-6]}$ the average rank one recognition rate drops drastically to $\approx 6\%$.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline htface_idiap_msceleb_inception_v2_centerloss_gray pola_thermal --preprocess-training-data # generating prior
$ bob bio htface htface_train_dsu siamese_inceptionv2_first_layer_nonshared_batch_norm pola_thermal  # Training DSU
$ bob bio htface htface_baseline
  siamese_inceptionv2_first_layer_nonshared_batch_norm pola_thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

It is possible to observe the same trends for **Incep. Res. v1 trained with Siamese Networks** (see Figure 5.19 (b)). The average recognition rates increase once depth is increased. With no adaptation, such DCNN has an average rank one recognition rate of 18.69%. The adaptation of β and W for $\theta_{t[1-1]}$ and $\theta_{t[1-2]}$ leads to an average rank one recognition rate of $\approx 26\%$ and $\approx 34\%$ respectively. With $\theta_{t[1-4]}$ the average rank one recognition rate decreases to $\approx 42\%$ Finally, for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ the average rank one recognition rates drops drastically to $\approx 21\%$ and $\approx 17\%$, respectively.

Using the thesis software this strategy can be triggered with the following bash commands:

```
$ bob bio htface htface_baseline
  htface_idiap_msceleb_inception_v1_centerloss_gray pola_thermal --preprocess-
  training-data # generating prior
$ bob bio htface htface_train_dsu
  siamese_inceptionv1_first_layer_nonshared_batch_norm pola_thermal  # Training
   DSU
$ bob bio htface htface_baseline
  siamese_inceptionv1_first_layer_nonshared_batch_norm pola_thermal
```

These command lines demonstrates just how to train $\theta_{t[1-1]}(\beta + W)$. To check how to train other DSUs, check[2].

As before, with Siamese Networks, it is also observed the same trends using **Triplet Networks** as training strategy. Adapting $\theta_{t[1-1]}$ for both, Incep. Res. v1 and Incep. Res. v2, the average rank one recognition rate improves to $\approx 23\%$ and $\approx 30\%$ respectively. For $\theta_{t[1-2]}$ such benchmark is improved to $\approx 24\%$ and $\approx 27\%$ respectively. Using Incep. Res. v1 the average rank one recognition rate drops to $\approx 26\%$ for $\theta_{t[1-4]}$ and drastically drops to $\approx 11\%$ and to $\approx 12\%$ for $\theta_{t[1-5]}$ and $\theta_{t[1-6]}$ respectively. For Incep. Res. v2 the average rank one recognition rates improves to $\approx 32\%$ for $\theta_{t[1-4]}$ and to $\approx 30\%$ for $\theta_{t[1-5]}$ and it drastically drops to $\approx 5\%$ for $\theta_{t[1-6]}$.

With this set of experiments it was possible to observe that, despite the adaptation of only the βs increase the recognition rates, the joint adaptation of β and W drastically increased even more such figure of merit confirming both hypotheses. It is possible to suggest that there are domain specific feature detectors and such feature detectors need to be taken in to account for the VIS to Pola Thermal task.

Table 5.11 shows the average rank one recognition rate comparing different configurations of the DSU approach. The best DSU (Incep. Res. v2 model $\theta_{t[1-4]}(W + \beta)$ trained with Siamese Networks) presented an average rank one recognition rate of 55.15%. Although this recognition rate is substantially higher than all the **Reproducible Baselines**, it is substantially lower than

Table 5.11 – Pola Thermal database - Average rank one recognition rate under different Face Recognition systems.

#	FR Algorithm	Average rank one rec. rate
FR Baselines		
1	Incep. Res. v1 - gray scaled	18.69%(2.1)
2	Incep. Res. v2 - gray scaled	27.29%(0.8)
Reproducible Baselines		
3	MLBP in [Liao et al., 2009]	15.61%(2.9)
4	Multiscale Feat. in [Liu et al., 2012]	20.81%(3.4)
5	ISV	9.63%(1.2)
6	GFK [Gong et al., 2012; Sequeira et al., 2017]	34.43%(2.3)
Non Reproducible Baselines		
7	PLS [Hu et al., 2016]	58.67% (n/a)
8	DPM [Hu et al., 2016]	80.54% (n/a)
9	CpNN [Hu et al., 2016]	**82.90% (n/a)**
DSU Adapt β		
10	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	42.08%(1.4)
11	Siam. Incep. Res. v2 $\theta_{t[1-4]}$	55.15%(1.3)
12	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	26.04%(1.4)
13	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	32.04%(4.1)
DSU Adapt $\beta + W$		
14	Siam. Incep. Res. v1 $\theta_{t[1-4]}$	42.07%(1.4)
15	Siam. Incep. Res. v2 $\theta_{t[1-4]}$	**55.15%(1.3)**
16	Trip. Incep. Res. v1 $\theta_{t[1-4]}$	26.06%(2.2)
17	Trip. Incep. Res. v2 $\theta_{t[1-5]}$	32.23%(2.0)

all **Non Reproducible Baselines**. For instance, the DPM and CpNN systems introduced by Hu et al. [2016] presents an average rank one recognition rate of 80.54% and 82.90% respectively.

5.4 Discussion

It this chapter two hypotheses were drawn. Hypothesis 5.1 argue that high level feature detectors from DCNNs trained with VIS images are potentially **domain independent** and that low level feature detectors are potentially **domain dependent** and the task of HFR can be assessed by adapting the low level layers for a particular target image modality. Hypothesis 5.1 argue that such **domain dependent** feature detectors might be embedded in the biases set of each low level feature detector. To approach these hypotheses a method called Domain Specific Units (DSU) was introduced. Given pairs of face images from different image modalities, this approach jointly learns specific features for a particular image modality.

Two methods to train such DSU were introduced and experiments were carried our using two different DCNN architectures trained, in the context of this thesis, with VIS images. Compared to a DCNN with no adaptation, the DSU approach systematically improved the

HFR recognition rates for all tested image domains, confirming Hypothesis 5.1. By applying DSU on the biases only, recognition rates were also improved, confirming Hypothesis 5.2. Moreover, such improvements were observed independently of the base DCNN architecture (Incep. Res. v1 or Incep. Res. v2) and training method (Siamese or Triplet training). By incrementally applying the DSU approach layer by layer ($\theta_{t[1-n]}$), it was possible to observe improvements, in terms of rank one recognition rate until gets to a point of overfit. Overall, for the Incep. Res. v1 it was possible to observe improvements until the layer set $\theta_{t[1-4]}$; for Incep. Res. v2 such improvements could be observed until the layer set $\theta_{t[1-5]}$. In both cases, the recognition rates started to decrease concomitantly when the number of free parameters started to exponentially grow (see Figure 5.5). Such models are possibly overfitted. Table 5.12 presents the number of free parameters that need to be learnt for each $\theta_{t[1-n]}$ and for both base architectures.

Table 5.12 – Number of free parameters learnt for each base DCNN adapting either β or $\beta + W$

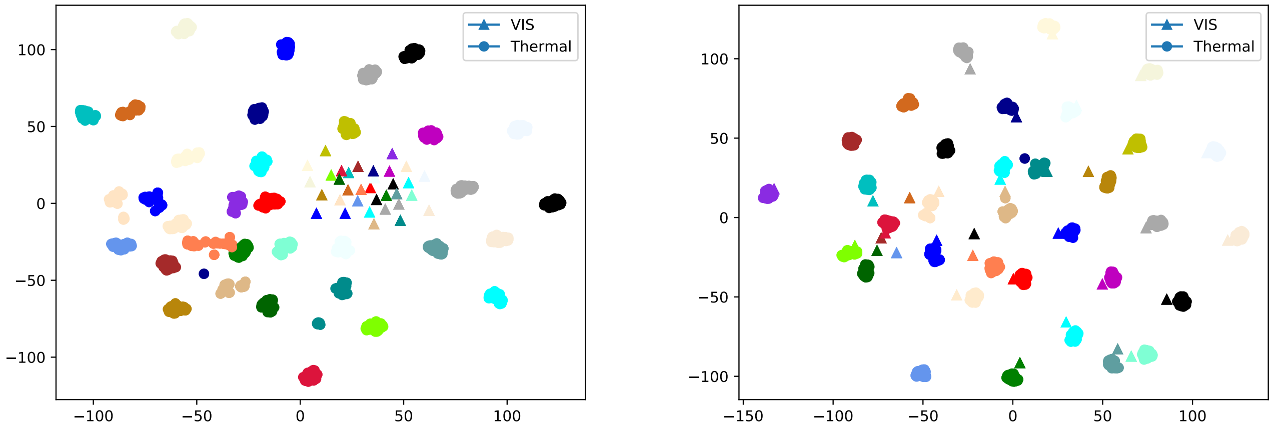
	Incep. Res. v1		**Incep. Res. v2**	
	Adapt β	Adapt $\beta + W$	Adapt β	Adapt $\beta + W$
$\theta_{t[1-1]}$	32	320	32	320
$\theta_{t[1-2]}$	208	33,264	208	33,264
$\theta_{t[1-4]}$	656	614,320	400	171,696
$\theta_{t[1-5]}$	1,616	1,000,560	928	439,488
$\theta_{t[1-6]}$	2,640	2,709,616	3,328	1,668,768

With respect to the training methods and the base architectures, it was observed that the Incep. Res. v2 associated with the Siamese training presented the highest recognition rates for most of the evaluated databases. This is possibly related with the fact that such DCNN presents the highest recognition rates for VIS images.

