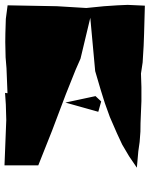


Universitat Autònoma de Barcelona

Face Classification Using Discriminative Features and Classifier Combination

A dissertation submitted by **David Masip Rodó** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informática**.

Bellaterra, June 16th, 2005

Director: **Dr. Jordi Vitrià i Marca**
Universitat Autònoma de Barcelona
Dept. Ciències de la Computació & Computer Vision Center

Centre de Visió per Computador

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 84–933652–3–8

Printed by Ediciones Gráficas Rey, S.L.

A la meva familia

"Un hombre sabio adecua su creencia a la evidencia"
David Hume, Investigación sobre el conocimiento humano.

De acuerdo, que hable del amor y de la muerte, pero expresandose
en terminos de matemáticas superiores, sobre todo los del álgebra de tensores. (...)
-Estas loco. Sobre el amor en el lenguaje matemático?
Pero se callo enseguida: el electrobardo se puso a recitar:

Un ciberneta joven potencias extremas
Estudiaba, y grupos unimodulares
De ciberias, en largas tardes estivales
Sin vivir del Amor grandes teoremas.

Huye...! Huye, Laplace que llenas mis días!
Tus versores, vectores que sorben mis noches!
A mi contraimagen! Los dulces reproches
Oir de mi amante, oh alma querías.
(...)

"El Electrobardo de Trul."
Stanislaw Lem. La Ciberíada.

"No. Try not. Do. Or do not. There is no try." - **Yoda**

Agraiments

Aquesta tesi ha estat realitzada gràcies a la beca FPI de formació de personal investigador FP2000-4960 del Ministeri de Ciència i Tecnologia associada al projecte TIC2003-00654. Gràcies a aquesta beca he pogut acabar el doctorat en un espai de temps raonable, i alhora m'ha permès realitzar una estada d'investigació a Bangor (Gales) de gran valor formatiu.

En primer lloc voldria agrair al meu director, en Jordi Vitrià, els consells, l'ajut i les correccions que m'ha ofert en la realització de aquest treball així com el seu interès i confiança dipositada en la meva persona des del primer dia. En Jordi em va despertar l'interès pel món de la investigació ja a l' època de la facultat, durant la realització del projecte de final de carrera. Sens dubte per mi ha estat el director de tesi ideal, estant sempre en el just punt mig a l'hora de demanar resultats per establir els objectius i donar sempre el marge de llibertat suficient als seus doctorands per assolir-los. Al Jordi doctor li haig d'agrair els seus abundants coneixements sobre la temàtica de la tesi, la seva capacitat per trobar solucions als problemes i la seva capacitat per predir quins mètodes i algorismes seran d'utilitat en els problemes plantejats. Al Jordi persona li haig d'agrair que sempre ha tingut un minut per qualsevol problema i qüestió que li he pogut plantejar.

De la mateixa manera haig d'agrair a tot el personal del centre, començant pel seu director, el Dr. Juanjo Villanueva i acabant per totes les persones que treballen fent tasques de suport, permetent que els doctorands puguem gaudir d'un entorn quasi privilegiat per a realitzar la nostra tesi. Moltes gràcies, especialment a la Montse, l'Anna Celia, la Mari Carmen, la Maria Jose, l'Ainhoa, en Pedro, i la Pilar.

De tots els companys del centre voldria fer menció especial a tres persones que han marcat decisivament el desenvolupament d'aquesta tesi: primer de tot agrair d'una manera especial les ensenyances rebudes del doctor Marco Bressan, amb qui vaig tenir la gran sort de treballar durant el meu primer any de tesi. En Marco és sense cap mena de dubte el mirall en què s'haurien de mirar tots els doctorands que comencen. Menció especial mereix en David Guillamet a qui no només haig de agrair la seva col.laboració en diversos treballs que hem emprès junts, sinó també la seva amistat dins i fora del centre, les xerrades bursàtils pel messenger i les sessions de saf setmanals, que han fet que la meva estada al centre sigui encara més divertida. I finalment agrair a Àgata Lapedriza el seu ajut imprescindible en la realització de la part final d'aquesta tesi. Els algorismes d'extracció de característiques externes d'imatges de cares per a la seva classificació presentats en aquest treball han estat fruit de la estreta col.laboració i del treball en equip en aquest darrer any, sense

l'Àgata res d'això hauria estat possible. A tots tres us dec molt mes del que us podré donar mai.

