

**FACE RECOGNITION BASED ON A  
COLLECTION OF BINARY CLASSIFIERS**



RAFAEL HENRIQUE VARETO

**FACE RECOGNITION BASED ON A  
COLLECTION OF BINARY CLASSIFIERS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: WILLIAM ROBSON SCHWARTZ  
COORIENTADOR: FILIPE DE OLIVEIRA COSTA

Belo Horizonte  
Outubro de 2017



RAFAEL HENRIQUE VARETO

**FACE RECOGNITION BASED ON A  
COLLECTION OF BINARY CLASSIFIERS**

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais - Departamento de Ciência da Computação. in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

ADVISOR: WILLIAM ROBSON SCHWARTZ  
CO-ADVISOR: FILIPE DE OLIVEIRA COSTA

Belo Horizonte

October 2017

© 2017, Rafael Henrique Vareto.  
Todos os direitos reservados.

Vareto, Rafael Henrique

V296f Face recognition based on a collection of binary classifiers / Rafael Henrique Vareto. — Belo Horizonte, 2017

xxii, 77 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação.

Orientador: William Robson Schwartz; Coorientador: Filipe de Oliveira Costa

1. Computação - Teses. 2. Visão computacional.  
3. Aprendizado de máquina. 4. Percepção visual.  
5. Reconhecimento de faces. 6. Vigilância. I.Orientador.  
II.Coorientador. III Título.

CDU 519.6\*82.10(043)



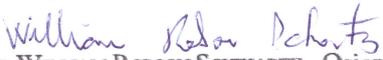
UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Face Recognition based on a Collection of Binary Classifiers

**RAFAEL HENRIQUE VARETO**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. WILLIAM ROBSON SCHWARTZ - Orientador  
Departamento de Ciência da Computação - UFMG

  
DR. FILIPE DE OLIVEIRA COSTA - Coorientador  
Pós-Doc/Departamento de Ciência da Computação - UFMG

  
PROF. GUILLERMO CÁMARA CHÁVEZ  
Departamento de Computação - UFOP

  
PROF. JEFERSSON ALEX DOS SANTOS  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 16 de Outubro de 2017.



# Acknowledgments

First and foremost, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily.

I am eternally thankful for all encouragement and assistance my parents and my sister provided, allowing me to focus on my research and studies. I am completely grateful for their prayers, support, respect and advice throughout the pursuit of the master degree.

I would like to thank deeply professor William Robson Schwartz for the outstanding orientation on my graduate study. Thank you for teaching and advising me for the last two years. Your vital advices helped me map out my following years as a postgraduate student, building up my future career as a researcher.

I must thank my colleagues whom I have worked with at the Federal University of Minas Gerais, specially those who have given your time and expertise: Antonio Nazaré Jr., Artur J. L. Correia, Carlos A. Caetano, Cássio E. Santos Jr., Filipe O. Costa, Gabriel R. Gonçalves, Jéssica S. Souza, Raphael C. Prates, Renan O. Reis, Rensso M. Colque, Samira S. Silva, Victor H. Melo. Also, a special thanks to Luan I. Santana, Pedro H. Matos, all my friends and colleagues from our Small Group (Célula Link) and Bible Study club.

I would like to thank the Brazilian National Research Council – CNPq (Grant #311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project).



*“Whatever you do, work at it with all your heart,  
as working for the Lord, not for human masters.”  
(Colossians 3:23)*



# Resumo

O reconhecimento de faces é um dos problemas mais relevantes em visão computacional quando consideramos sua importância em áreas como vigilância, ciência forense e psicologia. De fato, um sistema de reconhecimento que representa o mundo real deve lidar com vários indivíduos desconhecidos e determinar se uma dada imagem está associada a um sujeito registrado em uma galeria de indivíduos conhecidos ou se dois rostos representam identidades equivalentes. Neste trabalho, não só combinamos funções de indexação, coleção de classificadores e histogramas para estimar quando imagens faciais pertencem à galeria, mas também modelamos a relação entre pares de faces para determinar se elas são da mesma pessoa. Os dois métodos propostos são avaliados em cinco datasets: FRGCv1, LFW, PubFig, PubFig83 e CNN VGGFace. Os resultados são promissores e mostram que o nosso método continua eficiente tanto na verificação e identificação de galeria aberta, independentemente da dificuldade dos datasets.



# Abstract

Face Recognition is one of the most relevant problems in computer vision as we consider its importance to areas such as surveillance, forensics and psychology. In fact, a real-world recognition system has to cope with several unseen individuals and determine either if a given face image is associated with a subject registered in a gallery of known individuals or if two given faces represent equivalent identities. In this work, not only we combine hashing functions, embedding of classifiers and response value histograms to estimate when probe samples belong to the gallery set, but we also extract relational features to model the relation between pair of faces to determine whether they are from the same person. Both proposed methods are evaluated on five datasets: FRGCv1, LFW, PubFig, PubFig83 and CNN VGGFace. Results are promising and show that our method continues effective for both open-set face identification and verification tasks regardless of the dataset difficulty.

**Keywords:** Artificial Neural Network, Support Vector Machine, Partial Least Squares, Open-set Face Identification, Face Verification, Machine Learning, Surveillance.



# List of Figures

|     |  |    |
|-----|--|----|
| 11  | Illustration of three face recognition tasks . . . . .                     | 2  |
| 31  | Pipeline overview of background methods . . . . .                          | 17 |
| 32  | Illustration of a SVM hyperplane between two different classes . . . . .   | 18 |
| 41  | Pipeline overview of the identification training stage - part I . . . . .  | 27 |
| 42  | Pipeline overview of the identification training stage - part II . . . . . | 28 |
| 43  | Illustration of querying known and unknown samples . . . . .               | 29 |
| 44  | Pipeline overview of the identification testing stage - part I . . . . .   | 30 |
| 45  | Pipeline overview of the identification testing stage - part II . . . . .  | 31 |
| 46  | Pipeline overview of the artificial neural network model . . . . .         | 33 |
| 47  | Pipeline overview of the verification approach . . . . .                   | 34 |
| 48  | Overview of relational disparity feature generation process . . . . .      | 35 |
| 49  | Pipeline overview of the verification training stage . . . . .             | 37 |
| 410 | Pipeline overview of the verification testing stage - part I . . . . .     | 40 |
| 411 | Pipeline overview of the verification testing stage - part II . . . . .    | 41 |
| 51  | Images extracted from FRGCv1 dataset . . . . .                             | 44 |
| 52  | Images extracted from LFW dataset . . . . .                                | 45 |
| 53  | Images extracted from PubFig dataset . . . . .                             | 45 |
| 54  | Images extracted from PubFig83 dataset . . . . .                           | 45 |
| 55  | Images extracted from VGGFace dataset . . . . .                            | 46 |
| 56  | ROC curves for the FRGCv1 dataset . . . . .                                | 49 |
| 57  | Open-set ROC curves for FRGCv1 and PubFig83 datasets . . . . .             | 53 |
| 58  | ROC curves for the LFW-A dataset . . . . .                                 | 60 |
| 59  | AUC vs. Runtime contrast with increasingly models . . . . .                | 65 |



# List of Tables

|     |  |    |
|-----|--|----|
| 51  | Comparison of HOG and VGGFace on PubFig83 . . . . .                      | 50 |
| 52  | Average AUC on FRGCv1 dataset . . . . .                                  | 52 |
| 53  | Comparison of single SVM, PLS and ANN classifiers . . . . .              | 54 |
| 54  | Variable number of hashing models on PubFig83 . . . . .                  | 55 |
| 55  | Variable percentage of known individuals on FRGCv1 and VGGFace . . . . . | 55 |
| 56  | Complete battery of experiments on FRGCv1 dataset . . . . .              | 56 |
| 57  | Complete battery of experiments on VGGFace dataset . . . . .             | 57 |
| 58  | Complete battery of experiments on PubFig83 dataset . . . . .            | 58 |
| 59  | Average EER on PubFig dataset . . . . .                                  | 61 |
| 510 | Accuracy evaluation with variable models on LFW-A and PubFig . . . . .   | 62 |
| 511 | Accuracy evaluation with variable number of samples per model . . . . .  | 63 |
| 512 | Runtime evaluation with variable models on LFW-A and PubFig . . . . .    | 64 |



# Contents

|  |           |
|--|-----------|
| Acknowledgments                                      | ix        |
| Resumo   | xiii      |
| Abstract   | xv        |
| List of Figures                                      | xvii      |
| List of Tables                                       | xix       |
| <b>1 Introduction</b>                                | <b>1</b>  |
| 1.1 Applications . . . . .                           | 3         |
| 1.2 Motivation . . . . .                             | 4         |
| 1.3 Objectives . . . . .                             | 4         |
| 1.4 Hypotheses . . . . .                             | 5         |
| 1.5 Scientific Contributions . . . . .               | 5         |
| 1.6 Dissertation Roadmap . . . . .                   | 6         |
| <b>2 Related Work</b>                                | <b>7</b>  |
| 2.1 Open-set Face Identification . . . . .           | 8         |
| 2.2 Face Verification . . . . .                      | 10        |
| 2.3 Hashing-based Approaches . . . . .               | 12        |
| <b>3 Background Concepts</b>                         | <b>17</b> |
| 3.1 Support Vector Machines . . . . .                | 18        |
| 3.2 Partial Least Squares . . . . .                  | 20        |
| 3.3 Random Maximum Margin Hashing . . . . .          | 22        |
| 3.4 Partial Least Squares for Face Hashing . . . . . | 23        |
| <b>4 Methodology</b>                                 | <b>25</b> |

|          |  |           |
|----------|--|-----------|
| 4.1      | Open-set Face Identification . . . . . | 26        |
| 4.1.1    | Training Stage . . . . .               | 27        |
| 4.1.2    | Testing Stage . . . . .                | 29        |
| 4.1.3    | Artificial Neural Network . . . . .    | 32        |
| 4.2      | Face Verification . . . . .            | 34        |
| 4.2.1    | Feature Extraction . . . . .           | 35        |
| 4.2.2    | Training Stage . . . . .               | 36        |
| 4.2.3    | Testing Stage . . . . .                | 38        |
| 4.3      | Technical Comparison . . . . .         | 39        |
| <b>5</b> | <b>Experiments</b>                     | <b>43</b> |
| 5.1      | Common Attributes . . . . .            | 43        |
| 5.1.1    | Datasets . . . . .                     | 43        |
| 5.1.2    | Feature Descriptors . . . . .          | 46        |
| 5.1.3    | Experimental Setup . . . . .           | 47        |
| 5.2      | Open-set Face Identification . . . . . | 47        |
| 5.2.1    | Evaluation Protocol . . . . .          | 48        |
| 5.2.2    | Method Evaluation . . . . .            | 50        |
| 5.3      | Face Verification . . . . .            | 59        |
| 5.3.1    | Evaluation Protocol . . . . .          | 59        |
| 5.3.2    | Method Evaluation . . . . .            | 60        |
| <b>6</b> | <b>Conclusions</b>                     | <b>67</b> |
|          | <b>Bibliography</b>                    | <b>69</b> |

# Chapter 1

## Introduction

Face recognition has been one of the most important tasks in computer vision and biometrics during the last decades. Besides, it is in great rush to help fight crime and terrorism due to its increasing international problem and threat. Because of the wide range of face recognition applications in several environments – e.g., access control, forensics, law enforcement, social media, surveillance systems – and the accessibility of feasible recording and storage technologies in the last years, face recognition tasks received significant attention from the scientific community.

Traditional face recognition approaches conventionally extract image features that correspond to facial components and fiducial points. In general, these methods would initially search for shape of the eyes, mouth contour, nose appearance, face silhouette to name a few and use them as discriminating features while exploring other face images. Particularly, surveillance systems count on quiet and passive acquisition by taking the face image deprived of cooperation or knowledge of the subject being framed, intensifying the recognition process difficulty. For this reason, the approaches developed for face recognition still have some limitations caused by real applications conditions, such as partial occlusion, illumination variation, and camera resolution [Zhang and Gao, 2009].

According to Chellappa et al. [2010], the face recognition problem can be divided into three closely-related categories (see Figure 11): *face verification (FV)*, where the goal is to determine whether a pair of images corresponds to the same subject; *closed-set face identification (CI)*, where we assume that every queried subject was previously cataloged, ensuring that the probe face holds a corresponding identity in the gallery set; and *open-set face identification (OI)*, which is similar to face identification with the difference that it does not guarantee that all query subjects are registered in the face gallery.

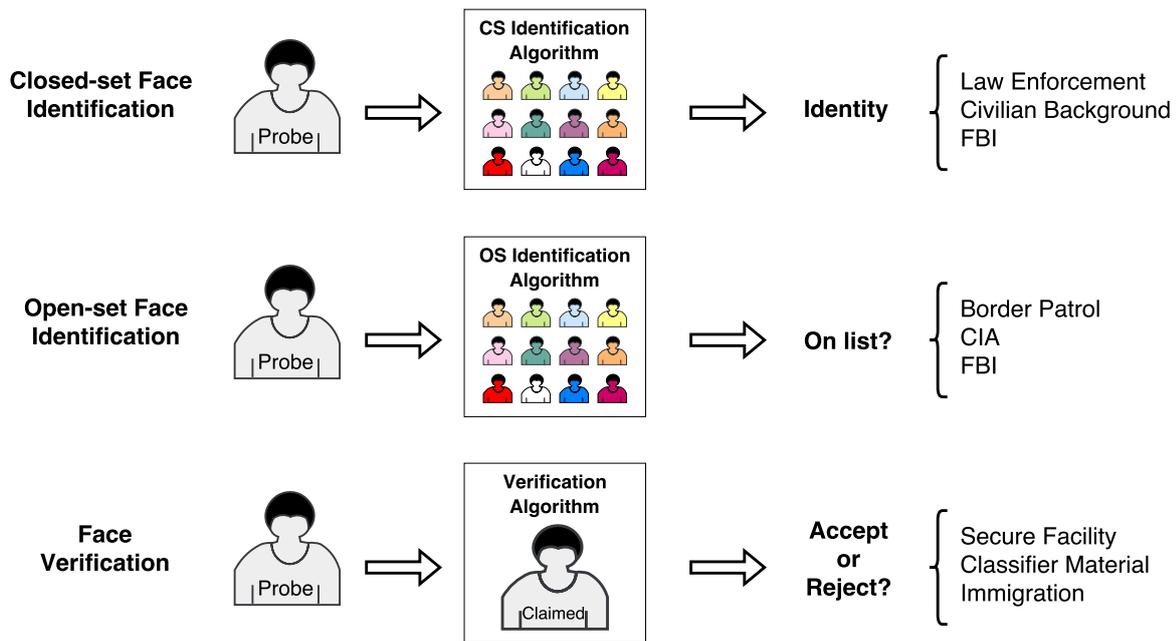


Figure 11: Illustration of three face recognition tasks: closed-set face identification, open-set face identification and face verification. In real scenarios, facial images go under many changes as we consider acquisition limitations and the typical human ageing. The main recognition challenges comprise image resolution, pose variations with respect to the camera, dark environments, light-induced saturation or glare, facial expressions, occlusions or disguises, facial hair, weight loss or gain, and any attribute inhering human beings.

Several researchers have developed approaches to improve the performance of automatic face recognition [Ahonen et al., 2006; Hayat et al., 2017; Klare and Jain, 2013; Lei et al., 2014; Lu et al., 2013; Schroff et al., 2015; Yi et al., 2013]. Basically, in a generic face identification system, the goal is to determine which one of a group of known individuals best matches a probe face sample. They are a suitable means of examining the separability among individuals' faces or finding similar persons. Purely closed-set identification applications are limited to cases where only enrolled persons are encountered. Open-set identification approaches behave like closed-set face identification for known individuals and also label persons not identified by the system into an "unidentified subject" category. In face verification, the goal is to check from a probe face sample whether a person is who she or he claims to be. Particularly, it requires distinguishing an alleged face image known to the recognition system from potentially face samples unknown to the system.

This work combines Locality-Sensitive Hashing (LSH), Support Vector Machine (SVM), Partial Least Squares (PLS) and Artificial Neural Networks (ANN) to get the best of the four worlds towards open-set face identification and face verification. LSH

was designed to solve near-neighbor search in high-dimensional spaces. It hashes data in such a way that similar items tend to map to the same “bucket”. SVMs are supervised learning methods that choose the hyperplane that maximizes the distance to the nearest data points. PLS weights feature descriptors to best discriminate throughout different classes, handling high-dimensional data and successfully overcoming the problem of having few samples per class. ANNs are composed of numerous highly interconnected processing elements (neurons) working in harmony to solve problems that demand machine learning. For the proposed approaches, we replace each LSH random projection either by SVM, PLS or ANN to obtain better discrimination between positive and negative samples. Then, a set of these learned classifiers are employed to solve open-set identification and face verification tasks.

## 1.1 Applications

As one of the most noninvasive biometrics, face recognition is employed in many pervasive computing tasks. It can be an interaction support to impaired users or anyone calling for special needs and assisted social interactions. Advertising agencies are exploring how facial recognition technology can assist in targeting their commercials and helping brands achieve a competitive edge. Video game consoles comprise cameras able to track people in a given space and identify individuals in addition to having the understanding of their activities [Mandal et al., 2014].

Physical access control may be the most obvious application for face identification and verification. Due to security concern, OI technology might be implemented at many areas around the world as it allows the police force and security companies to monitor suspicious and undesirable persons. FV is usually combined with turnstiles and door-locking mechanisms to grant/deny access to office buildings and/or restricted areas [Ashbourn, 2014]. Mobile phones are composed of cameras that are able to verify their users’ identities as another method of authentication to unlock devices. Payment companies are slowly introducing smartphone selfies to authenticate the payment of bills and checks [Okamoto et al., 2015].

In forensics, automated recognition systems replace the manual examination of facial images and videos in search of a match with numerous mugshots. Additionally, surveillance is the application domain holding most face recognition interest. OI can be applied without the subject’s active participation and, indeed, without the subject’s knowledge. Consequently, it is the most widely deployed biometric procedure for video data when safety cannot be taken for granted [Galbally et al., 2014].

## 1.2 Motivation

There is a great demand for face verification and open-set face identification since several problems either cannot or do not have awareness of all possible persons' identities. Strictly speaking, in most real-world scenarios, there is a limited knowledge of known persons in comparison to innumerable unknown individuals.

For a clear understanding, think of an identification application for law enforcement agencies where lawbreakers' identities are doubtless of interest; however, a large number of law-abiding individuals are not of concern. For that reason, an open-set algorithm should dismiss all irrelevant subjects and focus only on identifying potential suspects. In a similar manner, consider a verification-oriented application deployed for access control, in which individuals constantly claim to have certain identity in request for entrance grant. For each query, the verification procedure has to determine whether the biometric samples belong to the claimed identity.

In respect to the aforementioned scenarios, ignoring unsought individuals and neglecting facility access to unknown subjects are both a challenge and a requirement when not all persons are acknowledged [Scheirer et al., 2013]. These challenges require robust and accurate recognition systems. Therefore, they ended up motivating us to study and design methods that can be efficiently applied to open-set face identification and face verification.

## 1.3 Objectives

This work aims at providing efficient and straightforward techniques for the face recognition tasks that deal with unknown individuals, namely face verification and open-set face identification. We can divide the objectives into three different parts. First, the idea is to propose methods that target mainly classification problems: in open-set face identification it is to determine whether a given subject is known and, for face verification, decide whether two faces have equivalent identity. Second, we demonstrate that the information coming from multiple classifiers can improve results. To this end, we evaluate how the methods' performance responds to variable number of hashing functions. Third, we investigate whether it is possible to offer a trade-off between accuracy and simplicity.

In the following chapters, we attempt to show that an embedding of binary classifier is a great alternative to conventional randomized search algorithms, such as LSH, where queries are executed very quickly.

## 1.4 Hypotheses

For the open-set face identification task, our main hypothesis is that vote list histograms proceed differently whether we present probe face images whose identities are enrolled in the gallery set or whether we examine unseen individuals. We presume that when a probe sample is known, most classifiers would vote for the correct identity or otherwise distribute the votes among distinct individuals that were originally registered in the gallery.

For the face verification task, we assume that modeling the relation between two face samples can be useful for increasing the robustness and performance. We propose an approach that compares a pair of faces by extracting relational features and computing the absolute difference between their feature vectors. We believe that any pair of features of the same subject would present small differences and this difference increases when face images come from different persons.

## 1.5 Scientific Contributions

This work is inspired on a method proposed by Santos Junior et al. [2016], which provides a scalable closed-set face identification approach to galleries with hundreds and thousands of subjects. However, instead of working on the closed-set face identification task, we focus on solving open-set face identification and face verification problems.

According to experimental results, our approach reports competitive matching accuracy in comparison with other state-of-the-art works on well-known datasets. The predominant contributions of this master thesis for open-set identification and face verification are:

- An adaptation of locality-sensitive hashing linked with different binary classifiers in a supervised learning setting;
- Easy-to-implement algorithms with a few trade-off parameters to be estimated;
- Fast approaches that are capable of handling the combination of diverse feature descriptors.
- Extensive experimental evaluation and discussion of the proposed algorithms on the two aforementioned face recognition subcategories.

The following publications are related to this work in which the former consists of a description of the open-set face identification method and the latter details the implementation of the face verification algorithm.

- Vareto, R., Silva, S., Costa, F., and Schwartz, W. R. (2017). Towards Open-Set Face Recognition using Hashing Functions<sup>1</sup>. In *International Joint Conference on Biometrics (IJCB)*.
- Vareto, R., Silva, S., Costa, F., and Schwartz, W. R. (2017). Face Verification based on Relational Disparity Features and Partial Least Squares Models. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*.

## 1.6 Dissertation Roadmap

The remainder of this work is organized as follows: In Chapter 2, we review the main open-set face identification and verification techniques, features and approaches that focus on improving face recognition results. Chapter 3 presents some background concepts for the research presented in this thesis: Support Vector Machines and Partial Least Squares. In Chapter 4, we describe the core techniques: embedding of classifiers for open-set face identification and face verification algorithms. Chapter 5 discloses the experiments executed to validate and the discussions regarding the two proposed approaches. Finally, in Chapter 6 we conclude this thesis with final remarks.

---

<sup>1</sup>Best Paper Runner-up Award at the International Joint Conference on Biometrics, 2017.

# Chapter 2

## Related Work

Research in automatic face recognition started during the 1960s. Around 50 years ago, IBM stated that computers users could be recognized at a computer terminal by something they know or memorize, by something they carry and even by a personal characteristic [Jain et al., 2008]. Contemporary years have witnessed a significant development on the face recognition field [Best-Rowden et al., 2014; Guo et al., 2011; Wright et al., 2009; Yi et al., 2013]. In addition, several modeling systems have been developed and deployed.