In the **VIS to Sketches** task, most of the improvements, in terms of recognition rates, were observed once adaptation were carried out with $W + \beta$. For instance, experiments with CUHK-CUFS, where the sketches are very reliable, the average rank one recognition rate was improved from 80.29% to 97.7%. For CUHK-CUFSF, where the sketch line is not aligned with its corresponding photo, the average rank one recognition rate was improved from 29.51% to 85.05%.

In the **VIS to NIR** task, more data are available and, for some databases, such data was captured in different conditions. Hence, different analysis can be made. Under constrained conditions, where subjects are closer to the camera, with neutral expression and no pose/illumination variations, most of the recognition rate improvements can be observed by adapting only β. For instance, experiments with NIVL database, the average rank one recognition rate with no adaptation using **Incep. Res. v2** is 90.00%. By doing the $\theta_{t[1-4]}(\beta)$ adaptation, which corresponds to 400 free parameters only, such figure of merit was improved to 92.5% (see 5.5). Experiments with LDHF, considering only 1m stand-off only, the $\theta_{t[1-1]}(\beta)$ DSU adaptation using **Incep. Res. v1** as a basis improved such figure of merit from 94.8% to 99.6% (see Table 5.6). This adaptation corresponds to 32 free parameters only. Finally, experiments using the FARGO dataset, considering only the controlled protocol (**mc**), the $\theta_{t[1-2]}(\beta)$ DSU adaptation using **Incep. Res. v1** as a basis improved the $FNMR@FMR = 1\%$ from 4.4% to 0.6%. This adaptation corresponds to 208 free parameters only.

Improvements could also be observed under more unconstrained scenarios. For instance, experiments with CASIA database, where NIR face images with several variations in pose and expression are recorded, the $\theta_{t[1-5]}(\beta + W)$ DSU adaptation using **Incep. Res. v2** as a basis improved the average rank one recognition rate from 73.8% to 96.3% (see Table 5.4). Experiments with LDHF, considering stand-offs above 1m, the DSU strategy improved such figure of merit from 75.6% to 98% from 60m stand-off. Considering 100m stand-off, such improvement was from 9.6% to 56.0% and from 2.8% to 22.8% for 150m (see 5.7). All those recognition rates were observed with **Incep. Res. v1** as a basis. Finally, experiments using the FARGO dataset, considering the protocol dark (**ud**), the $\theta_{t[1-2]}(\beta + W)$ DSU adaptation using Incep. Res. v2 as a basis improved the $FNMR@FMR = 1\%$ from 4.0% to 2.4%. Considering the protocol outside (**uo**), improvements were marginal. For instance, the $\theta_{t[1-1]}(\beta)$ DSU adaptation using **Incep. Res. v1** as a basis improved the $FNMR@FMR = 1\%$ from 2.0% to 1.7% (see Table 5.9).



(a) Not adapted

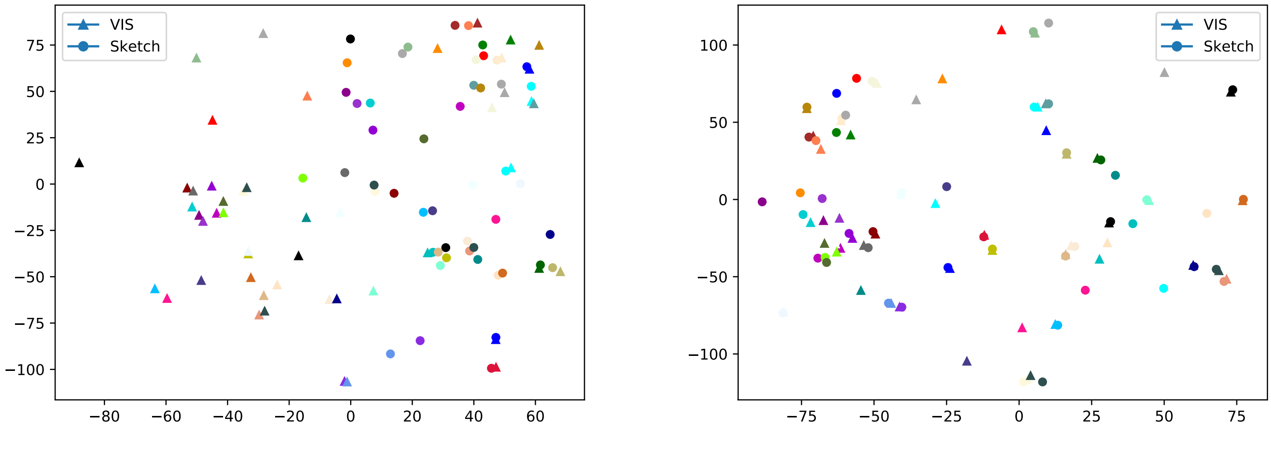
(b) DSU adapted

Figure 5.20 – t-SNE scatter plots from the test set of the Thermal database before and after DSU adaptation. Each color is one different identity and each shape is one of the two image modalities

Improvements could also be observed in the **VIS to Thermal** task. For instance, experiments with the Pola Thermal database, the $\theta_{t[1-4]}(\beta + W)$ DSU adaptation using Incep. Res. v2 as a basis, improved the average rank one recognition rate from 27.29% to 55.15% (see Table 5.11). Experiments with the Thermal database, the $\theta_{t[1-5]}(\beta + W)$ DSU adaptation using Incep. Res. v2 as a basis, improved the average rank one recognition rate from 31.09% to 77.74% (see Table

5.10).

It was possible to observe the discriminability power of DSUs using different benchmarks, such as CMC, DET curves, FNMR and rank one recognition rate. Figure 5.20 illustrates how the 128d embeddings from some samples of the test set using the Thermal database are distributed before (Figure 5.20 (a)) and after (Figure 5.20 (b)) the $\theta_{t[1-4]}(\beta + W)$ DSU adaptation using as a reference the Incep. Res. v2 architecture. This scatter plot is generated using t-Distributed Stochastic Neighbor Embedding (t-SNE) [Maaten and Hinton, 2008], which is a non-linear dimensionality reduction technique well suited for the visualization of high-dimensional data. In the t-SNE plots, each color is a different identity and each shape is a different image modality. It is possible to observe how image modalities are allocated in two big clusters (one for each image modality) in Figure 5.20 (a). On the other hand, in Figure 5.20 (b) it is possible to observe that most of the embeddings are clustered by the identities. Same effect can be observed in Figures 5.21 (a) and (b) using the embeddings from the CUHK-CUFSF databases before and after the $\theta_{t[1-5]}(\beta + W)$ DSU adaptation using as a reference the Incep. Res. v2 architecture. This highlights the effectiveness of the DSU adaptation. Furthermore, those embeddings can potentially be used as a front-end to another layer of classification. An use case of this is carried out in Appendix C.