Igualment vull agrair als companys de despatx que he tingut durant aquests 4 anys: Oriol Ramos, Carme, David Rotger, Jaume Amores, Jaume Garcia, Francesc, Silvia, Albert i Agnés la seva companyia i amistat. I en especial agrair a Jaume, Misael, Aura, Alicia (I do?), Joan, Ignasi, Dani Rowe, la seva paciència amb mi durant aquests darrers dos anys aguantant les meves bromes pesades envers ells.

Aquest agraïment es fa extensiu a la resta de companys del centre que sempre estan disposats a donar un cop de mà quan fa falta ([juanma, rfelip, botey, felipe, maria, ramon, joan, enric, josep, oriol, robert, daniel, debora, vicente, raquel, mjose, pilar, montse, mcmerino, pedro, xevi, poal, silvia, petia, anton, anna, carme, ricardo, cristina, sergio, mathias, fernando, tothom]@cvc.uab.es).

Tambe vull agrair a tots els meus amics els bons moments que em fan passar, permetent-me descarregar de forma divertida la tensió que es pot originar quan hi ha un deadline i s'ha de acabar una feina, especialment als meus grans amics Oscar, Esteve, i companys aficionats als decks com Eloi, Jose, Diego, Oriol, Dani, i un llarg etc. Als companys del futbol de sempre, Enrique, Jofre, Roger, Eloi, Esteve, Albert, Fermin, Aitor,... i d'ara Raul, Jar, Andrea, Giuseppe, Steve, Josep, Tonim, Todd, Fernando, ... (no us preocupeu nois, algun dia guanyarem algo...). Tambe vull tenir un record especial per una gran dona, la Rosa, que ens ha aguantat tant de temps els nostres costums "frikis", vals un imperi.

Un record especial mereixen tots els que m'han ajudat al llarg de la meva carrera universitària, en especial a Eloi, Ferran, Xiao Yuan, Guillem, Veronica, Alex Morillas, Aitor, Fermin, ... i als meus ex companys de iSOCO Tonie, Jess, Ruben, Natalia, Begoña, M. Carmen, Jesus, Xavi Drudis, Jar, Tonim, Maite, i un llarg etc...

Voldria agrair especialment a la doctora Ludmila I. Kuncheva les ensenyances i el temps que va invertir en mi en la meva estada a Bangor. El que vaig aprendre allí no té preu des d'un punt de vista metodològic, i el llibre i documentació que em va proporcionar han estat els eixos principals d'aquest treball. Però a la Lucy li haig d'agrair encara més la seva dedicació i preocupació a nivell personal per a fer-me sentir com a casa. Lucy, thanks for being my mum in the U.K.

Un agraïment especial mereixen dos amics que sobresurten per sobre de tots, l'Albert Sancho i l'Eloi Puertas, segurament les persones de fora del meu entron familiar amb qui més temps he compartit, i més m'han marcat. També vull tenir un record per la Cristina, la meva petitona, que amb qui he compartit un espai de temps meravellós.

I si hi ha dues persones que son i han estat importants en la meva vida, aquestes son el meu pare i al meu germà. Al meu pare li haig de agrair tot, l'educació que m'ha donat, a costa de patir ell tot tipus d'esforços i sacrificis, però sobretot les seves ensenyances, la seva capacitat de superació davant les adversitats (malauradament masses), i l'escala de valors que ha sabut transmetre als seus fills. El seu llegat no es pot calcular i per molt que li agraeixi tinc la sensació que mai sera suficient. I al meu germà Jordi, probablement la persona amb més talent que mai coneixeré, li haig d'agrair que m'hagi ajudat i suportat durant tants anys i que ho continuï fent. Finalment vull tenir un record ben especial per la meva mare, al cel sigui, per molts anys que visqui mai et deixaré d'estimar.

