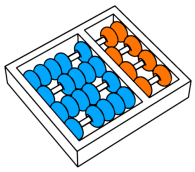
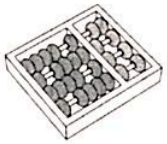


Samuel Botter Martins

"A Fast and Robust Negative Mining Approach for User Enrollment in Face Recognition Systems"

"*Uma Abordagem Eficiente e Robusta de Mineração de Negativos para Cadastramento de Novos Usuários em Sistemas de Reconhecimento Facial*"

CAMPINAS
2015



University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Samuel Botter Martins

"A Fast and Robust Negative Mining Approach for User Enrollment in Face Recognition Systems"

Supervisor:
Orientador(a): Prof. Dr. Alexandre Xavier Falcão

Co-Supervisor:
Co-orientador(a): Dr. Giovani Chiachia

"Uma Abordagem Eficiente e Robusta de Mineração de Negativos para Cadastramento de Novos Usuários em Sistemas de Reconhecimento Facial"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a Mestre degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

This volume corresponds to the version of the Dissertation submitted to examining board by Samuel Botter Martins, under the supervision of Prof. Dr. Alexandre Xavier Falcão.

Este exemplar corresponde à versão da Dissertação apresentada à banca examinadora por Samuel Botter Martins, sob orientação de Prof. Dr. Alexandre Xavier Falcão.

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M366f

Martins, Samuel Botter, 1990-
A fast and robust negative mining approach for user enrollment in face recognition systems / Samuel Botter Martins. – Campinas, SP : [s.n.], 2015.

Orientador: Alexandre Xavier Falcão.
Coorientador: Giovani Chiachia.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Reconhecimento de padrões. 3. Processamento de imagens. 4. Identificação biométrica. I. Falcão, Alexandre Xavier,1966-. II. Chiachia, Giovani,1981-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Uma abordagem eficiente e robusta de mineração de negativos para cadastramento de novos usuários em sistemas de reconhecimento facial
Palavras-chave em inglês:
Machine learning
Pattern recognition
Image processing
Biometric identification
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Alexandre Xavier Falcão [Orientador]
Hélio Pedrini
Rodrigo Fernandes de Mello
Data de defesa: 13-03-2015
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Defesa de Dissertação de Mestrado em Ciência da Computação, apresentada pelo(a) Mestrando(a) **Samuel Botter Martins**, aprovado(a) em **13 de março de 2015**, pela Banca examinadora composta pelos Professores(as) Doutores(as):

Prof(a). Dr(a). Rodrigo Fernandes de Mello
Titular

Prof(a). Dr(a). Hélio Pedrini
Titular

Prof(a). Dr(a). Alexandre Xavier Falcão
Presidente

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

A Fast and Robust Negative Mining Approach for User Enrollment in Face Recognition Systems

Samuel Botter Martins[1]

March 13, 2015

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Alexandre Xavier Falcão (Supervisor/*Orientador*)
- Prof. Dr. Hélio Pedrini
  Institute of Computing - UNICAMP
- Prof. Dr. Rodrigo Fernandes de Mello
  Institute of Mathematics and Computer Science - USP
- Prof. Dr. Anderson Rocha
  Institute of Computing - UNICAMP (Substitute/*Suplente*)
- Prof. Dr. David Menotti Gomes
  Computing Deparment - Federal University of Ouro Preto (Substitute/*Suplente*)

[1]Financial support: Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91 2013–2015

Abstract

Automatic face recognition has attracted considerable attention from the industry and academy due to its wide range of applications, such as video surveillance, access control, online transactions, suspect identification, *etc.* The recent progress in face recognition systems motivates the use of deep learning techniques and user-specific face representation and classification models for unconstrained scenarios, which present considerable variations in pose, face appearance, illumination, *etc.* Automatic face recognition systems make possible to build annotated face datasets through user enrollment. However, as the face datasets grow, it becomes crucial to reduce the number of negative samples used to train user-specific classifiers, due to processing constraints and responsiveness. Such a discriminative learning process during the enrollment of new individuals has implications in the design of face recognition systems. Even though it might increase recognition performance, it may affect the speed of the enrollment, which in turn may affect the user experience. In this scenario, it is important to select the most informative samples in order to maximize the performance of the classifier. This work addresses this problem by proposing a discriminative learning method during user enrollment with the challenges of not negatively affecting the speed and reliability of the process, and so the user experience. Our solution combines high-dimensional representations from deep learning with an algorithm for rapidly mining negative face images from a large mining set to build an effective classification model based on linear support vector machines for each specific user. The negative mining algorithm has shown to be robust in building small and effective training sets with the most informative negative samples for each given individual. We evaluate our approach on two unconstrained datasets, namely PubFig83 and Mobio, and show that it is able to attain superior performance, within interactive response times, as compared to five other baseline approaches that use the same classification scheme. The results indicate that our approach has potential to be exploited by the industry with minimum impact to the user experience. Moreover, the algorithm is application-independent. Hence, it may be a relevant contribution for biometric systems that aim to maintain robustness as the number of users increases.

Resumo

Sistemas automáticos de reconhecimento de faces tem atraído a atenção da indústria e da academia, devido à gama de possíveis aplicações, tais como vigilância, controle de acesso, *etc.* O recente progresso em tais sistemas motiva o uso de técnicas de aprendizado em profundidade e classificadores específicos para cada usuário em cenários de operação não-controlado, que apresentam variações consideráveis em pose, iluminação, *etc.* Sistemas automáticos de reconhecimento de faces possibilitam construir bases de imagens anotadas por meio do processo de cadastramento de novos usuários. Porém, à medida que as bases de dados crescem, torna-se crucial reduzir o número de amostras negativas usadas para treinar classificadores específicos para cada usuário, devido às limitações de processamento e tempo de resposta. Tal processo de aprendizado discriminativo durante o cadastramento de novos indivíduos tem implicações no projeto de sistemas de reconhecimento de faces. Apesar deste processo poder aumentar o desempenho do reconhecimento, ele também pode afetar a velocidade do cadastramento, prejudicando, assim, a experiência do usuário. Neste cenário, é importante selecionar as amostras mais informativas buscando maximizar o desempenho do classificador. Este trabalho resolve tal problema propondo um método de aprendizado discriminativo durante o cadastramento de usuários com o objetivo de não afetar a velocidade e a confiabilidade do processo. Nossa solução combina representações de alta dimensão com um algoritmo que rapidamente minera imagens faciais negativas de um conjunto de minerção grande para assim construir um classificador específico para cada usuário, baseado em máquinas de vetores de suporte. O algoritmo mostrou ser robusto em construir pequenos e eficazes conjuntos de treinamento com as amostras negativas mais informativas para cada indivíduo. Avaliamos nosso método em duas bases contendo imagens de faces obtidas no cenário de operação não-controlado, chamadas PubFig83 e Mobio, e mostramos que nossa abordagem é capaz de alcançar um desempenho superior em tempos interativos, quando comparada com outras cinco abordagens consideradas. Os resultados indicam que o nosso método tem potencial para ser explorado pela indústria com mínimo impacto na experiência do usuário. Além disso, o algoritmo é independente de aplicação, podendo ser uma contribuição relevante para sistemas biométricos que visam manter a robustez à medida que o número de usuários aumenta.

À minha doce e amada avó, Rosalina.

Agradecimentos

Em primeiro lugar, gostaria de agradecer a Deus pela saúde e por abençoar todas as etapas deste trabalho, desde o processo de inscrição até o presente dia.

Sou grandiosamente grato ao meu orientador Prof. Alexandre Falcão pela acolhida, apoio, motivação, orientação e, principalmente, pela amizade adquirida. Fica aqui a minha profunda admiração por você e o desejo de ainda aprender e trabalhar muito contigo durante essa minha pequena e jovial carreira que está se iniciando agora.

De igual maneira, agradeço ao meu amigo e co-orientador Dr. Giovani Chiachia a quem devo grande parte de meu aprendizado neste mestrado. Obrigado pela ajuda sempre presente, os conselhos e por proporcionar um enorme crescimento pessoal e profissional.

Sem o acolhimento da UNICAMP e o fomento da Samsung (No. 8.248/91), este trabalho tampouco seria possível. Obrigado pelo apoio neste projeto.

Aos colegas de meu grupo de pesquisa Nikolas, Renzo, John, David e Thiago pela colaboração, apoio e conversas jogadas ao vento. Agradeço grandemente ao meu amigo Felipe Louza, que foi como um irmão mais velho para mim. Obrigado cara por todos os conselhos e suporte em vários momentos de preocupação.

Aos tantos outros amigos antigos e novos que me motivaram a buscar meus objetivos, em especial ao meu amigo Ederlon pelo companheirismo, preocupação e amizade desde o dia em que o conheci... “bora matar essa matéria cara?”.

Minha família é sem sombra de dúvidas uma das grandes responsáveis, se não a maior, por eu seguir firme neste trabalho e em minha vida. Às minhas lindas irmãs Michele e Mariele e aos meus cunhados Vinícius e Raphael, meu muito obrigado pelo suporte e conselhos. Aos meus pais e grandes hérois Beth e César, que estiveram comigo em todos os momentos, cabeçadas, fracassos e conquistas. À minha doce e amada avó Rosalina, que com seu olhar carinhoso e seu abraço amoroso formaram este simples jovem cheio de sonhos. Com os olhos marejados de lágrimas, fica aqui a minha eterna gratidão.

Por fim, agradeço a minha linda e amada noiva Fabiana por tudo... pelo amor, suporte, colo, ..., e por deixar a minha vida mais feliz a cada dia. Te agradeço meu amor pelos incontáveis dias em que você me apoiou e me ajudou a suportar e superar todas as dificuldades. Em pouco tempo, seremos apenas um. Te amo.

"Just keep me where the light is..."
John Mayer - Gravity

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The face is our primary attention focus in social intercourse, playing a major role in conveying identity and emotions. The earliest work on face recognition can be traced back at least to the 1950s in psychology [1] and to the 1960s in the engineering literature [2]. However, research on automatic machine recognition of faces really started in the 1970s after the seminal work of Kanade [3].

Over the past two decades, face recognition has been an important area of research. Such an effort may be explained by the wide range of applications that require face recognition, such as video surveillance, access control, online transactions, suspect identification, *etc.* As a consequence, many systems and approaches have been proposed for face recognition, and some of them have achieved state-of-the-art performances in specific applications [4, 5, 6].

An important component in typical biometric systems, such as face recognition, is *user enrollment*, which is responsible for capturing appropriate biometric readings of a new user to be enrolled in the system and for storing this data either in raw format or as feature vectors or user models. This process is directly related to the approach used to match biometric samples and ultimately recognize the users. For example, a common matching approach consists of computing pairwise distances from a probe sample to gallery samples. The enrollment process in this case essentially consists of storing valid gallery samples — or their corresponding feature vectors — in the system database for later distance computation. While pairwise matching approaches have been largely used in biometrics, modern face recognition systems usually adopt more sophisticated mechanisms, often relying on learning tasks to extend or replace the approach with models that increase the overall robustness of the system by leveraging ever-growing collections of face images [5, 6, 7].

From a certain perspective, these learning tasks lead to models of two types: User-Independent (UI) and User-Specific (US). UI models do not require access to gallery sam-

ples for their training and therefore can be built offline, even prior to system deployment. Time and memory requirements to learn these models are usually not a matter of concern for the system operation, since the learning task is decoupled from the operation. Principal Component Analysis (PCA) [8] and Linear Discriminant Analysis (LDA) [9] applied on face datasets available at development time are common examples of UI models.

US models incorporate gallery samples into the learning task and are usually built with discriminative techniques executed during user enrollment [5] or at matching time [10, 11]. Time and memory demanded by the learning task in this scenario is critical, since they can affect the system responsiveness, which in turn can adversely impact the user experience. One of such approach is to learn a discriminative binary classifier that assumes the enrolling user as the positive class and a set of face images from unrelated individuals — e.g., from other individuals in a previously curated large face dataset — as the negative class. In this case, pairwise matching is replaced by predicting the class to which a probe sample belongs according to the discriminative US model.

1.1 Challenges and Objectives

The recent progress in automatic face recognition systems [6, 12, 13] motivates the use of high-dimensional feature spaces, user-specific face representation, and classification models for *unconstrained* scenarios, *i.e.*, scenarios where the face images present a large range of the variation in pose, lighting, expression, background, among others.

Face recognition systems make possible to build potentially huge annotated face datasets through user enrollment. For example, an industry of mobiles that provides a face recognition system in its devices may build a face dataset with the images from enrolled users. Nevertheless, as the face dataset grows, it becomes crucial to reduce the number of negative samples used to train user-specific classifiers, due to aspects such as processing constraints and responsiveness. Indeed, a high number of negative samples makes impractical or impossible to use all of them for the training of a user-specific classifier for each user. Such a discriminative learning process during the enrollment of new individuals has implications in the design of face recognition systems. Even though it might increase recognition performance, it may affect the speed of the enrollment, which in turn may affect the user experience. In this scenario, it is important to select the most informative samples in order to maximize the performance of the classifier.

The simplest approach for this problem is to select n samples from the potentially huge negative set at random, where n is the maximum possible number of negative samples for the training of an US model in line with the system limitations. However, there are no guarantees that the most informative negative samples will be selected, so that it is possible to build a classifier with poor performance if uninformative samples are chosen.

On the other hand, Negative Mining (NM) has been extensively used in Computer Vision, especially for object detection [14, 15, 16, 17]. Felzenszwalb et al. [16] present a NM method for object detection systems that uses Support Vector Machines (SVMs) [18] as basis. The method iteratively solves a sequence of training problems using a relatively small number of examples from a large negative set. However, the training set may have a considerable growth in some iterations reaching a high processing time, because it does not constrain the number of negative training samples to a maximum.

Papa et al. [19], in turn, propose a generic training sample mining algorithm that exploits the use of SVMs and Optimum-Path Forest (OPF) classifiers [20]. This algorithm fixes a maximum number of training samples and switches non-prototype training samples by misclassified validation samples of the same label. Its main drawback is that it does not capture the most informative negative examples when the positive class is extremely unbalanced with respect to the negative class, because the classifiers tend to be biased to the negative class, by resulting in a few (or no) classification error(s) of negative samples.

In view of the challenges presented, this work aims to study an effective and efficient negative mining approach that overcomes the limitation presented in [16, 19], in order to select the most informative negative samples for each individual being enrolled in the recognition system, leading to robust discriminative US models without undesirably impacting the user experience.

1.2 Methods and Contributions

We have studied a deep learning visual representation called HT-L3-1st [13], which is based on the use of Convolutional Networks (ConvNets) [21], for processing visual information. HT-L3-1st has the property of outputting high-dimensional features vectors and it has achieved the state-of-the-art performance in challenging face recognition problems [5, 6, 13, 22]. Some variants from HT-L3-1st were also implemented, such as the one that applies supervised techniques to learn filter weights in a given layer of the network [6].

We also conducted a study about some clustering algorithms in order to build an unsupervised negative mining approach. Since these algorithms do not require labeled data, the mining process may be previously executed at development time, by selecting the most general informative negative samples from a curated large face dataset. In this spirit, two approaches were investigated. The first one simply applies a clustering technique on the negative set and then selects the nearest samples from each group center, assuming that these are the most representative samples from their respective clusters. The second approach builds a hierarchical clustering of the negative set. Preliminary experiments presented unsatisfactory results for the addressed problem, which can be explained by the fact that clustering techniques based on neighborhood and proximity can not behave

well for feature vectors with high dimensionality [23]. Thus, it is necessary to use a low-dimensional visual representation, which would result in less effective classifiers for the addressed problem when compared to the HT-L3-1st descriptor. However, we believe that such cluster-based negative mining approaches might be promising for scenarios where the feature vectors have low dimensionality in order to obtain well-behaved clusters.

Notwithstanding the value of the these studies, the key contribution of this work is a new method for rapidly mining negative face images from a large mining set during user enrollment in order to build more effective US models based on linear Support Vector Machines (SVMs) [18]. The algorithm has shown to be fast (a few seconds) and robust in iteratively mining a much smaller and effective subset of negative training samples, according to a criterion based on distances to SVM decision boundaries. Our approach has similarities with [16, 19], however, we propose a new strategy to mine informative negative samples for a given user (positive class) being enrolled in the recognition system.