This chapter presents an overview regarding the main approaches employed in two face recognition tasks. Besides, it also provides proper insights on hashing-based approaches that might be employed to perform search on large scale face galleries. This chapter reading is advisable for anyone who plans to delve into face recognition or wants to become familiar with some of the state-of-the-art methods. Therefore, this chapter is written with two major motivations: prepare the reader for a complete understanding of the proposed approaches and compile relevant approaches available for automated open-set face identification and face verification.

The following sections are not a complete literature review but a summary of some of the most recent works on open-set face identification, face verification and hashing-based approaches. More appropriate information on face recognition may be found in the work of Jain and Li [2011] or in other well-endorsed published documents [Ashbourn, 2014; Jain et al., 2008]. The works described herein show that accurate and robust face recognition continues posing several challenges to computer vision and pattern recognition scientists, especially under unconstrained scenarios. The chapter is divided into three parts, according to the main problems addressed: *(i)* open-set face identification, *(ii)* face verification, and *(iii)* hashing-based approaches, which might be applied to face recognition to search in large galleries.

## 2.1 Open-set Face Identification

Only recently has open-set recognition been investigated in the literature. Specifically for face identification, few works focus on finding thresholds that must be satisfied so that probe individuals are identified as belonging to the gallery, i.e., a known subject. On the other hand, there have been more works intended to solve closed-set problems, either in unrestrained scenarios or in relatively small datasets [Tan and Triggs, 2010; Yi et al., 2013; Zhu et al., 2015b]. It is needless to say that these studies accomplished substantial progress over the last ten years. However, face recognition is still far from being solved since many applications have failed to work properly on open-set scenarios (watch-list task).

The work of Kamgar-Parsi et al. [2011] consists in performing open-world face recognition by learning a classifier for every single subject from a watch-list. Each subject’s model is likely to accept images from corresponding targets and reject everybody else. Liao et al. [2014] develop a protocol to explore all face images available in the Labeled Faces in the Wild (LFW) dataset [Learned-Miller et al., 2016] under verification and open-set identification scenarios. The authors concluded that open-set recognition persists unresolved for large galleries, requiring further attention and effort so that adequate learning algorithms can be assembled.

Support Vector Machines (SVM) are broadly used in image retrieval problems. With the advent of one-class SVM [Schölkopf et al., 2001], some researchers became involved in open-set tasks [Cevikalp and Triggs, 2012; Costa et al., 2012, 2014; Mygdalis et al., 2015] as it seems reasonable to train a classifier with only known positive data. Scheirer et al. [2013] provide thorough supervised open-set recognition formalization. The authors expand existing one-class and binary SVMs with linear kernels to address both generalized open-set recognition and face verification tasks. They introduce the open space risk concept and focus on reducing the error function combining the empirical risk over training data with the risk model for the open space.

The work proposed by Scheirer et al. [2014] also explores the concept of open space risk as it restricts the classification to accommodate non-linear classifiers in multi-class settings. The authors introduce a variation to the standard SVM, designated Weibull-calibrated SVM (W-SVM), an algorithm that combines the useful properties of statistical extreme value theory for score calibration with one-class and binary support vector machines. Experiments demonstrate that W-SVM is considerably better than common binary and multi-class support vector machine formulations. Scherreik and Rigling [2016] detail the probabilistic open-set SVM, a straightforward machine learning algorithm derived from W-SVM that attempts to determine independent

probability classification thresholds for each class. In cases of restricted training time or too much data, one-class SVM faces some issues since it generates kernel models and, therefore, it is not very scalable.

Jain et al. [2014] announce a new algorithm for open-set recognition where models encompass multiple known classes as well as provide the capability of detecting new classes or rejecting unknown categories. Santos Junior and Schwartz [2014] propose five different methods in a single work: one of them stands on discriminating face samples between known and background set. The four remaining methods are based on identification responses. They encompass *Chebyshev* inequality, SVM classifiers and least squares. The approaches explore different features in the data, such as commonplace attributes among enrolled subjects, margin separation or distribution patterns between identification responses. From the five proposed algorithms, only three presented promising results when few subjects are known. Moreover, they do not attain satisfactory accuracy in large gallery sets.

Bendale and Boulton [2015] introduce a formal definition to the open-world problem. The authors think of Nearest Non-Outlier (NNO), an algorithm that continuously update its inner model with additional unseen object categories and no need to retrain the entire model. The algorithm adds object categories incrementally while detecting outliers and managing open space risk. NNO minimizes the risk of falling into open space, a space sufficiently far from any known positive training sample. The algorithm rejects a query  $q$  as an outlier when it is too far from any training sample. As consequence, it labels  $q$  as *unknown* when all classes reject  $q$  as an outlier. The authors explain that as the number of classes in NNO incrementally increases, the performance on both closed and open-set tasks seem to converge. It indicates that adding new classes to the model may be limited by the open-space risk.

The work of Zhang and Patel [2016] proposes a new Sparse Representation and Classification algorithm (SRC). The authors introduce a training stage to the SRC algorithm so that it can be adapted to tackle open-set recognition problems by seeking the sparsest representation in terms of the training data. Results demonstrate that this method is not recommended for datasets with variations in pose and resolution. Bendale and Boulton [2016] expose a method that adapts deep networks to handle open-set recognition as it introduces a new network layer, called OpenMax, capable of estimating the likelihood of an input being from an unknown class. They use scores from the penultimate layer to estimate whether the input is distant to any known training data. Both methods depicted in this paragraph require large training sets in order to accomplish high recognition performance either to span unexpected conditions that may occur in the test set or train deep networks.

All works described in this section are related to open-set identification problems. Some of them learn classifiers taking into account only known positive data, others explore feature open spaces, collection of hashing functions and even artificial neural networks. On the contrary of our proposed method for open-set face identification, most of these methods require massive data to generate good representational models. Besides, some methods only support a restricted number of known classes or face images, excluding even those datasets that do not reach the million scale in terms of individuals. Our method is not evaluated at colossal scale, however it successfully manages datasets containing thousands of individuals and does not require much data for presenting good face discrimination.

## 2.2 Face Verification

Face verification is a largely explored research topic, so numerous works in the literature have been studied in the last years [Guo et al., 2011; Hu et al., 2015; Wagner et al., 2012; Wright et al., 2009]. For the most part, this section focuses on the face verification task with unconstrained face images, i.e., an environment where images are taken having no standard expression, pose, or lighting condition.

Simonyan et al. [2013] detect facial landmarks in favor of aligning and cropping face images before extracting compact feature descriptors derived from fisher vectors on densely sampled SIFT features. Ding et al. [2016] design a new feature descriptor that computes the first derivative of Gaussian operator to lessen illumination effects before detecting feature patterns at both holistic and landmark levels. Landmark detection-based methods may attain higher performance at the cost of massive labeled training data, which seldom is available in practical applications.

Taigman et al. [2014] come up with a facial alignment algorithm found on the detection of fiducial points and facial 3D modeling. They also introduce a deep neural architecture with nine layers to represent face images in a generalized manner. Similarly, Zhu et al. [2015a] present a method that normalizes poses and expressions in pursuance of canonical-view face images. In that work, the authors search for facial landmarks that are later used for meshing the entire image into a 3D object. Three-dimensional models tend to work well, but depending on the subject's pose, information rendered from three-dimensional techniques may end up hindering the recognition performance. Besides, if the faces contain occluded regions, these regions are generally mirrored, resulting in poor normalization results.

Chen et al. [2013] propose a two-step scheme to obtain sparse linear projections. The method compresses the original space into a low-dimensional feature so that a sparse matrix that maps high-dimensional features into a low-dimensional representation can be learned. Barkan et al. [2013] build high-dimensional face representations using hand-crafted feature descriptors such as LBP and SIFT and employ different dimensionality reduction techniques in LFW’s supervised and unsupervised cases. In the final step, multiple representations and image features are combined together using uniform weighting of cosine similarities. Ouamane et al. [2015] adopt a rich multi-scale facial texture representation to enhance performance. The authors propose a new dimensionality reduction technique that transforms the problem of face verification under weakly labeled data into a generalized eigenvalue problem. In general, methods comprised of high-dimensional spaces bring along several obstacles that may prevent further exploration, such as training, computation and storage issues.

Hu et al. [2014] present a deep metric learning method that aims at learning a Mahalanobis distance metric, maximizing inter-class variations and minimizing intra-class variations. A deep neural network learns hierarchical nonlinear transformations to fit a pair of face images into the same feature subspace so that discriminating information can be spotted. Zheng et al. [2015] propose a linear cosine similarity metric learning method based on triangle inequalities and gradient functions. Cost and gradient functions are handled as a mathematical problem, which is solved with an optimization algorithm. Metric learning methods do not usually hold the nonlinear manifolds faces images lie on. Furthermore, nonlinear mapping functions are not explicitly acquired, causing scalability problems.

Sun et al. [2013] introduced a hybrid convolutional network that learns relational visual features so that identity similarities can be pointed out. The network computes local visual features from two face images that are processed through multiple layers for the sake of extracting high-level holistic features. The work presented by Ding and Tao [2015] proposes a deep learning framework to represent faces using multi-modal information. The framework is made up of complementary convolutional neural networks that extract features, which are concatenated with a three-layer stacked auto-encoder. Neural networks are usually hard to train and regularly require the tuning of numerous parameters. Depending on the problem, there are simpler and faster alternatives that may attain better performance, such as support vector machines and decision trees.

Cevikalp and Triggs [2010] represent images as points in a linear feature space and characterize each image set by a convex geometric region. A kernel trick allows the approach to be extended to implicit feature mappings, thus handling complex and

nonlinear manifolds of face images. Yang et al. [2013] create an efficient algorithm to solve regularized nearest points with very low time complexity as they model with an efficient iterative solver. Masi et al. [2016] describe a new procedure of enriching an existing dataset with important facial appearance variations by reshaping the faces it contains. The new synthesized images hold new intra-class facial appearance variations and are an alternative to expensive data collection and labeling.

Former approaches compute low-level features [Dalal and Triggs, 2005; Lowe, 2004; Ojala et al., 2002] whereas others generate mid-level features [Huang et al., 2012b; Lee et al., 2009]. On the contrary, Wen et al. [2016] propose a new loss function, namely center loss, to efficiently enhance the discriminative power of deeply learned features in neural networks. Very few methods attempt to handle cross-age face verification, for instance, Du and Ling [2015] proposes an algorithm that excludes distracting features in a fine-grained level while pre-serving discriminative ones.

The aforementioned works in this section compute similarities between pairs of images to determine whether they represent a unique person. In pursuance of good verification results, some authors detect facial landmarks while others explore high-dimensional feature spaces. Few of them either generate facial 3D modeling of train deep neural networks. Nevertheless, none of them explores embedding of binary classifiers or evaluate their method on cross-dataset scenarios as we do in our proposed algorithm for face verification.

## 2.3 Hashing-based Approaches

When it comes to large galleries, a natural alternative is to replace objects and shapes by their feature vectors and, then, apply some sort of indexing and search strategy in the new low-dimensional space. When it is not employed, searching for the right identity in a dataset containing countless individuals seems pointless. A stable refinement step casts aside individuals enrolled in the gallery that are improbable to correspond to the probe sample identity with low computational penalty.

Hashing-based approaches employed to the face domain usually return a list of relevant candidates that were previously enrolled in a gallery set taking into account their similarity to a query face image. A common associated procedure, the nearest neighbor search [Jegou et al., 2011], consists of pre-processing an entire set as it calculates distances between probe and gallery samples. In most cases, we are not interested in finding just the closest neighbor but several nearest neighbors. The  $k$ -nearest neighbor ( $k$ -NN) search finds the  $k$  most similar gallery set images from the

probe image whereas radius nearest neighbor ( $r$ -NN) search returns all gallery-enrolled images located closer than some distance  $r$  from the query image.

Brute-force nearest neighbor approaches compute the distance of the probe face image to all gallery samples. This aspect is unfeasible for large galleries because it requires adequate algorithms to solve it efficiently. With that in mind,  $kd$ -trees are capable of efficiently perform either  $r$ -NN or  $k$ -NN searches [Silpa-Anan and Hartley, 2008; Zaklouta et al., 2011]. When processing a probe image, the algorithm looks for the closest-corresponding leaf. Then, it searches other face images stored in that leaf before it scans nearby leafs in pursuance of similar faces. The search stops when the distance from the probe image to the leaf is higher than the worst gallery face sample found so far since it indicates that remaining leafs are not going to improve search results.  $kd$ -trees are good search algorithms in low-dimensional spaces; however its efficiency decreases when dimensionality grows.

Spectral hashing [Fowlkes et al., 2004] generates compact binary hash codes for Approximate Nearest Neighbor (ANN) search. It reduces the computational cost based on spectral partitioning making it feasible to apply them to very large gallery sets. Shakhnarovich [2005] design a method to learn a weighted Hamming embedding where the traditional Hamming distance is replaced with its weighted version in order to reduce the collision likelihood of non-neighboring face samples. Kulis and Darrell [2009] propose data-dependent and bit-correlated hash functions created to reduce the cost function measuring the difference between the metric and reconstructed distance of the corresponding binary embeddings in the Hamming space. The method occasionally falls into poor local optima and is not evaluated on large-scale datasets. Ge et al. [2013] optimize space decomposition with two different approaches: a method that does not assume any data distribution and breaks the original problem into a pair of sub-problems, and a second method that guarantees an optimal solution if the input data satisfies a Gaussian distribution. Some experiments show that the approach is exhaustive and may damage ANN performance [Kalantidis and Avrithis, 2014]. Furthermore, high-dimensional data may result in poor performance due to small variation in distance values, stopping the method to differentiate similar and dissimilar visual feature samples.

The work of Muja and Lowe [2014] generates multiple hierarchical cluster trees to arrange binary vectors and searches for the nearest neighbors simultaneously over multiple trees by traversing each tree in a best-first manner. Pham et al. [2015] address the feature indexing problem using a linked-node  $m$ -ary tree (LM-tree) structure to promptly build queries for both Exact and Approximate Nearest Neighbor search (ENNS/ANNS) . The method produces a polar-space-based method of data

decomposition in pursuance of the LM-tree. Along these lines, it narrows down the search space by means of pruning rules. In the last stage, the authors bring up a bandwidth search method to explore tree nodes.

Wang et al. [2015] conclude that scalable face recognition has not been properly addressed yet, so they employ a fast-filtering procedure, which uses an approximation method to return a list of candidates. In the end, the authors employ a slow pairwise comparison that outputs a more accurate candidate list. Their method is outperformed by the approach of Chen et al. [2016], a method that learns a robust model from a large dataset characterized by face variations and generalizes well to other datasets. Tang et al. [2017] propose a new supervised deep hashing algorithm for scalable face image retrieval. The method is based on classification and quantization errors as it synchronously learns feature representation, face indexing and classification models. A deep convolutional network is introduced to learn discriminating feature representations, generate hash codes and predict images labels.

The last direction of relevant work lies on Locality-Sensitive Hashing (LSH) [Datar et al., 2004; Kulis and Grauman, 2012], a family of embedding approaches that reduces the dimensionality of high dimensional data. It is data independent, that is, it does not explore the data distribution. With LSH, there is a high probability feature descriptors map to the same hashing location when they are located in neighboring regions in the feature space. In other words, there is a great chance similar faces end up having close hash codes.

Locality sensitive hashing was introduced by Indyk and Motwani [1998] to solve NN search problems. The concatenation of all these functions is expected to reduce the chance of collision among different-person face images. So, with an increasingly number of hashing functions, it provides higher precision. LSH establishes a family of hash functions  $\mathcal{H}$ , containing arguments  $r, c, p_1, p_2$  when  $p_1 > p_2$  and  $c > 1$ . Thus, for any two feature descriptors  $p$  and  $q$ , a hash function  $h \in \mathcal{H}$  satisfies the following conditions:

$$\text{if } d(p, q) \leq r \text{ then } \textit{Probability}(h(p) = h(q)) \geq p_1$$

$$\text{if } d(p, q) \geq cr \text{ then } \textit{Probability}(h(p) = h(q)) \leq p_2$$

Therefore,  $r$  represents the maximum distance  $d(p, q)$  that associates  $p$  and  $q$  to equivalent buckets with probability  $p_1$ . On the contrary, the second condition guarantees that faraway feature descriptors are not likely to be mapped to the same bucket. Typical LSH employs  $k$  hash functions  $\{h_1, \dots, h_k\}$  – particularly,  $k$  random data-independent hyperplanes designed to hash input feature vectors – originated from sampling  $p$ -stable distributions.

On the contrary of conventional hashing algorithms that prevent collisions, locality sensitive hashing maximizes the chance of similar-person collisions. LSH indexes all feature descriptors in hash tables and searches for near descriptors via hash table lookup. Consequently, each reference feature descriptor  $x$  is placed into a bucket  $h(x)$ . According to Wang et al. [2012], given the probe face feature descriptor  $q$ , the items lying in the bucket  $h(q)$  are described as near items of  $q$ . There are distinct LSH families for different distances or similarities, including Hamming distance, Jaccard coefficient to name a few.

When a gallery set is composed of millions of subjects, the value of  $k$  should be large enough to reduce collisions between different persons, increasing precision and reducing recall. On the other hand, a large value for  $k$  also reduces the collision between feature descriptors from the same subject. To improve recall, hash functions can be gathered in  $l$  groups  $\{g_1, \dots, g_l\}$  of  $k$  hash functions ( $g_i = \{h_{i,1}, \dots, h_{i,k}\}$ ). The combination results in  $l$  hash tables and  $l \times k$  hash functions. Due to theoretical guarantees for random projection-based LSH, many large-scale search applications based on LSH have been developed [Kulis and Grauman, 2009, 2012; Wang et al., 2010].

The approaches described in Chapter 4 replace LSH's random projections with binary classifiers, which are suitable to provide discriminability among individuals known during the training step. Instead of associating a conventional binary string to every single face image sample, the proposed methods compute weighted binary strings whenever probe images emerge. The weights associated to each string bit are obtained from response values of the corresponding binary classifiers. As the approach regards an embedding of classifiers, the string eventually becomes a vector of response values. Then, the similarity score is updated every time a new bit is estimated for an unknown test sample. As a result, our methods operate in a very similar way to LSH without focusing on face hashing since the nature of this work is to certify whether an identity is known or if two face images correspond to the same individual.



# Chapter 3

## Background Concepts

Hashing methods for multi-dimensional indexing have been widely employed in the computer vision field [Irie et al., 2014; Wang et al., 2013; Zhang et al., 2012] as it seems to be a trend to embed visual features into compact hash codes.

This chapter presents some fundamental background concepts for the research topic of this work. It starts with a quick description of two machine learning methods, Support Vector Machines (SVM) and Partial Least Squares (PLS), in Sections 3.1 and 3.2, respectively. Then it summarizes an interesting method that combines Locality-Sensitive Hashing (LSH) with multiple support vector machine models. Last, it refers to another hashing method that replaces the conventional SVM with binary partial least squares models.

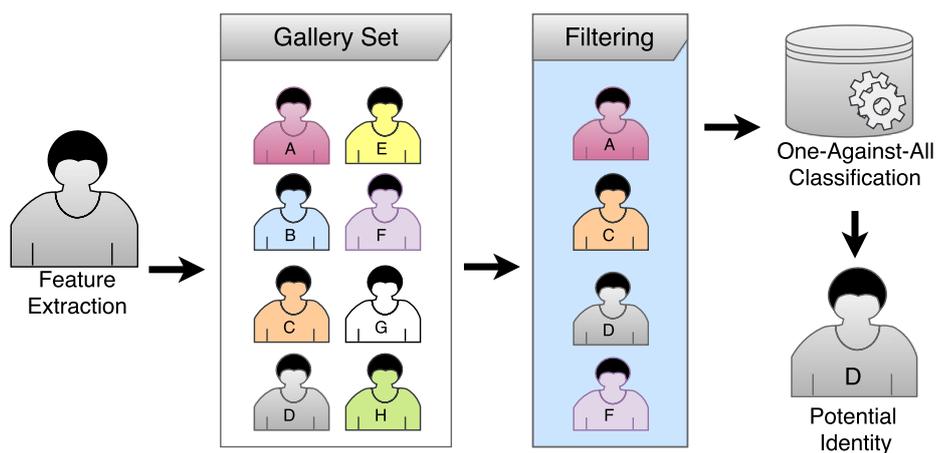


Figure 31: This is the overview of the two background methods under the face image retrieval scenario. The procedure starts with the feature extraction of the entire gallery set. Note that the filtering step reduces the number of evaluations in the classification step with reduced computational cost and, consequently, is the main contribution of both works.

Approaches described in Sections 3.3 and 3.4 are a complementary way to reduce computational time, memory usage and distance calculation cost rather than an alternative to indexing structures. A pipeline overview of the background methods in the face image retrieval context is presented in Figure 31. Instead of increasing the likelihood of colliding neighboring visual features, Random Maximum Margin Hashing [Joly and Buisson, 2011] and Partial Least Squares for Face Hashing [Santos Junior et al., 2016] target data scattering on account of learning hash functions and offering discriminability among items (i.e., subjects for the face recognition domain) in the feature space. By training based on exclusively random fractions of the data, disregarding the training samples proximity, the following methods show that it is feasible to consistently build independent hash functions.

### 3.1 Support Vector Machines

The field of machine learning has gone through intense advancement since the advent of kernel tricks for Support Vector Machine (SVM) during the 1990s. SVM is a collection of supervised learning methods that are broadly used for classification, regression and outliers detection. It is quite effective in high-dimensional spaces even in cases when the space dimensionality is greater than the number of samples [Steinwart and Christmann, 2008].

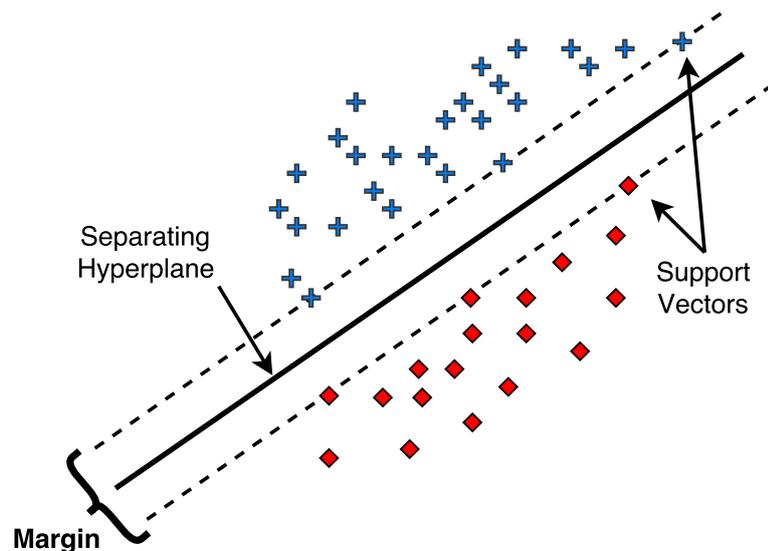


Figure 32: Illustration of a maximum-margin hyperplane and margins for an SVM trained on data points (samples) from two different classes. Data points on the margin are commonly called support vectors.