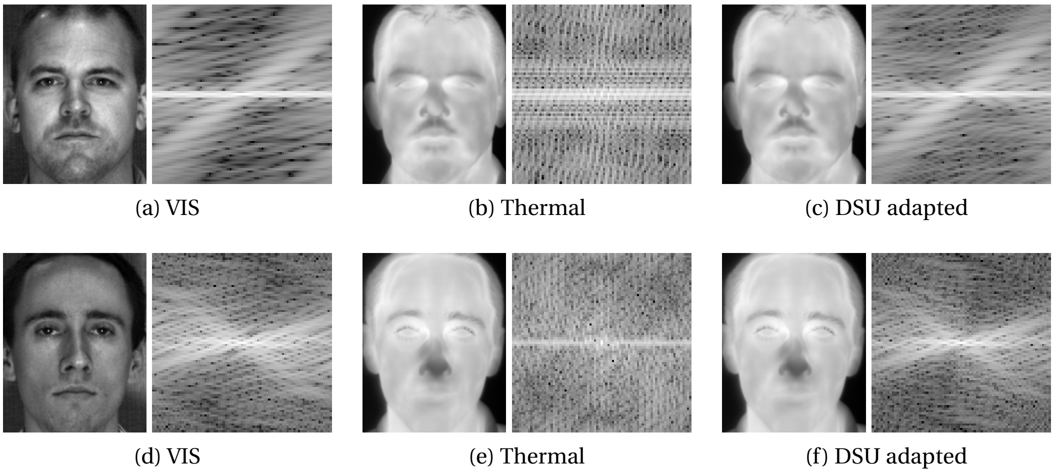


(a) Not adapted

(b) DSU adapted

Figure 5.21 – t-SNE scatter plots from the test set of the CUHK-CUFSF database before and after DSU adaptation. Each color is one different identity and each shape is one of the two image modalities

To better visualise the output of the DSU feature detectors a possible interpretation is offered in Figure 5.22, where the layer Conv2d_1a_3x3 from Incep. Res. 2 is analysed. This layer contains 32 convolutional filters. In Figure 5.22 (a) presents a VIS input with its corresponding FFT (Fast Fourier Transform) response after the convolution of the 12th layer. The same output is presented in Figure 5.22 (b), but now the input is a Thermal image from the same identity. Finally, Figure 5.22 (c) presents the output of the same filter on the same layer, but now DSU adapted for thermal images. It is possible to observe that the FFT output from (a) and (c) presents similar frequecy responses than compared with (a) and (b). The same trend can be observed with the 18th filter from the same layer and same DCNN, where Figure 5.22 (d) presents the FFT from a VIS input and Figures 5.22 (e) and (f) presents the FFT responses considering thermal images as input with and without DSU adaptation respectively.


(a) VIS

(b) Thermal

(c) DSU adapted

(d) VIS

(e) Thermal

(f) DSU adapted

Figure 5.22 – Fourier transform over the Incep. Res. v2 Conv2d_1a_3x3 convoluted images. (a) and (d) corresponds to VIS images convoluted with feature detectors from θ_s. (b) and (e) corresponds to Thermal images convoluted with feature detectors from θ_t before the DSU adaptation. (c) and (f) corresponds to Thermal images convoluted with feature detectors from θ_t after the DSU adaptation.

6 Conclusions and Future Work

The field of research coined as Heterogeneous Face Recognition (HFR) consists in matching face images from different image modalities, such as photographs with sketches, infra-red images, thermograms, etc. The key difficulty in the comparison of faces in this conditions is that images from the same subject may differ in appearance due to changes in image domain. Robust solutions for the HFR task can increase recognition rates in more covert scenarios, such as recognition at a distance or at nighttime, or even in situations where no real face exist (face search using sketches).

In this thesis the HFR task was addressed in three different directions. First, the assessment of some state-of-the-art face recognition systems (trained with VIS images only) was carried out for the HFR task. To the best of our knowledge, such extensive evaluation was never carried out in the literature. Second, an approach that leverages from well stablished crafted features was proposed. In this approach it was hypothesized that within class variations between faces sensed in different image modalities can be modelled and suppressed in the Gaussian Mixture Models mean-supervector space. Third, a strategy that leverages from very accurate DCNNs trained using VIS images only was proposed. In the approach coined as Domain Specific Units (DSU), it was hypothesized that high level feature detectors from those DCNNs are domain independent and their low level features detectors can be adapted from a particular image modality. Furthermore, all these approaches are publicly available in the thesis software package[1] and were implemented within Bob framework[2], an open source framework for signal processing and machine learning maintained and developed during my thesis.

6.1 Experimental Findings

The proposed techniques were applied on three different image modalities covering eight different databases. Each approach with its respective performances was presented in details. Furthermore, each baseline is reproducible via a command line interface along the software

[1] https://gitlab.idiap.ch/bob/bob.thesis.tiago

[2] https://www.idiap.ch/software/bob/

package that comes with this thesis.

The experimental findings of this thesis are listed below.

1. Face Recognition baselines based on the recently published DCNNs architectures trained with large scale set of VIS images presented some discriminative power in all image domains. Considering each task individually:

   (a) **VIS to Sketch**: High recognition rates could be observed in sketches where few shape distortions are perceived. However, recognition rates degrades once such distortions gets increased.

   (b) **VIS to NIR**: High recognition rates could be observed once constrained NIR images are used as probes. For instance, such constrained scenarios encompass mugshot images taken in irregular illuminated environments, with no variations in pose and expression. Once the aforementioned factors are into play, recognition rates starts to decrease.

   (c) **VIS to Thermal**: This is the most challenging task and, although these DCNN models does not use thermal images, recognition rates of $\approx 30\%$ could be observed.

2. Compared with the Face Recognition Baselines, Reproducible Baselines, some Non Reproducible Baselines (see Chapter 3) and Session Variability Modeling approach (see Chapter 4), the DSU strategy presented the highest recognition rates in all image modalities. Considering each task individually:

   (a) **VIS to Sketch**: Compared with its prior DCNN, recognition rate improvements could be observed adapting the convolutional kernels and biases ($W+\beta$). For instance, for the CUHK-CUFSF dataset, where the sketches are more challenging, it was possible to observe the average rank one recognition rate to be improved from 29.51% to 85.05%.

   (b) **VIS to NIR**: High recognition rates could be observed adapting only the biases (β) once constrained (no illumination, pose and expression variations) NIR images are used as probes. For some cases, substantial improvements could be observed adapting only 32 free parameters. Once the aforementioned factors are into play, substantial improvements could be observed adapting both, convolutional kernels and biases ($W+\beta$). For instance, it was possible to observe an improvement from 73.8% to 96.3% using the CASIA database.

   (c) **VIS to Thermal**: Substantial improvements could also be observed in this challenging task. By adapting the convolutional kernel and biases it was possible to observe the average rank one recognition rate to be improved from 31.09% to 77.74% using the thermal database.

3. The GMM Intersession Variability Modeling approach (ISV) presented high recognition rates only in the VIS to NIR task once constrained NIR images are used as probes.

6.2 Related Publications

During the course of this thesis we have published/submitted the following publications:

Journal Articles

- T. d. F. Pereira, A. Anjos and S. Marcel, "Heterogeneous Face Recognition Using Domain Specific Units," in IEEE Transactions on Information Forensics and Security. doi: 10.1109/TIFS.2018.2885284

- Guillaume Heusch, Tiago de Freitas Pereira, and Sebastien Marcel. A comprehensive exper- imental and reproducible study on selfie biometrics in multistream and heterogeneous settings. (Paper Submitted to) - IEEE Transactions on Biometrics, Behavior, and Identity Science, 2019

Conference Proceedings

- FREITAS PEREIRA, TIAGO, and SÉBASTIEN MARCEL. "Heterogeneous Face Recognition using Inter-Session Variability Modelling." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016.

- FREITAS PEREIRA, TIAGO, and SÉBASTIEN MARCEL. "Periocular biometrics in mobile environment." Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. IEEE, 2015.

- ANJOS, A.; GUNTER, M.; de FREITAS PEREIRA, T.; KORSHUNOV, P.; MOHAMMADI, A. and MARCEL, S. (2017). "Continuously reproducing toolchains in pattern recognition and machine learning experiments."

- SEQUEIRA, ANA, et al. "Cross-Eyed 2017: Cross-Spectral Iris/Periocular Recognition Competition." IEEE/IAPR International Joint Conference on Biometrics. No. EPFL-CONF-233586. IEEE, 2017.

- BEVERIDGE, J. ROSS, et al. "The ijcb 2014 pasc video face and person recognition competition." Biometrics (IJCB), 2014 IEEE International Joint Conference on. IEEE, 2014.

6.3 Related Software

During the course of this thesis I contributed to several open source communities related to open science. In this section it is described the most relevant ones.

6.3.1 Bob

Bob[3] is a free machine learning and signal processing library created by the Biometric Security and Privacy Group of Idiap Research Institute whose continuous development and collaboration was carried out during the period of my thesis. This is an open source and extensible toolbox which provides efficient implementations of several machine learning algorithms as well as a framework to help researchers to conduct reproducible research publications.