We evaluate the new approach on two unconstrained datasets, namely PubFig83 [5] and Mobio [24], and conduct an array of experiments by increasingly mining thousands of available images in order to simulate the potentially huge dataset scenario, which is not available. Results show that the proposed approach can attain significantly superior performance with respect to five other baselines, which rely on the same classification scheme, without negatively affecting the user experience. Moreover, given that the approach can be split into client and server tiers — requiring low bandwidth between the tiers — it is also well suited to modern face recognition systems that operate on budgeted devices. This work has been recently submitted to *IEEE Signal Processing Letters* [25].

1.3 Text Organization

In Chapter 2, we present background information on concepts explored in this work. Sections 2.1 and 2.2 summarize the main contributions in the fields of face recognition and negative mining according to the literature. Moreover, in view of providing a deeper explanation of the techniques used in the experiments, the subsequent sections cover, respectively, Convolutional Networks (ConvNets), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) in detail. Chapter 3 depicts the proposed linear SVM-based negative mining approach for user enrollment and previous attempts carried out with the purpose of developing negative mining approaches based on clustering algorithms. All details about the considered face datasets, evaluation protocol, experimental setup, and results are presented in Chapter 4. Finally, a compilation of our contributions and experimental findings, along with new directions to this research line, are presented in Chapter 5.

Chapter 2

Background

This chapter summarizes the main studies and concepts in the fields of face recognition with emphasis in user enrollment and negative mining, as well as briefly presents background information on concepts that are required for understanding the next chapters.

2.1 Face Recognition

Over the past two decades, the problem of automatic face recognition has attracted considerable attention from the industry and academy, and its study has promoted an impressive advance in basic and applied research and applications [26]. Face recognition technology can be used for user authentication, video surveillance, photo camera applications, among others. The earliest work on the topic can be traced back at least to the 1950s in psychology [1], but research on automatic face recognition started in 1973 with the seminal work of Kanade [3].

Since then, dramatic advances have been made in the performance of face recognition algorithms operating on images acquired under relatively controlled conditions [27]. Indeed, under such conditions, automated face recognition can surpass human performance in the task of matching pairs [27]. However, variations in the appearance of a given face due to illumination, viewing conditions, facial expressions, *etc.*, complicate the recognition process [26], and such an *unconstrained* recognition scenario still imposes challenges to state-of-the-art methods in spite of the constant progress in the development of robust face recognition systems capable to operate under such conditions [5, 6, 10, 28, 29].

The typical operation modes of a face recognition system are presented in Section 2.1.1. General considerations about the process of face recognition are discussed in Section 2.1.2.

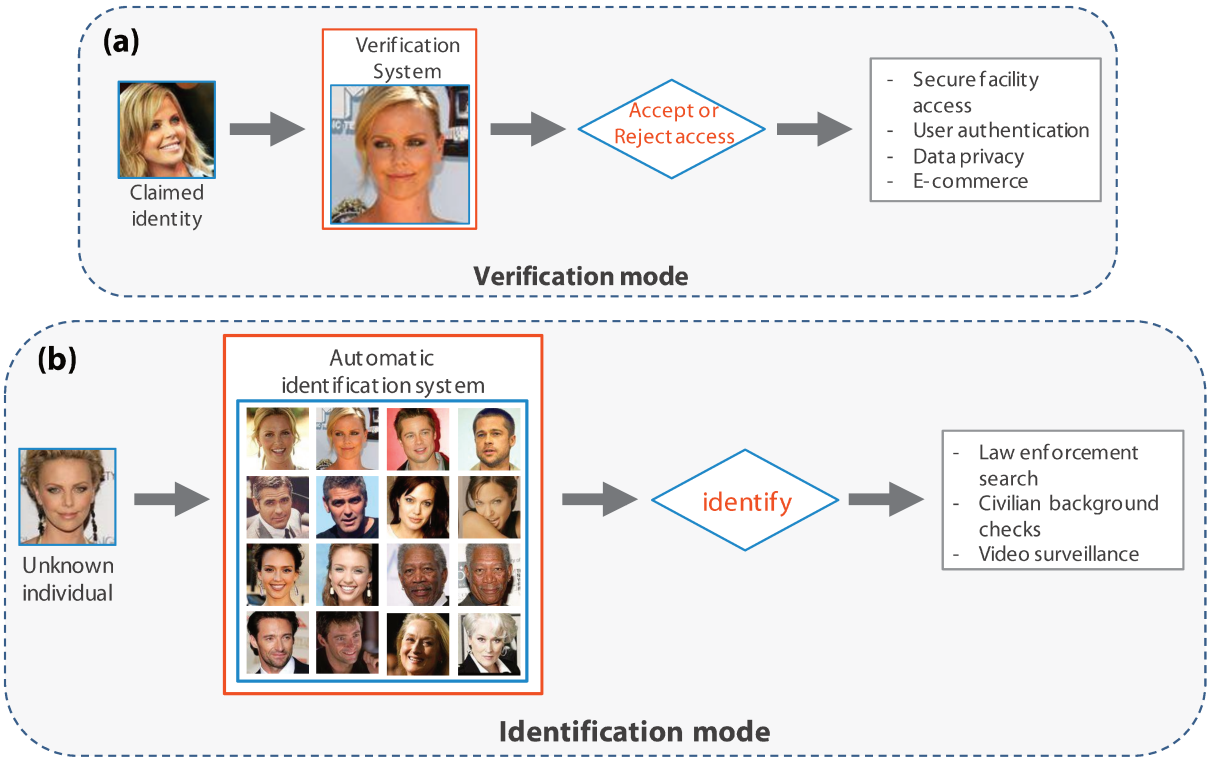


Figure 2.1: This schematic diagram shows the structure of two face recognition operation modes. (a) The verification task determines if the person pictured in a face image matches a claimed identity. (b) The identification task determines the person's identity of a probe image. This figure was inspired in [26], and their face images were extracted from the PubFig83 dataset [5].

2.1.1 Modes of Operation

Face recognition systems basically works either on verification or identification modes. In verification mode, the basic figurative question asked from the user perspective is: *"Can the system confirm that I am who I say to be?"*. The system validates the identity of a person by comparing the captured image with her own record(s) stored in the system's database. In such a system, an individual who desires to be recognized claims an identity and the system typically conducts a one-to-one comparison to determine whether the person really is who he/she claims to be. The verification task is aimed at applications requiring user interaction in the form of an identity claim, such as security facility access and user authentication. Figure 2.1(a) illustrates the concept.

In identification mode, the question in turn is: *"Can the system say who I am?"*. The system recognizes an individual by searching the records of all users in the database for a match. Hence, one-to-many comparisons are conducted by the system trying to determine

the individual's identity without the subject having to claim for it. The identification task is mostly aimed at applications not requiring user interaction, such as surveillance applications (Figure 2.1(b)).

We call *gallery* a set of enrolled images from known individuals that may be used, for example, to build models or to do pair matching. A *probe* image, in turn, is an image submitted to the system for comparison with the gallery.

Both operation modes can be further split into *closed-set* and *open-set* recognition scenarios. The first one considers that probe faces will always be of somebody that already belongs to the gallery. On the other hand, in the *open-set* scenario, probe faces may be of individuals not previously enrolled in the system and therefore not belonging to the gallery. In this work, experiments are carried out in the *open-set* scenario and results are reported by assuming that the system is operating in verification mode, even though the proposed approach naturally extends to operates in the *closed-set* scenario and identification mode.

2.1.2 The Process of Face Recognition

According to Li and Jain [4], face recognition systems usually involve four steps, as depicted in Figure 2.2(a): face detection (localization), face preprocessing (face alignment/normalization, light correction, and *etc.*), feature extraction, and matching.

Face detection segments the face areas from the background by coarsely estimating its location and scale in a given scene. In the case of videos, detected faces may also need to be tracked using a face tracking component [30].

The aim of the *face preprocessing* step is to refine the location and to normalize the faces provided by the face detection, so that a robust feature extraction can be achieved. Depending on the application, face preprocessing includes alignment (translation, rotation, scaling) and light normalization/correlation [4].

Subsequent to the preprocessing step, *feature extraction* is performed on the stable face image to derive effective information that is useful for distinguishing among faces of different individuals. Eigenfaces [31], Fisherfaces [32], and Local Binary Patterns (LBP) [33] are well known facial feature extraction methods.

Feature matching is the ultimate step of the recognition process. The feature vector obtained from feature extraction is matched to classes (individuals) of facial images already enrolled in the database. Matching algorithms vary from fairly obvious Nearest Neighbor classifiers to advanced classification schemes like Neural Networks and Support Vector Machines (SVMs) [18].

For the sake of clarity, we present a more detailed face recognition system architecture in Figure 2.2(b), which is based on Figure 2.2(a) and is essentially divided into two stages:

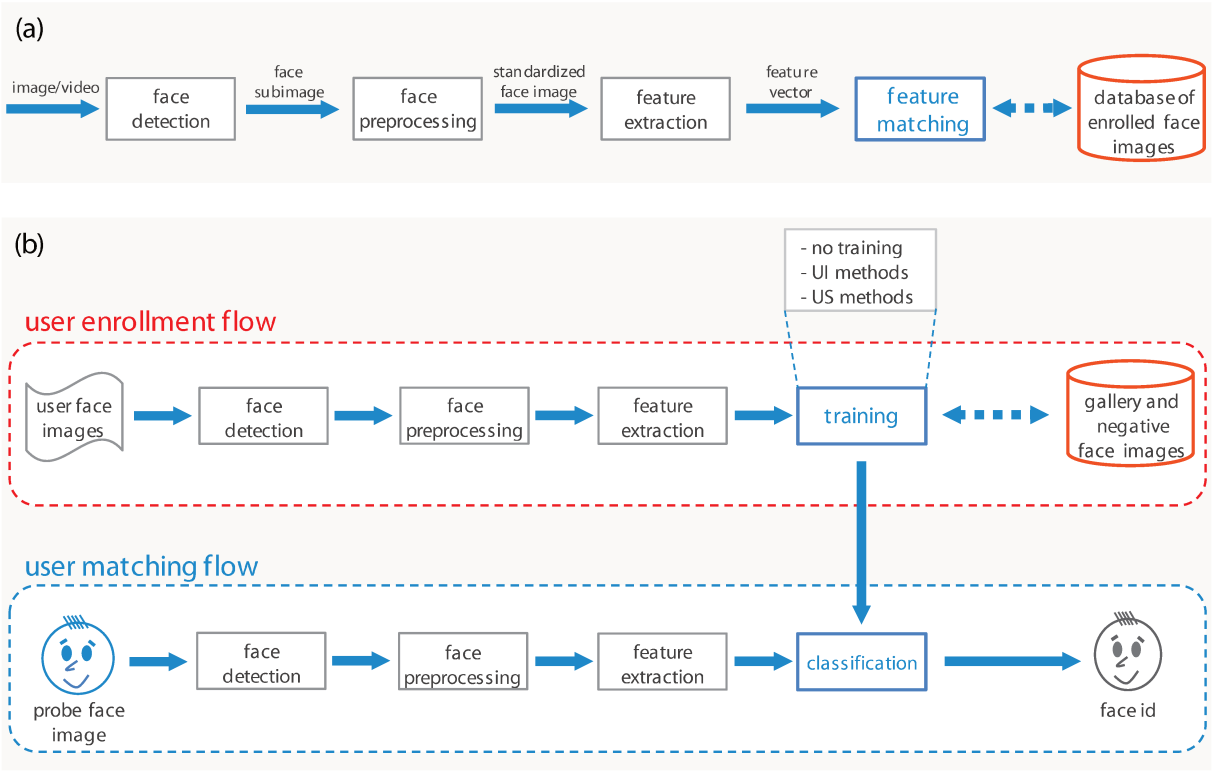


Figure 2.2: (a) A typical face recognition system architecture proposed by Li and Jain [4]. (b) An expansion from (a) that divides the face recognition process into user enrollment (top panel) and user matching (bottom panel) stages. The first consists of all tasks from the image/video capture of the user to be enrolled until the storage of the resulting process output, whereas the latter aims to classify an user probe face image according to the chosen operation mode and classification criterion.

"user enrollment" and "user matching". In both stages, the first three tasks are exactly the same as in Figure 2.2(a).

User enrollment consists of tasks for capturing appropriate biometric readings of a new user to be enrolled in the system and for storing this data either in raw format or as feature vectors or user models of two types: User-Independent (UI) and User-Specific (US). User matching, in turn, aims to classify an user probe image according to the chosen operation mode and the learned classifier.

Essentially, UI models do not require access to gallery samples for their training and therefore can be built offline at development time. Time and memory requirements to learn these models are usually not a matter of concern for the system operation, since the learning task is decoupled from the operation. Common examples of UI models are Eigenfaces [31] and Fisherfaces [32] applied on negative face datasets available at development time. In these cases, feature vectors extracted at enrollment time are projected onto the

previously learned subspace and the resulting projections are stored in the database for subsequent matching.

On the other hand, US models incorporate gallery samples into the learning task and are usually built with discriminative techniques executed during user enrollment. One of such approach is to learn a discriminative binary classifier that assumes the enrolling user as the positive class and a set of face images from other individuals in a previously curated large face dataset as the negative class. Time and memory demanded by the learning task in this scenario is critical, since they can affect the system responsiveness, which in turn can adversely impact the user experience. In this case, plain feature matching is replaced by predicting the class to which a probe sample belongs according to the discriminative classifiers. US models can also be learned at matching time, as proposed in [10, 34]. In Figure 2.3, we present some examples of face recognition systems that are categorized with respect to their user enrollment processes and training strategies.

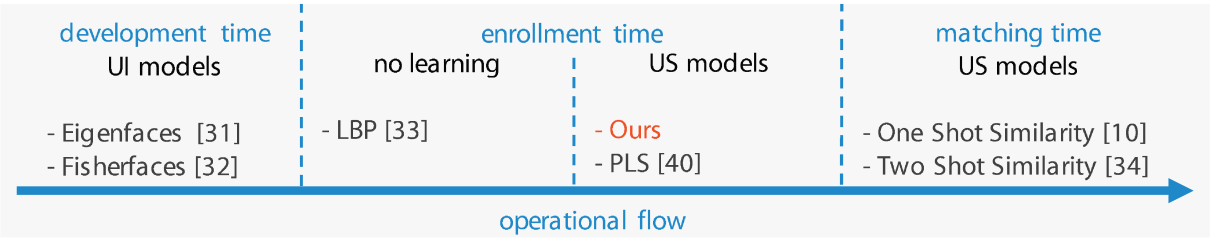


Figure 2.3: Examples of face recognition systems categorized according to their user enrollment processes and training strategies.

The vast majority of face recognition systems operate with UI models, previously built without regarding particularities in the appearance of the individuals to be recognized [12, 35, 36]. While such strategy may avoid the burden of using complicated learning tasks in the operational scenario — and it is well aligned with the evaluation protocol of a number of public face recognition benchmarks [11, 37, 38] — it completely disregards the opportunity of leveraging gallery samples to build better models and improve the overall system performance. In this spirit, several works have proposed robust face recognition systems based on US models that operate in *open-* and *closed-set* scenarios [5, 6, 22, 39, 40]. However, works based on US models are usually targeted at recognition performance and often employ time and memory demanding procedures to build them. Therefore, they do not assume user enrollment as a fundamental time constrained process in user interactive face recognition systems, and may be impractical for real applications.

In this work, we present a linear SVM-based negative mining approach for US model building at enrollment time. Linear SVMs present good behavior to high-dimensional data, which is the focus of the work. Our approach selects informative negative samples from the database with respect to a given user being enrolled in the system in order to

improve the system performance under memory and time constraints.

2.2 Negative Mining

Binary classification is a fundamental task in many data analysis applications such as human detection in video [41], person identity verification in biometry [6, 5], and tumor malignancy indication in medical diagnosis [42, 43]. In such applications, the number of negative samples is usually much greater than the number of positive ones, so that an extremely unbalanced training set usually makes the classifier to present a poor performance on positive test samples. For example, when designing a mobile face recognition system, the industry (designer) may have an external huge negative dataset, which consists of face images from millions of users, resulting in an important knowledge that can be used to build robust classifiers. The high number of negative samples makes impractical or impossible to use all of them for training the binary classifier for each user, due to aspects as processing constraints and responsiveness.

In order to compose an effective training set, which might still contain a reasonable difference between the number of negative and positive samples, the main idea is to mine the most informative negative samples. This section describes the current state of the problem in face recognition and the main negative mining techniques that could be explored in this context.