MOLTES GRÀCIES A TOTS

Resum

A mesura que la tecnologia avança, van apareixent nous dispositius que es poden integrar a la nostra vida diària. Alguns d'aquests dispositius incorporen petites càmeres que es poden utilitzar en aplicacions que donin valor afegit als nous sistemes encastats. En aquest context, s'ha generat un gran interès en el desenvolupament de sistemes automàtics d'aprenentatge i classificació mitjançant càmeres i mètodes de visió per computador. Una de les aplicacions més destacades en aquest camp és el reconeixement de cares. Les tècniques de reconeixement facial han estat històricament aplicades a sistemes de seguretat i control d'accés, com a mètode alternatiu de verificació de la identitat en sistemes biomètrics no intrusius. Aquest fet ha despertat l'interès dels fabricants de nous dispositius mòbils dotats de camera per a substituir el sistema d'autentificació basat en pins, requerint alts percentatges d'encert. Més recentment, han aparegut altres aplicacions que fan ús de classificació automàtica d'imatges facials, com ara el reconeixement de gènere en aplicacions de publicitat reactiva, o el reconeixement d'expressions facials en el disseny d'interfícies d'usuari avançades.

Tradicionalment, els mètodes basats en l'aparença han obtingut els millors resultats en el problema de la classificació de cares. Normalment s'acostuma a tractar cada imatge com un vector d'alta dimensionalitat que conté tots els pixels de la cara, i posteriorment s'aplica un procés d'aprenentatge d'un classificador en l'espai definit per les cares d'entrenament. Un dels problemes que sorgeixen en aquest model, és el que s'anomena "maledicció de la dimensionalitat" (curse of dimensionality), que provoca que es necessiti un número inabastable d'imatges d'entrenament per modelar els paràmetres del classificador. Per tal de mitigar aquest problema, s'utilitzen mètodes d'extracció de característiques que redueixen considerablement la dimensionalitat del problema. Per altra banda, l'extracció de característiques permet alhora reduïr la redundància inherent en les dades visuals, eliminar part del soroll present en imatges naturals i, el més important, permet aprendre característiques discriminants en les imatges, per tal que la posterior etapa de classificació sigui més efectiva, fent-la més robusta davant de canvis en la il.luminació i oclusions parcials.

En la primera part d'aquesta tesi, s'introdueixen els sistemes de combinació de classificadors per a derivar una nova família de tècniques d'extracció de característiques. S'han introduït 3 noves tècniques d'extracció de característiques que utilitzen l'algorisme de l'Adaboost per a generar una projecció lineal que extreu característiques dis-

criminants donat un conjunt de dades d'entrenament. A diferència de les tècniques clàssiques d'extracció de característiques que es poden trobar en la bibliografia, el principal avantatge de les tècniques introduïdes és el fet que no fan cap assumpció en les dades a classificar. A més, a la part experimental que valida els mètodes proposats, es pot concloure que la família de tècniques que es presenten és especialment adequada per a dades de alta dimensionalitat, com és el cas dels problemes de classificació de cares.