As of today Bob is composed of 102 components and most of them are focused on biometric related tasks such as: face/heterogeneous face recognition, speaker recognition, finger/-palm vein recognition, presentation attack detection, template protection, diarization among others.

In the subsections below it is listed a set of related software packages developed and maintained in context of this thesis

API for databases

Bob provides an API to programmatically query and access **samples**, **protocols** and **metadata** for any kind of pattern recognition task. Those packages are called database packages[4]. Follow below a list of all database packages developed in the context of this thesis:

- bob.db.cuhk_cufs[5] https://gitlab.idiap.ch/bob/bob.db.cuhk_cufs
- bob.db.cuhk_cufsf [5]: https://gitlab.idiap.ch/bob/bob.db.cuhk_cufs
- bob.db.nivl [5]: https://gitlab.idiap.ch/bob/bob.db.nivl
- bob.db.cbsr_nir_vis_2: https://gitlab.idiap.ch/bob/bob.db.cbsr_nir_vis_2
- bob.db.ldhf [5]: https://gitlab.idiap.ch/bob/bob.db.ldfh
- bob.db.pola_thermal [5]: https://gitlab.idiap.ch/bob/bob.db.pola_thermal
- bob.db.fargo: https://gitlab.idiap.ch/bob/bob.db.fargo
- bob.db.ijba: https://gitlab.idiap.ch/bob/bob.db.ijba
- bob.db.ijbc: https://gitlab.idiap.ch/bob/bob.db.ijbc
- bob.db.msceleb: https://gitlab.idiap.ch/bob/bob.db.msceleb
- bob.db.pericrosseye: https://gitlab.idiap.ch/bob/bob.db.pericrosseye
- bob.db.eprip: https://gitlab.idiap.ch/bob/bob.db.eprip

[3] https://www.idiap.ch/software/bob/

[4] https://www.idiap.ch/software/bob/docs/bob/docs/stable/#database-interfaces

[5] First public protocol for this database

Other software components

Follow below a list of other software components developed and maintained along the course of this thesis. Those software components encompass the implementation and/or integration of machine learning and signal processing algorithms:

- bob.bio.htface: https://gitlab.idiap.ch/bob/bob.bio.htface
- bob.bio.base: https://gitlab.idiap.ch/bob/bob.bio.base
- bob.bio.face: https://gitlab.idiap.ch/bob/bob.bio.face
- bob.bio.face_ongoing: https://gitlab.idiap.ch/bob/bob.bio.face_ongoing
- bob.bio.gmm: https://gitlab.idiap.ch/bob/bob.bio.gmm
- bob.bio.spear: https://gitlab.idiap.ch/bob/bob.bio.spear
- bob.bio.caffe_face: https://gitlab.idiap.ch/bob/bob.bio.caffe_face
- bob.learn.tensorflow: https://gitlab.idiap.ch/bob/bob.learn.tensorflow
- bob.ip.tensorflow_extractor: https://gitlab.idiap.ch/bob/bob.ip.tensorflow_extractor
- bob.learn.em: https://gitlab.idiap.ch/bob/bob.learn.em
- bob.ip.mtcnn: https://gitlab.idiap.ch/bob/bob.ip.mtcnn
- bob.ip.dlib: https://gitlab.idiap.ch/bob/bob.ip.dlib
- bob.bio.challenge_uccs: https://gitlab.idiap.ch/bob/bob.bio.challenge_uccs
- bob.paper.tifs2018_dsu: https://gitlab.idiap.ch/bob/bob.paper.tifs2018_dsu
- bob.thesis.tiago: https://gitlab.idiap.ch/bob/bob.thesis.tiago
- bob.math: https://gitlab.idiap.ch/bob/bob.math
- bob.learn.linear: https://gitlab.idiap.ch/bob/bob.learn.linear

6.3.2 Contributions to other software libraries

Along the course of this thesis, contributions to other open source software libraries was also carried out. Follow below the list of the most relevant ones:

1. Tensorflow: https://github.com/tensorflow/tensorflow/pull/11824
2. Caffe: https://github.com/BVLC/caffe/pull/4194
3. Scikit Learn: https://github.com/scikit-learn/scikit-learn/pull/4761

6.4 Directions for Future Work

The following items are some possible directions for future extensions of this thesis.

1. The loss functions used to train the DSU approach (see Chapter 5) expects pairs of images from the same identity sensed in different image modalities. This limit the extensibility of this approach to other image domains, since a data collection needs to be carried out taking into account this requirement and this can be time consuming. Strategies to "break" this requirement shall be studied. This raises a fundamental question on **what is domain and what is identity**. One possibility approach this issue would be to craft a specific loss function that relies in other outputs of the DCNN rather than its embedding.

2. The embeddings from the DSU approach could be used as a front-end to another layer of classification, such as PLDA [El Shafey et al., 2013], Extreme Value Machine (EVM) [Rudd et al., 2018] and specially the ISV approach from Chapter 4. This might increase recognition rates.

3. The prior DCNNs used in this thesis were trained using a large scale VIS image dataset which presents high recognition rates in the VIS task using several databases as benchmark. The impact of the quality of this prior DCNN in the HFR task (in terms of recognition rates) shall be studied.

4. In Chapter 5 a glimpse about what the DSU feature detectors are learning was introduced using FFTs. A deep analysis about the interpretability of those feature detectors shall be studied. One possibility would be the usage of Layer-wise Relevance Propagation [Montavon et al., 2018].

A Thesis Software Package

In this appendix installation details of the thesis software package is provided.

The thesis software is based on conda[1] and it is compatible with Linux and MacOS 64-bit operating systems. The **first step** is to install conda >= 4.4 (miniconda is preferred) in the target computer.

Once conda is installed, go to the terminal and type:

```
$ conda create --name bob_thesis_tiago --override-channels \
-c https://www.idiap.ch/software/bob/conda -c defaults \
python=3 bob_thesis_tiago
```

This will create a new conda environment and install the thesis software and all its dependencies.

Once the software is installed, the **second step** is to activate the environment that was just created with the aforementioned command line:

```
$ conda activate bob_thesis_tiago
```

Finally, the **last step** is to test the command line interface with the following command:

```
$ bob bio htface --help
```

The sequence of command lines to reproduce all the experiments of this thesis is available at: https://gitlab.idiap.ch/bob/bob.thesis.tiago.

[1] https://conda.io/

B Training Inception Resnet for VIS Face Recognition

In this appendix details on how to train the Incep. Res. v1 and Incep. Res. v2 models with VIS images used in the chapters 3 and 5 are provided. Both architectures are depicted in Figure 3.11 and were training with RGB and gray scaled images.

As mentioned in Chapter 3, the detected faces from MS-Celeb[Guo et al., 2016] dataset are used. Such dataset contains a substantial amount of mislabeling as can be observed in Figure B.1. In the context of this thesis, this dataset was pruned in a semi-automatic manner. First, only face images detected with the MTCNN face detector [Zhang et al., 2016] are considered. All those faces are detected, cropped, aligned and stored. Then, all the detected faces from a particular identity are pruned using the DBScan clustering algorithm [Ester et al., 1996]. Only samples from high density (minimum of 10 samples) are considered and the rest is discarded. Finally, those pruned set of samples were again pruned, but in this stage the pruning was manual. The outcome of this pruning resulted in a dataset of 8M samples with 87,662 identities. For both architectures 160 × 160 cropped faces are used. This pruning strategy as well as the annotation tool is published here [1].

Both architectures were crafted using tensorflow[2] and they are available on the Bob Framework via this component[3]. The RMSProp optimizer is used as a solver[4] with mini-batches of 90 samples. The learning rate is kept to 0.1 for 65 epochs. Then, it is decreased to 0.01 for 15 epochs and finally decreased once more to 0.001 until the end of the training. In total all the DCNNs are trained for 250 epochs. The weight sum between the center and cross entropy loss proposed by Wen et al. [2016] is used as loss function.

With the software thesis a command line interface is provided to train such DCNNs and an example on how to trigger this tool is described in the code snippet below.