Figure 3.2 demonstrates how SVM searches for optimal separating hyperplanes among classes as it attempts to maximize the margins between classes' closest points. Boundary-lying points are called support vectors and the centered-margin is the optimal separating hyperplane.

$$\vec{w} \cdot x - b = 0 \quad (3.1)$$

$$\frac{|\vec{w} \cdot x - b|}{\|\vec{w}\|} = \frac{\pm 1}{\|\vec{w}\|} \quad (3.2)$$

Linear SVM considers a training dataset of  $n$  points of the form  $(x_1, y_1), \dots, (x_n, y_n)$  where  $y_i$  is either the  $-1$  or  $+1$  label, which points out the class  $x_i$  belongs to. Point  $x_i$  is a  $p$ -dimensional feature vector. The maximum-margin hyperplane, which dissociates data points having  $y_i = +1$  from those with  $y_i = -1$ , is usually noted as a set of points  $x \in X$  satisfying Equation 3.1 where  $\vec{w}$  is the normal vector to the hyperplane. The distance between the separating hyperplane to the positive and negative support vector is respectively  $\frac{+1}{\|\vec{w}\|}$  and  $\frac{-1}{\|\vec{w}\|}$ , as demonstrated in Equation 3.2. To that end,  $\frac{2}{\|\vec{w}\|}$  is the total distance between the support vectors in such a way that maximizing the distance within the support vector hyperplanes is equivalent to minimizing  $\|\vec{w}\|$ .

SVM selects the hyperplane that best segregates two classes. When more than a single hyperplane is able of segregating classes well, maximizing the distances between nearest data points of each class and the hyperplane is the main criterion employed for identifying the best hyperplane. Having a proper margin is a synonym for robustness since low-margin hyperplanes hold higher probabilities of miss-classification. Linear SVMs are efficient when applied on linearly separable or linearly distributed data points in a way that soft-margin-based machines effectively cope with noise and outliers.

When SVM cannot establish a linear separation, data points are taken into a higher-dimensional space by through a *kernel trick* where they may become linearly separable. The addition of kernel methods enables SVM to run on higher-dimensional feature spaces with no data recalculation in the new space [Cristianini and Shawe-Taylor, 2000]. Kernel functions estimate the inner products from images of all pairs of data in the original space since this arithmetic is computationally cheaper than the generation of new coordinates. The kernel trick challenge is to find a mapping transformation  $\phi = \mathbb{R}^p \rightarrow \mathbb{R}^q, q > p$ , so that data points become linearly separable in  $\mathbb{R}^q$ . With a mapping function  $\phi$ , the new classification pipeline first transforms a training set  $S \in \mathbb{R}^p$  into  $S' \in \mathbb{R}^q$  prior to training a linear SVM classifier and, at

testing stage, a probe sample  $x$  is first transformed to  $x' = \phi(x)$  before it is presented to the classifier. Higher-dimensional feature space usually intensifies support vector machines' generalization error, although given a reasonable number of samples the algorithm maintains its good performance.

## 3.2 Partial Least Squares

Partial Least Squares (PLS) is a fast and effective regression technique based on covariance [Rosipal and Krämer, 2006; Wold, 1985]. It captures the relationship between observed variables (predictors) through latent variables and associates aspects from principal component analysis and multiple regressions. PLS is capable of fitting many response variables in a unique model. Consequently, it models the response variables in a multivariate manner so the results can significantly contrast with those calculated for the response values individually.

Components of a partial least squares model are chosen based on the amount of variance they explain in the predictors and between predictors and their responses. PLS works very well when the number of explanatory variables is both high and likely to be correlated and does not require a large quantity of training samples. The latter aspect is the main motivation for employing PLS in this work, since there are not many samples available for learning the models, preventing the employment of deep learning techniques in the process [Bengio et al., 2013]. In general, when observed variables are highly correlated, the number of components in the PLS model tends to be far lower than the number of predictors.

$$X = TP^T + E \quad (3.3)$$

$$Y = UQ^T + F \quad (3.4)$$

The purpose of partial least squares is to create latent variables as a linear combination of the independent zero-mean variables  $X$  and  $Y$  [Wold, 1985]. More precisely,  $X$  represents a matrix of feature descriptors whereas  $Y$  describes a vector of response variables. Then, PLS searches for latent vectors that can be simultaneously decomposed into Equations 3.3 and 3.4 in order to identify the maximum covariance between these variables. Matrix  $T_{n \times p}$  portrays latent variables from feature vectors and matrix  $U_{n \times p}$  denotes latent variables from target values. Variables  $P_{p \times d}$  and  $Q_{1 \times d}$  can be compared to the loading matrices from principal component analysis. Eventually, variables  $E$  and  $F$  represent residuals [Wold et al., 1987].

---

**Algorithm 1:** NIPALS( $X_{n \times d}, Y_{n \times 1}, p$ )

---

**Data:**  $X_{n \times d}$  and  $Y_{n \times 1}$  stand for visual feature vectors and target values, respectively, holding  $n$  samples,  $d$  dimensions and  $p$  factors.

```

1 for  $i \leftarrow 1$  to  $p$  do
2   start  $u_i$  randomly or with some column of  $X$ 
3   repeat
4      $w_i \leftarrow X'u_i / \|X'u_i\|$ 
5      $t_i \leftarrow Xw_i$ 
6      $q_i \leftarrow Y't_i / \|Y't_i\|$ 
7      $u_i \leftarrow Xq_i$ 
8   until convergence;
9    $b_i \leftarrow u_i t_i / \|t_i\|$ 
10   $p_i \leftarrow X't_i / \|t_i\|$ 
11   $X \leftarrow X - t_i p_i$ 
12   $Y \leftarrow Y - b_i (t_i p_i')$ 
13  return  $T, P, U, Q, W, B$ 
14 end

```

---

The typical algorithm for calculating PLS regression components/factors is the Non-linear Iterative PLS (NIPALS). Algorithm 1 presumes that the  $X$  and  $Y$  variables have been altered to have means of zero. An alternate mechanism for PLS components is the SIMPLS algorithm [De Jong, 1993]. SIMPLS algorithm was literally derived to solve specific objective functions, like maximizing covariance. Andersson [2009] observed that NIPALS is among the most stable algorithms for PLS and, while slightly less accurate, SIMPLS is faster.

$$\beta = W(P^T W)^{-1} T^T Y. \quad (3.5)$$

$$\hat{y} = \bar{y} + \beta^T (x - \bar{x}) \quad (3.6)$$

Even though the distinctions between NIPALS and SIMPLS are of theoretical significance, the practical implications of their difference may well be of minor relevance. Therefore, the PLS for regression we adopted employs NIPALS algorithm to estimate the low-dimensional data representation [Rosipal and Krämer, 2006]. NIPALS computes the highest covariance between latent variables  $T$  and  $U$  and produces a matrix of weight vectors  $W_{d \times p}$  which determines the regression coefficients vector  $\beta$  using least squares as detailed in Equation 3.5. The PLS regression output for query image's feature vector is given by Equation 3.6 where  $\bar{y}$  is the sample mean of  $Y$  and  $\bar{x}$  the average values of  $X$ .

### 3.3 Random Maximum Margin Hashing

Joly and Buisson [2011] come up with the Random Maximum Margin Hashing (RMMH), an approach that learns hash functions with the maximum margin criterion. According to the scheme, positive and negative labels are arbitrarily generated by randomly sampling  $m$  data-independent hyperplanes and randomly labeling half of the items as positive and the other half as negative.

The authors state that the independence deficit among hash functions is the main matter affecting the efficiency of data-dependent hashing methods in contrast with data-independent ones. For that reason, they propose an algorithm that uniformly distributes hash values and boosts the collision probability of close visual features while reducing the collision probability of irrelevant pairs of feature descriptors.

According to the authors, data-dependent hash functions can hold up the performance of a method as the number of irrelevant collisions continues to occur even with an increasingly number of  $m$  hash functions (hash code length). To overcome this issue, they train balanced and independent binary partitions of the feature space. For each hash function, RMMH selects all subjects from the dataset, which comprises the set  $S$ , and randomly labels half of them with  $+1$  (positive partition) and the other half with  $-1$  (negative partition). These positive and negative samples are conventionally denoted as  $x^+$  and  $x^-$ , respectively. The hash function  $h(x)$  is then computed by training a binary classifier  $h_\theta(x)$  in the following:

$$h(x) = \operatorname{argmax}_\theta \sum_{i=1}^{\frac{|S|}{2}} h_\theta(x_i^+) - h_\theta(x_i^-)$$

With RMMH's balanced strategy,  $h(x)$  attempts to minimize the probability a feature sample *spills over* the boundary between positive and negative classes by maximizing the margin between  $x^+$  and  $x^-$ . Hyperplanes that maximize the margin in Euclidean space among arbitrary balanced samples are easily generated with a SVM algorithm, justifying its choice over other classifiers. According to the authors, large margin classifiers have low capacity and, consequently, contribute for a better generalization capability, especially on restricting overfitting occurrences.

During the search procedure, which is very similar to the one from LSH, hash codes are put side by side with the Hamming distance ranked accordingly. However, the authors do not focus on finding the  $k$ -nearest neighbors in the feature space, but the most relevant classes. They relax the classification tolerance to the top five best ranked classes, resulting in a small list of candidates.

## 3.4 Partial Least Squares for Face Hashing

Santos Junior et al. [2016] concentrate efforts on the scalable face identification, that is, when galleries contain several individuals, making it unsuitable for conventional identification algorithms to respond under low computational time.

Similarly to RMMH, Partial Least Squares for face hashing (PLSh) is also inspired by the family of methods regarded as Locality-Sensitive Hashing (LSH), the most widely employed large-scale image retrieval in the literature; and PLS, a technique extensively used to analyze quantitative data, business management and several works regarding face recognition.

PLSh consists in filtering subjects in the gallery. The filtering approach provides a shortlist to the face identification so that it evaluates only the subjects presented in that shortlist. The method encompasses data dependent hash functions and hash functions generated independently from each other. Independently generated hash functions are required to produce uniform distributed binary codes among subjects in the gallery. Data dependent hash functions provide better performance in general since they consider characteristics of the data, such as discriminability among different labels and dimensions, resulting in more consistent hash functions [Indyk and Motwani, 1998].

In the training stage, subjects in the face gallery are randomly divided into two balanced subgroups  $x^+$  and  $x^-$ , as it is done for RMMH. This process is repeated  $m$  times. In other words, a number of  $m$  PLS regression models (also denoted *hashing functions*) are learnt to distinguish individuals in subset  $x^+$  from individuals in  $x^-$ . The association of a subject to one of the two subsets consists in sampling from a Bernoulli distribution with probability  $p = 0.5$ , indicating a fair sampling distribution<sup>1</sup>. Therefore, it can be viewed as a bit in the Hamming embedding and the Bernoulli distribution guarantees that Hamming strings are evenly distributed among all subjects in the face gallery.

During testing, a probe sample is presented to each one of the  $m$  hashing functions in order to obtain a regression response  $r$ . A vote list, having size analogous to the number of subjects and its elements initially filled with zeros, is established and continuously updated with regression values  $r_i$  according to the subject index  $i$  added to the positive subset of the corresponding hash model. In the end, the list of subjects is sorted in descending order and the top candidates are presented for face identification.

---

<sup>1</sup>The Bernoulli distribution is a yes-no question case of the binomial distribution where a single experiment is conducted, outputting a single bit of information whose value is *true/positive* with probability  $p$  and *false/negative* with probability  $1 - p$ . If  $p = 0.5$ , it implies that a sample has equivalent probability of being labeled *true/positive* or *false/negative*.



# Chapter 4

## Methodology

This chapter describes the methods employed in the proposed approaches: an ensemble of classifiers is created as we group several classifiers that enhance the overall performance. For that purpose, it is only required that each classifier alone outperforms randomness. The methods were inspired by the ideas detailed in Chapter 3, but adapted for the face recognition scenario.

We implement two approaches that convert the original data into a metric space where a Hamming distance seems to represent well the similarity between gallery and probe images. The algorithms reported here use an embedding approach in conjunction with binary classifiers, and achieve competitive query performance. Our methods offer a balance between simplicity and learning speed on one hand and accuracy and flexibility of the learned similarity concept on the other hand. To our knowledge, this is the first combination of binary support vector machines or partial least squares models with locality-sensitive hashing to open-set face identification and face verification.

A practical advantage of the approach for open-set face identification presented in this work is that the search for faces resembling a given probe image is reduced to a standard search in the metric embedding space and thus it can be executed quickly. Given a dataset of numerous faces and a probe image, the idea is to retrieve identities from the dataset that are similar to the query without comparing it to every subject sample enrolled in the gallery. Therefore, the embedding of classifiers retrieves with high probability individuals from the gallery set with smallest Hamming distance to the probe face image.

We consider a variation of partial least squares for face hashing in open-set tasks. The proposed methods differ from PLS<sub>h</sub> on how vote-list histograms are interpreted and on how hash models are partitioned. Despite of their easy implementation, the approaches present competitive performance with other literature open-set methods.

Although both proposed approaches only differ in few aspects, we recommend that readers explore each method separately. For that reason, some sections may sound redundant as we do not expect anyone to browse the entire work in pursuance of a complete comprehension of both methods.

## 4.1 Open-set Face Identification

The approach disclosed in this section takes advantage of fitting features from high-dimensional spaces into a more compact space. Instead of having LSH splitting the feature space by establishing random regressions, we appraise Support Vector Machines (SVM), Partial Least Squares (PLS) and Artificial Neural Networks (ANN) for discriminability enhancement and classification.

Like most supervised learning problems, the method is based on three canonical steps: extracting features, training and testing. Features are obtained with Histograms of Oriented Gradients [Dalal and Triggs, 2005] and VGGFace CNN [Parkhi et al., 2015] feature descriptors. The approach analyzes feature vectors and their corresponding identities to learn an inferred function for every single hashing model, which are used to generate vote list histograms whenever a query is requested. Figures 41, 42, 44 and 45 present the pipeline of the proposed approach.

To determine whether a face image is enrolled in the gallery of individuals, visual features are obtained from a query face image. Next, the extracted features are presented to each hashing function in order to avoid comparing the probe feature vector to all gallery subjects. A vote list histogram is hence set up with size in accordance with the number of individuals enrolled in the gallery set during training time. If the algorithm establishes that a probe image corresponds to an enrolled identity, the vote list histogram turns into a list of candidates since the method only considers the individuals that closest match the query face image.

The list of candidates is a subset of the vote list histogram. In the face identification stage, one could consider the *one-against-all* classification scheme described by [Schwartz et al., 2012]. This strategy attributes +1 to all samples interrelated to the chosen subject and  $-1$  to the samples from all remaining individuals. In other words, all features corresponding to the selected individual are used as examples, while all unlike visual features are adopted as counterexamples. Samples belonging to subject will receive positive responses. This process is reproduced for every single subject in the list of candidates.

### 4.1.1 Training Stage

We start the training stage by randomly partitioning all subjects registered in the gallery set into two disjoint collections: positive and negative sets. In pursuance of a balanced division, samples are drawn from a Bernoulli distribution (see Chapter 3.4 for footnote reference) in the interest of associating a gallery subject with the positive class when the distribution value gets closer to one or with the negative class otherwise.

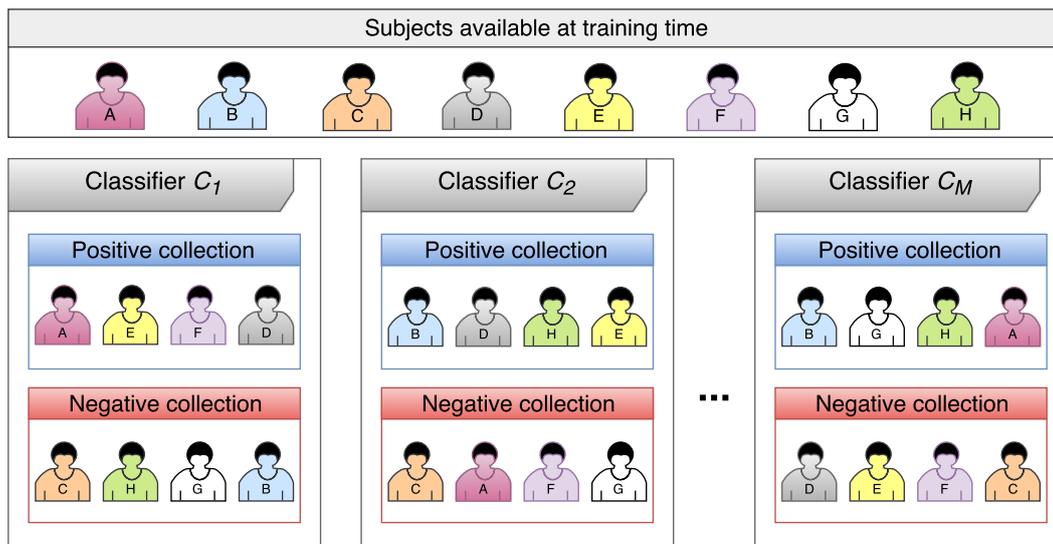
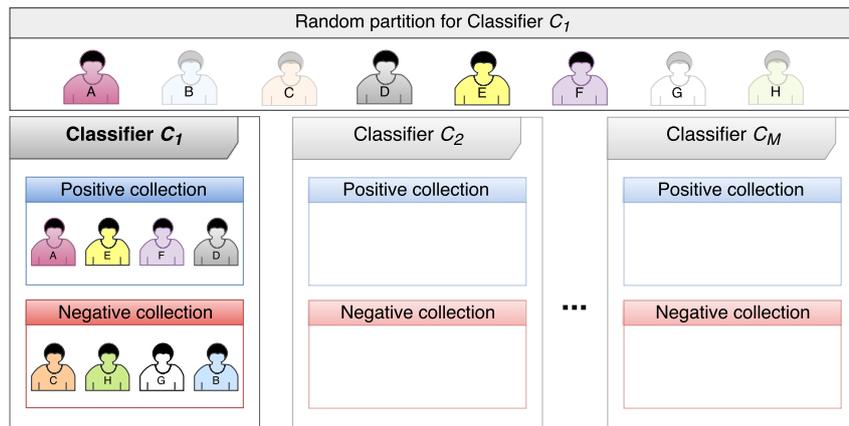


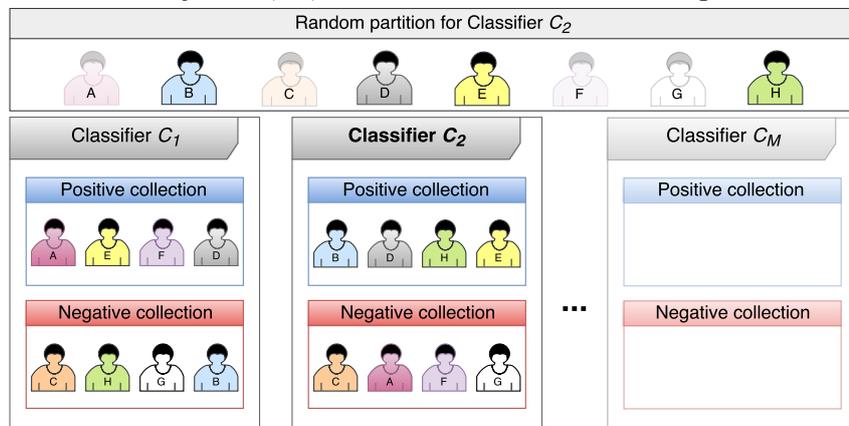
Figure 41: *Training (part I)*: Feature descriptors are acquired for all subjects' samples before partitioning them into positive and negative sets. Then, different classification models are generated containing different subjects in each collection. As can be noticed in this example, each classifier shares the very same eight individuals; however, their distributions among positive and negative sets are totally unequal.

Just as we split all subjects into positive and negative sets, we guarantee that each collection contains approximately the same number of individuals. Furthermore, we make sure that all samples belonging to a certain individual reside in the same class. At that moment we execute a learning algorithm so that a single model is created for both positive and negative collections in a binary classification fashion.

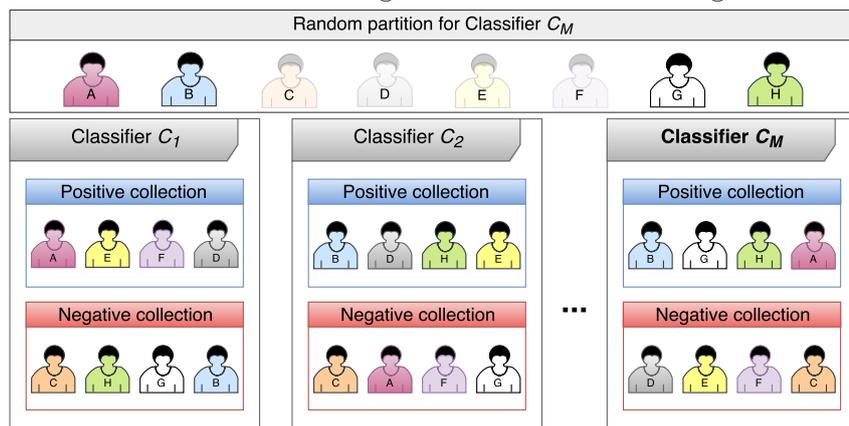
The generation of binary classifiers for hashing is repeated several times and, therefore, each classification model contains different individuals belonging to the positive and negative collections. These classifiers are essential because they determine whether a query face sample is a member of the positive or negative class. Feature descriptors are obtained from samples in the positive set with target values equal to +1, in contrast to samples in the negative set, which hold target values equal to -1. These features must be extracted and combined with their corresponding target values so that classification models can be successfully learnt.



(a) A random partition splits the dataset so that individuals A, D, E and F are added to the positive collection and subjects B, C, G and H are added to the negative set in classifier  $c_1$ .



(b) A new random partition splits the dataset so that individuals B, D, E and H are added to the positive collection and the remaining ones are added to the negative set in classifier  $c_2$ .



(c) The final random partition splits the dataset so that individuals A, B, G and H are added to the positive set and the left individuals are inserted into the negative set in classifier  $c_m$ .

Figure 42: *Training (part II)*: Different classification models are generated containing different subjects in each collection. In each partition, opaque individuals are added to the positive set whereas translucent subjects are added to the negative set.

### 4.1.2 Testing Stage

At the time we move to the testing stage, we engage the same feature descriptors employed during the training step to extract features from query face images. We present this feature vector to each one of the classifiers  $c_i \in C$  in exchange for the response value  $r_i$ . Given a probe face image, we start a vote list replete with zero values. Each position in the vote list histogram corresponds to a subject from the gallery of known subjects that is required for training. Each classification model has a record of which individuals were randomly categorized as pertaining to the positive and negative set.