En la major part dels treballs passats referents a classificació de cares, es fa servir únicament la informació interna per a aprendre el classificador. Descartant la informació localitzada al cabell, front, i ambdues zones laterals. Per altra banda, s'han publicat estudis psicològics que destaquen que el sistema visual humà dóna una gran importància a les característiques externes. A la segona part d'aquesta tesi es planteja un sistema per modelar computacionalment la informació externa d'imatges facials, per tal de poder-hi aplicar les mateixes tècniques de classificació que s'apliquen a la informació interna. El principal problema que presenten les característiques externes d'imatges facials és l'absència absoluta d'alineació entre els pixels que les formen. Donada la gran diversitat que es pot originar en els diferent estils de cabell i pentinat, el mateix pixel (i per tant coordenada en un vector de característiques) no significa el mateix entre imatges d'individus diferents. En aquesta tesi s'introdueix un algoritme per a extreure característiques externes de cares mitjançant l'adaptació d'un algorisme "Top-down" utilitzat en el camp de la segmentació d'imatges. Donats un conjunt d'imatges prou diverses d'entrenament, se n'extreuen petits fragments de les característiques externes de manera que es cobreixi al màxim possible tots els possibles pentinats que puguin aparèixer. El conjunt de petits fragments escollit es el que constitueix el model après de les característiques externes. El problema d'extracció de les característiques externes d'una nova imatge es converteix doncs en trobar la millor manera de recobrir la nova imatge amb els fragments del model. Per a realitzar aquesta tasca s'ha fet servir l'algorisme NMF, que proporciona una representació "sparse" de les dades, de manera que només alguns fragments del model romanen actius en cada reconstrucció (aquells més adequats a la persona que s'analitza). La sortida de l'algorisme és un vector de característiques amb el pes que té cada fragment del model en el procés de reconstrucció, de manera que els problemes d'alineament queden inherentment resolts. Els experiments realitzats amb la nova tècnica introduida mostren uns resultats molt prometedors per a l'aplicació de les característiques externes en la classificació d'imatges facials.

Finalment, es conclou la tesi mostrant alguns possibles mètodes de combinació de les característiques internes i externes en classificació de cares, on es pot apreciar que hi ha una millora dels resultats causats per l'aportació extra d'informació per part de les característiques externes, especialment en imatges on apareixen oclusions parcials o canvis sobtats en la illuminació.

Abstract

As technology evolves, it allows the development of new electronic devices that can be embedded in our everyday life. Some of these devices incorporate small cameras that can be used in applications to become an added value for the new embedded systems. In this context, the development of automatic learning and classification systems using cameras and computer vision methods has become the focus of attention. One of the most important applications on this field is face recognition. Face recognition techniques have been historically applied to surveillance systems and access control, as an alternative method to identity verification in non intrusive biometric systems. Manufacturers of new mobile devices equipped with cameras have started to test the replacement of the actual pin based identifying methods for face verification, requiring high accuracies. And recently, new applications using face classification have arised, such as gender recognition in reactive publicity, or gesture recognition on the design of user-friendly interfaces.

Traditionally, appearance based methods have reached the best results in face classification problems. Normally, each image is treated as a high dimensional vector containing all the pixel values, and then a classifier is trained on the subspace defined by the training faces. This model usually suffers from the "curse of dimensionality" problem, that consists on the need of an extremely large number of training samples to properly learn the parameters of a probabilistic classifier. In order to mitigate this problem, we use feature extraction methods that considerably reduce the data dimensionality. On the other hand, feature extraction techniques allow also to reduce the inherent redundancy of visual data, and also eliminates part of the noise present in natural images. Moreover, feature extraction allows to learn invariant discriminant characteristics in images, in order to improve the posterior classification step, being more robust against illumination changes and partial occlusions.

In the first part of this thesis, we introduce the classifier combination methods in order to derive a new family of feature extraction techniques. Actually, three new feature extraction methods have been introduced using the Adaboost algorithm to generate a linear projection that extracts discriminant features given a data training set. Opposite to classic feature extraction techniques found in the bibliography, we do not make any statistical assumption on the data to classify. Moreover, we conclude in the experimental validation of the proposed methods that they are specially suitable for high dimensional data, as is the case of face classification.

In the most part of previous face classification works, only the internal information is used to learn the classifier, discarding the information located hair, forehead,

and both lateral zones. On the other hand some psychological studies suggest that human visual system gives a lot of importance to external features in the recognition problems. In the second part of this thesis we introduce a system to computationally model the external information of facial images, in order to apply the same classification techniques applied to the internal ones. The main problem of external features in facial images is the lack of alignment among the pixels that compose the image. Given the extreme diversity originated from different subjects and hair-styles, the same pixel (the same coordinate in the feature vector) does not mean the same in images from different subjects. In this thesis we introduce an algorithm to extract external features adapting a Top-down segmentation algorithm. Given a training set of diverse enough images, we extract small fragments from the external zones, in such a way that we can cover the maximum diversity of hair-styles that can appear. The set of small fragments constitutes the learned model of external features. Therefore, the external feature extraction problem of a new unseen images is simplified to finding the optimal cover of the image using fragments from the model. To achieve this goal we have used the NMF (non negative matrix factorization algorithm), that yields an sparse representation of the data, in such a way that only a small subset of the fragments are active at each reconstruction (those more appropriate to cover the person under analysis). The output of the algorithm is a feature vector with the weight of each model fragment in the reconstruction process, so alignment problems become inherently solved. The experiments performed using this technique show encouraging results in order to be applied in face classification problems.