2.2.1 State of the Problem in Face Recognition

Face recognition has become an important technological development topic in the industry of ATMs, mobile devices, TVs, *etc.* Considering a person identity verification system based on cloud services, for example, one can imagine the huge face image set in the cloud that results from the enrollment of a myriad of users. In order to build a robust binary classifier (gallery model) for a given user, one can easily avoid the face images of this user (positive samples) in the negative set. The most informative negative samples are very likely the face images similar to those of the user. However, their identification in a huge data set is the main challenge. As far as we know, this problem seems to not have caught any attention in the literature of face recognition.

The fastest and simplest strategy is to randomly select a given number of negative samples for the training set. However, what is the ideal number of negative samples? How lucky is it to get the most informative ones for that particular user? It should be clear that this strategy is efficient, but it might not be effective. We are interested in the most effective and efficient negative mining (NM) approach that answers both questions.

To the best of our knowledge, our work is the first one to propose negative mining for

user-specific gallery model building at enrollment time. Perhaps the most related work to ours is [40], where the authors propose the use of Partial Least Squares (PLS) [44] to build US models. Nevertheless, our work differs from [40] in at least two fundamental ways (see Section 3). First, we mine negative *samples* instead of negative *individuals*. Second, and more importantly, we do not build US models against gallery samples in a *closed-set* scenario. Instead, we rely on a previously curated large face dataset to mine negative samples to build the models. This not only avoids the burden of gallery maintenance, which is the focus of [40], but it is also more realistic, since it is aligned with face recognition in the open-set scenario.

2.2.2 Techniques and Strategies

In spite of not being well explored in the enrollment process of biometric systems, Negative Mining (NM) has drawn the attention of researchers in the computer vision literature [14, 15, 16, 17, 19, 45, 46] due to the need for treatment of huge negative sets.

Essentially, a common NM approach consists of two steps. First, a binary classifier is trained using the positive samples and an initial random subset of negative samples. The second step is inspired on the *bootstrapping* procedure [47], and consists of *mining* negative samples by giving more importance to the "hard" ones — *i.e.*, the incorrectly classified negative examples — thereby improving the training set. A new classifier is then trained and this procedure may be repeated a few times. Dalal and Trigs [14], for example, use only one mining step by adding all the false positives that are found. Dolar et al. [45] use 2 of them, while other works use more [19, 46, 48].

Felzenszwalb et al. [16] present a general negative mining method for object detection systems that uses classical SVMs and latent SVMs. The method iteratively solves a sequence of training problems using a relatively small number of hard examples from a large training set. The innovation of this approach is a theoretical guarantee that it leads to the *exact* solution of the training problem defined by the large training set. This requires a margin-sensitive definition of hard examples. Additional details about the method, with theorems and their proofs, are presented in [15].

The authors define the hard instances relative to a model β as the examples that are incorrectly classified or inside the margin of the classifier. Similarly, the easy samples are the samples that are correctly classified and outside the margin. Examples on the margin (support vectors) are neither hard nor easy. Figure 2.4 presents a block diagram with the algorithm steps for the case of keeping all positive examples in the training set and mining the negatives. The numbers in the figure represent the algorithm steps and are detailed below.

The method starts building an initial *training set* Z_1, with randomly selected negative

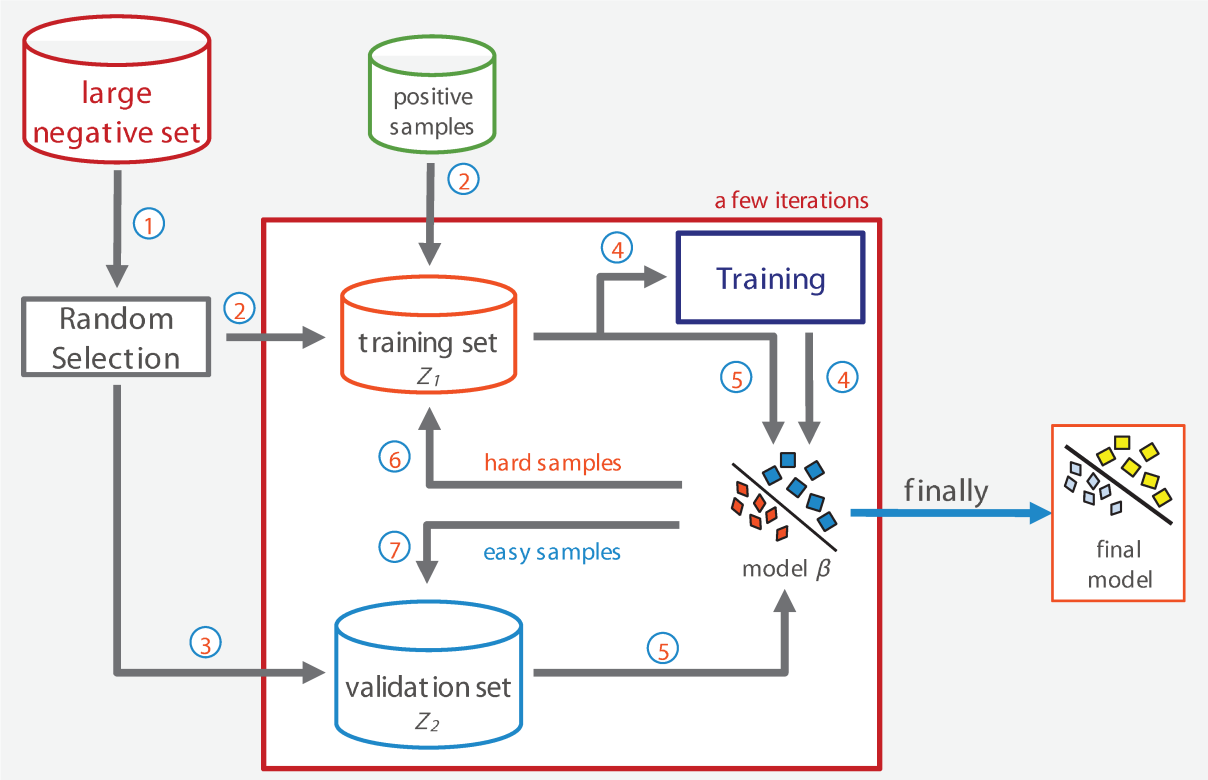


Figure 2.4: Block diagram from the NM approach proposed by Felzenszwalb et al. [16]. Initially, a training set Z_1 and a validation set Z_2 are built. In each iteration, the method removes easy training negative samples from Z_1 and adds all hard examples from Z_2 to Z_1. Similarly, all easy negative samples from Z_1 are added to Z_2. The method stops when there are no hard examples in Z_2.

samples and all the positive examples, and an initial *validation set* Z_2, with the remaining negative samples from the large negative set (steps 1–3). In each iteration, the algorithm trains a model β by using the training set Z_1 (step 4), and finds the hard and easy samples from Z_1 and Z_2 (step 5). If there are no hard examples in Z_2, then the method stops and returns β. The easy training negative samples are removed from Z_1 and all hard instances from Z_2 are added to Z_1 (step 6). Similarly, all easy negative samples from Z_1 are added to Z_2 (step 7).

The shortcoming of this approach is that the training set may have a considerable growth in some iterations, reaching a high processing time (see Section 4.4), because it does not fix a maximum number of training negative samples. Thus, this method is impractical for the problem addressed in this work.

Papa et al. [19] proposed a generic training sample mining algorithm also inspired in the bootstrapping idea, by exploiting the use of SVM and Optimum-Path Forest

(OPF) [20] classifiers. This algorithm is slightly different from the previous approach, because it considers a fixed number of training samples. Moreover, this method is able to be applied in multi-class problems. A pseudocode for the approach can be defined as in Algorithm 1.

Algorithm 1 Learning Algorithm

Input: *Large dataset Z labeled by λ, maximum number of training samples c, and number n of iterations.*
Output: *The best OPF/SVM classifier.*
Auxiliary: *Number of classes g, false positive and false negative arrays, FP and FN, of sizes g, list LM of misclassified samples, and variables α, and Acc.*

```
1.  Z_1 ← rand_selection(Z, c)
2.  Z_2 ← Z \ Z_1
3.  For i ← 1 to n do
4.      LM ← ∅
5.      Train OPF/SVM with Z_1
6.      For j ← 1 to g do
7.          FP(j) ← 0 and FN(j) ← 0
8.      For each sample t ∈ Z_2 do
9.          Use the classifier obtained in Line 5 to classify t, resulting in the label α
10.         If α ≠ λ(t) then
11.             FP(α) ← FP(α) + 1
12.             FN(λ(t)) ← FN(λ(t)) + 1
13.             LM ← LM ∪ t
14.     Compute Acc by Equation 2.3 and save the current instance of the classifier and
15.     its accuracy
16.     While LM ≠ ∅ do
17.         Remove t from LM
18.         Replace t by a random non-prototype sample of the same class in Z_1
19. Return the instance of the classifier with highest system accuracy
```

Initially, a large dataset Z is splitted into a training set Z_1 and a validation set Z_2 with $|Z_1|$ and $|Z_2|$ samples, respectively (Lines 1–2). The function $rand\_selection(N, c)$ selects c randomly samples from the dataset Z with a same percentage of samples per class. The idea is to use the validation set Z_2 to improve the sample composition in Z_1 without increasing its size, *i.e.*, $|Z_1| = c$.

In each iteration, a classifier is learned using Z_1, the validation examples are classified, and the system accuracy is computed (Lines 4–14). The accuracy of each classifier is measured by taking into account that the classes may have different sizes in Z_2. Let

$Z_2(i)$, $i = 1, 2, ..., g$, be the set of samples in Z_2 from each class i, then

$$e_{i,1} = \frac{FP(i)}{|Z_2| - |Z_2(i)|} \quad \text{and} \quad e_{i,2} = \frac{FN(i)}{|Z_2(i)|}, i = 1, ..., g \tag{2.1}$$

where $FP(i)$ is the number of samples from other classes that were classified as being from the class i in Z_2 (false positives), and $FN(i)$ is the number of samples from the class i that were incorrectly classified as being from other classes in Z_2 (false negatives). The errors $e_{i,1}$ and $e_{i,2}$ are used to define

$$E(i) = e_{i,1} + e_{i,2}, \tag{2.2}$$

where $E(i)$ is the partial sum error of class i. Finally, the accuracy Acc of the classification is written as

$$Acc = \frac{2g - \sum_{i=1}^{g} E(i)}{2g} = 1 - \frac{\sum_{i=1}^{g} E(i)}{2g}. \tag{2.3}$$

The loop in Lines 16–18 exchanges the misclassified samples from Z_2 for random non-prototype training samples. Particularly, the support vectors are the prototypes in a SVM model. The best OPF/SVM classifier is the one with highest accuracy along the n iterations.

The main drawback of this method is that it does not capture the most informative negative examples when the positive class is extremely unbalanced with respect to the negative class, which is a typical scenario, for example, in modern face recognition systems. This is because the classifiers tend to be biased to the negative class, resulting in a few (or no) classification error(s) of negative samples.

The negative mining approach proposed in this dissertation overcomes the problems of both methods presented in this review with a new strategy to mine relevant negative samples for a given user (positive class) being enrolled in the recognition system.

2.3 Convolutional Networks

Deep Learning (DL) has caught a lot of attention recently due to breakthrough results in a number of important vision problems [6, 49, 50, 51]. Deep Learning techniques enable to learn multi-layered data representations for categorization — therefore the term deep — directly from a labeled training dataset, without requiring descriptor specifications from an expert in the application domain.

In this work, we are particularly interested in the visual representation called HT-L3-1st [13], which is based on the use of Convolutional Networks (ConvNets) [21] for processing visual information.

Figure 2.5(a) shows a schematic diagram of the HT-L3-1st model, which basically consists of a deep representation through the concatenation of three feed-forward layers. At the end of the process, a linear Support Vector Machine (SVM) is learned with the resulting feature vectors. As shown in Figure 2.5(b), each layer executes four fundamental operations in the order that they appear, as discussed below. The set of all hyperparameters required in the operations is called network architecture.

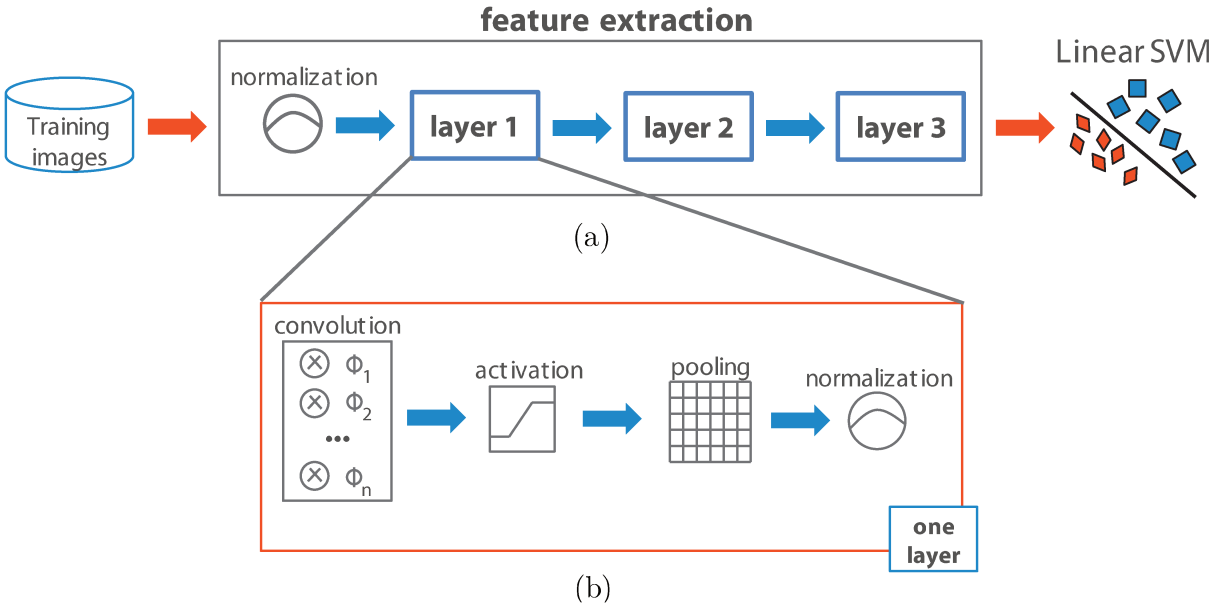


Figure 2.5: A schematic diagram of the HT-L3-1st model, which basically consists of a deep representation through the combination of three feed-forward layers (a), so that each layer executes four fundamental operations in the input image $\hat{I}$, following the order shown in (b). The symbols shown in (a) and (b) correspond to the hyperparameters of the network, which are detailed in Section 2.3.1. This figure was adapted from [13].

2.3.1 Fundamental Operations

This section presents the fundamental operations of a single-layered ConvNet that can be viewed as linear and non-linear image processing operations. When stacked, these operations essentially extract higher level representations, named *multiband images*, whose pixel attributes are concatenated into high-dimensional feature vectors for pattern categorization (recognition). In this dissertation, ConvNets are described from an image processing perspective, with terms like image *domain*, image *band*, *etc.*

Let $\hat{I} = (D_I, \vec{I})$ be a multiband image, where $D_I \subset Z^2$ is the image domain and $\vec{I}(p) = \{I_1(p), I_2(p), \ldots, I_m(p)\}$ is the attribute vector of a m-band pixel $p = (x_p, y_p) \in D_I$. The fundamental operations are described as follows.

Filter Bank Convolution

Initially, let $\mathcal{A}(p)$ be a squared region centered at p with size $L_{\mathcal{A}} \times L_{\mathcal{A}}$, such that $\mathcal{A} \subset D_I \times D_I$ and $q \in \mathcal{A}(p)$ if $\max(|x_q - x_p|, |y_q - y_p|) \leq (L_{\mathcal{A}} - 1)/2$. Let $\Phi_i = (\mathcal{A}, \vec{W}_i)$ be a filter with weights $w_{i,j}(q)$ associated with pixels $q \in \mathcal{A}(p)$. We represent the weights of multiband filters as vectors $\vec{W}_i(q) = \{w_{i,1}(q), w_{i,2}(q), \ldots, w_{i,m}(q)\}$ for each filter i of the bank. A multiband filter bank $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_n\}$ is a set of filters $\Phi_i = (\mathcal{A}, \vec{W}_i)$, with $i = \{1, 2, \ldots, n\}$.