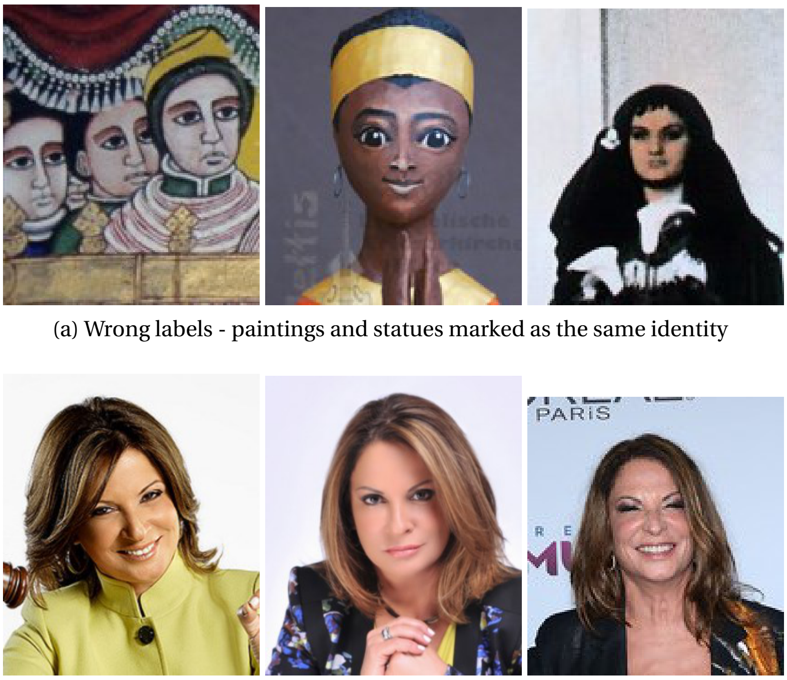
```
bob tf train <CONFIG>
```

[1] http://gitlab.idiap.ch/bob/bob.db.msceleb

[2] https://www.tensorflow.org/

[3] https://gitlab.idiap.ch/bob/bob.learn.tensorflow

[4] tensorflow.org/api_docs/python/tf/train/RMSPropOptimizer



(a) Wrong labels - paintings and statues marked as the same identity

(b) Correct labels - Samples marked as the same identity

Figure B.1 – Samples from the MSCeleb dataset

Developed in the context of this thesis, the input of this command line program is a python script. In this script details of loss, solver, inputs, etc needs to be provided. The code in the end of this appendix is an example of script used as input.

Experiments on VIS images database

Experiments under three different face databases are presented. The first one is the MOBIO database [Marcel et al., 2010]. The MOBIO database is made of videos recorded from 152 people in 6 different sites from 5 different countries. Such database is focused in the task of face/speaker verification using mobile phones. The second database is the Label Faces in the Wild (LFW) [Learned-Miller et al., 2016]. This database is one of the main references in unconstrained face recognition. In this appendix the unconstrained set of protocols are used. Finally, the last database is the IARPA Janus Benchmark C (IJB-C) database [5]. The IJB-C database is a mixture of frontal and non-frontal images and videos (provided as single frames) from 3531 different identities. The verification protocol is used in this work

Table B.1 presents the Half Total Error Rates (HTER) in the development and evaluation set using the MOBIO database. The same protocol applied in Günther et al. [2016] is applied in this experiment. Please, refer to Günther et al. [2016] for further details.

[5] https://www.nist.gov/programs-projects/face-challenges

Table B.1 – Mobio - HTER% using the mobio-male protocol

#	FR Algorithm	Sets	
		dev	eval
FR Baselines			
1	Incep. Res. v1 - gray scaled	1.35	0.53
2	Incep. Res. v1 - rgb	2.07	0.73
3	Incep. Res. v2 - gray scaled	0.94	0.37
4	Incep. Res. v2 - rgb	0.33	0.29
5	Baseline from [Günther et al., 2016]	3.20	7.50

Table B.2 presents the ten fold average True Positive Identification Rate (TPIR), as well as with its standard deviation, under different thresholds (estimated under different FMR operation points) using the unrestricted protocol from the LFW database. [Learned-Miller et al., 2016].

Table B.2 – LFW - TPIR% under different FMR thresholds

#	FR Algorithm	FMR thresholds		
		0.1	0.01	0.001
FR Baselines				
1	Incep. Res. v1 - gray scaled	98.12 (0.39)	97.18 (0.6)	67.75 (8.01)
2	Incep. Res. v1 - rgb	99.41 (0.29)	98.95 (0.35)	81.15 (9.30)
3	Incep. Res. v2 - gray scaled	99.01 (0.25)	98.88 (0.50)	80.01 (12.12)
4	Incep. Res. v2 - rgb	99.77 (0.19)	99.18 (0.43)	77.75 (30.82)
5	Facenet from [Schroff et al., 2015]	99.6 (0.66)	98.37 (0.82)	93.13 (3.71)

Table B.3 presents the True Positive Identification Rate (TPIR) under different thresholds (estimated under different FMR operation points) using the IJB-C database.

Table B.3 – LFW - TPIR% under different FMR thresholds

#	FR Algorithm	FMR thresholds		
		0.1	0.01	0.001
FR Baselines				
1	Incep. Res. v1 - gray scaled	98.5	92.05	59.10
2	Incep. Res. v1 - rgb	99.1	92.45	65.10
3	Incep. Res. v2 - gray scaled	97.1	90.01	60.40
4	Incep. Res. v2 - rgb	99.0	91.55	62.53
5	Facenet from [Schroff et al., 2015]	97.14	85.94	64.98

In all three experiments, under constrained and unconstrained scenarios, it is possible to observe very high recognition rates. Those recognition rates are competitive with respect to the open source state of the art in face recognition. All these baselines are available for reproducibility in the following package[6].

[6] https://gitlab.idiap.ch/bob/bob.bio.face_ongoing

```
from bob.learn.tensorflow.network import inception_resnet_v2_batch_norm
from bob.learn.tensorflow.estimators import LogitsCenterLoss
from bob.learn.tensorflow.dataset.tfrecords import shuffle_data_and_labels_image_augmentation
from bob.learn.tensorflow.utils.hooks import LoggerHookEstimator
from bob.learn.tensorflow.utils import reproducible
import tensorflow as tf

# HYPER PARAMETERS
learning_rate = 0.1
data_shape = (182, 182, 3)  # size of atnt images
output_shape = (160, 160)
data_type = tf.uint8; batch_size = 90; epochs = 65
architecture=inception_resnet_v2_batch_norm

alpha=0.90; factor=0.02; steps = 2000000

model_dir = "./"
tf_record_path = "./"
n_classes = 87662


# Creating the tf record
def train_input_fn():
    return shuffle_data_and_labels_image_augmentation(tf_record_path, data_shape, data_type, batch_size, epochs=epochs,
                                                      output_shape=output_shape,
                                                      buffer_size=2*(10**4),
                                                      random_flip=True,
                                                      random_brightness=False,
                                                      random_contrast=False,
                                                      random_saturation=False,
                                                      per_image_normalization=True,
                                                      random_rotate=True,
                                                      gray_scale=True)

session_config, run_config,_,_,_ = reproducible.set_seed(log_device_placement=False)
run_config = run_config.replace(save_checkpoints_steps=2000)

optimizer = tf.train.RMSPropOptimizer(learning_rate, decay=0.9, momentum=0.9, epsilon=1.0)
estimator = LogitsCenterLoss(model_dir=model_dir,
                             architecture=architecture,
                             optimizer=optimizer,
                             n_classes=n_classes,
                             embedding_validation=embedding_validation,
                             validation_batch_size=validation_batch_size,
                             alpha=alpha,
                             factor=factor,
                             config=run_config)

hooks = [tf.train.SummarySaverHook(save_steps=1000,
                                   output_dir=model_dir,
                                   scaffold=tf.train.Scaffold(),
                                   summary_writer=tf.summary.FileWriter(model_dir) )]
```

C Domain Specific Units, Special Case for Unconstrained Face Recognition

In this appendix an use case of the DSU approach for unconstrained and open set face recognition is presented. This work was submitted as part of the 2nd Unconstrained Face Detection and Open Set Recognition Challenge presented in the ECCV 2018 [1].

The UCCS dataset used in this challenge was collected over several months using Canon 7D camera fitted with Sigma 800mm F5.6 EX APO DG HSM lens, taking images at one frame per second, during times when many students of the University of Colorado were walking on the sidewalk. The images captured cover various weather conditions such as sunny versus snowy days and also contain various occlusions such as sunglasses, winter caps or even occlusion due to tree branches or poles as can be seen in Figure C.1.