As we compare a probe sample to all classification models, we increment each model's response value  $r_i$  on the vote list only for those subjects belonging to the positive set. In the end, we sort the vote list in decreasing order in behalf of keeping individuals with higher probability of matching the probe sample on the top of the vote list ranking. In essence, we want to find out how much the top scorer, namely the leading subject of the sorted vote list, stands out from all other individuals.

When a classifier  $c_i$ 's response score  $r_i$  is closer to  $+1$ , it indicates that the query image sample is very much alike the subjects in the positive collection. The algorithm votes for individuals from  $c_i$ 's positive class as it only increases their bins in the vote list. Additionally, if classifier  $c_i$ 's score is closer to  $-1$ , then the query image sample probably resembles subjects in the negative set, which results in a subtraction of the vote list bins that correspond to  $c_i$ 's positive subjects.

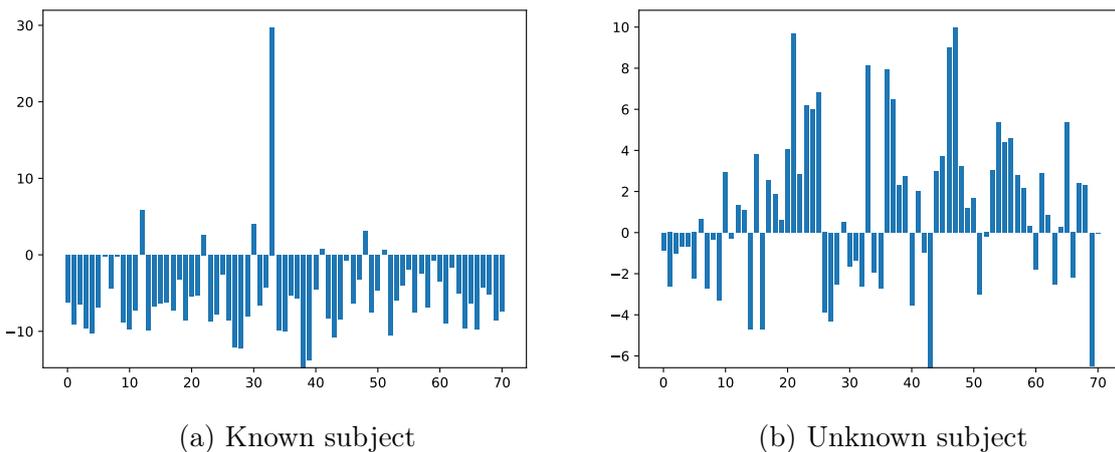
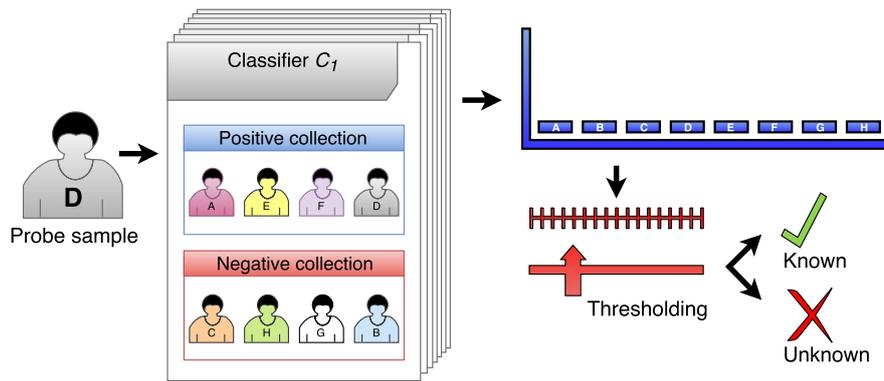
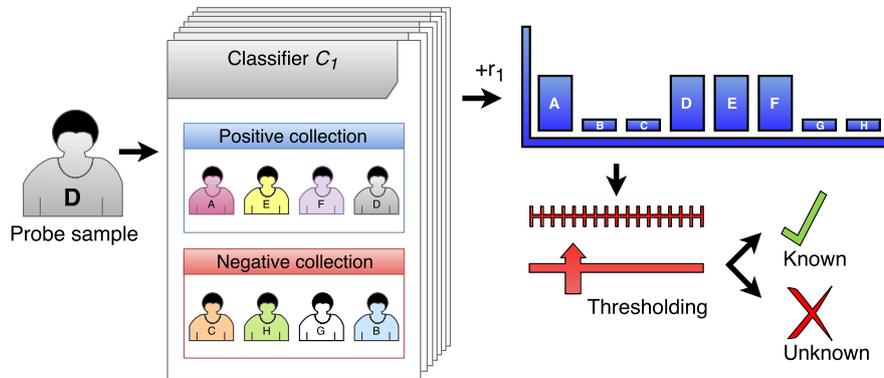


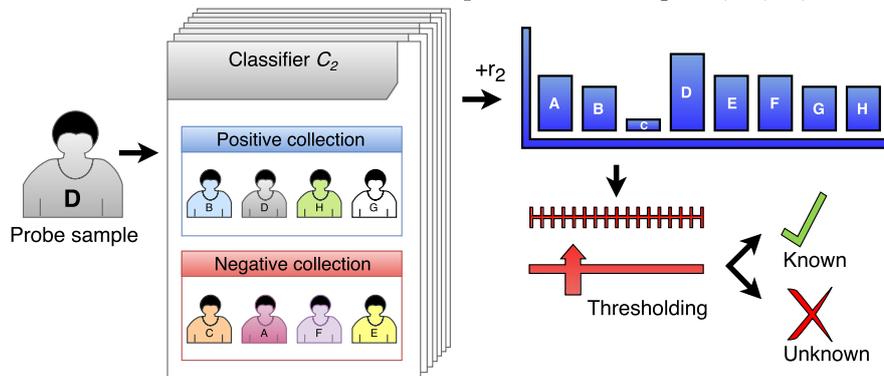
Figure 43: Two queries for the same individual of the FRGCv1 dataset when the subject is either registered in the gallery set (a) or an unknown person (b). A considerable number of subjects from the gallery set turns into candidates when there is no clue what the identity for the query image is.



(a) A vote list histogram is initialized containing all individuals enrolled in the gallery set during training time. The probe sample is projected to every classifier in search for a response value  $r$ .

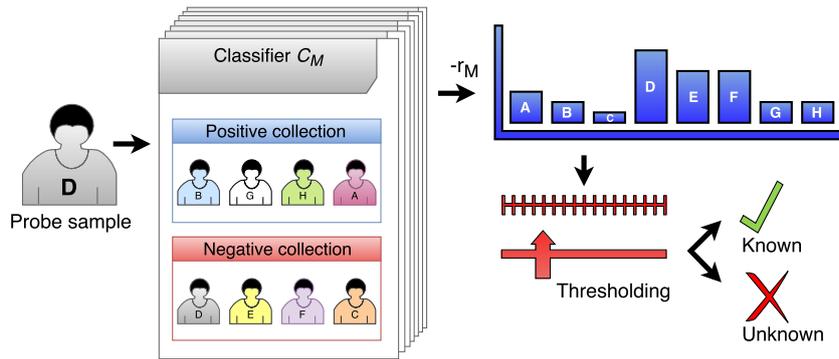


(b) The probe sample's feature vector is presented to classifier  $c_1 \in C$  in exchange for a positive response value  $r_1$ . Bins in the vote list that correspond to the subjects from  $c_1$ 's positive collection are incremented with the response score. E.g.: A, D, E, and F.

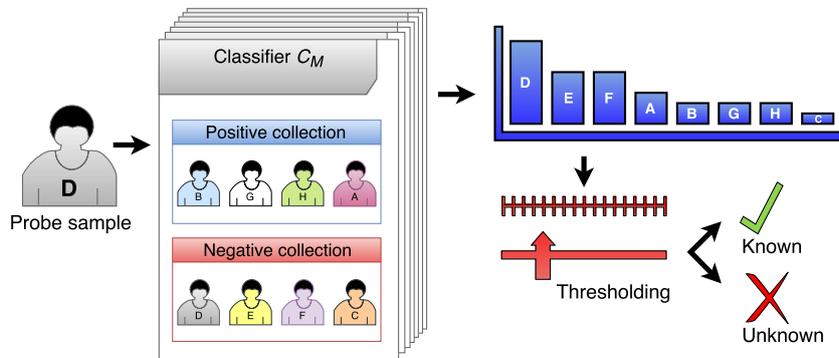


(c) The probe sample's feature vector is now presented to classifier  $c_2 \in C$  in exchange for a positive response value  $r_2$ . Bins in the vote list that correspond to the subjects from  $c_2$ 's positive collection are incremented with the response score. E.g.: B, D, H and G.

Figure 44: *Testing (part I)*: Same features descriptors employed in the training stage are used to extract features from the query image. The probe feature is compared to all classifiers and their response values are used to increment a vote list. Illustration continues on the following page.



(a) The probe sample's feature vector is presented to classifier  $c_m \in C$  in exchange for a negative response value  $r_m$ . Bins in the vote list that correspond to the subjects from last classifier's positive collection  $c_m$  are decremented with the response score in cases when it is not greater than zero. E.g.: B, G, H and A.



(b) The vote list is sorted in decreasing order in the interest of computing the ratio of the top scorer to the remaining individuals. If the ratio is higher than a specified threshold, we declare the probe sample as a known subject. If not, the pipeline halts since there is not any reason to continue searching for a subject with low probability of being previously enrolled in the gallery set.

Figure 45: *Testing (part II)*: The probe feature is compared to all classifiers (hashing functions) and their response values are used to increment a vote list. If the ratio of the top scorer to a fraction of the remaining subjects satisfies a threshold, it is considered a known individual. Otherwise, the probe sample is classified as unknown due to the absence of a highlighted bin in the vote list histogram.

Consider score  $r_i$  being +1 for any classifier every time the corresponding identity to a probe sample lies in the positive collection: even when other positive-class individuals from classifier  $c_i$  receive equivalent score, their respective bins in the vote list will eventually differ due to the random partition of all subjects at training time. Therefore, any chance of repeatedly having two or more subjects in the positive class declines as the number of  $M$  classifiers escalates, providing each gallery set subject a *unique* binary signature during training time. Figures 44 and 45 clarify how the testing stage works in the identification task.

Figure 43 illustrates two different face queries: one corresponds to querying a gallery-enrolled individual and the other searches for a subject that has no corresponding identity. Notice that there is a highlighted bin in Figure 43a when the query image has a matching identity, a prospect that it probably represents the probe sample’s identity. On the contrary, in Figure 43b several bins are incremented when there is not a single subject from the gallery that corresponds to the probe sample.

Different from the method of Santos Junior et al. [2016] that sorts the vote list in descending order of incremented scores to generate a list of candidates for face identification, we arrange the vote list in the interest of computing the ratio of the top scorer to the remaining individuals (the ratio estimation is detailed in Chapter 5.2.1), if the ratio is higher than a threshold, we declare the probe sample as a known subject, as it would be for Figure 43a. The proposed method also differs from the original implementation in some aspects: we store both positive and negative collections for every regression model. Moreover, we increment regression values on the vote list even when they are negative in an attempt to intensify the difference among the vote list scores.

### 4.1.3 Artificial Neural Network

In pursuance of better recognition results, we propose a third approach by replacing the standard Partial Least Squares (PLS) and Support Vector Machines (SVM) with Artificial Neural Network (ANN) classifiers. ANNs are extensively applied to research due to their capacity of dealing with problems stochastically, which often allows approximate solutions for optimization problems like fitness approximation. Elementary ANNs are feed-forward artificial neural networks comprising at least two layers of nodes and capable of distinguishing non-linearly separable data. In fully-connected artificial neural networks, each node  $i$  in one layer connects with a certain weight  $w_{ij}$  to every node  $j$  in the following layer [Yegnanarayana, 2009].

Most ANN models target at modeling the relationship of observable variables and determining whether a probe sample’s identity is enrolled in the gallery set at training time. The idea of applying artificial neural network models in place of PLS or SVM-based hashing functions was encouraged by the works of Lin et al. [2012] and Mouazen et al. [2010]. Despite the fact the authors evaluate the performance of machine learning algorithms on the study of respiratory ventilation and soil properties, they inferred that artificial neural networks could equal or exceed the performance of regression-based approaches.

As depicted in Figure 46, we propose a small network architecture with three

layers: input, hidden and output layer. Each node is a neuron with a nonlinear activation function that is connected to every neuron in the previous layer. The hidden layer is set up with rectification non-linearity (ReLU) and the last layer is equipped with a soft-max function. Therefore, each node in one layer connects with a certain weight  $w$  to every node in the subsequent layer. This was the chosen architecture because it reported the best results in an exploratory experiment with several other architectures, considering different numbers of neurons and depths.

The employment of an artificial neural network is exclusive to the identification task. Our goal is to generate a new collection of classifiers as an alternate approach to PLS and SVM embeddings. Similarly, for a probe sample, the network outputs scores closer to +1 indicating there is a high probability of the probe sample being in the positive collection. Otherwise, it returns response scores closer to  $-1$ .

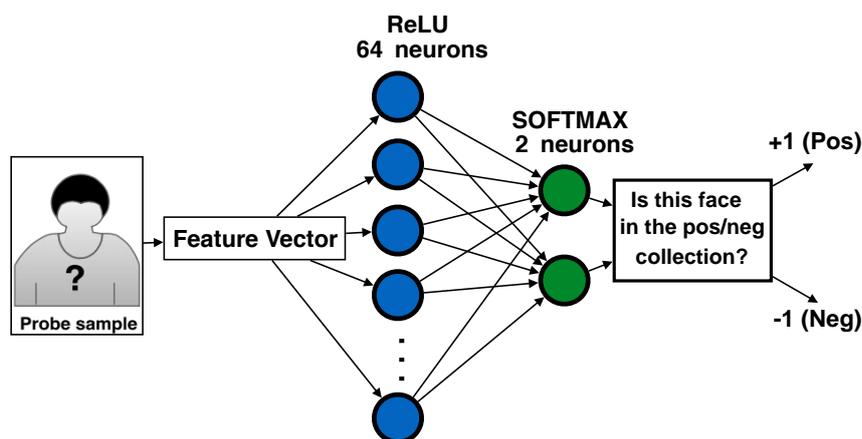


Figure 46: The three-layer artificial neural network designed to be in place of each PLS or SVM model. The network is fed up with feature vectors to learn weights that will determine whether the probe is closer to the positive or negative collection.

We present a *shallow* artificial neural network to enable very rapid binary training and low implementation complexity in order to guarantee little execution time contrast to the linear regression models previously described. Due to its training speed and few tunable parameters, the method has great applicability for approaches that contain an embedding of classifiers or require frequent retraining and online training. The network architecture for classification is a three-layer neural network holding an input layer, a hidden layer of nonlinear units, and a linear output layer. It works very much alike the PLS approach: a matrix of weight vectors is calculated considering the results of each ANN model and the regression response for a query image's feature vector is then returned.

## 4.2 Face Verification

In this section, we describe the proposed approach for performing face verification. The method compares pairs of face images as it extracts relational features with VGGFace CNN descriptor [Parkhi et al., 2015], assuming the hypothesis that the relation between two faces are valuable for increasing the performance of the verification task. Figure 47 illustrates the designed face verification process, described in details in the next sections.

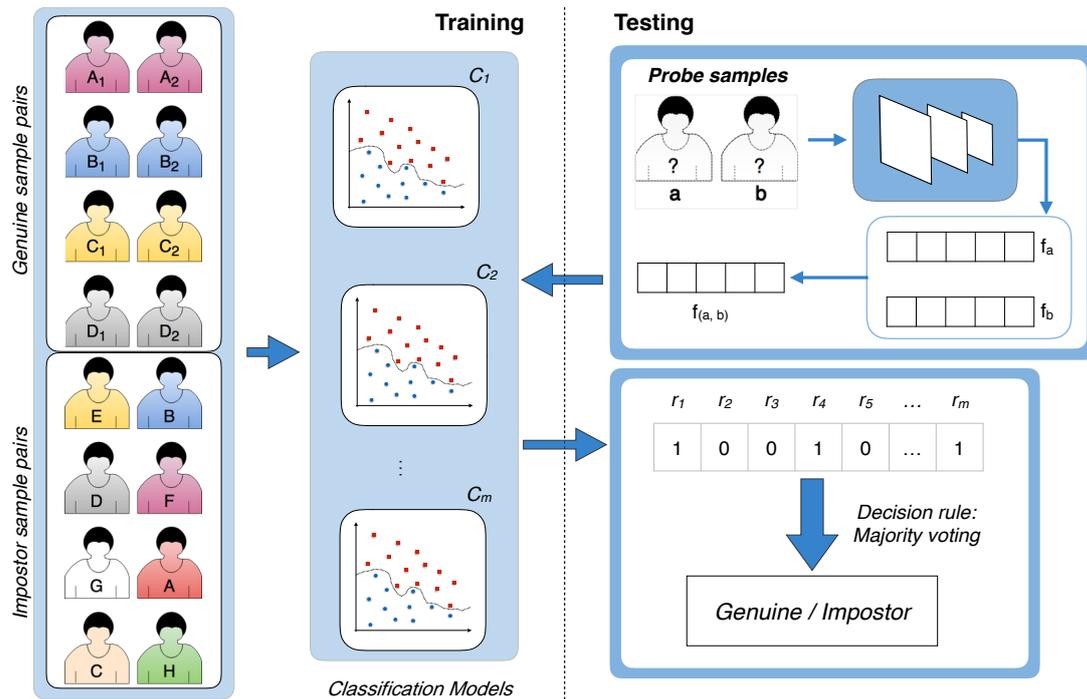


Figure 47: Overview of the proposed face verification approach. *Training:* Disparity feature vectors are obtained for all pair of subjects before they are partitioned into genuine (same) and impostor (not-same) sets. Then, different classification models are learned containing different feature samples in each collection. *Testing:* The disparity features are extracted from a pair of testing images to compose a feature vector which is then classified by all models  $c_i \in C$  and their response values  $r_i$  are used to estimate the label (genuine or impostor), based on a majority voting scheme.

Following sections specify thorough aspects of the three main stages for face verification: a distinctive extraction of features and the conventional training and testing procedures. We are inclined to believe that pairs of features of same individuals hold little differences. On the other hand, this difference increases while comparing images from different persons. Multiple classification models based on SVM or PLS are employed to determine if the given pair of images belongs to the same subject (*genuine*) or to different subjects (*impostor*).

### 4.2.1 Feature Extraction

Different from considering a feature vector for each image independently, we extract relational features for pair of faces as follows: First, the verification approach extracts deep features for all images employing the VGGFace convolutional neural network descriptor [Parkhi et al., 2015]. Then, it computes the absolute difference between them and stores this new feature vector in order to build and execute the classifier. Figure 48 illustrates the process of extracting disparity features among two face images.

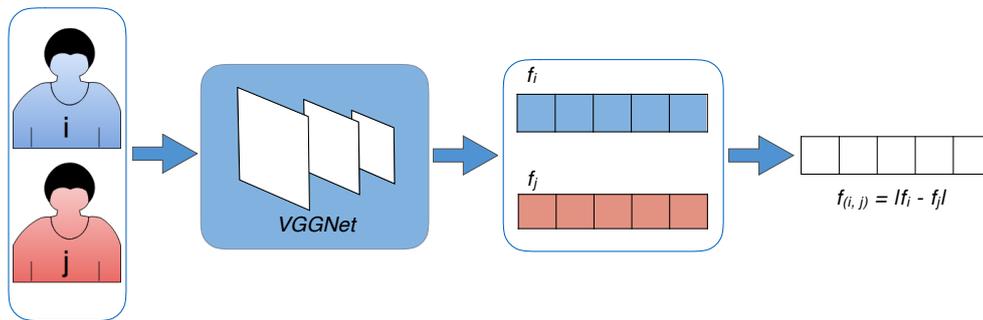


Figure 48: Feature extraction overview for a pair of face images. Two face images  $i$  and  $j$  input a trained convolutional neural network that extract their correspondent features vectors. Next, the algorithm computes the absolute difference between the two descriptors. This operation outputs a new feature vector, denominated disparity feature vector, which determines how similar the two image samples are. Note that when images  $i$  and  $j$  have identical identities, their disparity feature vector is added to the genuine set, otherwise it is stored in the impostor set.

For the face verification task, features extracted directly from face images are never presented to any classifier as they are only employed to compute relational feature vectors. Whenever two random pictures belong to the very same subject, the resulting disparity feature vector is exclusive to positive collections or to the negative ones, otherwise.

After the feature extraction process, we perform the training and testing classification steps, as depicted on Figure 47. From now on, we refer to feature vectors derived from the absolute difference as *disparity feature vectors* or simply *disparity features*. The main hypothesis lies behind the assumption that two face images of the same subject hold small differences. However, this difference is likely to rise when we cope with a pair of images from different subjects. Feature vectors that represent a pair of faces from the same person are labeled as *same person* (genuine) and feature vectors extracted from a pair of faces of different people are labeled as *not-same persons* (impostor).

### 4.2.2 Training Stage

This approach learns classification models capable of differentiating same-subject probe images from contrasting ones. The training stage randomly samples disparity feature vectors that were previously stored in two disjoint sets: *same* and *not-same*. While the former relates to pairs of samples from the same subject, the latter refers to pairs of samples from different subjects. In pursuance of a balanced division, these disparity feature samples are drawn from a uniform distribution.

The positive class contains only features selected from the *same* collection and the negative class only contains samples selected from the *not-same* collection. Then, a binary classification model is learned considering the selected samples and classifies a pair of probe samples as being same and not-same. The generation of binary classifiers is repeated  $m$  times<sup>1</sup> by selecting different disparity feature vectors from the *same* and *not-same* classes to capture different aspects of the data and allow the complementarity among the classifiers. Disparity feature descriptors are obtained from samples in the positive set with target values equal to  $+1$ , in contrast to samples in the negative set, which hold target values equal to  $-1$ . Relational disparity features must be computed and joined with their corresponding same/not-same target values. Thus, the classification models can be successfully learnt as shown in Figure 49.

The proposed face verification approach falls into a technique called Bootstrap Aggregating [Breiman, 1996], well-known for the acronym *bagging*. Bootstrap aggregating methods use multiple homogeneous learning algorithms by combining classification models trained on randomly generated training sets to achieve better predictive performance than the composing classifier alone. Breiman et al. [1994] demonstrate that neural nets, classification and regression trees are unstable, resulting in favorable results when *bagging* is employed.