The weights of a filter Φ_i are randomly generated from a uniform distribution, and normalized to zero mean and unit norm, in order to ensure that they are spread over the unit sphere. In the DL literature, this weights are usually learned by backpropagation for a given architecture. Since we are learning the architecture, we decided to take the proposed approach in [13], which estimates a random orthonormal basis of weight vectors.

The convolution between an input image $\hat{I}$ and a filter Φ_i produces a band i of the filtered image $\hat{J} = (D_J, \vec{J})$, where $D_J \subset D_I$ and $\vec{J} = (J_1, J_2, \ldots, J_n)$, such that for each $p \in D_J$,

$$J_i(p) = \sum_{\forall q \in \mathcal{A}(p)} \vec{I}(q) \cdot \vec{W}_i(q), \tag{2.4}$$

where $\cdot$ represents the inner product.

In Figure 2.6, we show an illustration of the convolution between a hypothetical m-band input image and a multiband filter bank with n filters, so that each one also has m bands. The resulting filtered image has n bands. The convolution may be interpreted as a projection of the input image in the direction given by $\vec{W}_i(q)$.

Activation

The activation operation considered in our networks is used in many state-of-the-art ConvNet architectures [13, 50] and simply creates an image $\hat{J}' = (D_J, \vec{J}')$ by

$$J'_i(p) = \max(J_i(p), 0), \tag{2.5}$$

where $p \in D_J$ are pixels of the image, and $i = \{1, 2, \ldots, n\}$ are the image bands.

In spite of its simplicity, this activation function plays an important role in the network information flow, specially when coupled with random filters initialized as described previously. The combination of random filters with zero mean and unit norm, and this activation rule aims to output a sparse code in order to improve the overall robustness of the features being extracted [52].

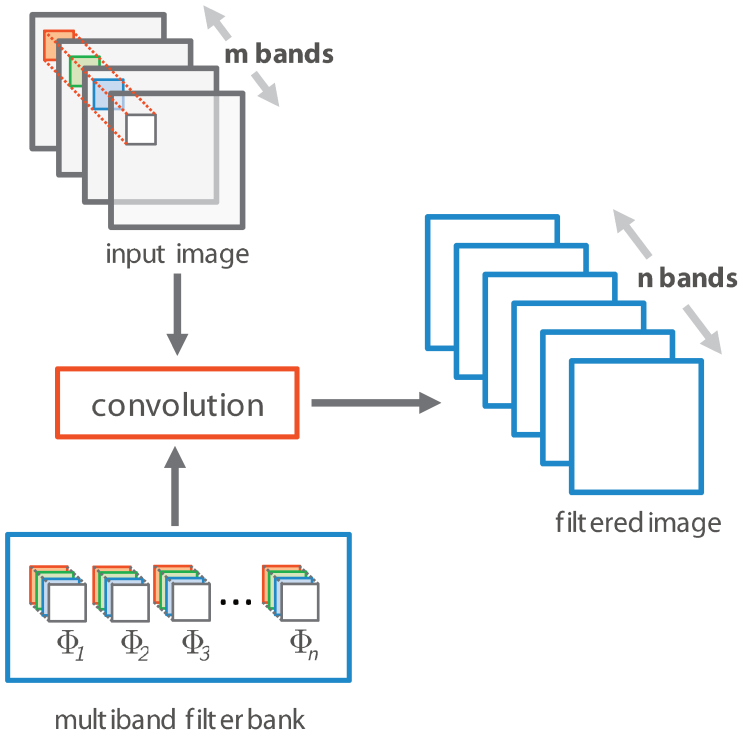


Figure 2.6: Illustration of the convolution operation in our ConvNet. A filtered image is obtained by convolving the m-band input image with the m-band filter bank of n filters. The filter weights from each band are multiplied by the corresponding band in the input image (inner product). For a better understanding of the operation, we have assigned a different color for each filter band and highlighted a specific region in the input image that follows the same sequence of colors.

Spatial Pooling

Spatial pooling is an extremely important operation in the ConvNet literature [21], since it aims at bringing translational invariance to the features by aggregating activations from the same filter in a given region [52].

Let $\mathcal{B}(p)$ be a pooling region of size $L_\mathcal{B} \times L_\mathcal{B}$ centered at pixel p and $D_K = D_J/s$ be a regular subsampling of every s pixels p in D_J. We call s the *stride* of the pooling operation. For example, given that $D_J \subset Z^2$, if $s = 3$, $|D_K| = |D_J|/9$. The pooling operation results in the image $\hat{K} = (D_K, \vec{K})$, which is defined as

$$K_i(p) = \sqrt[\alpha]{\sum_{\forall q \in \mathcal{B}(p)} J_i'(q)^\alpha}, \tag{2.6}$$

where $p \in D_K$ are pixels in the new image, $i = \{1, 2, \ldots, n\}$ are the image bands, and α is a hyperparameter that controls the sensitivity of the operation. We can see this pooling

operation as a L_α-norm of values in $\mathcal{B}(p)$.

Figure 2.7 shows a toy example of how the pooling operation is done over 3x3 pooling regions of an image band pixel centers represented by blue circles. We are showing a stride of 3, so that there is no overlapping between regions.

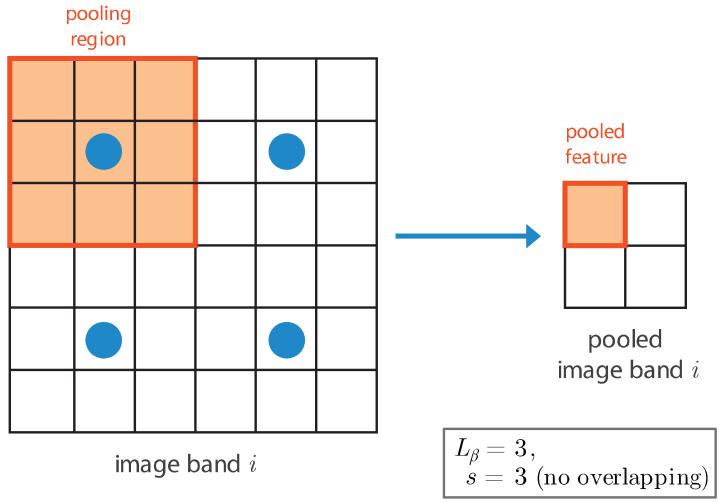


Figure 2.7: A toy example of a pooling operation over 3x3 pooling regions of an image band with pixel centers represented by blue circles. We considered a stride of 3, so that there is no overlapping between regions.

Divisive Normalization

The last operation of each layer from the considered network is the divisive normalization, a mechanism widely used in top-performing ConvNets [13, 50], which is based on gain control mechanisms found in cortical neurons [53]. The divisive normalization is also the first operation in the feature extraction process, as shown in Figure 2.3(a).

This operation is defined within a squared region $\mathcal{C}(p)$ of size $L_\mathcal{C} \times L_\mathcal{C}$ centered at pixel p, such that

$$Y_i(p) = \frac{K_i(p)}{\sqrt{\sum_{j=1}^{n} \sum_{\forall q \in \mathcal{C}(p)} K_j(q) K_j(q)}} \tag{2.7}$$

for each pixel $p \in D_Y \subset D_K$ of the resulting image $\hat{Y} = (D_Y, \vec{Y})$.

Divisive normalization promotes competition among pooled filter bands, such that high responses will prevail even more over low ones, further strengthening the robustness of the ConvNet output feature vector $\vec{Y}$ [52].

Note that, $D_I \subset D_J \subset D_Y \subset D_K$ implies that after each layer the image domain reduces, but usually the number of bands increases. Such reduction in the image domain

happens not only due to the stride parameter, but also because we do not consider regions in which the adjacency window is not entirely inside the image domain.

2.3.2 Summary of all ConvNet Hyperparameters

As shown in Figure 2.5(b), a single layer in our networks consists of four previously presented operations in a total of six hyperparameters detailed as follows:

- $L_{\mathcal{A}}$ filter size;
- n number of filters;
- $L_{\mathcal{B}}$ pooling size;
- s pooling stride;
- α pooling sensitivity;
- $L_{\mathcal{C}}$ normalization size;

Additionally, a ConvNet performs an input normalization prior to processing of the first layer, which requires one more hyperparameter: the input normalization size $L_{\mathcal{C}in}$.

Therefore, our three-layered ConvNet has a total of 19 hyperparameters, determining its architecture and behavior.

2.4 Principal Component Analysis

Principal Component Analysis (PCA) is a technique which is widely used to reduce the dimensionality or the noise in a dataset, while retaining the most variance, by finding patterns within it [54]. The origin of PCA can be traced back to Pearson [8] in 1901, but the modern instantiation was formed by Hotelling [55].

PCA computes a set of new orthogonal variables with the decreasing variances within the dataset, producing principal components. The first principal component is the linear combination of the original dimensions that has the maximum *variance*. Hence, the n^{th} principal component is the linear combination with the highest variance subjected to being *orthogonal* to the $n-1$ first principal components.

2.4.1 PCA Transformation

As mentioned before, PCA is mostly used as a dimensionality reduction technique based on the extraction of interesting information from multidimensional data. Specifically, PCA attempts to find a new representation of the original set by constructing a set of orthogonal vectors — the principal components — spanning a subspace of the initial space [54].

These principal components, or basis vectors in the transformed space, can be calculated as follows [56]. Let X be the $N \times M$ data matrix, so that the columns $x_1, ..., x_M$ are observations of a signal embedded in $\mathbb{R}^N$. The PCA basis Φ is obtained by solving the eigenvalue problem

$$\Lambda = \Phi^T \Sigma \Phi \tag{2.8}$$

where Σ is the covariance matrix of the data,

$$\Sigma = \frac{1}{M} \sum_{i=1}^{M} x_i x_i^T \tag{2.9}$$

$\Phi = [\phi_1 ..., \phi_m]^T$ is the eigenvector matrix of Σ, and Λ is the diagonal matrix with eigenvalues $\lambda_1 \geq ... \geq \lambda_N$ of Σ on its main diagonal. In this manner, ϕ_j is the eigenvector corresponding to the j^{th} largest eigenvalue, being λ_j also the variance of the data projected on it.

Therefore, to extract k principal components of the data, one must project the data onto Φ_k - the first k columns of the PCA basis Φ, which correspond to the k highest eigenvalues of Σ. This can be seen as a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^k$ that retains the maximum energy (variance) of the signal. Another important property of PCA is that it *decorrelates* the data with the covariance matrix of $\Phi_k^T X$ always being diagonal. This comes from the orthogonality of the principal components Φ previously mentioned [56].

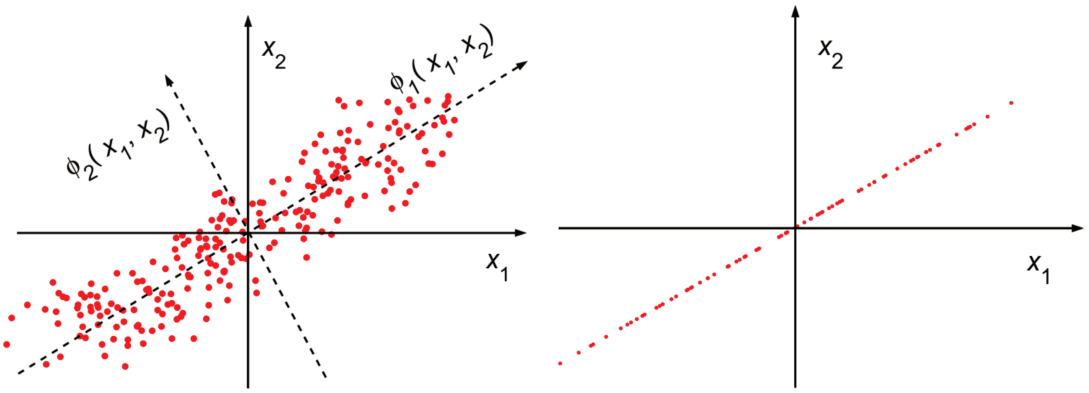
Figure 2.8 shows an example of the PCA transformation in a two-dimensional dataset. The axis labeled ϕ_1 in (a) corresponds to the direction of maximum variance and it is chosen as the first principal component. The second principal component is the remaining perpendicular axis ϕ_2. In a higher-dimensional space, in turn, the selection process would continue dictated by the variances of the projections.

Figure 2.8(b) shows how the original data is expressed only with the first principal component. Even being the most discriminant way for a one-dimensional projection of the dataset, it is possible to note some loss of information.

When the data matrix X is small, PCA is not very expensive to calculate. Nevertheless, as X grows, the computation of Σ (Equation 2.9) becomes quite expensive. Fortunately, PCA may be implemented via an iterative method called Singular Value Decomposition (SVD) [56]. The SVD of an $N \times M$ matrix $X (N \geq M)$ is given by

$$X = UDV^T \tag{2.10}$$

where the $N \times M$ matrix U and the $M \times M$ matrix V have orthonormal columns, and the $M \times M$ matrix D has the square root of the eigenvalues of XX^T on its diagonal entries, the singular values of X.



(a) PCA Basis (b) PCA reduction to 1D

Figure 2.8: The outline of PCA. (a) The solid axis represent the original basis, while the dashed axis represents the PCA basis. (b) The projection of the data using only the first principal component [56].

It can be shown that $U = \Phi$, so SVD allows efficient and robust computation of PCA without the need to estimate the data covariance matrix. When the number of examples M is much smaller than the dimension N, this is a definitive advantage.

2.5 Linear Discriminant Analysis

Originally developed in 1936 by Fisher [9], Linear Discriminant Analysis (LDA) [1] is a well-known technique that has been used successfully in many statistical pattern recognition problems [57, 58, 59, 60]. LDA attempts to project all the data points into new space, normally of lower dimension, which maximizes the *between-class* separability while minimizing the *within-class* variability. This technique is commonly used for dimensionality reduction in the pre-processing step or as a linear classifier.

2.5.1 LDA Transformation

Initially, let Φ be a projection matrix with linearly independent columns, such that each one is a projection basis vector.

LDA starts finding the *between-class* and *within-class* scatter matrices. Let g be the number of classes (sample groups), and $x_{i,j}$ the *jth* sample from the class i. Each sample

[1] A considerable part of this Section was based on the lectures of the course "Intelligent Data Analysis and Probabilistic Inference" from the Department of Computing – Imperial College, London: `http://www.doc.ic.ac.uk/~dfg/ProbabilisticInference/Bayesian.html`

group i has a class mean $\bar{x}_i$, which is defined

$$\bar{x}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}, \tag{2.11}$$

where N_i is the number of examples in class i. Let Σ_i be covariance matrix from the class i, and $\bar{x}$ the grand mean for the whole data set, such that

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T, \tag{2.12}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{g} N_i \bar{x}_i = \frac{1}{N} \sum_{i=1}^{g} \sum_{j=1}^{N_i} x_{i,j}, \tag{2.13}$$

where N is the total number of samples of all classes. The *between-class* and *within-class* scatter matrix are defined as follows

$$S_b = \sum_{i=1}^{g} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T, \tag{2.14}$$

$$S_w = \sum_{i=1}^{g} (N_i - 1)\Sigma_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T. \tag{2.15}$$

The S_w matrix is computed by pooling the estimates of the covariance matrices of each class. Since each Σ_i has rank $N_i - 1$, its rank can be at most $N - g$.

The main objective of LDA is to find a projection matrix Φ_{lda} that maximizes the determinant ratio of S_b to the determinant of S_w (Fisher's criterion [9]), that is

$$\Phi_{lda} = \arg\max_{\Phi} \frac{\left|\Phi^T S_b \Phi\right|}{\left|\Phi^T S_w \Phi\right|}. \tag{2.16}$$

Fisher's criterion tries to find the projection that maximizes the variance of the class means and minimizes the variance of the individual classes.

It has been shown that Φ_{lda} is in fact the solution of the following eigensystem problem:

$$S_b \Phi - S_w \Phi \Lambda = 0. \tag{2.17}$$

Multiplying by the inverse of S_w:

$$S_w^{-1} S_b \Phi - S_w^{-1} S_w \Phi \Lambda = 0 \tag{2.18}$$

$$S_w^{-1} S_b \Phi - \Phi \Lambda = 0 \tag{2.19}$$

$$S_w^{-1} S_b \Phi = \Phi \Lambda \tag{2.20}$$

Thus, if S_w is a non-singular matrix, and can be inverted, then the Fisher's criterion is maximized when the projection matrix Φ_{lda} is composed of the eigenvectors of:

$$S_w^{-1} S_b \tag{2.21}$$

There will be at most $g-1$ eigenvectors with non-zero real corresponding eigenvalues, because there are only g points to estimate S_b.