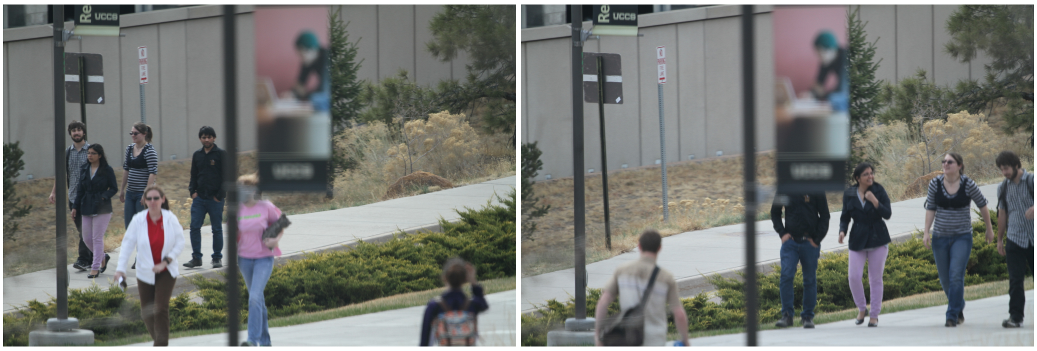


Figure C.1 – Example images of the UCCS dataset[1]

In total around 70,000 face regions were manually cropped and part of those were labeled into 1,732 identities. This data was split into training, validation and test set where the training and the validation sets were provided for fine tune possible submissions. The test set is used to report recognition rates.

There are two major challenges in this database. First, faces inside of those captured frames contains strong variations on poses, levels of blurriness and occlusion as can be observed in

[1] http://vast.uccs.edu/Opensetface/

Figure C.2. A second challenge arises from the fact of being an open-set problem [Jain and Li, 2011, p.551], where unknown people will be seen during testing and must be rejected.

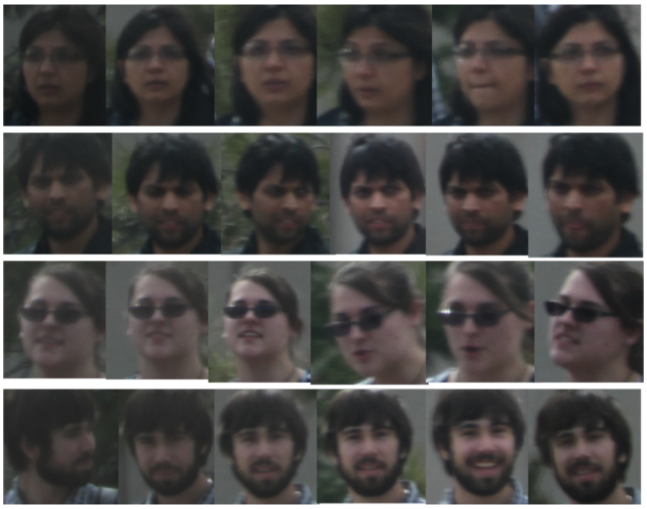


Figure C.2 – Examples of pose, occlusion and blurriness variations of the UCCS dataset[1]

For this work it is hypothesized that those sources of blurriness is another image modality and that recognition rates using an arbitrary pre-trained DCNN can be improved by using the DSU strategy from chapter 5.

For this contest submission, detected faces are cropped and scaled to 160 × 160 and feed into the Inception Resnet v2 CNN (see chapter 3). Then, two DSUs are trained using Siamese Networks strategy: $\theta_{t[1-1]}$ and $\theta_{t[1-2]}$ (see 5.1) where both convolutional kernels and biases were adapted. The 128-d embeddings are used as a front-end to a PLDA probabilistic model [El Shafey et al., 2013] where enrollment and scoring are carried out.

Figure C.3 presents the Detection & Identification Rate curve on the test set from 5 different systems submitted to the contest, which was published in an anonymized manner. It is possible to observe very low identification rates for all submitted systems once the number of false identifications varies from 1 to 100. Moving the decision threshold to 1000 False Identifications it is possible to observe an identification rate increase in all systems. However, the best submitted systems (A2 and A3) presents an identification rate of ≈ 50%. The DSU approach $\theta_{t[1-2]}$ presents the best identification rates in the range of 8,000 to 1,000 False Identification with an identification rate of ≈ 78%. The source code submitted for this contest was made open source and can be accessed [2].

With this contest it was possible to observe that open-set and unconstrained face recognition is an open problem in biometrics and computer vision research in general. With a minimum number of False Identifications all submitted systems presented very low identification rates, which limit the application of this technology in real world scenarios.

[2] https://gitlab.idiap.ch/bob/bob.bio.challenge_uccs

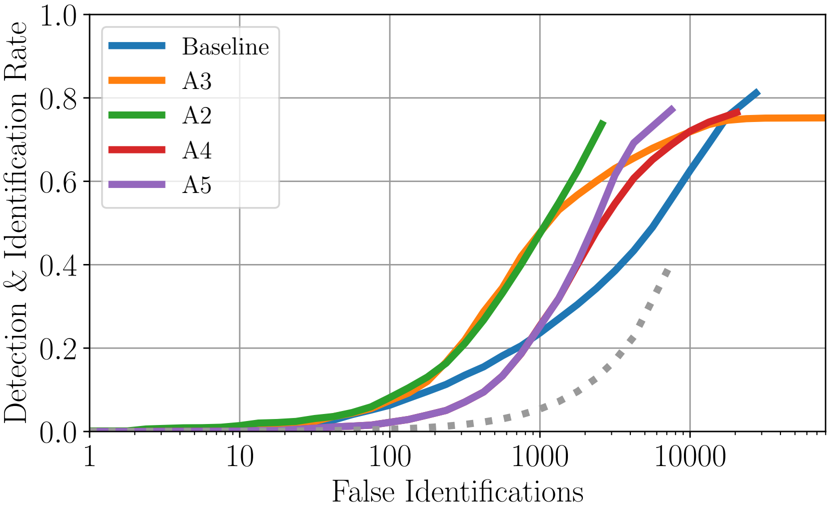


Figure C.3 – Detection & Identification Rate curve published in 2nd Unconstrained Face Detection and Open Set Recognition Challenge. The systems A4 for and A5 stands for the DSU $\theta_{t[1-1]}$ and $\theta_{t[1-2]}$ respectively.

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

André Anjos, Manuel Günther, Tiago de Freitas Pereira, Pavel Korshunov, Amir Mohammadi, and Sébastien Marcel. Continuously reproducing toolchains in pattern recognition and machine learning experiments. In *Thirty-fourth International Conference on Machine Learning*, August 2017. URL https://www.idiap.ch/software/bob/. https://openreview.net/group?id=ICML.cc/2017/RML.

Enrique Bailly-Bailliére, Samy Bengio, Frédéric Bimbot, Miroslav Hamouz, Josef Kittler, Johnny Mariéthoz, Jiri Matas, Kieron Messer, Vlad Popovici, Fabienne Porée, et al. The banca database and evaluation protocol. In *International conference on Audio-and video-based biometric person authentication*, pages 625–638. Springer, 2003.

Peter N Belhumeur, Joao P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *European Conference on Computer Vision*, pages 43–58. Springer, 1996.

John Bernhard, Jeremiah Barr, Kevin W Bowyer, and Patrick Flynn. Near-ir to visible light face matching: Effectiveness of pre-processing options for commercial matchers. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–8. IEEE, 2015.

Himanshu S Bhatt, Samarth Bharadwaj, Richa Singh, and Mayank Vatsa. On matching sketches with digital face images. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE, 2010.

Himanshu S Bhatt, Samarth Bharadwaj, Richa Singh, and Mayank Vatsa. Memetically optimized mcwld for matching sketches with digital face images. *IEEE Transactions on Information Forensics and Security*, 7(5):1522–1535, 2012.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Thirimachos Bourlai, Nathan Kalka, Arun Ross, Bojan Cukic, and Lawrence Hornak. Cross-spectral face verification in the short wave infrared (swir) band. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1343–1347. IEEE, 2010.

Fabien Cardinaux, Conrad Sanderson, and Samy Bengio. User authentication via adapted statistical models of face images. *IEEE Transactions on Signal Processing*, 54(1):361–373, 2006.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

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

Tiago de Freitas Pereira and Séastien Marcel. Periocular biometrics in mobile environment. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–7. IEEE, 2015.

Tiago de Freitas Pereira and Sébastien Marcel. Heterogeneous face recognition using inter-session variability modelling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 111–118, 2016.