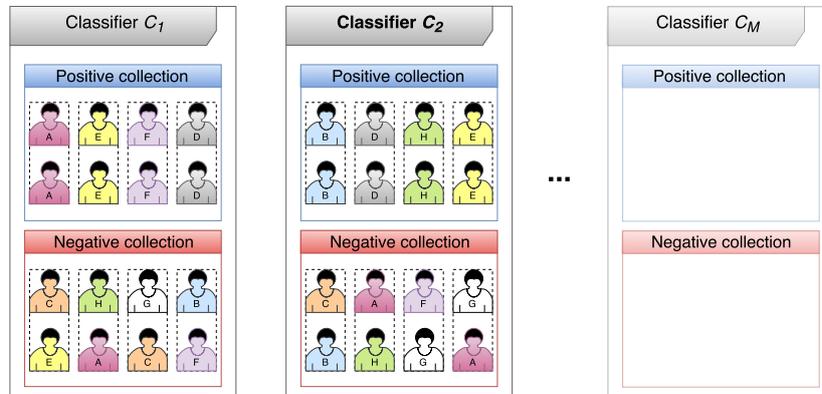
The considered training set  $S$  is composed of disparity feature vectors, which derives from the dataset's original image features (shown on Figure 48). The face verification approach holds a sequence of classification models  $\{c_i \in C : |C| = M\}$ , a series of different training subsets  $s_i \in S$  consisting of  $F$  independent disparity feature vectors each. Every classifier inputs  $f : f < F$  disparity features to compose each of the two classes, an operation that is repeated  $M$  times. The pairs of probe feature vectors that compose disparity features are drawn at random so that they may come out repeated times in any particular subset  $s_i$ . *Bagging* is able to improve overall accuracy due the stable construction of the classification set  $C$  where a minor variation between subsets  $s_i$  and  $s_j$  results in big changes for classifiers  $s_i$  and  $s_j$ , respectively.

---

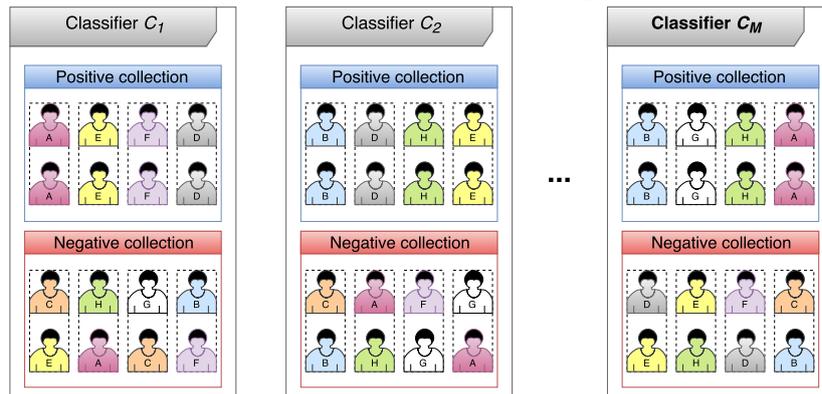
<sup>1</sup>The number  $m$  of hashing models is a parameter defined by the user.



(a) First random partition splits same-subject disparity features so that samples from individuals A, D, E and F are added to the positive set in classifier  $c_1$ . Relational features derived from distinct subjects are added to the negative collection.



(b) Second random partition splits same-subject disparity features so that samples from individuals B, D, E and H are added to the positive collection in classifier  $c_2$ . Relational features derived from distinct subjects are added to the negative set.



(c) Final random partition splits same-subject disparity features so that samples from individuals A, B, G and H are added to the positive collection in classifier  $c_m$ .

Figure 49: *Training*: In each partition, same-individual relational features are added to the positive set whereas unequal-individual disparity features are added to the negative collection.

### 4.2.3 Testing Stage

In the testing stage, the method computes the disparity feature vector among the target and query samples and presents the new feature to each of the  $M$  binary classifiers, which results the response value  $r_i$ , where  $i = 1, 2, \dots, M$ . In the end, it computes the majority voting to find which label must be designated to the probe, i.e., genuine or impostor. The former label is attributed when the pair of samples is classified as belonging to the same subject.

When a classifier  $c_i$ 's response score  $r_i$  is closer to  $+1$ , it indicates that the probe disparity feature vector is very similar to features from the positive collection, which are generated from a pair of same-identity extracted features. On the other hand, if classifier  $c_i$ 's score is closer to  $-1$ , then the query disparity feature vector probably resembles disparity features in the negative collection. As we present a probe disparity feature vector to all learning models, we store each model's response value  $r_i$ , which corresponds to the output of classifier  $c_i \in C$ . Eventually, we sample<sup>2</sup> each response value  $r_i$  in order to obtain a discrete value  $r_i^*$ , resulting in a binary response vector  $\vec{r}$  of size  $M$  when appended together. The prediction is described in Figures 410 and 411.

Rather than just outputting *same* or *not-same* binary labels, the algorithm computes ultimate response  $u$  as the ratio between the number of positive matches ( $r_i^* > 0$ ) to the total number of binary classification models in the following form:

$$u = \frac{\sum_{i=0}^M r_i^*}{M} : r_i^* > 0, r_i^* \in \vec{r}$$

Due to sampled elements in vector  $\vec{r}$ , each  $r_i^* \in \vec{r}$  holds either  $+1$  or  $-1$  values. Then,  $u$  containing values greater than  $0.5$  indicates that most classifiers consider the probe disparity feature to be *same* subject, and *not-same* otherwise. Therefore, we obtain a probability estimate of the positive class (target score), which is used to compute the Receiver Operating Characteristic (ROC) curves.

Similar to the open-set face identification approach, the probability of frequently having two or more classifiers composed of similar relational disparity features among their positive and negative collections decreases when the number of disparity feature rises in each classifiers' positive and negative set. However, no vote list histogram is generated since there is not any need of establishing a ranking of candidates. As a consequence, the proposed method does not require keeping records of feature vectors' labels due to the fact they are not employed in the testing stage.

---

<sup>2</sup>Sampling is the process of measuring the instantaneous values of continuous-time signal in a discrete form.

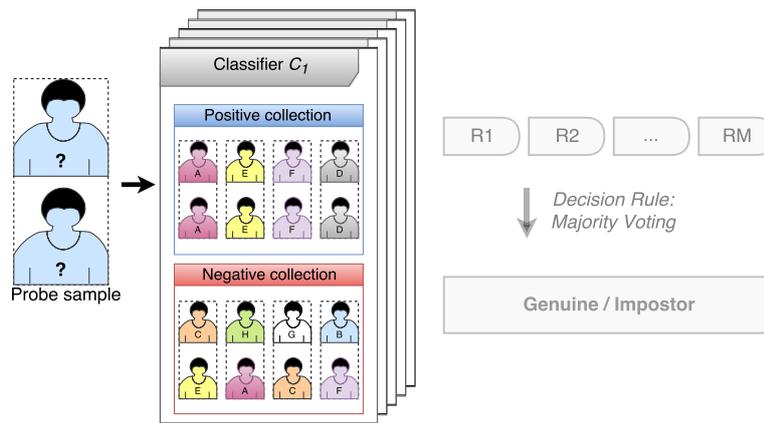
## 4.3 Technical Comparison

The proposed approaches for *open-set face identification* and *face verification* share several common aspects. Nevertheless, they differ in many others as well. The methods are developed for different purposes, once that open-set identification aims at finding the identity that best matches a probe face image if it indeed exists, while the verification simply informs whether a pair of face images correspond to the same individual or not.

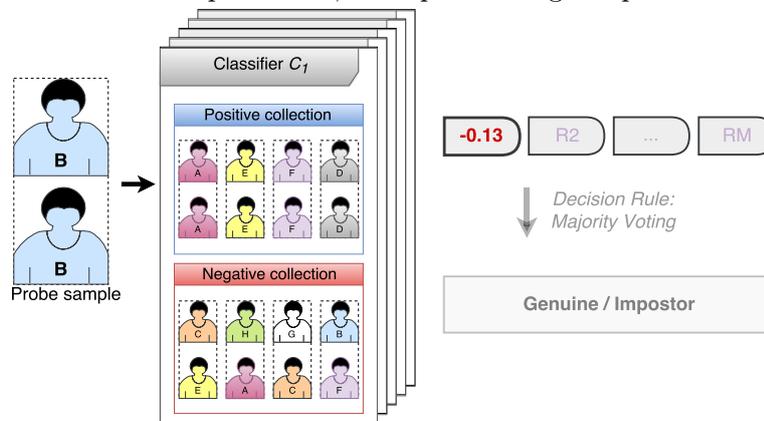
The first difference is the feature extraction process itself: open-set face identification method utilizes low-level features in addition to VGGFace CNN descriptor, the only descriptor employed on the face verification task. The latter focuses on exploiting similarities among pair of features vectors. Particularly, the face verification algorithm addresses a new feature descriptor derived from the absolute difference of the original feature vectors.

The most relevant distinction is on how feature descriptors are split for the training stage, preceding the process of generating hashing functions. In the open-set face identification algorithm, all samples of an individual are randomly marked as either *positive* or *negative* in each hashing model. Therefore, if a subject  $s$  has all its samples in the *positive* set of hashing model  $h$ ,  $s$  cannot be assigned to the *negative* set of same hashing model  $h$ . On the other hand, the generation of hashing functions in the face verification algorithm concerns whether a random pair of images have identical labels. Consequently, if disparity feature vectors are originated from genuine sample pairs, they can only be allotted to multiple *positive* sets (regardless of the hashing functions) or *negative* collections otherwise.

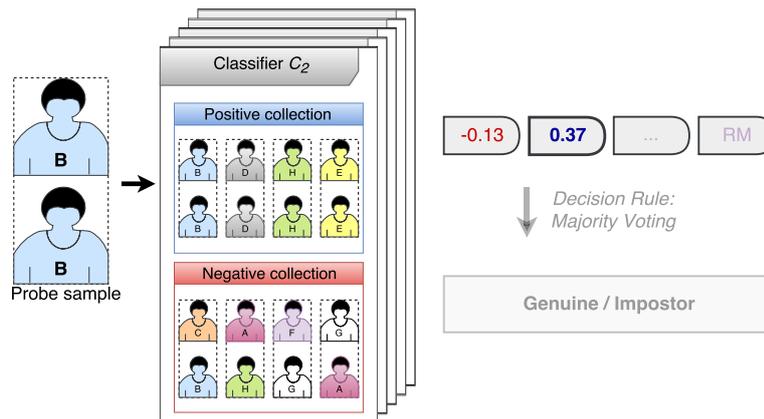
Finally, open-set face identification method designates probe samples as known if they satisfy a certain threshold (described with details in Chapter 5.2.1), aiming at selecting the best individuals to compose a list of candidates which will be delivered to a closed-set identification method for each known probe sample. On the other hand, the face verification method loops through all hashing models to find out whether the number of positive responses surpasses the quantity of negative responses. Both methods consist of embedding of classifiers. Also, they are an adaptation of Bootstrap Aggregating [Breiman, 1996], which usually works well when small changes in a training subset can cause large changes in a classifier. The proposed approach does not learn an embedding containing heterogeneous machine learning models. More precisely, when a list of classifiers  $C$  is generated, all its classifiers  $c_i \in C$  share a homogeneous nature in terms of learning methods since only one out of SVM, PLS and FCN is adopted (uniform).



(a) A disparity feature vector is computed for a pair of probe face images before projecting onto all classifiers. In this exemplification, both probe images represent identity  $B$ .

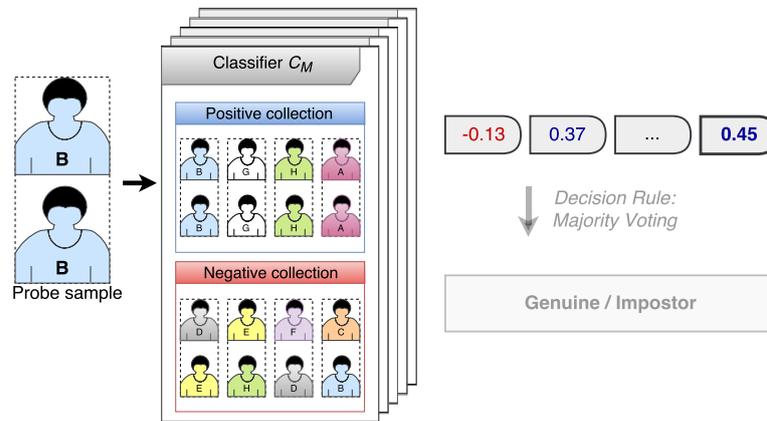


(b) The disparity feature vector is presented to classifier  $c_1 \in C$  in exchange for an arbitrary negative response value  $r_1$  of  $-0.13$ . Negative responses indicate that probe samples are likely to diverge.

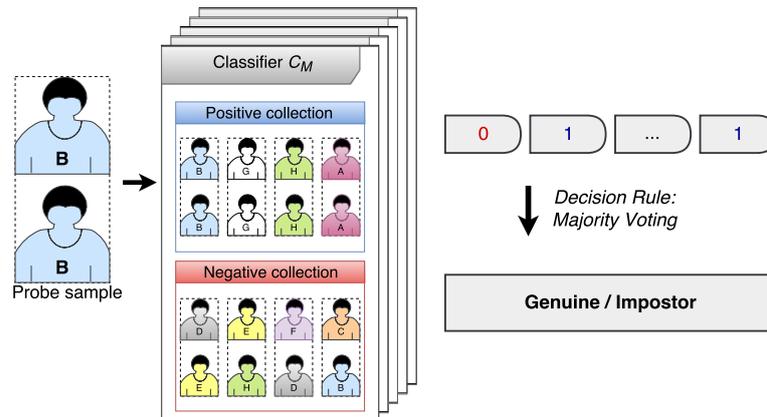


(c) The disparity feature vector is presented to classifier  $c_2 \in C$  in exchange for an arbitrary positive response value  $r_2$  of  $+0.37$ . Positive responses indicate that the probe samples are likely to be identical.

Figure 410: *Testing (part I)*: The proposed feature extraction is employed on training and testing stages. The probe disparity feature is compared to all classifiers and their response values are stored for a posterior decision rule.



(a) The disparity feature vector is now presented to classifier  $c_m \in C$  in exchange for an arbitrary positive response value  $r_m$  of  $+0.45$ . Positive responses indicate that the probe samples are likely to be identical.



(b) Each response value  $r_i$  is sampled in order to obtain a corresponding discrete vector  $\vec{r}$ . The method then computes the ratio of the amount positive scores to the number of elements in the response vector, that is,  $|\vec{r}|$ .

Figure 411: *Testing (part II)*: The probe disparity feature is compared to all classifiers (hashing functions) and their response values are stored for a voting decision rule. In case of a positive ratio, both probe images are considered to share equivalent identities. Otherwise, the probe samples are classified as not sharing the same identity since most classification models vote for diverging identities.



# Chapter 5

## Experiments

This chapter presents the experimental evaluation of the approaches described in Chapter 4. For the sake of keeping the explanation clear, the methods' assessment is depicted independently. Section 5.1 details the datasets and the feature descriptors used for face identification and verification. Section 5.2 details the evaluation protocol and the recognition evaluation for the open-set face identification task. Finally, Section 5.3 describes the experiment protocol performed for face verification, as well as the obtained results.

### 5.1 Common Attributes

For a clear reading and avoid going over repetitive sections, we group related experimental information and present them in this section. Initially, there is the summary of five datasets and two feature descriptors engaged in both developed methods. We selected three of them as benchmarks for the open-set identification method and the two remaining are reference datasets for the verification algorithm. The datasets were released at different occasions and, consequently, they hold distinctive attributes and so do the feature descriptors. Subsequently, we simultaneously expose the experimental setup for open-set face identification and face verification seeing that they share comparable parameters.

#### 5.1.1 Datasets

In favor of demonstrating the effectiveness of our methods, we select datasets with different characteristics, ranging from frontal cropped images taken under controlled scenarios to images in the wild with lighting and pose variations.

The algorithm proposed for open-set face identification is evaluated on a recent non-constrained dataset and on two well-known datasets. We evaluate our open-set identification method on the datasets FRGCv1, PubFig83 and VGGFace. For the sake of demonstrating the capability of the proposed face verification approach, we chose two challenging datasets with different aspects, such as age discrepancies and expression diversities. We evaluate our verification method on the datasets LFW and on PubFig.

**Face Recognition Grand Challenge v1.0 (FRGCv1):** FRGCv1 [Phillips et al., 2005] consists of more than five thousand images distributed to 152 subjects, six different experiments and two facial expressions: smiling and neutral. The controlled frontal face images were captured in a studio setting under two lighting conditions whereas uncontrolled images were taken in changing illumination conditions on either hallways or outside. We only evaluate the methods on three of them, experiment one, two and four, since experiments three, five and six do not correspond to 2D face recognition. Experiments *one* and *two* only contain controlled images whereas experiment *four* considers a gallery with one controlled still picture for each subject plus a probe set having multiple uncontrolled images.



Figure 51: Frontal face images extracted from Face Recognition Grand Challenge v1.0 dataset.

**Labeled Faces in the Wild (LFW):** LFW dataset [Huang and Learned-Miller, 2014; Huang et al., 2007] can be considered the genuine state-of-the-art benchmark for face verification. It also comprises face images aligned with an unsupervised deep feature algorithm, commonly known as LFW-A or deep-funneled LFW [Huang et al., 2012a]. This dataset contains approximately 13,000 uncontrolled face images of more than five thousand individuals. In contrast to the majority of existing face datasets, these images were taken in entirely unconstrained situations with non-cooperative individuals. Thus, there is also large divergence in pose, lighting, expression, scene, and camera. For fair comparison, the creators of LFW suggest reporting performance as a 10-fold cross validation using splits they have randomly generated. As other works on face verification, we used deep-funneled LFW face images (LFW-A).

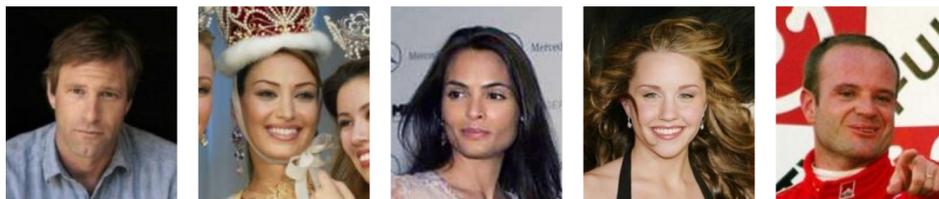


Figure 52: Uncontrolled face images extracted from Labeled Faces in the Wild dataset.

**Public Figures (PubFig):** PubFig [Kumar et al., 2009] is larger than the LFW in terms of image samples, consisting of nearly 60,000 images of 200 subjects gathered from across the Internet. PubFig was released long ago and they do not distribute image files due to copyright issues. Thus, only 26,787 out of 58,797 initially images remain available as links to these files are gradually disappearing over time. The database is considered as very difficult as it evidences vast diversity in pose, lighting, facial expression, age, gender, and ethnicity. The PubFig dataset is divided into two units, the evaluation set with 140 subjects, designed to evaluate methods, and the development set with 60 individuals, which holds no overlap with the evaluation set.



Figure 53: Uncontrolled face images extracted from Public Figures dataset.

**Public Figures 83 (PubFig83):** PubFig83 [Pinto et al., 2011] is a fragment of the original PubFig dataset. In fact, it incorporates the subjects from PubFig holding more than 100 samples. Therefore, PubFig83 is composed of at least 100 samples for each one of the 83 individuals, totaling a minimum of 8,300 image samples. It comprises several uncontrolled images with pose and expression variations. The images were captured in non-restrained situations with non-cooperative individuals.



Figure 54: Uncontrolled face images extracted from Public Figures 83 dataset.

**VGGFace Dataset (VGGFace):** VGGFace [Parkhi et al., 2015] contains about 2.6 million samples of more than 2,600 celebrities and public figures collected from the web. Its initial list of public figures was taken out of the Internet Movie Data Base (IMDB) celebrity list. Due to its massive size and high training time, we arbitrary select a portion of the original VGGFace containing a thousand subjects with 15 samples each. The individuals are chosen according to alphabetical ordering of all subjects followed by the selection of the first 1000 individuals. Samples are also sorted in ascending order and the first fifteen available images for each subject are selected.

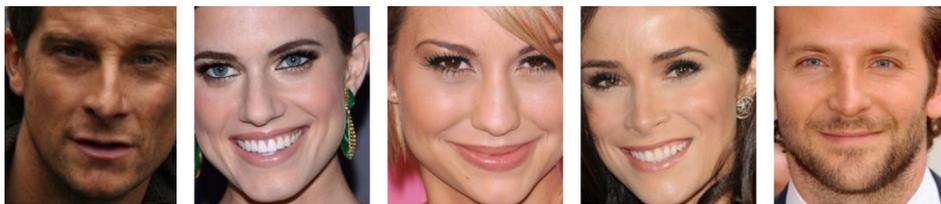


Figure 55: Uncontrolled face images extracted from VGGFace dataset.

### 5.1.2 Feature Descriptors

Two feature descriptors are utilized with the open-set identification method: HOG and VGGFace. In the verification approach we employ the VGGFace descriptor only. The former was designed for object detection whereas the latter is based on convolutional neural networks for face detection and recognition.

**Histogram of Oriented Gradients (HOG):** HOG [Dalal and Triggs, 2005] generates descriptors that comprise shape information in the form of histograms. Before feature extraction, images are re-scaled to  $128 \times 144$  pixels and each sample is decomposed into a set of overlapping blocks which features are extracted from. Each block is  $16 \times 16$  pixels, with an 8-pixel stride and an  $8 \times 8$ -pixel cell size. After extracting features for all blocks, descriptors are concatenated in a feature vector and that turns into a feature descriptor.

**VGGFace CNN descriptor (VGGFace):** VGGFace is computed using the implementation of Parkhi et al. [2015], which is derived from the VGG-Very-Deep-16 CNN architecture [Simonyan and Zisserman, 2015], an artificial neural network that comprises a long sequence of convolutional layers. VGGFace is a stack of  $3 \times 3$ -convolutional filters followed by three fully-connected and a soft-max layer. All hidden layers are equipped with the rectification non-linearity (ReLU). We do not employ

any sort of fine tuning towards the datasets mentioned in Section 5.1.1. Instead, we consider the network already learned using the standard training weights generated by Parkhi et al. [2015].

### 5.1.3 Experimental Setup

We explore the Scikit-Learn library for Python, a simple and efficient open-source tool for data analysis and mining that provides fundamental versions of PLS and SVM. TensorFlow is the adopted neural network library for our open-set ANN-based algorithm version, high-leveled with Keras API for a fast experimentation.