2.5.2 LDA versus PCA

LDA is closely related to PCA because both look for linear combinations of variables which best explain the data [61], and are commonly used for dimensionality reduction.

PCA is an *unsupervised* method, since it does not take into account the class label from the samples to compute the directions — principal components — that maximize the variance in a dataset (see Section 2.4). In contrast, LDA is a *supervised* technique that finds a linear combination of features — linear discriminants — that best separates samples of distinct classes.

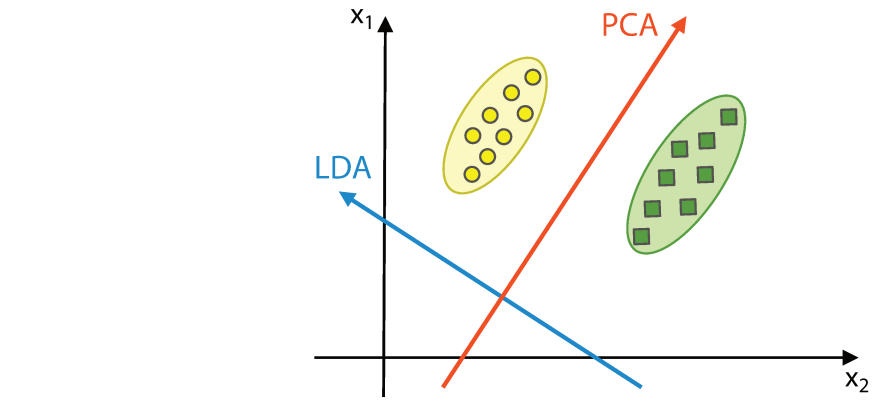
Figure 2.9(a) shows a toy example that illustrates the difference between PCA and LDA in a simple dataset with two classes. It was considered only the first basis in each technique.

PCA treats the data as a whole and its axes indicate where the maximum variation actually lies. It does not consider any division into classes, so that the class distribution on the projection axis can have a considerable overlapping, as shown in Figure 2.9(b).

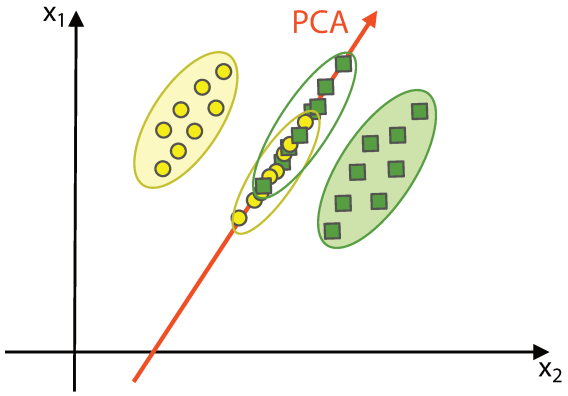
LDA in turn aims to find a linear combination of features that best separates samples of distinct classes. Figure 2.9(c) shows how the original data is expressed with the only linear discriminant. It is possible to observe that the classes are well separate.

Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task, where the class labels are known, this may not always be warranted and may sometimes lead to faulty system design, especially if the size of the learning dataset is small [61].

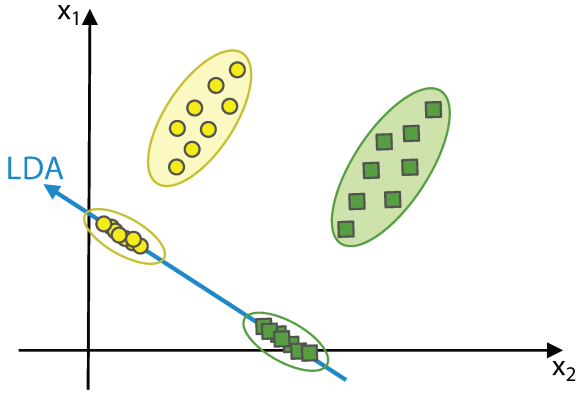
Martinez et al. [61] presented comparisons between classification accuracies for image recognition after using PCA and LDA. The results showed that PCA tends to outperform LDA if the number of samples per class is relatively small. In practice, it is also not



(a) PCA and LDA 1D Basis



(b) PCA reduction to 1D



(c) LDA reduction to 1D

Figure 2.9: A toy example that compares the PCA and LDA transformations. (a) The red and blue axes represent the PCA and LDA 1D basis, respectively. (b) The projection of the data on the only principal component. (c) The projection of the data on the only linear discriminant.

uncommon to use both LDA and PCA in combination, *e.g.*, PCA for dimensionality reduction followed by an LDA. A wide discussion about the compassion between PCA and LDA is presented in [61].

2.6 Support Vector Machines

Commonly considered as the first practical derivation of statistical learning theory, Support Vector Machines (SVMs) [18] represent at the present time a field of research that has a large choice of topics to work on, and many of the issues are conceptual rather than merely technical [62, 63]. Over the last years, its scope has widened significantly, both

in terms of new algorithms, such as kernel methods, and in terms of a deeper theoretical understanding [62].

In this dissertation, the covered theory is only introductory and the discussed concepts is narrowly related to the primal optimization of linear SVMs for pattern recognition problems. A considerable part of this section was condensed from the study about SVMs provided by Yu and Kim in [64].

2.6.1 Motivation

One of the fundamental problems of learning theory is stated as: given two classes of known objects, assign one of them to a new unknown object. A *linear classifier* reaches this by building a decision boundary based on a linear combination of feature values. Considering a given empirical dataset, this problem can be formalized as follows [62]:

$$(x_1, y_1), ..., (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}, \tag{2.22}$$

such that $\mathcal{X}$ is some nonempty set usually referred to as the *domain* from where the *patterns* x_i (also known as *samples, inputs*, or *instances*) are taken, and y_i are called *labels, targets*, or *outputs*. In such a problem, there are only two classes of patterns — which for mathematical convenience are labeled by $+1$ and -1 — such that for a new pattern $x \in \mathcal{X}$, the corresponding $y \in \{\pm 1\}$ has to be predicted. In other words, it indicates that a y has to be chosen, so that (x, y) is in some sense similar to the training examples, which leads to the notions of *similarity* in $\mathcal{X}$ and in $\{\pm 1\}$.

In the same spirit, binary SVMs are classifiers which discriminate instances of two classes. Each instance is represented by a n-dimensional vector. A linear classifier aims to separate the classes with an hyperplane, so that each instance belongs to only one.

Figure 2.10 illustrates two linearly separable groups of instances (training dataset) and only three of many possible hyperplanes that correctly classify (or separate) the groups. The best hyperplane is the one that achieves maximum separation between the two classes, *i.e.*, the hyperplane which has the largest margin. The margin is the summation of the shortest distance from the separating hyperplane to the nearest instances from both classes [64]. If such a hyperplane exists, it is known as the *maximum-margin hyperplane* and the linear classifier it defines is known as a *maximum margin classifier*.

2.6.2 Linear SVM

Given a training set Z, such that

$$Z = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}, \tag{2.23}$$

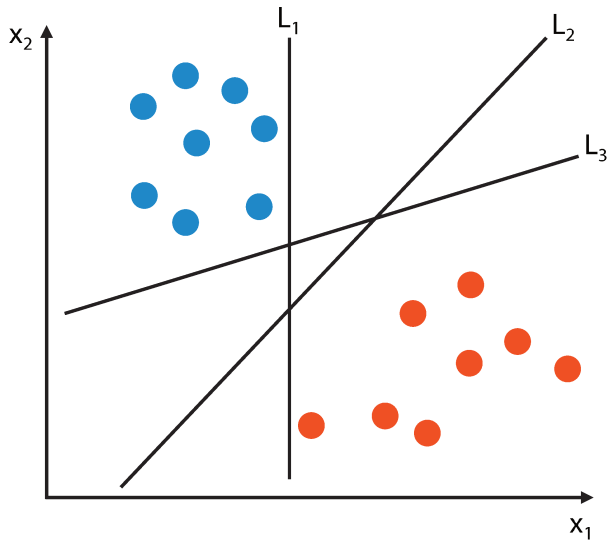


Figure 2.10: Some linear classifiers (hyperplanes) in a two-dimensional space. Each hyperplane separates the two linearly separable groups of instances.

where $\mathbf{x}_i$ is a n-dimensional real vector, and y_i is either 1 or -1 indicating the class to which the instance $\mathbf{x}_i$ belongs. Any hyperplane can be defined as the set of instances $\mathbf{x}$, such that

$$\mathbf{w} \cdot \mathbf{x} - b = 0, \tag{2.24}$$

where $\cdot$ denotes the dot product, $\mathbf{w}$ is the normal vector to the hyperplane, and b is the bias, which will be computed by SVM in the training process. The classification function $F(\mathbf{x})$, in turn, takes the form

$$F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b. \tag{2.25}$$

If the training set is linearly separable — *i.e.*, there is at least a linear classifier that correctly classifies all the training instances — SVM selects two hyperplanes that separate the data, so that there are no instances among them, and then try to **maximize their distance**. The region bounded by them is called *margin*, and can be described as

$$\mathbf{w} \cdot \mathbf{x} - b = 1, \tag{2.26}$$

and

$$\mathbf{w} \cdot \mathbf{x} - b = -1. \tag{2.27}$$

The distance between these two margins is $\frac{2}{\|\mathbf{w}\|}$, and the instances that lie on the ones are called *support vectors*.

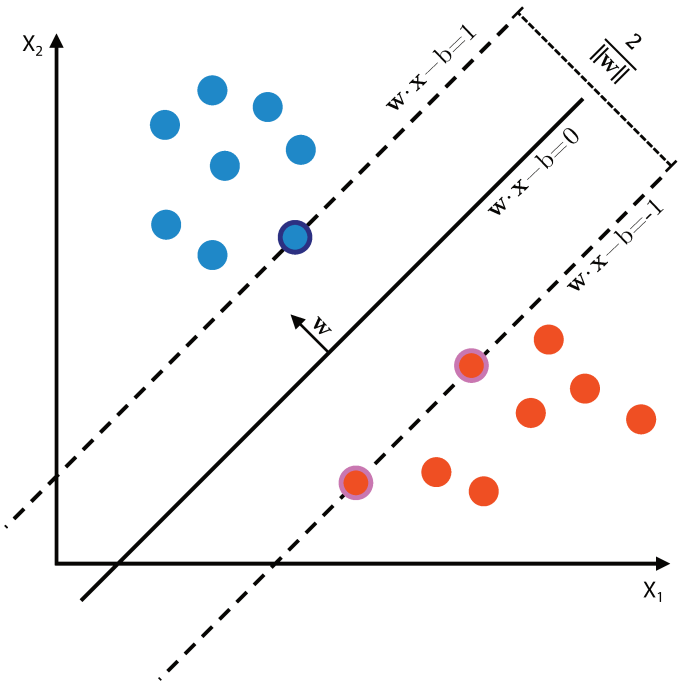


Figure 2.11: A toy example of maximum-margin hyperplane for an SVM trained with instances from two classes in a two-dimensional space. The optimal hyperplane (Equation 2.31a) is shown as a solid line, and the margins as dashed lines. The instances that lie on the margins are called *support vectors*. The problem being separable: there exists a normal vector $\mathbf{w}$ and a bias b, such that $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \forall(\mathbf{x}_i, y_i) \in Z$, where Z is the training set.

Since there are no instances into the margins, and $F(\cdot)$ must return positive values for positive instances from Z and negative values otherwise, we may define that, for every instance $\mathbf{x}_i$ in Z,

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1, \text{ if } y_i = 1, \tag{2.28}$$

and

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1, \text{ if } y_i = -1. \tag{2.29}$$

These conditions can be revised into:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \forall(\mathbf{x}_i, y_i) \in Z. \tag{2.30}$$

Maximizing the margin becomes minimizing $\|\mathbf{w}\|$. Therefore, the training task in SVM becomes a constrained optimization problem defined as

$$\underset{\mathbf{w},b}{\text{minimize}}\ Q(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{w}\|^2 \tag{2.31a}$$

$$\text{subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1, \forall(\mathbf{x}_i, y_i) \in Z. \tag{2.31b}$$

The factor of $\frac{1}{2}$ is used for mathematical convenience.

Figure 2.11 presents a toy example of maximum-margin hyperplane for an SVM trained with instances from two classes in a two-dimensional space.

2.6.3 Solving the Constrained Optimization Problem

The constrained optimization problem shown in Equation 2.31 is called *primal problem*. The objective function presented in Equation 2.31a is a *convex* function of $\mathbf{w}$, and the constraints are *linear* in $\mathbf{w}$. The constrained optimization problem may then be solved by using the method of Lagrange multipliers [65]. First, we construct the Lagrange function

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1], \tag{2.32}$$

where the auxiliary non-negative variables α are called Lagrange multipliers. The solution to the constrained optimization problem is determined by the saddle point of the Lagrange function $J(\mathbf{w}, b, \alpha)$, which has to be minimized with respect to $\mathbf{w}$ and b, and it also has to be maximized with respect to α [64]. Formally, we can define

$$\arg\min_{\mathbf{w},b}\max_{\boldsymbol{\alpha}\geq 0}\left\{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1]\right\}. \tag{2.33}$$

Differentiating $J(\mathbf{w}, b, \alpha)$ in terms of $\mathbf{w}$ and b, and setting the results equal to zero, we have the following two optimality conditions

$$\begin{aligned} &\text{Condition 1: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \\ &\text{Condition 2: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} = 0 \end{aligned} \tag{2.34}$$

The Condition 1 may then rewritten as

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i, \tag{2.35}$$

and the Condition 2 yields

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{2.36}$$

The solution vector $\mathbf{w}$ is defined in terms of an expansion that involves the m training instances.

Chapter 3

Proposed Negative Mining Approach

In this Chapter, we present an effective and efficient negative mining approach for face recognition systems in order to build US models during user enrollment. We first detail preliminary attempts regarding the development of cluster-based negative mining methods in Section 3.1 to then present the proposed linear SVM-based negative mining in Section 3.2.

3.1 Preliminary Attempts

Our first idea for the development of effective and efficient negative mining approaches was to use unsupervised learning algorithms, more specifically clustering techniques. Since these algorithms do not require labeled data, the mining process may be previously executed at development time, by selecting the most informative negative samples (in general) from a curated large face dataset. Such an informative negative subset, in turn, could then be used for the training of UI and US models at enrollment time. In this spirit, two approaches were investigated.

Figure 3.1 shows the pipeline of the first one. From a potentially huge dataset of negative face images, the method relies on a suitable selection of samples to create a large mining set with respect to processing constraints. Random selection was considered for such task. A cluster-based negative mining approach is then applied on the large mining set, outputting a negative subset with the most informative samples from the large mining set at development time. This approach simply applies a cluster technique on the large mining set and selects the nearest samples to each group center, assuming that these are the most representative samples from their respective clusters (Figure 3.2).

The second proposed approach consists of organizing the potentially huge negative set into a hierarchical clustering [66], as shown in Figure 3.3, in order to generate a large and informative mining set — which will be used in a negative mining approach — or the final

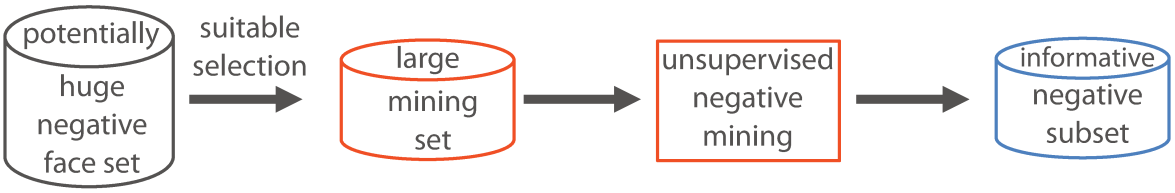


Figure 3.1: Pipeline of the first cluster-based negative mining approach proposed. From a potentially huge dataset of negative face images, we first select a set of images suitable for mining with respect to processing constraints. A cluster-based negative mining approach is applied on the large mining set, outputting a small negative subset with the most informative negative samples from the large mining set at development time.