Tiago de Freitas Pereira, André Anjos, and Sébastien Marcel. Heterogeneous face recognition using domain specific units. *IEEE Transactions on Information Forensics and Security*, page 13, February 2019.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Tejas Indulal Dhamecha, Praneet Sharma, Richa Singh, and Mayank Vatsa. On effectiveness of histogram of oriented gradient features for visible to near infrared face matching. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1788–1793. IEEE, 2014.

Laurent El Shafey, Chris McCool, Roy Wallace, and Sébastien Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1788–1794, July 2013. doi: 10.1109/TPAMI.2013.38. URL https://pypi.python.org/pypi/xbob.paper.tpami2013.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Christian Galea. *Face Photo-Sketch Recognition using Deeply-Learned and Engineered Features*. PhD thesis, University of Malta, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

Manuel Günther. Statistical gabor graph based techniques for the detection, recognition, classification, and visualization of human faces. 2011.

Manuel Günther, Dennis Haufe, and Rolf P Würtz. Face recognition with disparity corrected gabor phase differences. In *International Conference on Artificial Neural Networks*, pages 411–418. Springer, 2012.

Manuel Günther, Laurent El Shafey, and Sébastien Marcel. Face recognition in challenging environments: An experimental and reproducible research survey. In Thirimachos Bourlai, editor, *Face Recognition Across the Imaging Spectrum*. Springer, 1 edition, February 2016.

Manuel Günther, Laurent El Shafey, and Sébastien Marcel. 2d face recognition: An experimental and reproducible research survey. Idiap-RR Idiap-RR-13-2017, Idiap, 4 2017.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

Hu Han, Brendan F Klare, Kathryn Bonnen, and Anil K Jain. Matching composite sketches to face photos: A component-based approach. *IEEE Transactions on Information Forensics and Security*, 8(1):191–204, 2013.

Simon Haykin. *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Learning invariant deep representation for nir-vis face recognition. In *AAAI*, volume 4, page 7, 2017.

Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Wasserstein cnn: Learning invariant features for nir-vis face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Guillaume Heusch, Tiago de Freitas Pereira, and Sebastien Marcel. A comprehensive experimental and reproducible study on selfie biometrics in multistream and heterogeneous settings. *(Paper Submitted to) - IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Shuowen Hu, Nathaniel J Short, Benjamin S Riggan, Christopher Gordon, Kristan P Gurton, Matthew Thielke, Prudhvi Gurram, and Alex L Chan. A polarimetric thermal database for face recognition research. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 119–126, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2nd edition, 2011.

Yi Jin, Jiwen Lu, and Qiuqi Ruan. Coupled discriminative feature learning for heterogeneous face recognition. *IEEE Transactions on Information Forensics and Security*, 10(3):640–652, 2015.

Dongoh Kang, Hu Han, Anil K Jain, and Seong-Whan Lee. Nighttime face recognition at large standoff: Cross-distance and cross-spectral matching. *Pattern Recognition*, 47(12): 3750–3766, 2014.

Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1435–1447, 2007.

Brendan Klare, Zhifeng Li, and Anil K Jain. Matching forensic sketches to mug shot photos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):639–646, 2011.

Brendan F Klare and Anil K Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1410–1422, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Erik Learned-Miller, Gary B Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. Labeled faces in the wild: A survey. In *Advances in face detection and facial image analysis*, pages 189–248. Springer, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Zhen Lei, Shengcai Liao, Anil K Jain, and Stan Z Li. Coupled discriminant analysis for heterogeneous face recognition. *IEEE Transactions on Information Forensics and Security*, 7(6): 1707–1716, 2012.

Hongyang Li, Huchuan Lu, Zhe Lin, Xiaohui Shen, and Brian Price. Lcnn: Low-level feature embedded cnn for salient object detection. *arXiv preprint arXiv:1508.03928*, 2015.

Stan Li, Dong Yi, Zhen Lei, and Shengcai Liao. The casia nir-vis 2.0 face database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 348–353, 2013.

Stan Z Li, Zhen Lei, and Meng Ao. The hfb face database for heterogeneous face biometrics research. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2009.

Shengcai Liao, Dong Yi, Zhen Lei, Rui Qin, and Stan Z Li. Heterogeneous face recognition from local structures of normalized appearance. In *International Conference on Biometrics*, pages 209–218. Springer, 2009.

Sifei Liu, Dong Yi, Zhen Lei, Stan Z Li, et al. Heterogeneous face image matching using multi-scale features. In *ICB*, pages 79–84, 2012.

Xiaoxiang Liu, Lingxiao Song, Xiang Wu, and Tieniu Tan. Transferring deep representation for nir-vis heterogeneous face recognition. In *Biometrics (ICB), 2016 International Conference on*, pages 1–8. IEEE, 2016.

Jiwen Lu, Venice Erin Liong, and Jie Zhou. Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1979–1993, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374 (2065):20150203, 2016.

Sébastien Marcel, Chris McCool, Pavel Matějka, Timo Ahonen, Jan Černocký, Shayok Chakraborty, Vineeth Balasubramanian, Sethuraman Panchanathan, Chi Ho Chan, Josef Kittler, Norman Poh, Benoît Fauve, Ondřej Glembek, Oldřich Plchot, Zdeněk Jančík, Anthony Larcher, Christophe Lévy, Driss Matrouf, Jean-François Bonastre, Ping-Han Lee, Jui-Yu Hung, Si-Wei Wu, Yi-Ping Hung, Lukáš Machlica, John Mason, Sandra Mau, Conrad Sanderson, David Monzo, Antonio Albiol, Hieu V. Nguyen, Li Bai, Yan Wang, Matti Niskanen, Markus Turtinen, Juan Arturo Nolazco-Flores, Leibny Paola Garcia-Perera, Roberto Aceves-Lopez, Mauricio Villegas, and Roberto Paredes. On the results of the first mobile biometry (mobio) face and speaker verification evaluation. In Devrim Ünay, Zehra Çataltepe, and Selim Aksoy, editors, *Recognizing Patterns in Signals, Speech, Images and Videos*, pages 210–225, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-17711-8.

Aleix M Martinez. The ar face database. *CVC Technical Report24*, 1998.

Chris McCool, Roy Wallace, Mitchell McLaren, Laurent El Shafey, and Sébastien Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2(3):117–129, September 2013. ISSN 2047-4938. doi: 10.1049/iet-bmt.2012.0059.

Christopher McCool and Sébastien Marcel. Parts-based face verification using local frequency bands. In *International Conference on Biometrics*, pages 259–268. Springer, 2009.

Geoffrey McLachlan and David Peel. Finite mixture models, willey series in probability and statistics, 2000.

Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker, 1988.

Kieron Messer, Josef Kittler, Mohammad Sadeghi, Sebastien Marcel, Christine Marcel, Samy Bengio, Fabien Cardinaux, Conrad Sanderson, Jacek Czyz, Luc Vandendorpe, et al. Face verification competition on the xm2vts database. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 964–974. Springer, 2003.

Ethan Meyers and Lior Wolf. Using biologically inspired features for face processing. *International Journal of Computer Vision*, 76(1):93–104, 2008.

John Lester Miller. *Principles of infrared technology*. Springer, 1994.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

Chunlei Peng, Xinbo Gao, Nannan Wang, and Jie Li. Superpixel-based face sketch–photo synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(2):288–299, 2017.

Tiago F Pereira, Marcus A Angeloni, Flávio O Simões, and José Eduardo C Silva. Video-based face verification with local binary patterns and svm using gmm supervectors. In *International Conference on Computational Science and Its Applications*, pages 240–252. Springer, 2012.

P Jonathon Phillips, Sandor Z Der, Patrick J Rauss, and Or Z Der. *FERET (face recognition technology) recognition algorithm development and test results.* Army Research Laboratory Adelphi, MD, 1996.

Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

Yann Rodriguez and Sébastien Marcel. Face authentication using adapted local binary pattern histograms. In *European Conference on Computer Vision*, pages 321–332. Springer, 2006a.

Yann Rodriguez and Sébastien Marcel. Face authentication using adapted local binary pattern histograms. In *9th European Conference on Computer Vision (ECCV)*, 2006b. IDIAP-RR 06-06.

Arun A Ross, Karthik Nandakumar, and Anil K Jain, editors. *Handbook of biometrics.* US: Springer, 2008.

Hiranmoy Roy and Debotosh Bhattacharjee. Local-gravity-face (lg-face) for illumination-invariant and heterogeneous face recognition. *IEEE Transactions on Information Forensics and Security*, 11(7):1412–1424, 2016.

E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult. The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):762–768, March 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2707495.