All identification and verification experiments are performed on a Intel Xeon E5-2630 CPU with 2.30 GHz and 16GB of RAM using Ubuntu 14.04 LTS operating system, no more than 12 GB of RAM was required though. Both identification and verification methods have mainly three parameters: the number of binary classification models ( $hm$ ), the number of subjects in each model’s positive or negative class ( $hs$ ), the number of PLS dimensions in the latent space ( $d$ ) or the SVM trade-off ( $c$ ) between training data errors and margin maximization.

Some experiments portray parameter observations varying  $hm$  from 10 to 500 in arbitrary steps. No more hashing models are created because the analysis focuses on the combination of accuracy with fast training/testing rather and including thousands of hashing models since it significantly increases computational time. In the face identification approach,  $hs$  is proportional to the percentage of known individuals in the known set, which is demonstrated on Table 55. For face verification,  $hs$  indicates the number of relational disparity features vectors in each model’s positive or negative class, which was implicitly provided by LFW’s protocols. SVM parameter  $c$  is set to 1 as it returned best results during tuning stage. Moreover, we also ranged PLS parameter  $d$  from 4 to 30 in a 2-step increase to conclude that it had little impact on our algorithm’s performance; therefore, we set  $d$  to 10.

## 5.2 Open-set Face Identification

In this section we assess the method proposed in Chapter 4.1, which resembles the combination of Locality-Sensitive Hashing (LSH) with Support Vector Machines (SVM), Partial Least Squares (PLS) and Artificial Neural Networks (ANN). The open-set face identification framework and experimental data are available online<sup>1</sup>.

---

<sup>1</sup><https://github.com/rafaelvareto/HPLS-HFCN-openset>

### 5.2.1 Evaluation Protocol

There is not a worldwide consensus when it comes to protocols for open-set face recognition. In this work, we propose a new protocol for both FRGCv1 and VGGFace datasets in addition to adopting an already-explored protocol for PubFig83.

**Literature Protocol:** We evaluate PubFig83 alone on a protocol exploited by some researchers [Carlos et al., 2013; Pinto et al., 2011; Santos Junior and Schwartz, 2014]. According to this convention, since PubFig83 holds at least 100 samples for each one of its 83 individuals, eight subjects are randomly selected to compose the unknown test set. Consequently, we consider the remaining 75 subjects of the original dataset as the known set (known set = 90%, unknown set = 10%). For the training stage, 90 samples are randomly selected from every subject in the known set to build the gallery set and the remaining ten samples form the test set (training set = 90% of the known set). This protocol is applied to experiments depicted in Table 51 and Table 54.

**Proposed Protocol:** We propose a new protocol for the experiments carried out with FRGCv1 and the subset of VGGFace. We partition the entire dataset, varying the known individuals set size in 10%, 50% and 90% of the complete face database. All remaining individuals become unseen classes during training time. For each subject in the known set, 50% of the samples are randomly selected for training and the remaining is left for testing. This protocol is applied to experiments exposed on Table 52 and Table 55.

**Evaluation Metric:** We consider both extensively employed Receiver Operating Characteristic (ROC) curves and its Area Under Curve (AUC) for all datasets. ROC curves usually present true positive rate on the  $Y$  axis, and false positive rate on the  $X$  axis. It indicates that the plot's top left corner is the optimal point. Good open-set recognition systems would present true positive rates for the ROC curve equal to one. Similarly, AUC ranges from zero to one, being preferable values approaching one.

For a more accurate evaluation of retrieval-based open-set biometric systems, we take into account the Detection and Identification Rate (DIR) and False Alarm Rate (FAR) as well. DIR is a probability estimate that a subject enrolled in the gallery is detected whereas FAR estimates the likelihood a non-enrolled individual is characterized as belonging to the gallery set. Plotting DIR vs. FAR produces a chart known as Open-set ROC, a metric generally used to evaluate approaches composed by filtering and identification steps.

**Threshold Selection:** An evaluation of three different thresholds is executed in the interest of finding out the one that best impacts our algorithms. Figure 56 shows the ROC curve for each threshold  $\tau$ , which are detailed below:

$$\tau_1 = \frac{\mathcal{H}_{TS_1}}{AVG(\mathcal{H}_{TS_2} + \mathcal{H}_{TS_3})} \quad (5.1)$$

$$\tau_2 = \frac{\mathcal{H}_{TS_1}}{\mathcal{H}_{TS_2}} \quad (5.2)$$

$$\tau_3 = \frac{\mathcal{H}_{TS_1}}{AVG(\mathcal{H}_{TS_2} + \dots + \mathcal{H}_{TS_p})}, p = \lceil 0.15 \times |\mathcal{H}| \rceil \quad (5.3)$$

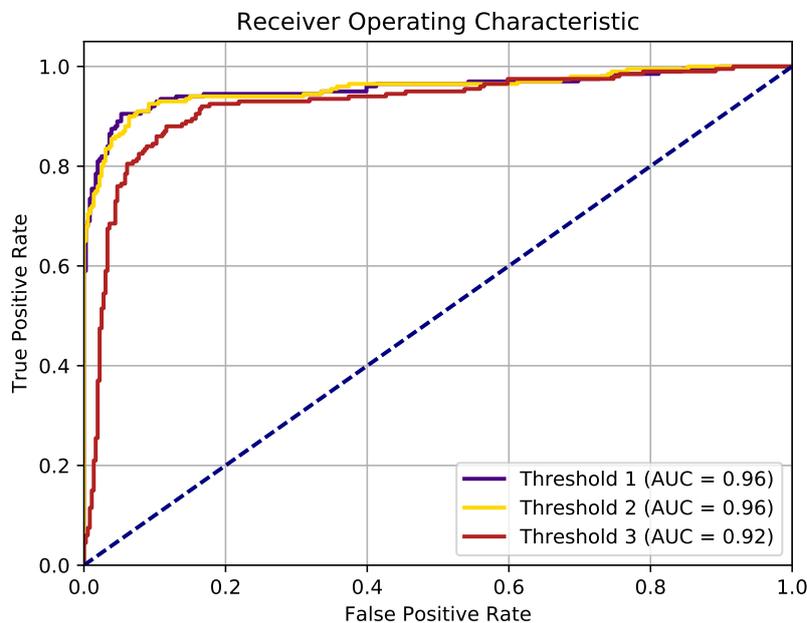


Figure 56: Average ROC curves for the FRGCv1 dataset on experiment one. We repeated this experiment five times, fixing variable  $p$  to 15% of all subjects in the gallery set.

Basically, they are based on the ratio of the vote-list histogram  $\mathcal{H}$ 's top scorer  $TS_1$  to the average of the succeeding subjects. The chart indicates that the ratio among only the leading three top scores does not drastically interfere in the area under the curve. Better results attained with  $\tau_1$  and  $\tau_2$ , demonstrating that the addition of many bins to the threshold calculation ( $\tau_3$ ) worsens the algorithm decision performance. We believe that when more individuals are considered in the threshold, it becomes less discriminative as most of the bins consist of low values, then the denominator is predisposed to hold small numbers due to the average of the 15% most relevant persons. To the remainder of the open-set identification experiments, we opt for threshold  $\tau_1$ .

## 5.2.2 Method Evaluation

We evaluate the open-set identification approach described in Chapter 4.1 for VGGFace, PubFig83 and experiment number four of FRGCv1 dataset. From now on, we refer to the combination of locality-sensitive hashing and support vector machine as HSVM. Equivalently, HPLS turns into the association of hashing methods with partial least squares, and HANN represents the embedding of artificial neural networks.

### 5.2.2.1 Descriptor Selection

In consonance with Section 5.1.2, there are two feature descriptors used in this approach: HOG and VGGFace. Table 51 presents a comparison between both descriptors on PubFig83 dataset. The considered approaches are HSVM, HPLS and HANN, which respectively stands for the proposed approaches alternating the classifier in SVM, PLS and ANN.

Table 51: Comparison between HOG and VGGFace descriptors on PubFig83 with HSVM, HPLS and HANN algorithms. It presents ROC’s area under curve (AUC), standard deviation (STD) as well as the number of executions (Execs) for experiments with one hundred binary classification models ( $hm = 100$ ).

|          | <b>Approach</b> | <b>AUC</b> | <b>STD</b> | <b>Execs</b> |
|----------|-----------------|------------|------------|--------------|
| PubFig83 | HSVM-HOG        | 0.510      | 0.011      | 10           |
|          | HSVM-VGG        | 0.940      | 0.010      | 10           |
|          | HPLS-HOG        | 0.658      | 0.014      | 10           |
|          | HPLS-VGG        | 0.940      | 0.020      | 10           |
|          | HANN-HOG        | 0.640      | 0.021      | 10           |
|          | HANN-VGG        | 0.973      | 0.004      | 10           |

As we can see in Table 51, the approaches using VGGFace CNN descriptor notably outperform HOG-based algorithms following the *Literature Protocol*. While HOG only holds shape information, VGGFace CNN descriptor comprises much more information related to faces since its network was previously trained on an unrestrained face dataset.

PubFig83 was chosen for the fact that it has no room for data influence, resulting then in a proper contrast, which would not be possible with FRGCv1 or VGGFace. HOG would be indicated for controlled frontal face datasets, like FRGCv1, due to its characteristic of encountering gradient orientations. VGGFace CNN descriptor is a network trained with VGGFace dataset. Therefore, for an unbiased HOG and VGGFace descriptor evaluation, FRGCv1 and VGGFace datasets would not be recommended.

### 5.2.2.2 Baseline: One-Class SVM

One-class SVM [Schölkopf et al., 2001] generates a spherical boundary around the data in the feature space. The idea is to add most data into the hyper-sphere so that it becomes an optimization problem. In our experiments, we execute the extended one-class WSVM algorithm proposed by Scheirer et al. [2014], which is publicly available in the form of a library called LibSVM<sup>2</sup>. The SVM learning code from LibSVM is the ground of other open source machine learning toolkits like Scikit-Learn library.

There are two key parameters we should concern when using one-class SVM:  $\gamma$ , which defines the shape of the hyper-sphere that represents the SVM kernel, and  $\nu$ , which is the upper bound on the amount of training errors and a lower bound of the quantity of support vectors. The best results were obtained with  $\gamma$ 's default value ( $\gamma = 1/n$ , where  $n$  is the length of the feature vectors. With respect to  $\nu$ , we perform a grid search in pursuance of the value that provides the best AUC for each experiment's protocol.

### 5.2.2.3 Literature Comparison

The performance assessment of the feature descriptors HOG and VGGFace is carried out independently. Santos Junior and Schwartz [2014] combine four descriptors: HOG, LBP, mean color and Gabor filters. WSVM symbolizes the one-class SVM of Scheirer et al. [2014]. We fix both the number of hashing models to 100 and the quantity of individuals in the known set to 50%, in accordance to the *Proposed Protocol* specified in Section 5.2.1.

We present the proposed face identification results in Table 52 in the form of HANN, HPLS and HSVM for artificial neural networks, partial least squares and support vector machines, respectively. Our method achieves very good results using two different feature descriptors: HOG and VGGFace. Table 52 shows experiments on the FRGCv1 dataset. There are blank cells in the first three rows because we did not reproduce the experiments implemented by Santos Junior and Schwartz [2014].

Histogram of Oriented Gradients is considered a low-level feature descriptor; however, it performed well with HPLS. We believe it can be explained by HOG's structure and FRGCv1's predominant characteristics since it encompasses high-resolution images acquired under partial controlled conditions and no pose variation. VGGFace was learnt considering more than two thousand unique individuals with all sorts of pose variations and expression changes. Therefrom, HOG outperforming VGGFace seems plausible for a dataset like FRGC.

---

<sup>2</sup><https://github.com/ljain2/libsvm-openset>

Table 52: Average area under curve (AUC), standard deviation (STD) and number of executions (Execs) for the experiment four of FRGCv1 dataset. We employ the *Proposed Protocol* with 100 hashing models, selecting 50% of the subjects to compose the known set (training and known test set).

|                     | <b>Approach</b>                                  | <b>AUC</b> | <b>STD</b> | <b>Execs</b> |
|---------------------|--|------------|------------|--------------|
| FRGCv1 Experiment 4 | Least Squares [Santos Junior and Schwartz, 2014] | 0.869      | -          | -            |
|                     | SVM-Single [Santos Junior and Schwartz, 2014]    | 0.853      | -          | -            |
|                     | Chebyshev [Santos Junior and Schwartz, 2014]     | 0.838      | -          | -            |
|                     | WSVM-VGG [Scheirer et al., 2014]                 | 0.862      | 0.014      | 10           |
|                     | WSVM-HOG [Scheirer et al., 2014]                 | 0.515      | 0.027      | 10           |
|                     | HSVM-VGG   | 0.871      | 0.016      | 10           |
|                     | HSVM-HOG   | 0.902      | 0.015      | 10           |
|                     | HPLS-VGG   | 0.863      | 0.020      | 10           |
|                     | HPLS-HOG   | 0.910      | 0.022      | 10           |
|                     | HANN-VGG   | 0.867      | 0.026      | 10           |
| HANN-HOG            | 0.613  | 0.105      | 10         |              |

#### 5.2.2.4 Identification Evaluation

On the contrary of the previous experiments that only notify whether individuals are known, this section focuses on assessing the complete identification pipeline. That is, we return the identity that best matches the probe sample when the algorithm considers that the subject is enrolled in the gallery set. We couple the algorithm proposed in Chapter 4.1 with another PLS for regression so that a single model can be learnt for each subject following a *one-against-all* classification scheme implemented in the work of Schwartz et al. [2012].

In the One-Against-All Partial Least Squares (OAAPLS) approach, samples from the subject are learnt with positive response equal to +1 and samples from other subjects with negative response equal to -1. In other words, when the  $i$ -th individual is considered, all other subject's samples are used as counterexamples. In this case, the PLS regression model is learnt considering feature descriptors extracted from samples in the positive set with target values equal to +1 against samples in the negative set with target values equal to -1.

In this experiment, we employ HPLS as the trigger to OAAPLS since it is only executed when HPLS considers a subject as known. OAAPLS learns the same training samples employed in HPLS. Therefore, we do not select the most similar individuals to compose a list of candidates. During the testing stage, the probe sample is presented to each OAAPLS model, obtaining the identity from the model that returns the highest score.

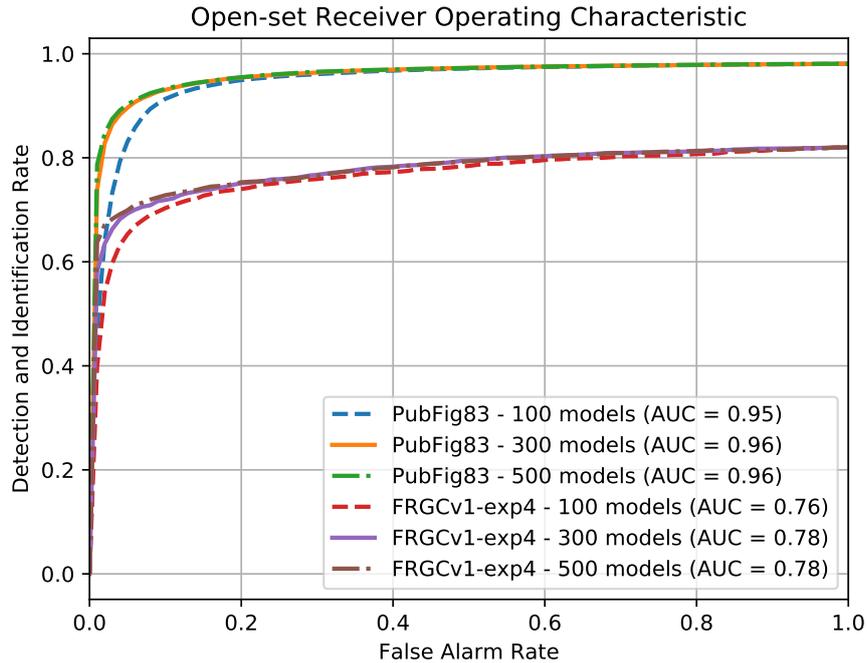


Figure 57: Open-set ROC curves for PubFig83 and FRGCv1 dataset on experiment four achieved with the combination of VGGFace features, HPLS and OAAPLS.

Interpreting open-set ROC curves resembles the form we discern the typical ROC curve. Therefore, the plot's top left corner is the optimal point. Figure 57 demonstrates that our method attained very good results for PubFig83. Even though FRGCv1 is composed of frontal faces, experiment four holds only one illumination-controlled image sample per individual for training step, which tends to be insufficient for a good representation. In this scenario, FRGCv1 turns out to be more challenging than PubFig83 due to the fact the latter has several samples per subject.

### 5.2.2.5 Single Classifier Evaluation

For the purpose of analyzing each classifier's behavior individually, experiments considering SVM, PLS and ANN are performed on the FRGCv1 experiment four, which considers a gallery with one controlled still picture for each subject plus a probe set having multiple uncontrolled images.

The objective behind this experiment is not to determine whether a subject is enrolled the gallery set. In fact, this experiment assesses how well a single classifier correctly matches the class a probe sample belongs to. If a subject  $s$  is randomly assigned to the positive collection, it is expected that the classifier outputs a positive response. In the same manner, if  $s$  belongs to the single classifier's negative collection, the classifier must return a negative response.

Table 53: Evaluation of SVM, PLS and ANN classifiers individually with features extracted using VGGFace. The presented results are: average hit rate (AVG), standard deviation (STD), minimum (MIN) and maximum (MAX). They are computed on 100-execution hit rates.

|        | Values | SVM (%) | PLS (%) | ANN (%) |
|--------|--------|---------|---------|---------|
| FRGCv1 | AVG    | 71.379  | 73.559  | 77.026  |
|        | STD    | 02.382  | 01.943  | 01.648  |
|        | MIN    | 66.419  | 70.614  | 73.245  |
|        | MAX    | 76.217  | 77.631  | 80.592  |

Hit rate is a synonym for *recall* and measures the proportion of positives that are correctly identified. Table 53 presents the hit rate for 100 executions. Results show that a ANN classifier alone provides better results than a PLS or SVM model. While a PLS model achieves a hit rate of approximately 73.56%, the SVM classifier attains 71.38% and the ANN-based classifier attains 77.03%. All methods hold tight standard deviation values and the variation between their minimum and maximum values remain close to the average.

### 5.2.2.6 Parameter Evaluation

To check how the three identification method versions respond to some parameter adjustments, we analyze the behavior of the approaches by varying the number of hashing models on the PubFig83 dataset as we adhere to the *Literature Protocol* and alternating the known individuals' set size for both VGGFace and FRGCv1 in accordance with the *Proposed Protocol*. Progressively adding hashing models results in extra random partitions, more models training and further probe feature projections at testing time. Initially, small number of classifiers does not seem adequate for discerning when an individual is registered in the gallery set. Table 54 and Table 55 expose how these parameters affect the implemented methods.

Table 54 shows great improvement in the initial classifier augmentation when varying from 10 to 50 binary classifiers. No significant accuracy improvement is noticed when more than 100 classifiers are established. The little increase in AUC for PubFig83 with increasingly hashing models may be justified by the fact that algorithms trained with multiple-sample gallery sets – for this experiment, PubFig83 holds around 90 samples per class and only 83 classes – are inclined to remain stable regardless of the number of hashing functions. If we reduce the number of samples per subject available at training time and increase the number of subjects enrolled in the gallery of known individuals, chances are more hashing models are required to keep AUC high.

Table 54: Variable number of hashing models: area under curve (AUC) and standard deviation (STD) for PubFig83, considering 75 randomly chosen subjects out of 83 in the known set and the eight left individuals compose the unknown test set in each of the 10 executions.

|          |          | Number of models | 10    | 20    | 30    | 50    | 100   | 300   | 500   |
|----------|----------|------------------|-------|-------|-------|-------|-------|-------|-------|
| PubFig83 | HSVM-VGG | AUC              | 0.683 | 0.816 | 0.881 | 0.908 | 0.940 | 0.966 | 0.972 |
|          |          | STD              | 0.028 | 0.019 | 0.018 | 0.013 | 0.010 | 0.007 | 0.005 |
|          | HPLS-VGG | AUC              | 0.743 | 0.866 | 0.885 | 0.932 | 0.940 | 0.960 | 0.968 |
|          |          | STD              | 0.030 | 0.024 | 0.029 | 0.021 | 0.020 | 0.006 | 0.004 |
|          | HANN-VGG | AUC              | 0.385 | 0.792 | 0.921 | 0.959 | 0.973 | 0.977 | 0.981 |
|          |          | STD              | 0.059 | 0.050 | 0.016 | 0.009 | 0.004 | 0.003 | 0.003 |

Table 55: Variable known individuals: area under curve (AUC) and standard deviation (STD) for FRGCv1 experiment four and VGGFace dataset. We secure 100 hashing models for HSVM, HPLS and HANN with VGGFace descriptor as we execute each algorithm 10 times.

|                                    |                                    | Known individuals | 10%   | 50%   | 90%   |
|------------------------------------|------------------------------------|-------------------|-------|-------|-------|
| FRGCv1 4                           | HANN-VGG                           | AUC               | 0.900 | 0.867 | 0.868 |
|                                    |                                    | STD               | 0.045 | 0.026 | 0.014 |
|                                    | HPLS-VGG                           | AUC               | 0.848 | 0.863 | 0.839 |
|                                    |                                    | STD               | 0.059 | 0.020 | 0.024 |
|                                    | HSVM-VGG                           | AUC               | 0.877 | 0.871 | 0.869 |
| STD                                |                                    | 0.021             | 0.016 | 0.011 |       |
| WSVM-VGG<br>Scheirer et al. [2014] | AUC                                | 0.866             | 0.862 | 0.848 |       |
|                                    | STD                                | 0.035             | 0.015 | 0.019 |       |
| VGGFace                            | HANN-VGG                           | AUC               | 0.987 | 0.976 | 0.965 |
|                                    |                                    | STD               | 0.003 | 0.004 | 0.006 |
|                                    | HPLS-VGG                           | AUC               | 0.978 | 0.961 | 0.926 |
|                                    |                                    | STD               | 0.005 | 0.003 | 0.005 |
|                                    | HSVM-VGG                           | AUC               | 0.967 | 0.943 | 0.725 |
|                                    |                                    | STD               | 0.014 | 0.006 | 0.004 |
|                                    | WSVM-VGG<br>Scheirer et al. [2014] | AUC               | 0.841 | 0.839 | 0.835 |
|                                    |                                    | STD               | 0.013 | 0.007 | 0.007 |

In general, the accuracy of a recognition system tends to reduce as we have more individuals enrolled in the gallery set. Besides, as this number increases, the computational cost usually skyrockets. Surprisingly, in Table 55, our open-set methods efficiency does not deteriorate with the enrollment of new subjects on the FRGCv1 dataset since having more samples increase the discriminability of classifiers when there are only few image faces per subject. FRGCv1 contains less than two hundred subjects and, as a result, raising the number of individuals in the known set from 10 to 90% does not severely affect the methods' accuracy.