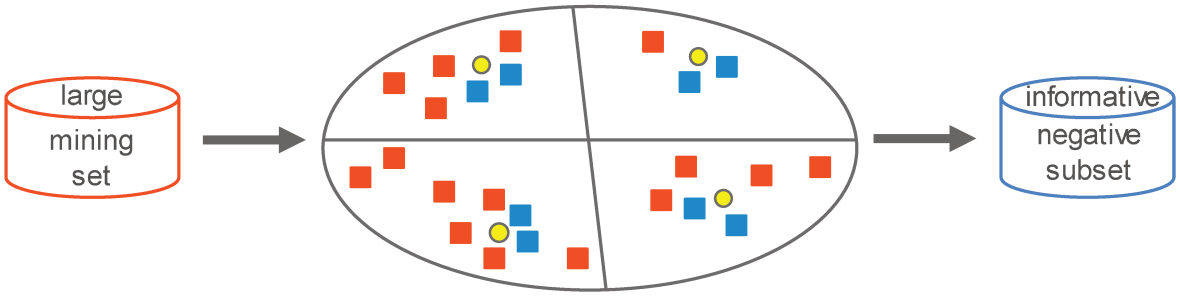


Figure 3.2: A toy example illustrating the negative mining process based on clustering. The yellow circles indicate the group centers and the squares are the samples. The large mining set is clustered and the nearest samples to each group center (blue squares) are selected to build an informative and small negative subset.

training negative subset with the most informative negative samples.

Assuming that the potentially huge negative set is labeled, at the first level, we use the class labels to partition the samples on subsets corresponding to each class, and then we run a clustering algorithm in each of these class-specific subsets. Alternatively, if the dataset is not annotated, we can randomly select samples to each partition and cluster them. The clustering algorithm then gives us the representative samples of each group that will be promoted to the next level of the hierarchy. This first selection may already correspond to a significant reduction in the dataset size, since only some representative samples from each partition will be selected.

On each of the following levels, we take all the promoted samples from the previous level and run again a clustering algorithm on them. This will result in new clusters with new representative samples to be promoted to the next level. This procedure is repeated until some criterion is satisfied, for example, that a number of samples in a given level is reached.

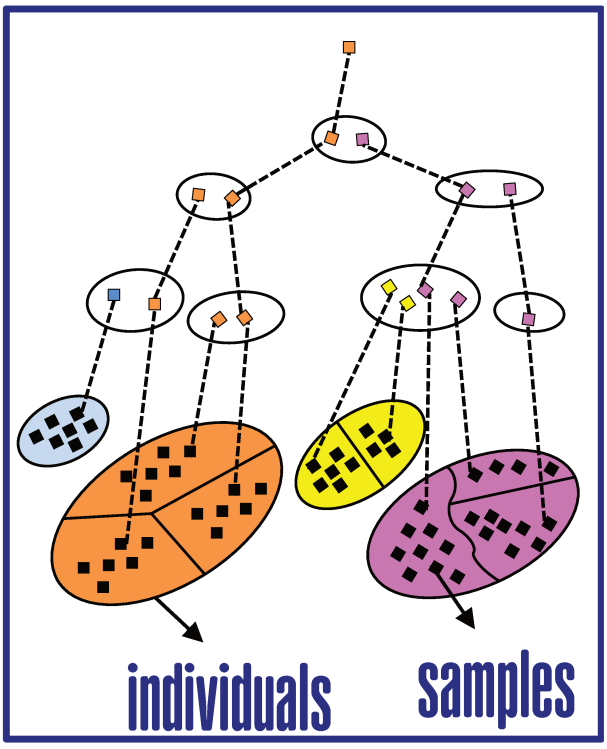


Figure 3.3: A hypothetical negative set organized into a hierarchical clustering. Assuming that the negative set is labeled, at the first level, we use the class labels to partition the samples on subsets corresponding to each class, and then we run a clustering algorithm in each of these class-specific subsets. On the following levels, we take all the promoted samples from the previous level and run again a clustering algorithm on them. This procedure is repeated until some criterion is satisfied.

The two proposed approaches were developed by using the Kmeans algorithm [67] for clustering, and the HT-L3-1st [13] for feature extraction (see Section 2.3). After the mining process, an User-Specific (US) model was trained for each individual (positive class) taking as input the union of the informative negative subset and the positive samples. Both methods were evaluated on PubFig83 [68] dataset according to the evaluation protocol presented in Section 4.2.

We have adopted some mining strategies for the first approach. Initially, we considered that the number of clusters is equal to the number of relevant samples. For example, if we want to build an informative negative subset with 100 samples, the method groups the large mining set in 100 clusters and then selects only the nearest sample of each centroid. The second strategy considers a fixed number of clusters. In other cases, we also select the nearest and the farthest samples from the centroids in order to retain representative and confusing samples of each group.

The second approach, in turn, builds a hierarchical clustering from the potentially huge negative set and selects a given number of samples, by starting from the top and walking down the hierarchy until achieving a level where the required number of negative samples is satisfied in order to build the final informative negative subset.

Both cluster-based negative mining approaches presented poor recognition performance, which can be explained by the fact that clustering techniques based on neighborhood and proximity cannot behave well with high-dimensional feature vectors [23]. Indeed, as the descriptor HT-L3-1st results in feature vectors with high dimensionality ($\sim$25,000 for the considered architecture), the performance of Kmeans was compromised, leading to few clusters with many samples and other clusters with very few samples. In order to deal with this limitation, we then reduced the dimensionality of the feature space using PCA as well as feature selection algorithms [69, 70], but the results were still unsatisfactory. Other metrics, such as Manhattan and Mahalanobis distances, were also evaluated, but they also resulted in no significant improvement.

In light of these results, we decided to focus on the investigation of linear SVM-based approaches, such as [16, 19], in order to develop an effective and efficient negative mining method, as presented in the next section. However, we believe that cluster-based negative mining approaches are promising for scenarios where the feature vectors have low dimensionality so that it is possible to obtain well-behaved clusters.

3.2 Proposed Linear SVM-based Negative Mining Approach

We propose a negative mining approach based on linear Support Vector Machines (SVMs) [18] with the following motivations. First, the ability to perform well with small sample sizes, especially in the case where the samples are represented by high-dimensional feature spaces, and second, we can train linear SVMs [18] quite fast under these circumstances.

Figure 3.4 illustrates user enrollment process of the proposed linear SVM-based negative mining approach. From a potentially huge dataset of negative face images, the algorithm relies on a suitable (under the time constraints) selection of samples to create a large mining set. Subsequently, it creates a small training set by identifying the most informative negative samples, with respect to the positive samples from a given user, to build an effective US model for that user in a few seconds.

A pseudocode of the proposed negative mining is presented in Algorithm 2. The algorithm considers gallery images of the individual being enrolled as the positive set P and a much larger negative mining set N from which a small set of c informative images must be iteratively mined within a given maximum processing time max_time. Indeed,

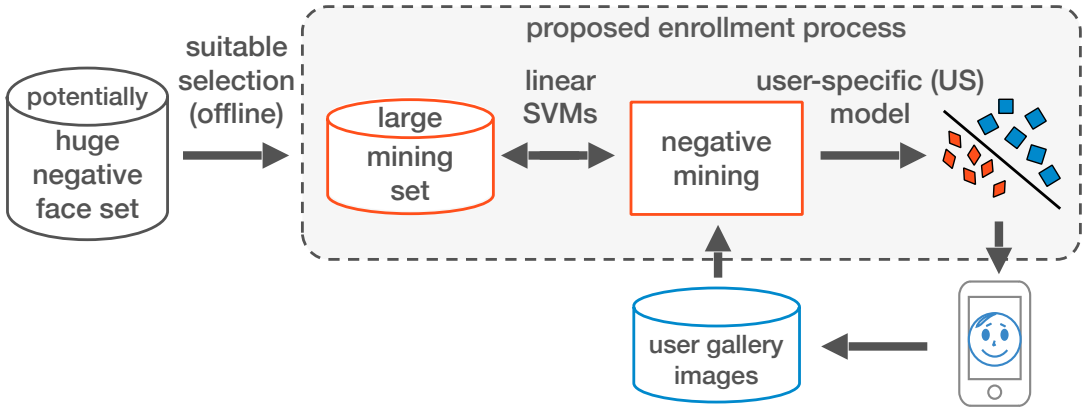


Figure 3.4: User enrollment process. From a potentially huge dataset of negative face images, the algorithm relies on a suitable (under the time constraints) selection of samples to create a large mining set. Subsequently, it creates a small training set by identifying the most informative negative samples, with respect to the positive samples from a given user, to build an effective US model for that user.

the mining set is split into a negative training set N_t ($|N_t| = c$) and a negative validation set N_v (Lines 1–2).

A linear SVM β is trained at each iteration by taking $P \cup N_t$ as input (Line 7). If the processing time from the algorithm right after the SVM training exceed the maximum processing time, it will be returned the model β_{out} which will correspond either to $None$ — no linear SVM could be trained within the maximum processing time — or to the model trained at the previous iteration (Lines 6, 8, 9, 12). Otherwise, the algorithm saves the current trained model in β_{out} (Line 10).

The signed distances to the SVM hyperplane of all samples in the negative training set — except support vector — and in the validation set are computed and inserted into the lists L_t and L_v, respectively (Lines 14–18). These lists are then sorted, according to the signed distance $\beta(.)$, for subsequent sample swapping (Lines 19–20).

Images are swapped between N_t and N_v according to a criterion based on an "informativeness" degree, which is exactly the signed distance $\beta(.)$ to the SVM hyperplane of the given iteration (Lines 22–30).

Given a sample $s \in N_t \cup N_v$, the assumption is that the greater $\beta(s)$ is, more informative for the gallery model s will be. Therefore, the least informative samples in N_t that are *not* support vectors are swapped with the most informative ones in N_v. If no improvement in the overall informativeness of N_t is observed in a given iteration — *i.e.*, no swaps occurred — or the maximum processing time max_time is reached, the algorithm terminates (Lines 31–34). There is no problem if the processing time exceed the

Algorithm 2 PROPOSED SVM-BASED NEGATIVE MINING

INPUT: Positive set P, large mining set N, maximum processing time max_time, and number of negatives to be mined c.
OUTPUT: Best model β_{out} for the positive set P.
AUXILIARY: Sets N_t, N_v, lists L_t, L_v, variables β, $swaps$, $stop$, s, t, ds, dt, $time_1$, $proc\_time$.

```
1.  N_t ← random selection of c samples from N
2.  N_v ← N \ N_t
3.  proc_time ← 0
4.  β_out ← None
5.  While proc_time < max_time
6.      time_1 ← point_time()
7.      β ← linear SVM trained on P ∪ N_t
8.      proc_time ← proc_time + (point_time() − time_1)
9.      If proc_time ≤ max_time
10.         β_out ← β
11.     Else
12.         Return β_out
13.     time_1 ← point_time()
14.     L_t ← empty list, L_v ← empty list
15.     For each s ∈ N_t not support vector
16.         insert (s, β(s)) into L_t
17.     For each t ∈ N_v
18.         insert (t, β(t)) into L_v
19.     L_t ← sort L_t by β(.) in increasing order
20.     L_v ← sort L_v by β(.) in decreasing order
21.     swaps ← 0, stop ← 0
22.     While L_t ≠ empty and L_v ≠ empty and stop ≠ 1
23.         remove (s, ds) from L_t head
24.         remove (t, dt) from L_v head
25.         If dt < ds
26.             N_t ← (N_t \ s) ∪ t
27.             N_v ← (N_v \ t) ∪ s
28.             swaps ← swaps + 1
29.         Else
30.             stop ← 1
31.     If swaps = 0
32.         Return β_out
33.     proc_time ← proc_time + (point_time() − time_1)
34. Return β_out
```

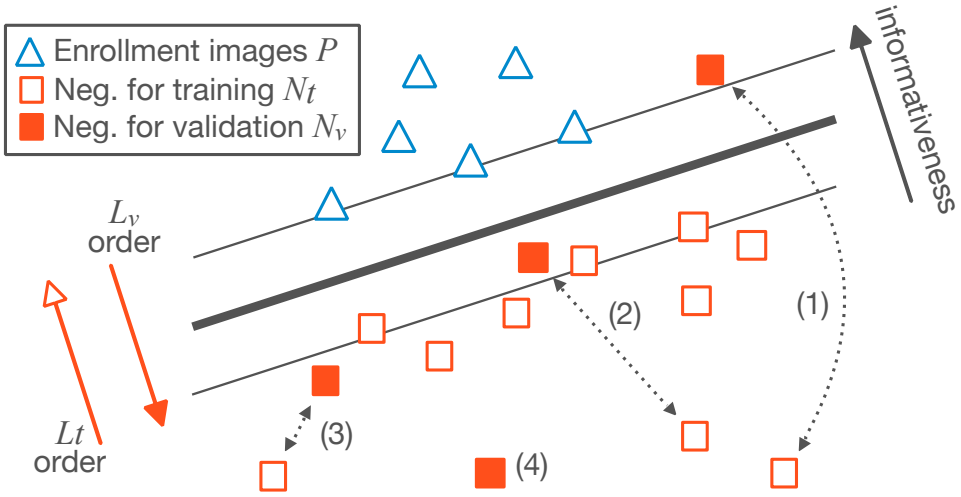


Figure 3.5: Mining process in a given iteration. The least informative samples in N_t that are *not* support vectors are swapped with the most informative ones in N_v, as indicated by the swapping sequence (1), (2), and (3). Swapping occurs no matter each side of the margin the negative samples are, which increases the ability of the method to operate well even in unbalanced learning scenarios. In (4), no swap occurs because such validation sample is less informative than any other available for swapping in N_t.

max_time before the informativeness checking (Line 31), because the model returned in Line 32 was trained within the allowed processing time (Line 7).

An important property of the approach as compared to [15, 19] is that correctly classified negative samples may also be swapped, which enables it to mine negative samples even in extremely unbalanced learning scenarios. Moreover, we can see from Algorithm 2 that its running time is dominated by the SVM training in Line 7, which can ranges from quadratic to cubic on the size of input training set, depending on the regularization constant C [18]. Given that the number of negative samples predominates over the number of positive samples in the idealized enrollment process, our expectation is that learning gallery models by iterating a few times the mining process with $c << |N|$ will probably speedup the enrollment process while not compromising the recognition performance.

In Figure 3.5, we illustrate the rationale of the swapping process. The possible swaps between negative samples of training and validation sets are enumerated according to the informativeness degree. The least informative samples in N_t that are *not* support vectors are swapped with the most informative ones in N_v, as indicated by the swapping sequence (1), (2), and (3). Swapping occurs no matter each side of the margin the negative samples are, which increases the ability of the method to operate well even in unbalanced learning scenarios (see Section 4.4). Finally, there is no swap in (4), because such validation sample is less informative than any other available for swapping in N_t.

Chapter 4

Experiments

In the absence of a really huge public face dataset, we simulated this scenario with the two unconstrained public face datasets described in Section 4.1. The evaluation protocol considered in this work is detailed in Section 4.2, and the baselines are presented in Section 4.3. Finally, the results are shown in Section 4.4.

4.1 Datasets

A face dataset plays a crucial role in the effective evaluation of a recognition algorithm. In its previous time, when automatic face recognition had just become a reality, these databases expressed their challenges, with mostly collections representing constrained scenarios. However, as pose, illumination, age, occlusion, or expression started to be considered, a new perspective to face recognition research was introduced mainly with the release of Labeled Faces in the Wild (LFW) [11], a dataset based on the original idea of collecting images of celebrities from the Internet with the only requirement that their faces were detectable by the Viola-Jones algorithm [71].

In this section, two public unconstrained face datasets are described. The first one is the PubFig83 [5] that is a refined version from the PubFig dataset [68]. The second one is the Mobio [24], a dataset recorded using mobile devices.

4.1.1 PubFig83

The PubFig83 dataset [5] is a subset of the PubFig dataset [68], which is, in turn, a large collection of real-world images of celebrities collected from the Internet. This subset was established and released to promote research on familiar face recognition from unconstrained images, and it is the result of a series of processing steps aimed at removing spurious face samples from PubFig, *i.e.*, non-detectable, near-duplicate, *etc.* In

addition, only persons for whom 100 or more face images remained were considered, leading to a dataset with 83 individuals. Each image has originally 100x100 pixels in size. To our knowledge, this is the publicly available face dataset with the largest amount of unconstrained, uncorrelated images per individual.