Daniel L Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. In *Advances in neural information processing systems*, pages 551–558, 1994.

Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Ana F Sequeira, Lulu Chen, James Ferryman, Peter Wild, Fernando Alonso-Fernandez, Josef Bigun, Kiran B Raja, Raghavendra Raghavendra, Christoph Busch, Tiago de Freitas Pereira, et al. Cross-eyed 2017: Cross-spectral iris/periocular recognition competition. In *Biometrics (IJCB), 2017 IEEE International Joint Conference on*, pages 725–732. IEEE, 2017.

Linlin Shen and Li Bai. A review on gabor wavelets for face recognition. *Pattern analysis and applications*, 9(2-3):273–292, 2006.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich. Going deeper with convolutions: Ieee conference on computer vision and pattern recognition (cvpr), 2015.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

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

Robbie Vogt and Sridha Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, 2008.

Robert J Vogt, Brendan J Baker, and Sridha Sridharan. Modelling session variability in text independent speaker verification. 2005.

Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1955–1967, 2008.

Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. In *International Conference on Computer Analysis of Images and Patterns*, pages 456–463. Springer, 1997.

Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *Workshop on faces in'real-life'images: Detection, alignment, and recognition*, 2008.

Xiang Wu, Lingxiao Song, Ran He, and Tieniu Tan. Coupled deep learning for heterogeneous face recognition. *arXiv preprint arXiv:1704.02450*, 2017.

Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

Rolf P Würtz. *Multilayer dynamic link networks for establishing image point correspondences and visual object recognition.* Deutsch Frankfurt am Main, 1995.

Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

He Zhang, Vishal M Patel, Benjamin S Riggan, and Shuowen Hu. Generative adversarial network-based synthesis of visible faces from polarimetrie thermal faces. In *Biometrics (IJCB), 2017 IEEE International Joint Conference on*, pages 100–107. IEEE, 2017.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10): 1499–1503, 2016.

Teng Zhang, Arnold Wiliem, Siqi Yang, and Brian Lovell. Tv-gan: Generative adversarial network based thermal to visible face recognition. In *2018 International Conference on Biometrics (ICB)*, pages 174–181. IEEE, 2018.

Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 513–520. IEEE, 2011.

Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 786–791. IEEE, 2005.

Tiago de Freitas Pereira

IDIAP Research Institute
Rue Marconi, 19
Martigny - 1920
Switzerland

Phone:	(+41) 76 764 7949
Email:	`tiago.pereira@idiap.ch`
Homepage:	`https://scholar.google.com.br/citations`

Personal

Born on August 7th, 1985.
Brazil

Education

PhD in Electrical Engineering, started in 2014
Field: Heterogeneous Face Recognition
Supervisor: Dr. Sébastien Marcel
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

MSc in Electrical Engineering, 2013
Field: Antispoofing in face authentication systems
Dissertation Title: A Comparative Study of Countermeasures to Detect Spoofing Attacks in Face Authentication Systems
Supervisor: Professor Dr. José Mario de Martino
School of Electrical and Computer Engineering, State University of Campinas (UNICAMP), Brazil

BSc in Computer Science, 2010
Scientific Project: Classification of sentences for automatic text simplification.
Funding Agency: FAPESP (Funding council for the state of São Paulo)
Supervisor: Professor Dr. Sandra Maria Aluísio
Institute of Mathematics and Computer Sciences, University of São Paulo (USP), Brazil

Employment

Idiap - Biometrics Group (http://www.idiap.ch/scientific-research/research-groups/biometric-person-recognition), Research Assistant, 2014-now
Developing research in the field of Heterogeneous Face Recognition, which consists in the comparison of faces sensed in different image modalities, such as photographs with near infra-red, sketches or thermogram images. Also responsible in the development and maintenance of the machine learning and signal processing library called Bob (http://idiap.github.io/bob/).

Samsung Research America (http://thinktankteam.info/) Ph.D. Intern, April-July 2017
Part of the Think Tank Team - a small team of interdisciplinary researchers, scientists, designers and engineers, passionate about inventing experience-centric future products and technologies. The group aim to transform their disruptive concepts into products that connect objects, environments, information and people. I was mainly focused in Machine Learning Research.

CPqD Telecom and IT Solutions (https://www.cpqd.com.br/en/), 2010-2014
Worked in a research project whose goal was to develop the technology of face and speaker authentication. I was mainly focused in the research in face authentication, although I was also supporting the speaker authentication team. The outcome of this project was a product called CPqD Smart Authentication.

Publications

Book Chapters

PEREIRA, TIAGO DE FREITAS ; ANGELONI, MARCUS DE ASSIS. Verificação Facial em Vídeos Capturados por Dispositivos Móveis. In: Luiz Antônio Pereira Neves;Hugo Vieira Neto;Adilson Gonzaga. (Org.). Avanços em Visão Computacional. 1ed.Curitiba: Omnipax, 2012, v. 1, p. 181-200.

Journal Articles

T. d. F. Pereira, A. Anjos and S. Marcel, "Heterogeneous Face Recognition Using Domain Specific Units," in IEEE Transactions on Information Forensics and Security. doi: 10.1109/TIFS.2018.2885284

FREITAS PEREIRA, TIAGO; KOMULAINEN, JUKKA; ANJOS, ANDRÉ; DE MARTINO, JOSÉ MARIO; HADID ABDENOUR; PENTIKÄINEN, MATTI and MARCEL SÉBASTIEN. "Face liveness detection using dynamic texture." EURASIP Journal on Image and Video Processing 2014, no. 1 (2014): 2.

Guillaume Heusch, Tiago de Freitas Pereira, and Sebastien Marcel. A comprehensive exper- imental and reproducible study on selfie biometrics in multistream and heterogeneous settings. (Paper Submitted to) - IEEE Transactions on Biometrics, Behavior, and Identity Science, 2019

Proceedings

FREITAS PEREIRA, TIAGO, and SÉBASTIEN MARCEL. "Heterogeneous Face Recognition using Inter-Session Variability Modelling." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016.

PEREIRA, T. F. ; ANJOS, A. R. ; MARTINO, J. M. ; MARCEL, S. . Can face anti-spoofing countermeasures work in a real world scenario?. In: 6th IAPR International Conference on Biometrics (ICB2013), 2013, Madrid, Spain. 6th IAPR International Conference on Biometrics (ICB2013), 2013.

ANJOS, A.; GUNTER, M.; de FREITAS PEREIRA, T.; KORSHUNOV, P.; MOHAMMADI, A. and MARCEL, S. (2017). "Continuously reproducing toolchains in pattern recognition and machine learning experiments."

SEQUEIRA, ANA, et al. "Cross-Eyed 2017: Cross-Spectral Iris/Periocular Recognition Competition." IEEE/IAPR International Joint Conference on Biometrics. No. EPFL-CONF-233586. IEEE, 2017.

FREITAS PEREIRA, TIAGO, and SÉBASTIEN MARCEL. "Periocular biometrics in mobile environment." Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. IEEE, 2015.

BEVERIDGE, J. ROSS, et al. "The ijcb 2014 pasc video face and person recognition competition." Biometrics (IJCB), 2014 IEEE International Joint Conference on. IEEE, 2014.

GUNTER, M., et al. ; The 2013 Face Recognition Evaluation in Mobile Environment. In: 6th IAPR International Conference on Biometrics (ICB2013), 2013, Madrid, Spain. 6th IAPR International Conference on Biometrics (ICB2013), 2013.

KHOURY, E., et al. ; The 2nd Competition on Counter Measures to 2D Face Spoofing Attacks. In: 6th IAPR International Conference on Biometrics (ICB2013), 2013, Madrid, Spain. 6th IAPR International Conference on Biometrics (ICB2013), 2013.

CHINGOVSKA, I., et al. ; The 2nd Competition on Counter Measures to 2D Face Spoofing Attacks. In: 6th IAPR International Conference on Biometrics (ICB2013), 2013, Madrid, Spain. 6th IAPR International Conference on Biometrics (ICB2013), 2013.

FREITAS PEREIRA, TIAGO ; ANJOS, ANDRÉ ; MARTINO, JOSÉ MARIO ; MARCEL, SÉBASTIEN . LBP-TOP Based Countermeasure against Face Spoofing Attacks. Lecture Notes in Computer Science. 1ed.: Springer Berlin Heidelberg, 2013, v. , p. 121-132.

Awards

Idiap PhD Student Award 2018

Ce document a été imprimé au Centre d’impression EPFL,
imprimerie climatiquement neutre, certifiée myClimate.