According to Table 55 we believe that the stable behavior observed on HANN and HPLS approaches on both datasets lies on their capability of remaining robust despite of parameter adaptation and dataset selection. VGGFace Dataset offers more challenging experiments as it has a lot more than 200 individuals, in fact, it is composed of one thousand subjects. As a consequence, there was a sudden accuracy drop running HSVM on VGGFace as the 100-model SVM embedding could not separate the training data linearly when assigning 90% of all VGGFace datasets as the known set.

### 5.2.2.7 Complete Evaluation

In order to provide a complete experimentation overview, the following tables contain the results obtained on three datasets involving the extraction of deep features.

Table 56: Complete battery of experiments on FRGCv1 dataset following the *Proposed Protocol*: number of hashing models (#Models), average area under the curve (AUC), standard deviation (STD), number of subjects selected to composed the known set (KS) and the percentage of samples from each subject in the known set that should be user for training (TS).

| <b>Face Recognition Grand Challenge v1.0: Experiment 4</b> |       |       |       |       |       |       |                     |                |  |
|--|-------|-------|-------|-------|-------|-------|---------------------|----------------|--|
| #<br>Models  | HPLS  |       | HSVM  |       | HANN  |       | KS                  | TS             |  |
|  | AVG   | STD   | AVG   | STD   | AVG   | STD   |                     |                |  |
| 10   | 0.694 | 0.064 | 0.717 | 0.011 | 0.634 | 0.093 | 15/152<br>subjects  | 50%<br>samples |  |
| 20   | 0.778 | 0.041 | 0.786 | 0.057 | 0.734 | 0.105 |                     |                |  |
| 30   | 0.765 | 0.057 | 0.826 | 0.032 | 0.819 | 0.071 |                     |                |  |
| 50   | 0.826 | 0.037 | 0.861 | 0.020 | 0.876 | 0.025 |                     |                |  |
| 100  | 0.848 | 0.059 | 0.877 | 0.021 | 0.900 | 0.045 |                     |                |  |
| 300  | 0.826 | 0.045 | 0.888 | 0.048 | 0.886 | 0.047 |                     |                |  |
| 500  | 0.842 | 0.039 | 0.900 | 0.037 | 0.904 | 0.039 |                     |                |  |
| 10   | 0.698 | 0.029 | 0.666 | 0.029 | 0.418 | 0.071 | 76/152<br>subjects  | 50%<br>samples |  |
| 20   | 0.786 | 0.018 | 0.784 | 0.019 | 0.727 | 0.027 |                     |                |  |
| 30   | 0.809 | 0.028 | 0.819 | 0.018 | 0.819 | 0.019 |                     |                |  |
| 50   | 0.823 | 0.019 | 0.837 | 0.021 | 0.848 | 0.013 |                     |                |  |
| 100  | 0.863 | 0.020 | 0.871 | 0.016 | 0.867 | 0.026 |                     |                |  |
| 300  | 0.870 | 0.016 | 0.898 | 0.015 | 0.888 | 0.024 |                     |                |  |
| 500  | 0.879 | 0.018 | 0.898 | 0.009 | 0.896 | 0.019 |                     |                |  |
| 10   | 0.616 | 0.034 | 0.640 | 0.025 | 0.383 | 0.032 | 136/152<br>subjects | 50%<br>samples |  |
| 20   | 0.758 | 0.023 | 0.774 | 0.022 | 0.703 | 0.040 |                     |                |  |
| 30   | 0.801 | 0.019 | 0.792 | 0.019 | 0.775 | 0.020 |                     |                |  |
| 50   | 0.813 | 0.026 | 0.847 | 0.018 | 0.836 | 0.023 |                     |                |  |
| 100  | 0.839 | 0.024 | 0.869 | 0.011 | 0.868 | 0.014 |                     |                |  |
| 300  | 0.862 | 0.022 | 0.895 | 0.011 | 0.878 | 0.015 |                     |                |  |
| 500  | 0.844 | 0.012 | 0.890 | 0.009 | 0.890 | 0.019 |                     |                |  |

Tables 56, 57 and 58 provide a complete battery of experiments on different datasets adopting the VGGFace CNN feature descriptor. In these experiments, we vary both the number of hashing models from 10 to 500 and the number of subjects in the known set, which may range in terms of 10, 50 and 90% of all individuals from the dataset depending on the protocol. For each subject randomly allotted in the known set, a percentage of his/her face samples are used for training and the remaining samples comprise the known probe sample set. Individuals that are not selected to form the known set end up constituting the unknown probe set.

Table 57: Complete battery of experiments on VGGFace dataset subset following the *Proposed Protocol*: number of hashing models (#Models), average area under the curve (AUC), standard deviation (STD), number of subjects selected to composed the known set (KS) and the percentage of samples from each subject in the known set that should be user for training (TS).

| VGGFace Dataset Subset: 1000 Subjects |       |       |       |       |       |       |                          |                |
|---------------------------------------|-------|-------|-------|-------|-------|-------|--------------------------|----------------|
| #<br>Models                           | HPLS  |       | HSVM  |       | HANN  |       | KS                       | TS             |
|                                       | AVG   | STD   | AVG   | STD   | AVG   | STD   |                          |                |
| 10                                    | 0.775 | 0.015 | 0.712 | 0.032 | 0.322 | 0.037 | 100/<br>1000<br>subjects | 50%<br>samples |
| 20                                    | 0.915 | 0.016 | 0.897 | 0.014 | 0.795 | 0.064 |                          |                |
| 30                                    | 0.939 | 0.008 | 0.933 | 0.011 | 0.939 | 0.013 |                          |                |
| 50                                    | 0.960 | 0.007 | 0.943 | 0.028 | 0.976 | 0.005 |                          |                |
| 100                                   | 0.978 | 0.005 | 0.967 | 0.014 | 0.987 | 0.003 |                          |                |
| 300                                   | 0.988 | 0.003 | 0.982 | 0.013 | 0.992 | 0.004 |                          |                |
| 500                                   | 0.988 | 0.002 | 0.979 | 0.015 | 0.992 | 0.003 |                          |                |
| 10                                    | 0.558 | 0.009 | 0.540 | 0.006 | 0.223 | 0.061 | 500/<br>1000<br>subjects | 50%<br>samples |
| 20                                    | 0.794 | 0.004 | 0.663 | 0.006 | 0.624 | 0.035 |                          |                |
| 30                                    | 0.875 | 0.004 | 0.776 | 0.008 | 0.872 | 0.012 |                          |                |
| 50                                    | 0.929 | 0.005 | 0.874 | 0.007 | 0.949 | 0.007 |                          |                |
| 100                                   | 0.961 | 0.003 | 0.943 | 0.006 | 0.976 | 0.004 |                          |                |
| 300                                   | 0.977 | 0.002 | 0.972 | 0.003 | 0.987 | 0.002 |                          |                |
| 500                                   | 0.976 | 0.001 | 0.977 | 0.004 | 0.987 | 0.001 |                          |                |
| 10                                    | 0.504 | 0.014 | 0.504 | 0.012 | 0.244 | 0.022 | 900/<br>1000<br>subjects | 50%<br>samples |
| 20                                    | 0.662 | 0.011 | 0.516 | 0.012 | 0.530 | 0.030 |                          |                |
| 30                                    | 0.758 | 0.006 | 0.529 | 0.010 | 0.788 | 0.030 |                          |                |
| 50                                    | 0.854 | 0.010 | 0.599 | 0.007 | 0.928 | 0.005 |                          |                |
| 100                                   | 0.926 | 0.005 | 0.725 | 0.004 | 0.965 | 0.006 |                          |                |
| 300                                   | 0.959 | 0.005 | 0.895 | 0.006 | 0.984 | 0.002 |                          |                |
| 500                                   | 0.965 | 0.003 | 0.925 | 0.004 | 0.984 | 0.001 |                          |                |

Remember that for the *Proposed Protocol* the known individuals set size vary progressively. Regardless of the known set size, half the samples of each subject in the known set builds the gallery set and the other half makes the known probe set.

Contrarily, the *Literature Protocol* establishes that 90% of the individuals are assigned to the known and 90% of these subjects' face samples must be used for training. Table 58 goes beyond the employed protocol and also reveals how the embedding of classifiers perform on PubFig83 dataset with a variable known set size.

Table 58: Complete battery of experiments on PubFig83 dataset partially following the *Literature Protocol*: number of hashing models (#Models), average area under the curve (AUC), standard deviation (STD), number of subjects selected to composed the known set (KS) and the percentage of samples from each subject in the known set that should be user for training (TS).

| Public Figures 83: 83 Subjects |       |       |       |       |       |       |                |             |
|--------------------------------|-------|-------|-------|-------|-------|-------|----------------|-------------|
| #<br>Models                    | HPLS  |       | HSVM  |       | HANN  |       | KS             | TS          |
|                                | AVG   | STD   | AVG   | STD   | AVG   | STD   |                |             |
| 10                             | 0.762 | 0.048 | 0.728 | 0.037 | 0.688 | 0.057 | 8/83 subjects  | 90% samples |
| 20                             | 0.741 | 0.059 | 0.778 | 0.034 | 0.86  | 0.05  |                |             |
| 30                             | 0.826 | 0.051 | 0.783 | 0.046 | 0.915 | 0.020 |                |             |
| 50                             | 0.855 | 0.060 | 0.820 | 0.047 | 0.945 | 0.014 |                |             |
| 100                            | 0.882 | 0.040 | 0.839 | 0.045 | 0.962 | 0.012 |                |             |
| 300                            | 0.899 | 0.033 | 0.807 | 0.034 | 0.969 | 0.010 |                |             |
| 500                            | 0.919 | 0.017 | 0.775 | 0.039 | 0.975 | 0.014 |                |             |
| 10                             | 0.803 | 0.023 | 0.765 | 0.026 | 0.464 | 0.076 | 41/83 subjects | 90% samples |
| 20                             | 0.875 | 0.027 | 0.868 | 0.015 | 0.804 | 0.047 |                |             |
| 30                             | 0.910 | 0.014 | 0.890 | 0.018 | 0.927 | 0.017 |                |             |
| 50                             | 0.935 | 0.015 | 0.932 | 0.011 | 0.966 | 0.008 |                |             |
| 100                            | 0.957 | 0.006 | 0.951 | 0.014 | 0.973 | 0.006 |                |             |
| 300                            | 0.970 | 0.005 | 0.973 | 0.006 | 0.983 | 0.004 |                |             |
| 500                            | 0.976 | 0.003 | 0.979 | 0.002 | 0.981 | 0.003 |                |             |
| 10                             | 0.743 | 0.030 | 0.683 | 0.028 | 0.385 | 0.059 | 75/83 subjects | 90% samples |
| 20                             | 0.866 | 0.024 | 0.816 | 0.019 | 0.792 | 0.050 |                |             |
| 30                             | 0.885 | 0.029 | 0.881 | 0.018 | 0.921 | 0.016 |                |             |
| 50                             | 0.932 | 0.021 | 0.908 | 0.013 | 0.959 | 0.009 |                |             |
| 100                            | 0.940 | 0.020 | 0.940 | 0.010 | 0.973 | 0.004 |                |             |
| 300                            | 0.960 | 0.006 | 0.966 | 0.007 | 0.977 | 0.003 |                |             |
| 500                            | 0.968 | 0.004 | 0.972 | 0.005 | 0.981 | 0.003 |                |             |

Although speed and low computational cost are largely desired, it was obtained as a consequence of handling collections of binary classifiers for a different purpose: discriminating known from unknown samples in regard to a gallery of registered individuals. Low computational cost and speed comes from the fact that the employment of simple binary classifiers as hashing functions provide compelling improvement over the brute-force approach, a process in which all subjects from the gallery are compared.

## 5.3 Face Verification

In this section, we evaluate our approach for face verification. The proposed method generates an embedding of binary Partial Least Squares (PLS) or Support Vector Machine (SVM) models coupled with majority voting to determine whether two faces belong to the same person. The face verification framework and experimental data are available online<sup>3</sup>. Note that the embedding of binary partial least squares classifiers or support vector machines for face verification are denoted on the experiments as HPLSV and HSVMV, precisely.

### 5.3.1 Evaluation Protocol

In addition to ROC curves and AUC metric, described in Section 5.2.1, we measure the effectiveness of our approach considering Equal Error Rate (EER), a measure usually employed on face verification and biometrics in general that indicates the value where the fraction of genuine samples classified as impostor (false rejection rate, FRR) is equal to fraction of impostor samples classified as genuine (false acceptance rate, FAR). The lower the equal error rate, the higher the accuracy of the biometric system.

For the evaluation performed on the LFW dataset, we use the protocol *unrestricted, labeled outside data* for all experiments. We show the ROC curve, its AUC and the standard deviation error (STD) on the deep-funneled LFW. The *unrestricted* protocol allows researchers to exploit identities in the training set so that it is possible to generate more training pairs and add them to the training stage. For the PubFig, we present the EER and the standard deviation.

Differently from many approaches that achieve state-of-the-art results following LFW’s *unrestricted, labeled outside data* protocol, we neither focus on grouping millions of images in the interest of learning discriminative face representations using convolutional neural networks [Ding and Tao, 2015; Masi et al., 2016; Schroff et al., 2015; Taigman et al., 2014] nor make use of additional face datasets to train *same/not-same* classifiers [Ding and Tao, 2015; Taigman et al., 2014]. These authors claim that employing either LFW-A or PubFig to produce more training pairs substantially overfits the training data due to their redundant characteristics. On contrary, we carry out a minimal training, working with pairs of images recommended by the dataset. The only outside data we use are the samples required in the learning process of the VGGFace CNN descriptor [Parkhi et al., 2015].

---

<sup>3</sup><https://github.com/rafaelvareto/HPLS-verification>

### 5.3.2 Method Evaluation

The algorithm proposed in Section 4.2 is evaluated with LFW-A and PubFig datasets following the *unrestricted, labeled outside data* protocol. Our experiments are grouped in two categories: same-dataset evaluation and cross-dataset evaluation. In the *same-dataset* evaluation, we follow the LFW and PubFig splits strictly with no use of additional labeled training examples to increase the amount of data available when learning the classification models, as we understand that outside datasets only for the purposes of extracting features is significantly different than using outside data to train classifiers. In the *cross-dataset* evaluation, we train the classifiers using PubFig development set and evaluate the performance on LFW splits for cross validation. We use the PubFig development set because it is entirely disjoint of LFW identities and PubFig evaluation set individuals.

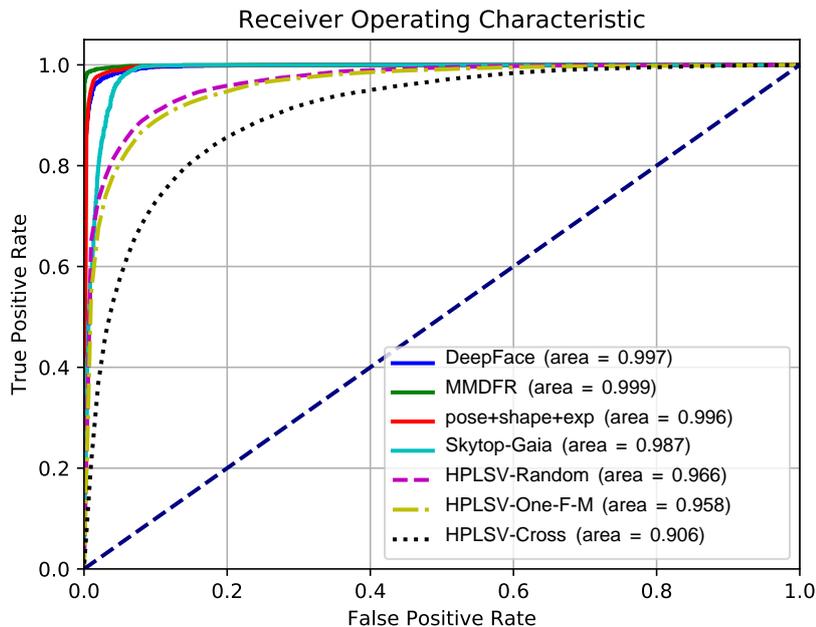


Figure 58: Average ROC curves for the LFW-A dataset and its respective area under the curve (AUC). Some curves represent experiments conducted with deep funneled face images. We repeat our experiments ten times for each setting. The plot considers the following methods: DeepFace [Taigman et al., 2014], MMDFR [Ding and Tao, 2015], Pose+Shape+Exp [Masi et al., 2016] and a commercial recognition system called SkyTop.

Table 59 shows the results on PubFig whereas Figure 58 shows the experiments on the LFW-A dataset. In these datasets, the cross-validation evaluation is adopted among the available folds, and we report the averaged results. Our approach was evaluated for different settings, described as follows.

- **Cross:** it categorizes a cross-dataset verification, in which the training stage uses images from PubFig development set and LFW folds are used during the testing stage. PubFig development set does not have a list of *same/not-same* training pairs. Therefore, training tuples were symmetrically sampled in a random manner.
- **Dev-Eval:** it is analogous to the cross-dataset experiment, but it comprises PubFig development set in the training stage and its evaluation set for testing. Thus, it does not constitute a cross-dataset experiment.
- **One-F-M:** the number of PLS models is associated with the number of folds in the cross validations scheme – one fold per PLS binary model. Particularly, in each iteration we pick a fold to test and each one of the remaining folds comprises a PLS model. Since the datasets have ten folds, we only generate nine PLS classification models.
- **Random:** it randomly allocates *same/not-same* training pairs into each PLS binary model as explained in Section 4.2.2, ensuring that all training pairs are evenly distributed among all models.

Table 59: Average equal error rate (EER) and standard deviation (STD) for the PubFig dataset. Top rows indicate approaches with state-of-the-art performance, our performance is shown in mid rows and bottom rows present other relevant methods.

|                | Approaches                        | EER (%)  | STD   |      |
|----------------|-----------------------------------|----------|-------|------|
| PubFig Dataset | DRM-WV [Hayat et al., 2015]       | 02.80    | 0.57  |      |
|                | RNP [Yang et al., 2013]           | 10.79    | 0.83  |      |
|                | HPLSV                             | Dev-Eval | 13.65 | 2.11 |
|                |                                   | Random   | 14.73 | 2.02 |
|                |                                   | One-F-M  | 16.63 | 3.05 |
|                | HSVMV                             | Dev-Eval | 13.89 | 1.97 |
|                |                                   | Random   | 14.64 | 2.08 |
|                |                                   | One-F-M  | 15.95 | 2.56 |
|                | CHISD [Cevikalp and Triggs, 2010] | 19.15    | 0.71  |      |
|                | GEDA [Harandi et al., 2011]       | 23.90    | 1.29  |      |

We observe that running the verification algorithm ten times for every setting, except for *One-F-M* once it is deterministic, provides fair stability and small standard deviation error. The results presented on Table 59 and on Figure 58 show that the method achieves comparable performance on both benchmarks making use of much less required data during the training stage and applying no data pre-processing algorithm.

### 5.3.2.1 Parameter Evaluation

We can note that the approach achieves good results even fixing the number of disparity features in every classification model to 100 samples in each positive and negative collection. To check how the method responds to some parameter adjustments, we analyze its accuracy behavior by varying the number of PLS and SVM-based binary classification models for both LFW-A and PubFig datasets under the *Random*, *Dev-Eval* and *Cross* settings as previously described in Section 5.3.2. Average results are gathered in Table 510 in virtue of they point up how the quantity of hashing models affects the proposed methods.

Table 510: Evaluation of our method’s performance (AUC) and standard deviation (STD) on LFW-A and PubFig datasets having an increasingly number of PLS and SVM classification models with disparity samples fixed to 100.

|                   |                 | Number of Models | 10    | 30    | 50    | 100   | 300   | 500   |
|-------------------|-----------------|------------------|-------|-------|-------|-------|-------|-------|
| H <sub>PLSV</sub> | Cross-LFW-A     | AUC              | 0.817 | 0.856 | 0.867 | 0.890 | 0.906 | 0.908 |
|                   |                 | STD              | 0.014 | 0.017 | 0.016 | 0.017 | 0.017 | 0.014 |
|                   | Dev-Eval-PubFig | AUC              | 0.926 | 0.938 | 0.938 | 0.940 | 0.942 | 0.942 |
|                   |                 | STD              | 0.014 | 0.014 | 0.014 | 0.014 | 0.012 | 0.012 |
|                   | Random-LFW-A    | AUC              | 0.942 | 0.953 | 0.953 | 0.954 | 0.966 | 0.966 |
|                   |                 | STD              | 0.012 | 0.012 | 0.012 | 0.012 | 0.011 | 0.011 |
|                   | Random-PubFig   | AUC              | 0.924 | 0.935 | 0.936 | 0.936 | 0.936 | 0.937 |
|                   |                 | STD              | 0.015 | 0.014 | 0.014 | 0.014 | 0.014 | 0.012 |
| H <sub>SVM</sub>  | Cross-LFW-A     | AUC              | 0.920 | 0.920 | 0.921 | 0.921 | 0.921 | 0.922 |
|                   |                 | STD              | 0.014 | 0.014 | 0.014 | 0.014 | 0.014 | 0.014 |
|                   | Dev-Eval-PubFig | AUC              | 0.932 | 0.936 | 0.936 | 0.936 | 0.937 | 0.937 |
|                   |                 | STD              | 0.016 | 0.014 | 0.014 | 0.014 | 0.012 | 0.014 |
|                   | Random-LFW-A    | AUC              | 0.952 | 0.955 | 0.955 | 0.956 | 0.956 | 0.956 |
|                   |                 | STD              | 0.009 | 0.010 | 0.009 | 0.010 | 0.010 | 0.010 |
|                   | Random-PubFig   | AUC              | 0.931 | 0.934 | 0.936 | 0.936 | 0.936 | 0.936 |
|                   |                 | STD              | 0.015 | 0.014 | 0.012 | 0.012 | 0.012 | 0.012 |

According to the results presented in Table 510, there is a constant improvement when the number of binary classification models is increased from 10 to 50, specially for the PLS-based approach, indicating the need for multiple hashing models. However, there is no large AUC improvement for both datasets when the number of models increases from 100 to 500. It may be justified by the fact that algorithms trained with few-sample or few-subject gallery sets – LFW and PubFig, respectively – are inclined to remain invariable because most PLS models may be very similar to one another. Then, adding more binary models only increases computational time without significant result improvement.

The cross-dataset setting can also be analyzed according to Table 510. We can see a slightly inferiority since training data (i.e., PubFig development set) are not aligned and the testing dataset images (i.e., LFW-A), are aligned. Such alignment is intended to lessen undesired pose variations as actual systems can rarely count on the cooperation of people being framed in order to assist the recognition process.