Its original evaluation protocol [5] is designed for *closed-set* recognition. The dataset is splitted into ten pairs of *training* and *test* sets with images selected randomly and without replacement. For each individual, 90 images were considered for training and the remaining 10 for test, which in the context of this work we may call gallery and probe images, respectively.

In Figure 4.1, we present images of four individuals in a given split of PubFig83. While here we only have space to show 10 (out of 90) gallery images of each individual, all their respective probe images are presented. Due to its unconstrained nature, we can observe that PubFig83 presents at the same time all factors of variation in face appearance: pose, expression, illumination, occlusion, hairstyle, aging, among others.

4.1.2 Mobio

The Mobio dataset used in this work is precisely the same used in the competition on unconstrained face recognition in mobile platforms organized as part of the *International Conference on Biometrics*, ICB'13 [22]. The dataset has 150 people with a female-male ratio of nearly 1:2 (100 males and 50 females). It is the result of an international collaboration, in which images from six institutions of five different countries were recorded using two types of mobile devices (laptop and mobile phone) in 12 distinct video sessions for each individual [1]. The dataset can be considered challenging in the sense that images were acquired without control over factors such as illumination, facial expression, and face pose. Moreover, in some cases, only parts of the face are visible.

Based on the gender of the individuals, its original evaluation protocol is split up into *female* and *male*. Still, for the sake of fairness, individuals in the dataset are divided into three subsets, namely the *training* set, the *development* set, and the *evaluation* set.

The *training* set has 50 individuals — 13 females and 37 males — with 192 images each and can be used for any purpose to aid the systems, from learning subspace models to leveraging score normalization. In addition, this is the only subset where gender can be combined according to the participant's needs.

The *development* set has 42 individuals — 18 females and 24 males — and can be used to tune the hyperparameters of the algorithm, *e.g.*, the number of projection vectors while learning subspaces, which similarity measure to use, *etc.* For each person in this set, there are five gallery images and 105 probe images.

[1] In particular for the competition, all images available were captured by mobile phones.

Figure 4.1: Images of four individuals in a given split of PubFig83. While here we only have space to show 10 (out of 90) gallery images of each individual, all their respective probe images are presented. Due to its unconstrained nature, we can observe that PubFig83 presents at the same time all factors of variation in face appearance: pose, expression, illumination, occlusion, hairstyle, aging, among others. This figure was adapted from [72].

The *evaluation* set, in turn, is used to assess the final system performance. It has 58 individuals — 20 females and 38 males — with samples arranged in exactly the same way as the *development* set, *i.e.*, five gallery images and 105 probe images.

In Figure 4.2, we present gallery and probe images of four individuals in the evaluation set. While we can clearly see variation in pose, expression, and illumination, we can also observe that the individuals are — to some extent — collaborating with the image acquisition process. More importantly, however, is to observe the difference in appearance among the gallery and the probe images. In fact, we can see that the five gallery images of each individual look quite similar. While this is a natural consequence from the fact that these images were recorded in the same session, this considerably diminishes the discriminative power of learning techniques operating on them [72].

4.2 Evaluation Protocol

Since the size of both datasets is too small when compared to a real scenario (millions or billions of images), we consider that the *mining* set of Figure 3.4 is already built. Evaluations are carried out in a realistic *open-set* scenario, in that no information of other gallery individuals is used for building US models of new individuals at enrollment time (see Section 2.1.1). The results are reported assuming that the system is operating in verification mode.

The Mobio protocol presented in Section 4.1.2 naturally addresses this scenario, and hence we report results using the union of its original *training* and *development* set as the *mining* set of Figure 3.4 — in a total of $14,010$ images — and its *evaluation* set as containing images of individuals under enrollment (gallery) — the user of the system for whom the False Acceptance Rates (FAR) and Correct Acceptance Rates (CAR) are calculated.

PubFig83 original evaluation protocol, however, is designed for *closed-set* face recognition. Therefore, we extended the protocol detailed in Section 4.1.1 by further splitting the dataset into two subsets: one simulates the *mining* set, containing images of 60 individuals chosen at random — in a total of $6,000$ images — and the other is equivalent to the *evaluation* (gallery) set, containing images of the remaining 23 individuals to report the FAR and CAR values. Each individual from the evaluation set has 90 gallery images and 10 probe images.

Each experiment is repeated ten times in order to enable us to report more reliable mean values of CAR and FAR, verifying the robustness (standard error of those values) of the algorithm with respect to different mining sets. Even though the *mining* set is fixed on Mobio, these ten experiments are valid, because the considered negative mining methods (see Section 4.3) use an initial random negative training set in each user enrollment which

Figure 4.2: Gallery and probe images from the MOBIO evaluation set. While we can clearly see variation in pose, expression, and illumination, we can also observe that the individuals are collaborating with the image acquisition process. More importantly, however, is to observe the difference in appearance among the gallery and the probe images. This figure was adapted from [72].

in turn will be different in each experiment. A similar scenario occurs on PubFig83 which additionally has a different *mining* set in each experiment.

All images were aligned by the position of the eyes according to [6]. For feature extraction, we use the HT-L3-1st descriptor (see Section 2.3 for details), which has the property of outputting high-dimensional features vectors, with $25,600$ elements for the considered architecture.

The execution times of all experiments were obtained in the same Intel I7-3770k machine with 32GB of RAM, and no experiment required memory swapping. We use LIBSVM [73] via Scikit-learn package [74] to train the SVMs (Algorithm 2, Line 7) with the regularization constant C fixed at 10^5 as in [5, 6, 22].

4.3 Compared Methods

We compared our approach with five others. The first two are User-Independent (UI) models built with PCA [31] and LDA [32], both methods applied in the entire mining set. These techniques are widely used to build offline face recognition models, during the conception of the recognition system. Both PCA and LDA implementations are from Scikit-learn [74], the number of retained projection vectors was according to the rank of the input covariance matrices, and the matching between face samples was done via cosine similarity [75].

The other compared methods were based on User-Specific (US) models. We started by comparing US models built with linear SVMs also using the entire mining set, as in PCA and LDA. Given that this approach is also based on linear SVMs, but uses all negative samples at disposal for learning (no negative mining), we may say that it represents a statistical upper bound for the proposed approach, which is based on a considerably smaller training set. Therefore, for clarity, we call it *expected upper bound*.

We then evaluate two negative mining approaches, one consisting of a *random selection* of the negative samples — and serving as baseline and sanity check for the proposed approach — and the other implementing the well known SVM-based negative mining criterion of Felzenszwalb et al. [16].

The processing time of each negative mining method corresponds to the sum of the time spent during user enrollment to mine the mining set and train the final linear SVM classifier which will be used to assess the final recognition performance.

4.4 Results and Discussion

Initially, we compared the performance between UI and US models with no negative mining. That is, all methods — PCA, LDA, and linear SVM (our expected upper bound) — using the entire mining set as input (Figure 3.4). The comparative results are shown in Figure 4.3. Since the mining set is fixed for Mobio, only one iteration was executed, so that the standard errors were not computed.

We can clearly see the enormous difference in Correct Acceptance Rate (CAR) between US and UI models in all scenarios of FAR for both datasets. The difference in performance is considerably more significant for PubFig83. These results confirm the superior performance of US over UI models, and the effectiveness of linear SVMs to deal with high-dimensional feature spaces in the unconstrained face recognition scenario. Indeed, the US modeling technique presented in Figure 4.3 produces a state-of-the-art results for Mobio dataset (vide the UC-HU method in [22]).

In Table 4.1, we present the experimental results of our negative mining approach and the considered baselines using a system that wrongly accepts only 0.01% of the test cases for (a) Pubfig83 and (b) Mobio.

The comparative results between UI and US models can be verified in the first three lines of Tables 4.1a and 4.1b. PCA and LDA dismiss learning during user enrollment, which explains the zeros in their learning times. However, this seriously compromises their performance. The linear SVM with no mining, on the other hand, can negatively affect the user experience, since it requires 52.72 seconds for Mobio, for instance. The larger the base is, the higher the processing time will be. Thus, negative mining methods are crucial to attain the "ceiling" CAR of the expected upper bound within an interactive time without affecting the user experience.

Such an interactive response time for user enrollment requires to mine the most informative samples into a small training set. One can observe from the fourth line in Table 4.1 the results of the three US models based on negative mining. The considered maximum processing times (third column in Table 4.1) were chosen based on the required time for the expected upper bound in order to be less than this one and yet interactive.

We consider a negative training set with 5% of the mining set (parameter c in Algorithm 2) for PubFig83 and 1% for Mobio. These values were chosen based on our power constraints. Thus, for each gallery individual being enrolled in the system, all negative mining methods use the same initial negative training set built randomly, which is used in all considered maximum processing times.

Since that the spent time by the Random Selection is less than all considered maximum processing times, it presents only one CAR value in Tables 4.1a and 4.1b. Indeed, it is the most efficient approach, but its ability to select informative negative samples for the

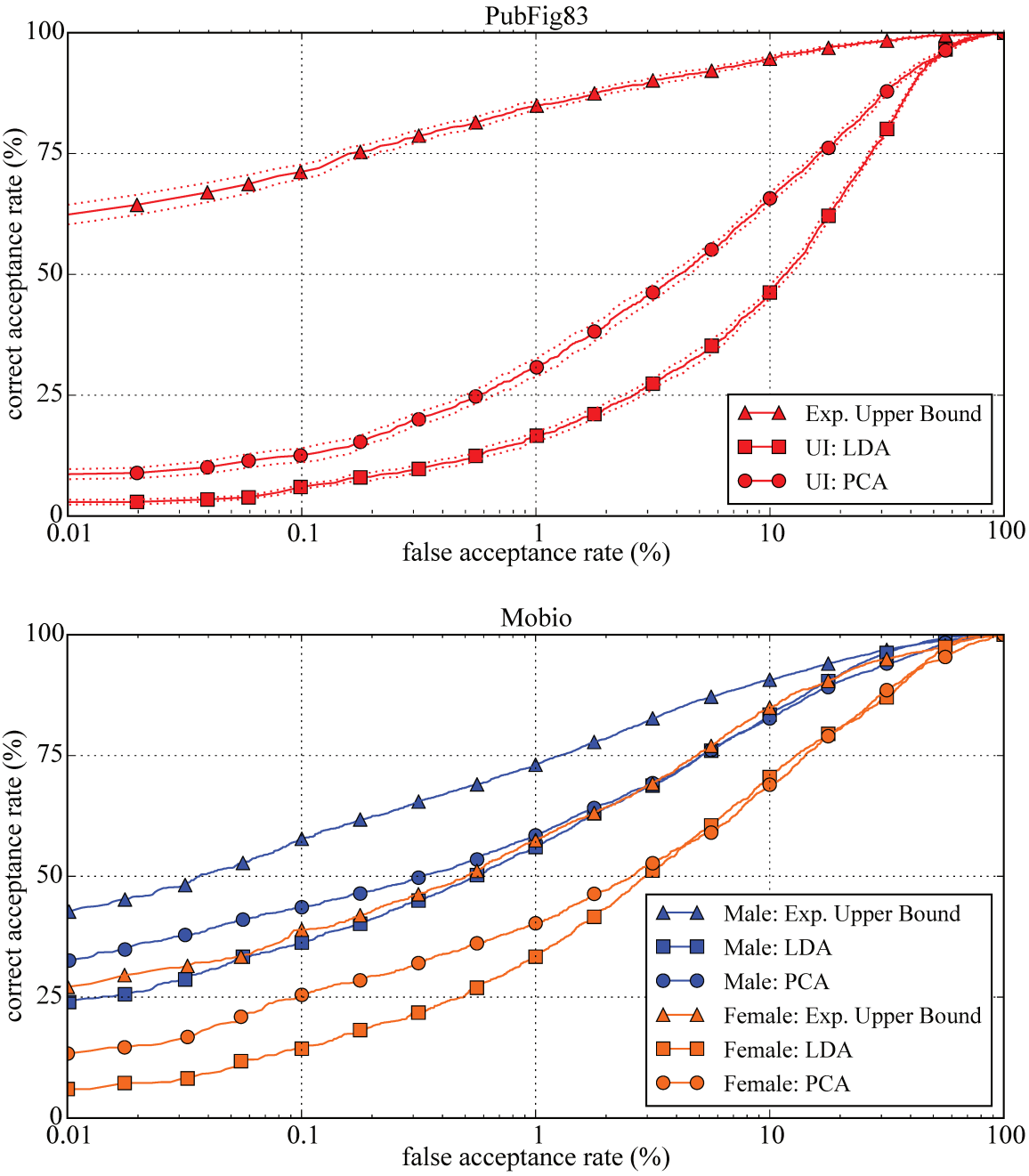


Figure 4.3: This figure illustrates the receiver operating characteristics (ROC) curve (with FAR in logarithmic scale) of an US classifier and two UI classifiers for PubFig83 and Mobio. Dashed lines correspond to standard errors. All methods were applied in the entire mining set. The US classifier performs substantially better than the UI classifiers along the ROC curve for both datasets, achieving recognition rates significantly higher in PubFig83.

Table 4.1: CAR as obtained with UI and US models in both datasets for a FAR fixed at 0.01%.

	learning approach	max. proc. time (secs)	neg. mining criterion	PubFig83
UI models	PCA	0.00	n/a	8.91 ± 1.08
	LDA	0.00	n/a	2.91 ± 0.51
US models	linear SVM (exp. up. bound)	14.35	n/a	64.39 ± 2.07
	Linear SVM-based negative Mining	2.00	random selection	34.35 ± 2.83
		4.00		
		6.00		
		8.00		
		2.00	Felzenszwalb et al. [16]	32.57 ± 2.83
		4.00		34.35 ± 2.92
		6.00		48.57 ± 4.66
		8.00		64.00 ± 1.86
		2.00	proposed in this work	39.09 ± 5.02
		4.00		**64.65± 1.71**
		6.00		64.43 ± 1.97
		8.00		64.52 ± 1.91

(a) PubFig83

	learning approach	max. proc. time (secs)	neg. mining criterion	Mobio male	Mobio female
UI models	PCA	0.00	n/a	32.58	13.33
	LDA	0.00	n/a	23.98	6.05
US models	linear SVM (exp. up. bound)	52.72	n/a	42.71	27.14
	Linear SVM-based negative Mining	4.00	random selection	34.66 ± 1.38	21.91 ± 0.52
		8.00			
		12.00			
		16.00			
		4.00	Felzenszwalb et al. [16]	34.66 ± 1.38	21.75 ± 0.52
		8.00		37.47 ± 1.11	27.65 ± 0.30
		12.00		42.25 ± 0.22	27.28 ± 0.08
		16.00		42.19 ± 0.30	26.75 ± 0.41
		4.00	proposed in this work	**45.00 ± 0.27**	27.73 ± 0.42
		8.00		43.72 ± 0.09	**27.93 ± 0.10**
		12.00		43.81 ± 0.02	27.65 ± 0.02
		16.00		43.83 ± 0.03	27.69 ± 0.03

(b) Mobio

training set is seriously affected.