Table 511: Evaluation of our method’s performance (AUC) and standard deviation (STD) on LFW-A and PubFig datasets having an increasingly number of relational disparity features and number of binary hashing models fixed to 100.

|       |                 | Number of Samples | 50    | 100   | 300   | 500   |
|-------|-----------------|-------------------|-------|-------|-------|-------|
| HPLSV | Cross-LFW-A     | AUC               | 0.898 | 0.890 | 0.893 | 0.894 |
|       |                 | STD               | 0.016 | 0.017 | 0.016 | 0.015 |
|       | Dev-Eval-PubFig | AUC               | 0.935 | 0.940 | 0.945 | 0.951 |
|       |                 | STD               | 0.014 | 0.014 | 0.010 | 0.012 |
|       | Random-LFW-A    | AUC               | 0.950 | 0.954 | 0.961 | 0.966 |
|       |                 | STD               | 0.012 | 0.012 | 0.007 | 0.008 |
|       | Random-PubFig   | AUC               | 0.933 | 0.936 | 0.938 | 0.941 |
|       |                 | STD               | 0.016 | 0.014 | 0.015 | 0.014 |
| HsvMV | Cross-LFW-A     | AUC               | 0.920 | 0.921 | 0.923 | 0.926 |
|       |                 | STD               | 0.014 | 0.014 | 0.014 | 0.014 |
|       | Dev-Eval-PubFig | AUC               | 0.933 | 0.936 | 0.941 | 0.945 |
|       |                 | STD               | 0.016 | 0.014 | 0.016 | 0.010 |
|       | Random-LFW-A    | AUC               | 0.951 | 0.956 | 0.963 | 0.967 |
|       |                 | STD               | 0.010 | 0.010 | 0.009 | 0.009 |
|       | Random-PubFig   | AUC               | 0.930 | 0.936 | 0.939 | 0.940 |
|       |                 | STD               | 0.015 | 0.013 | 0.014 | 0.016 |

Average results in Table 511 indicate how influential the quantity of relational disparity features for each binary hashing model is. The experiment comprises an accuracy analysis with a progressive augmentation of disparity samples in PLS and SVM classification models for both LFW-A and PubFig datasets under the *Random*, *Dev-Eval* and *Cross* settings. Even though Tables 510 and 511 portray different experiments, they have a comparable performance. For most experimental settings in Table 511 there are stable enhancements when the number of disparity features ranges from 50 to 500 samples. However, differently from generating more binary classifiers, increasing the amount of relational disparity features does not seem to draw the accuracy towards converging values. It probably happens because incrementing the number of relational disparity features does not over-fit the classifier but, in fact, augments its discriminating capability. Therefore, additional samples make classifiers predict future observations more accurately.

### 5.3.2.2 Time Consumption Evaluation

The execution time or CPU runtime of a specified task is commonly characterized as the time spent by the system executing that task, including the time spent running system services on its behalf. That is, the runtime comprises the moment the Python script execution begins in conjunction with the program’s entry point and the corresponding memory setup. On the contrary, the carried-out evaluation only takes into account the time required for training and testing. It neither considers the feature extraction process nor the setup time required to perform the learning and predicting stages. We focus mainly on the machine learning runtime since it tends to be the most time-demanding process in machine learning tools on the grounds that we are not interested in computing the time taken by third-party modules.

Table 512: Evaluation of our method’s runtime (RUN), measured in seconds, and standard deviation (STD) on LFW-A and PubFig having an increasingly number of PLS and SVM classification models with subject samples fixed to 100.

|       |                 | Number of Models | 10     | 30     | 50     | 100     | 300     | 500     |
|-------|-----------------|------------------|--------|--------|--------|---------|---------|---------|
| HPLSV | Cross-LFW-A     | RUN              | 6.049  | 17.581 | 28.999 | 59.864  | 171.757 | 275.088 |
|       |                 | STD              | 0.088  | 0.171  | 0.244  | 2.281   | 1.707   | 5.920   |
|       | Dev-Eval-PubFig | RUN              | 4.726  | 13.593 | 22.477 | 44.541  | 148.622 | 235.986 |
|       |                 | STD              | 0.207  | 0.480  | 0.484  | 0.537   | 16.731  | 6.034   |
|       | Random-LFW-A    | RUN              | 6.540  | 18.697 | 30.652 | 63.763  | 179.239 | 289.118 |
|       |                 | STD              | 0.116  | 0.247  | 0.340  | 1.863   | 1.650   | 6.608   |
|       | Random-PubFig   | RUN              | 5.879  | 15.004 | 24.552 | 46.801  | 137.307 | 228.446 |
|       |                 | STD              | 0.397  | 0.383  | 0.443  | 0.264   | 1.332   | 1.634   |
| HSVMV | Cross-LFW-A     | RUN              | 10.390 | 30.622 | 52.462 | 100.467 | 288.832 | 478.059 |
|       |                 | STD              | 0.237  | 0.606  | 0.936  | 2.446   | 10.874  | 8.183   |
|       | Dev-Eval-PubFig | RUN              | 7.516  | 21.751 | 36.465 | 70.966  | 213.170 | 366.171 |
|       |                 | STD              | 0.170  | 0.327  | 0.432  | 0.510   | 2.103   | 14.694  |
|       | Random-LFW-A    | RUN              | 10.387 | 29.954 | 49.917 | 97.678  | 289.168 | 464.783 |
|       |                 | STD              | 0.065  | 0.209  | 1.331  | 1.519   | 10.521  | 2.103   |
|       | Random-PubFig   | RUN              | 9.207  | 23.396 | 36.668 | 71.379  | 208.221 | 345.276 |
|       |                 | STD              | 0.125  | 0.194  | 0.232  | 0.590   | 1.246   | 2.613   |

In this experiment we demonstrate how a similar parameter adaptation affects the computational cost in terms of time. This analysis investigates which value for the number of hashing models provides best trade-off between low execution time and high area under the curve. According to Table 512 one can note there is a near-linear runtime rise indicating that execution time grows accordingly to the number of binary models in every step, which represents a constant scale factor commonly illustrated by straight lines on a graph.

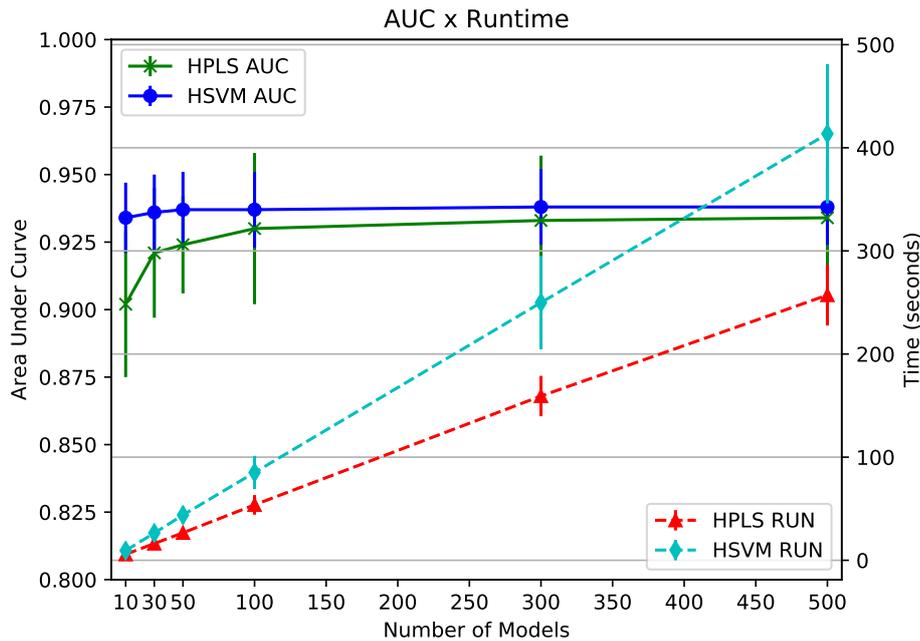


Figure 59: A comparison between the area under the curve (AUC) and the execution time (RUN). Straight lines represent the experiments comprising an accuracy evaluation whereas dashed lines comprehend the average time taken. Each curve/line involves the mean values of proposed approach on LFW-A and PubFig datasets.

Figure 59 contrasts the results demonstrated in Table 510 and 512. According to the chart, there is a linear growth in the lines corresponding to the runtime evaluation. Contrarily, the accuracy-related curves resemble a horizontal line, which denotes a consistent performance that is not highly dependent on the number of binary classifiers. It is not required to run the verification approach with either hundreds or thousands of models to attain great results. Actually, with only a 100-model embedding, results are comparable to executions with more classifiers. We conclude that it is only possible for the reason that aggregating classifiers improves the stability and accuracy of the employed machine learning algorithms, reducing variance and avoiding over-fitting.

Although the proposed approach has not outperformed state-of-the-art methods, the experiments show that it attains favorable results. Furthermore, the approach remains stable even under different domains with limited number of training samples. The cross-dataset evaluation of PubFig development set and LFW folds demonstrates that the method can consistently achieve promising results while maintaining satisfactory generalization ability. Overall, this work confirms that there is no need of large amount of data in pursuance of quality results on the chosen benchmarks. With few thousands of face images, simple but robust algorithms can achieve very accurate results.



# Chapter 6

## Conclusions

The use of embedding of classifiers has been widely used for fast image retrieval. Throughout this work, two methods were proposed and analyzed in favor of answering a single question: *Can methods originally designed for fast image retrieval be successfully applied to binary classification?*

We were inspired by the potential of simple binary classifiers and how locality-sensitive hashing splits the feature space. We decided to take advantage of their speed and low computational cost to determine either if a probe face sample is known for the open-set face identification task or if two face images represent the same person for the face verification task. Experiments were carried out in a variety of datasets attained satisfactory results. One of the main advantages of our two methods is their simplicity and practical deployment since only two key parameters deeply influence performance: the number of hashing functions and the number of subjects for each hashing model.

In this work, we did not concentrate on determining the correct identity of an individual, but we actually focused on the classification problem only. Our identification approach does not perform an incremental enrollment without retraining all hashing models, which may be restrictive. In any case, the cross-dataset verification evaluation showed that our method bears great generalization capability.

As future directions, a complete pipeline for open-set face identification, considering the generation of a list of candidates as well as evaluating the identification method on huge galleries is an applicable extension. For the face verification approach, either other datasets could be incorporated in the training stage in pursuance of better facial discrimination or fine-tune some of the last layers of the VGGFace CNN descriptor.



# Bibliography

- Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *Transactions on Pattern Analysis and Machine Intelligence*.
- Andersson, M. (2009). A comparison of nine pls1 algorithms. *Journal of Chemometrics*.
- Ashbourn, J. (2014). *Biometrics: Advanced identity verification: the complete guide*. Springer.
- Barkan, O., Weill, J., Wolf, L., and Aronowitz, H. (2013). Fast high dimensional vector multiplication face recognition. In *International Conference on Computer Vision*.
- Bendale, A. and Boulton, T. (2015). Towards open world recognition. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Bendale, A. and Boulton, T. E. (2016). Towards open set deep networks. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *Transactions on Pattern Analysis and Machine Intelligence*.
- Best-Rowden, L., Han, H., Otto, C., Klare, B. F., and Jain, A. K. (2014). Unconstrained face recognition: Identifying a person of interest from a media collection. *Information Forensics and Security*.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123--140.
- Breiman, L. et al. (1994). Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6):2350--2383.
- Carlos, G., Pedrini, H., and Schwartz, W. R. (2013). Fast and scalable enrollment for face identification based on partial least squares. In *AFGR*. IEEE.

- Cevikalp, H. and Triggs, B. (2010). Face recognition based on image sets. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Cevikalp, H. and Triggs, B. (2012). Efficient object detection using cascades of nearest convex model classifiers. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Chellappa, R., Sinha, P., and Phillips, P. J. (2010). Face recognition by computers and humans. *Computer*.
- Chen, D., Cao, X., Wen, F., and Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Chen, J.-C., Patel, V. M., and Chellappa, R. (2016). Unconstrained face verification using deep cnn features. In *Winter Conference on Applications of Computer Vision*. IEEE.
- Costa, F. d. O., Eckmann, M., Scheirer, W. J., and Rocha, A. (2012). Open set source camera attribution. In *Conference on Graphics, Patterns and Images*, pages 71--78. IEEE.
- Costa, F. d. O., Silva, E., Eckmann, M., Scheirer, W. J., and Rocha, A. (2014). Open set source camera attribution and device linking. *Pattern Recognition Letters*.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Computational Geometry*. ACM.
- De Jong, S. (1993). Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*.
- Ding, C., Choi, J., Tao, D., and Davis, L. S. (2016). Multi-directional multi-level dual-cross patterns for robust face recognition. *Transactions on Pattern Analysis and Machine Intelligence*.
- Ding, C. and Tao, D. (2015). Robust face recognition via multimodal deep face representation. *Transactions on Multimedia*.

- Du, L. and Ling, H. (2015). Cross-age face verification by coordinating with cross-face age verification. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the nystrom method. *Transactions on Pattern Analysis and Machine Intelligence*.
- Galbally, J., Marcel, S., and Fierrez, J. (2014). Biometric antispoofing methods: A survey in face recognition. *IEEE Access*.
- Ge, T., He, K., Ke, Q., and Sun, J. (2013). Optimized product quantization for approximate nearest neighbor search. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Guo, H., Schwartz, W. R., and Davis, L. S. (2011). Face verification using large feature sets and one shot similarity. In *International Joint Conference on Biometrics*. IEEE.
- Harandi, M. T., Sanderson, C., Shirazi, S., and Lovell, B. C. (2011). Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Hayat, M., Bennamoun, M., and An, S. (2015). Deep reconstruction models for image set classification. *Transactions on Pattern Analysis and Machine Intelligence*.
- Hayat, M., Khan, S. H., Werghi, N., and Goecke, R. (2017). Joint registration and representation learning for unconstrained face identification. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Hu, J., Lu, J., and Tan, Y.-P. (2014). Discriminative deep metric learning for face verification in the wild. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Hu, S., Choi, J., Chan, A. L., and Schwartz, W. R. (2015). Thermal-to-visible face recognition using partial least squares. *Journal of the Optical Society of America A*.
- Huang, G., Mattar, M., Lee, H., and Learned-Miller, E. G. (2012a). Learning to align from scratch. In *Advances in Neural Information Processing Systems*. NIPS.
- Huang, G. B. and Learned-Miller, E. (2014). Labeled faces in the wild: Updates and new reporting procedures. Technical report, Department of Computer Science, University of Massachusetts – Amherst, USA.

- Huang, G. B., Lee, H., and Learned-Miller, E. (2012b). Learning hierarchical representations for face verification with convolutional deep belief networks. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Department of Computer Science, University of Massachusetts – Amherst, USA.
- Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM symposium on Theory of computing*. ACM.
- Irie, G., Li, Z., Wu, X.-M., and Chang, S.-F. (2014). Locally linear hashing for extracting non-linear manifolds. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Jain, A. K., Flynn, P., and Ross, A. A. (2008). *Handbook of Biometrics*. Springer.
- Jain, A. K. and Li, S. Z. (2011). *Handbook of face recognition*. Springer.
- Jain, L. P., Scheirer, W. J., and Boulton, T. E. (2014). Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*. Springer.
- Jegou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *Transactions on Pattern Analysis and Machine Intelligence*.
- Joly, A. and Buisson, O. (2011). Random maximum margin hashing. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Kalantidis, Y. and Avrithis, Y. (2014). Locally optimized product quantization for approximate nearest neighbor search. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Kamgar-Parsi, B., Lawson, W., and Kamgar-Parsi, B. (2011). Toward development of a face recognition system for watchlist surveillance. *Transactions on Pattern Analysis and Machine Intelligence*.
- Klare, B. F. and Jain, A. K. (2013). Heterogeneous face recognition using kernel prototype similarities. *Transactions on Pattern Analysis and Machine Intelligence*.
- Kulis, B. and Darrell, T. (2009). Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*. NIPS.

- Kulis, B. and Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *International Conference on Computer Vision*. IEEE.
- Kulis, B. and Grauman, K. (2012). Kernelized locality-sensitive hashing. *Transactions on Pattern Analysis and Machine Intelligence*.
- Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *International Conference on Computer Vision*. IEEE.
- Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., and Hua, G. (2016). *Labeled faces in the wild: A survey*. Springer.
- Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*. ACM.
- Lei, Y., Bennamoun, M., Hayat, M., and Guo, Y. (2014). An efficient 3d face recognition approach using local geometrical signatures. *Pattern Recognition*.
- Liao, S., Lei, Z., Yi, D., and Li, S. Z. (2014). A benchmark study of large-scale unconstrained face recognition. In *International Joint Conference on Biometrics*. IEEE.
- Lin, M.-I. B., Groves, W. A., Freivalds, A., Lee, E. G., and Harper, M. (2012). Comparison of artificial neural network (ann) and partial least squares (pls) regression models for predicting respiratory ventilation: an exploratory study. *European journal of applied physiology*.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*.
- Lu, J., Tan, Y.-P., and Wang, G. (2013). Discriminative multimanifold analysis for face recognition from a single training sample per person. *Transactions on Pattern Analysis and Machine Intelligence*.
- Mandal, B., Chia, S.-C., Li, L., Chandrasekhar, V., Tan, C., and Lim, J.-H. (2014). A wearable face recognition system on google glass for assisting social interactions. In *Asian Conference on Computer Vision*. Springer.
- Masi, I., Trãn, A. T., Hassner, T., Leksut, J. T., and Medioni, G. (2016). Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*. Springer.

- Mouazen, A., Kuang, B., De Baerdemaeker, J., and Ramon, H. (2010). Comparison among principal component, partial least squares and back propagation neural network analyses for accuracy of measurement of selected soil properties with visible and near infrared spectroscopy. *Geoderma*, 158(1):23–31.
- Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Transactions on Pattern Analysis and Machine Intelligence*.
- Mygdalis, V., Alexandros, I., Tefas, A., and Pitas, I. (2015). Large-scale classification by an approximate least squares one-class support vector machine ensemble. In *Trustcom/BigDataSE/ISPA*. IEEE.
- Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*.
- Okamoto, T. et al. (2015). *Financial Cryptography and Data Security*. Springer.
- Ouamane, A., Bengherabi, M., Hadid, A., and Cheriet, M. (2015). Side-information based exponential discriminant analysis for face verification in the wild. In *Conference on Automatic Face and Gesture Recognition Workshop*. IEEE.
- Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*. IEEE.
- Pham, T.-A., Barrat, S., Delalandre, M., and Ramel, J.-Y. (2015). An efficient tree structure for indexing feature vectors. *Pattern Recognition Letters*.
- Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). Overview of the face recognition grand challenge. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Pinto, N., Stone, Z., Zickler, T., and Cox, D. (2011). Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In *Computer Vision and Pattern Recognition Workshops*. IEEE.
- Rosipal, R. and Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, latent structure and feature selection*. Springer.
- Santos Junior, C., Kijak, E., Gravier, G., and Schwartz, W. R. (2016). Partial least squares for face hashing. *Neurocomputing*.

- Santos Junior, C. and Schwartz, W. R. (2014). Extending face identification to open-set face recognition. In *Conference on Graphics, Patterns and Images*. IEEE.
- Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2013). Toward open set recognition. *Transactions on Pattern Analysis and Machine Intelligence*.
- Scheirer, W. J., Jain, L. P., and Boult, T. E. (2014). Probability models for open set recognition. *Transactions on Pattern Analysis and Machine Intelligence*.
- Scherreik, M. D. and Rigling, B. D. (2016). Open set recognition for automatic target classification with rejection. *Transactions on Aerospace and Electronic Systems*.
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*.
- Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Schwartz, W. R., Guo, H., Choi, J., and Davis, L. S. (2012). Face identification using large feature sets. *Transactions on Image Processing*.
- Shakhnarovich, G. (2005). *Learning task-specific similarity*. PhD thesis, Massachusetts Institute of Technology.
- Silpa-Anan, C. and Hartley, R. (2008). Optimised kd-trees for fast image descriptor matching. In *Conference on Computer Vision and Pattern Recognition*, pages 1--8. IEEE.
- Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher vector faces in the wild. In *British Machine Vision Conference*.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.
- Sun, Y., Wang, X., and Tang, X. (2013). Hybrid deep learning for face verification. In *International Conference on Computer Vision*. IEEE.

- Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing*.
- Tang, J., Li, Z., and Zhu, X. (2017). Supervised deep hashing for scalable face image retrieval. *Pattern Recognition*.
- Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., and Ma, Y. (2012). Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *Transactions on Pattern Analysis and Machine Intelligence*.
- Wang, D., Otto, C., and Jain, A. K. (2015). Face search at scale: 80 million gallery. Technical report, Department of Computer Science, Michigan State University – East Lansing, USA.
- Wang, J., Kumar, S., and Chang, S.-F. (2010). Semi-supervised hashing for scalable image retrieval. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Wang, J., Kumar, S., and Chang, S.-F. (2012). Semi-supervised hashing for large-scale search. *Transactions on Pattern Analysis and Machine Intelligence*.
- Wang, J., Liu, W., Sun, A. X., and Jiang, Y.-G. (2013). Learning hash codes with listwise supervision. In *International Conference on Computer Vision*. IEEE.
- Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*. Springer.
- Wold, H. (1985). Partial least squares. *Encyclopedia of Statistical Sciences*.
- Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*.
- Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *Transactions on Pattern Analysis and Machine Intelligence*.
- Yang, M., Zhu, P., Van Gool, L., and Zhang, L. (2013). Face recognition based on regularized nearest points between image sets. In *Conference on Automatic Face and Gesture Recognition Workshop*. IEEE.

- Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.
- Yi, D., Lei, Z., and Li, S. Z. (2013). Towards pose robust face recognition. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Zaklouta, F., Stanciulescu, B., and Hamdoun, O. (2011). Traffic sign classification using kd trees and random forests. In *International Joint Conference on Neural Networks*. IEEE.
- Zhang, H. and Patel, V. (2016). Sparse representation-based open set recognition. *Transactions on Pattern Analysis and Machine Intelligence*.
- Zhang, X. and Gao, Y. (2009). Face recognition across pose: A review. *Pattern Recognition*.
- Zhang, X., Zhang, L., and Shum, H.-Y. (2012). Qsrank: Query-sensitive hash code ranking for efficient e-neighbor search. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Zheng, L., Idrissi, K., Garcia, C., Duffner, S., and Baskurt, A. (2015). Triangular similarity metric learning for face verification. In *Conference on Automatic Face and Gesture Recognition Workshop*. IEEE.
- Zhu, X., Lei, Z., Yan, J., Yi, D., and Li, S. Z. (2015a). High-fidelity pose and expression normalization for face recognition in the wild. In *Conference on Computer Vision and Pattern Recognition*. IEEE.
- Zhu, X., Yan, J., Yi, D., Lei, Z., and Li, S. Z. (2015b). Discriminative 3d morphable model fitting. In *Conference on Automatic Face and Gesture Recognition*. IEEE.