The method of Felzenszwalb *et al.* [16] presents similar CAR values with respect to the Random Selection for the lowest considered processing times. This is a consequence of its mining criterion, which may allow the number of negatives in the training set to

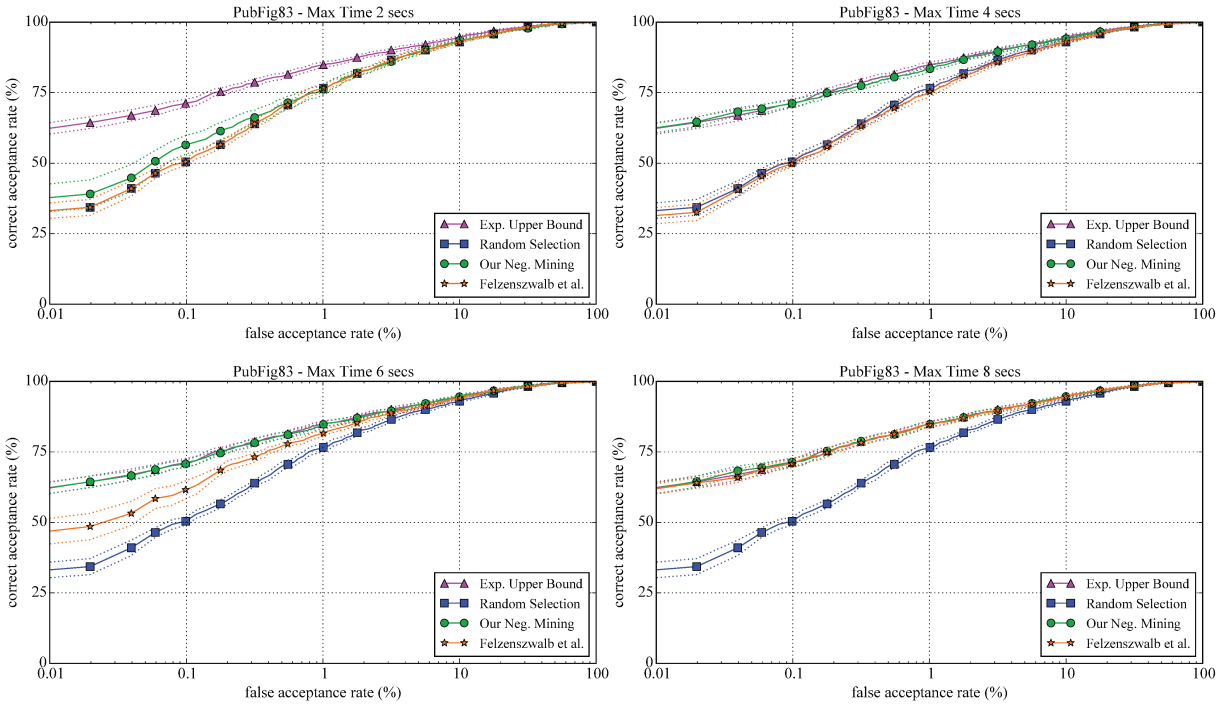


Figure 4.4: System performances of our approach and five other baseline methods in PubFig83 for the considered maximum processing times and for different points of FAR. Intervals correspond to standard errors.

grow arbitrarily, resulting in the execution of only a few iterations. As the maximum time increases, its recognition rate also increases, outperforming the Random Selection and reaching values near the "ceiling" CAR of the expected upper bound in some cases.

Table 4.1 clearly shows that our negative mining approach is able to attain superior recognition performance within a fraction of the time required by the expected upper bound, especially in the Mobio dataset. As compared to the Felzenszwalb's method, our negative mining is also preferred in both aspects, recognition and time performance. We believe that our mining criterion is more robust to critical negative samples, since we may also mine important correctly classified samples outside the SVM margin (see Figure 3.5), while these samples are ignored in the mining process of [16]. Moreover, the processing time of each iteration from our method tends to be approximately constant, since the negative training set size is fixed. The bold numbers in Table 4.1 show that for both datasets, our method can achieve the expected upper bound recognition performance without affecting user experience (in interactive time).

In Figure 4.4, we present a Receiver Operating Characteristic (ROC) curve (mean error values with standard errors) for each considered maximum processing time with the behavior of *random selection*, Felzenszwalb *et al.*'s method, and *our* negative mining

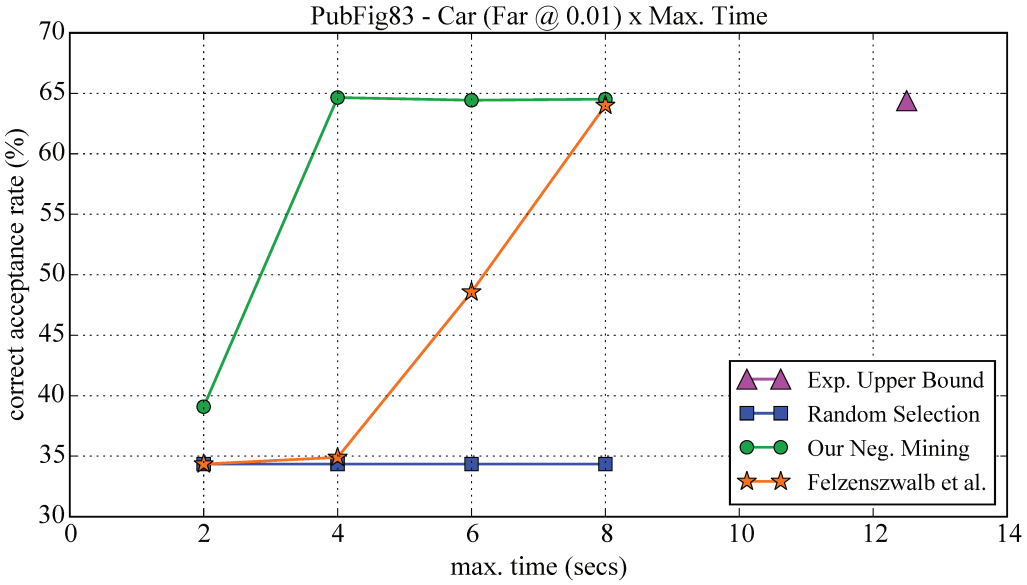


Figure 4.5: Recognition rate and maximum processing time of all negative mining methods for a system with FAR at 0.01% on PubFig83, as shown in Table 4.1.

approach at different operating points (as in Table 4.1). As previously mentioned, our method already reaches the "ceiling" CAR of the expected upper bound, requiring only 4 seconds for user enrollment. Moreover, this behavior is valid to all operating points, from a FAR of 0.01% to a FAR of 10% and beyond.

Finally, it is possible to better visualize recognition and time behaviors of the compared methods in Figure 4.5 which makes clear the advantages of our approach from both perspectives, recognition and time performances.

Chapter 5

Conclusion and Future Work

Modern face recognition systems have reached expressive results in the *unconstrained* scenario with the use of high-dimensional feature spaces and user-specific (US) classification models. However, as the face datasets grow, the training time of these systems becomes prohibitive. The problem becomes even more critical, when considering the training time during user enrollment.

In this context, negative mining methods are necessary to considerably reduce the training set by mining the most informative negative samples from the large face dataset, and at the same time maximize the performance of the system in recognizing the user under enrollment. However, the majority of these methods either fail to capture the most informative negative examples or require high processing times, which in turn affects the user experience for enrollment.

In this work, we addressed these issues by proposing a fast negative mining approach suitable for application during user enrollment in modern face recognition systems. Initially, we developed two cluster-based negative mining methods to gather informative negative samples for later US model learning, but both methods presented unsatisfactory results for high-dimensional feature spaces. In spite of the obtained results, we believe that unsupervised mining is promising in scenarios where the feature vectors have low dimensionality.

Subsequently, we proposed a negative mining approach based on linear SVM given its ability to perform well in the case where the samples are represented by high-dimensional feature vectors. Our approach has shown to be fast and robust in mining the most informative negative samples for a given individual, being enrolled in the system, according to a criterion based on distances to SVM decision boundaries, and outputs an US model in a few seconds.

We compared our approach with well-known methods in terms of recognition and time performances. In the absence of a really huge public face dataset, we simulated

this scenario with two public unconstrained face datasets, namely PubFig83 and Mobio. The results showed that the proposed linear SVM-based negative mining approach excels on these datasets, achieving superior recognition rates within interactive response times. This was not the case of the other baseline approaches, which performed worse when the maximum processing time allowed is low. Based on such results, we may conclude that our approach has potential to be exploited by the industry with minimum impact to the user experience.

We argue that smart user enrollment, coupled with learning tasks to leverage data at disposal, is a promising idea to consider in modern biometrics systems, and this dissertation presents a practical and effective approach for that. To the best of our knowledge, this is the first work to propose negative mining for user-specific gallery model building at enrollment time. This work was recently submitted to *IEEE Signal Processing Letters* [25].

Given that our negative mining approach can be split into client and server tiers — requiring low bandwidth between the tiers — it is also well suited to face recognition systems that operate on budgeted devices. Moreover, our algorithm is application-independent, so that we may conclude that it is a relevant contribution for biometric systems that aim to maintain robustness as the number of users increases.

In the short term, we envision the extension of our linear SVM-based negative mining approach to other problem domains, such as remote sensing and mobile systems. Indeed, as the amount of available data has considerably grown in many problems, the use of efficient and effective techniques for the selection of representative samples from large databases becomes increasingly essential. Finally, we also intend to further investigate the applicability of cluster-based negative mining approaches studied in problems that use low-dimensional visual representations.

Bibliography

[1] I. S. Bruner and R. Tagiuri, *The perception of people*, vol. 2. Addison-Wesley, 1954.

[2] W. W. Bledsoe, "The model method in facial recognition," tech. rep., Panoramic Research Inc., Palo Alto, CA, 1964.

[3] T. Kanade, *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Kioto University, 1973.

[4] A. K. Jain and S. Z. Li, *Handbook of Face Recognition*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[5] N. Pinto, Z. Stone, T. Zickler, and D. Cox, "Scaling-up Biologically-Inspired Computer Vision: A Case-Study on Facebook," in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[6] G. Chiachia, A. X. Falcão, N. Pinto, A. Rocha, and D. Cox, "Learning person-specific representations from faces in the wild," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. PP, no. 99, pp. 1–1, 2014.

[7] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

[8] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[9] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[10] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *International Conference on Computer Vision*, pp. 897–902, 2009.

[11] G. B. Huang, M. Mattar, T. Berg, and E. Learned-miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," tech. rep., University of Massachusetts, Amherst, October 2007.

[12] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] N. Pinto and D. Cox, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," in *IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG)*, 2011.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[15] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[16] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[17] J. Valmadre, S. Sridharan, and S. Lucey, "Learning detectors quickly using structured covariance matrices," *CoRR*, vol. abs/1403.7321, 2014.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] J. Papa, A. X. Falcão, C. T. N. Suzuki, and N. D. A. Mascarenhas, "A discrete approach for supervised pattern recognition," in *Combinatorial Image Analysis (IWCIA)*, 2008.

[20] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] M. Gunther, A. Costa-Pazo, C. Ding, E. Boutellaa, G. Chiachia, H. Zhang, M. de Assis Angeloni, V. Struc, E. Khoury, E. Vazquez-Fernandez, D. Tao, M. Bengherabi, D. Cox, S. Kiranyaz, T. de Freitas Pereira, J. Zganec-Gros, E. Argones-Rua, N. Pinto, M. Gabbouj, F. Simoes, S. Dobrisek, D. Gonzalez-Jimenez, A. Rocha, M. Neto, N. Pavesic, A. Falcão, R. Violato, and S. Marcel, "The 2013 Face Recognition Evaluation in Mobile Environment," in *International Conference on Biometrics (ICB)*, 2013.

[23] J. Walters-Williams and Y. Li, "Comparative study of distance functions for nearest neighbors," in *Advanced Techniques in Computing Sciences and Software Engineering*, pp. 79–84, Springer, 2010.

[24] C. McCool, S. Marcel, A. Hadid, M. Pietikainen, P. Matejka, J. Cernocky, N. Poh, J. Kittler, A. Larcher, C. Levy, D. Matrouf, J.-F. Bonastre, P. Tresadern, and T. Cootes, "Bi-modal person recognition on a mobile phone: Using mobile phone data," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012.

[25] S. B. Martins, G. Chiachia, and A. X. Falcão, "A fast and robust negative mining approach for enrollment in face recognition systems," *IEEE Signal Processing Letters (SPL)*, Jan 2015. Submitted for publication.

[26] R. Chellappa, P. Sinha, and P. J. Phillips, "Face recognition by computers and humans," *IEEE Computer Biometrics Compendium*, vol. 43, no. 2, pp. 46–55, 2010.

[27] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe, "Frvt 2006 and ice 2006 large-scale experimental results," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 5, pp. 831–846, 2010.

[28] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[29] G. Goswami, R. Bhardwaj, R. Singh, and M. Vatsa, "Mdlface: Memorability augmented deep learning for video face recognition," in *IEEE International Joint Conference on Biometrics (IJCB)*, 2014.

[30] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognition: A review," *Journal Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 103–135, 2005.

[31] M. Turk and A. Pentland, "Face Recognition using Eigenfaces," in *Computer Vision and Pattern Recognition (CVPR)*, 1991.

[32] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 19, no. 7, pp. 711–720, 1997.

[33] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 12, pp. 2037–2041, 2006.

[34] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[35] Q. Liao, J. Z. Leibo, and T. Poggio, "Learning invariant representations and applications to face verification," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[36] J. Z. Leibo, Q. Liao, and T. Poggio, "Subtasks of Unconstrained Face Recognition," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics (VISIGRAPP)*, 2014.

[37] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[38] P. J. Phillips, J. R. Beveridge, B. A. Draper, G. Givens, A. J. O'Toole, D. Bolme, J. Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer, "The good, the bad, and the ugly face challenge problem," *Journal Image and Vision Computing*, vol. 30, no. 3, pp. 177–185, 2012.

[39] W. R. Schwartz, H. Guo, and L. S. Davis, "A robust and scalable approach to face identification," in *European Conference on Computer Vision (ECCV)*, vol. 6316, pp. 476–489, 2010.

[40] G. P. Carlos, H. Pedrini, and W. R. Schwartz, "Fast and scalable enrollment for face identification based on partial least squares," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

[41] N. A. Ogale, "A survey of techniques for human detection from video," *Survey, University of Maryland*, 2006.

[42] T. M. Jørgensen, A. Tycho, M. Mogensen, P. Bjerring, and G. Jemec, "Machine-learning classification of non-melanoma skin cancers from image features obtained by optical coherence tomography," *Skin Research and Technology*, vol. 14, no. 3, pp. 364–369, 2008.

[43] L. Li, Q. Zhang, Y. Ding, H. Jiang, B. H. Thiers, and J. Z. Wang, "Automatic diagnosis of melanoma using machine learning methods on a spectroscopic system," *BMC medical imaging*, vol. 14, no. 1, p. 36, 2014.

[44] V. E. "Vinzi, W. W. Chin, J. Henseler, and H. Wang, *"Handbook of Partial Least Squares"*. "Springer", "2010".

[45] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *British Machine Vision Conference (BMVC)*, 2009. doi:10.5244/C.23.91.

[46] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, pp. 1532–1545, Aug 2014.

[47] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.

[48] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[49] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[50] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[51] J. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *International Conference on Computer Vision (ICCV)*, 2014.

[52] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. X. Falcão, and A. Rocha, "Deep Representations for Iris, Face, and Fingerprint Spoofing Detection," *ArXiv e-prints*, 2014.

[53] W. S. Geisler and D. G. Albrecht, "Cortical neurons: Isolation of contrast gain control," *Vision Research*, vol. 32, no. 8, pp. 1409–1410, 1992.

[54] L. Batina, J. Hogenboom, and J. J. van Woudenberg, "Getting more from pca: First results of using principal component analysis for extensive power analysis," in *Topics in Cryptology* (O. Dunkelman, ed.), vol. 7178 of *Lecture Notes in Computer Science*, pp. 383–397, Springer Berlin Heidelberg, 2012.

[55] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *The Journal of Educational Psychology*, vol. 24, 1933.

[56] S. Z. Li and A. K. Jain, eds., *Handbook of Face Recognition*, ch. 7 (Face Recognition in Subspaces, by G. Shakhnarovich and B. Moghaddam), pp. 141–168. Springer, 2004.

[57] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," in *Audio- and Video-based Biometric Person Authentication* (J. Bigün, G. Chollet, and G. Borgefors, eds.), vol. 1206 of *Lecture Notes in Computer Science*, pp. 125–142, Springer Berlin Heidelberg, 1997.

[58] L. "Chen, H. M. Liao, M. Ko, J. Lin, and G. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713 – 1726, 2000.

[59] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using lda-based algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, 2003.

[60] J. Ye, T. Li, T. Xiong, and R. Janardan, "Using uncorrelated discriminant analysis for tissue classification with gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 1, no. 4, pp. 181–190, 2004.

[61] A. M. Martinez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 23, no. 2, pp. 228–233, 2001.

[62] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[63] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[64] H. Yu and S. Kim, "Svm tutorial—classification, regression and ranking," in *Handbook of Natural Computing*, pp. 479–506, Springer, 2012.

[65] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[66] S. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[67] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[68] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *IEEE International Conference on Computer Vision (ICCV*, 2009.

[69] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[70] M. Kolar and H. Liu, "Feature selection in high-dimensional classification," in *International Conference on Machine Learning (ICML)*, pp. 329–337, 2013.

[71] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[72] G. Chiachia, A. Rocha, and A. X. Falcão, *Learning Person-Specific Face Representations*. PhD thesis, University of Campinas (Unicamp), Campinas, Brazil, 2013.

[73] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[75] X. Jiang, B. Mandal, and A. Kot, "Eigenfeature regularization and extraction in face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, no. 3, pp. 383–394, 2008.