Finally, we conclude this thesis showing some possible methods to combine internal and external features in face classification, where it can be observed that there is an improvement on the results due to contribution of the information from the external features, specially in images with occlusions or changes in the illumination.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The evolution of the available computational resources accelerates every year, making feasible new applications dealing with face classification. Simultaneously, the amount of space and energy needed by actual computers decreases, allowing a progressive integration in our everyday life. Examples of applications that make use of face classification are face recognition applied to surveillance systems, gender and ethnicity recognition applied to reactive marketing and gesture recognition in user friendly interfaces.

Today one of the most challenging applications of computer vision is to integrate visual systems in uncontrolled environments, for improving our quality of life. The final step of this progressive integration should be a world where technologies become completely indistinguishable from the environment, making the computers disappear. This new paradigm is known as ubiquitous computing [183] and includes several areas of computer science: networking, sensors, video, operative systems, etc. Face classification systems will be a piece of this global model, yielding important information about identity, gender, or human interaction.

The topic of face recognition has been studied in a lot of disciplines: psychologists, computer vision and cognitive researchers, plastic surgeons, police force and justice agents have studied this problematic under their own point of view. Nevertheless, it seems that human visual system has solved this problem, with some limitations [178, 46]. Although it is not the goal of this thesis to imitate the natural face recognition process, a brief overview of some relevant aspects of human face recognition will be given in this introductory section.

The first question arising is: learn the humans to recognize faces, or are face recognition strategies "hard-wired" in human brain? In this context, Johnston and Ellis [71] showed that babies with less than 10 minutes of life show stronger preferences for face patterns than for any other kinds. In 48 hours, babies can recognize their mothers face, and in 10 weeks face patterns are the stimulus where babies pay more attention. Then the face recognition capabilities are increasingly improved until they

Figure 1.1: The "Thatcher illusion" made by Thompson [169], the expression of the face with its eyes and mouth inverted changes from 'pleasant' to 'grotesque' as the stimulus is rotated from 180 to 0.

reach the age of twelve years old approximately. Also the experiments performed under gesture and expression changes show that children under 11 years old are more influenced by salient features than adults, that use a more structural strategy.

Also it has been shown that humans recognize easier subjects from their same race [167], so the learning of face recognition seems to be own-race biased by subjects seen by each person during his whole life. Also it has been shown that face recognition of negative and rotated images is difficult for the human visual system [57] (decreasing more than 10% the accuracy). In another work by Thompson [169] the "Thatcher illusion" was introduced in order to show that we do not have the same capability when measuring the spatial relations between facial features in rotated images. Figure 1.1 shows an example of this illusion, looking at both face upside-down, they seem both real faces, but when the images are rotated back 180, the second one becomes grotesque. Therefore, perhaps strategies that are useful for general object recognition are not the best ones for face recognition, that should be studied using specific techniques.

In addition it has been shown that motion can be useful for face recognition, given that motion acts as an additional cue to the visual system to extract 3D structure of faces [54]. Actually, some studies performed on patients injured in the brain suggests that there are completely separate modules in the human brain dedicated to face classification tasks [45, 46]. They studied perceptual responses from patients with prosopagnosia. People suffering this perceptual illness are not able to recognize faces, although they can recognize the remaining objects as usual.

Other studies made by psychology researchers show an interesting perceptual phenomenon: humans recognize faster caricatures than original face images [55, 133]. A computational model exploiting this fact has been introduced in [129].

On the other hand, it has been shown that humans are able to recognize gender with high accuracy rates (more than 95%). Some studies [187] show that the time

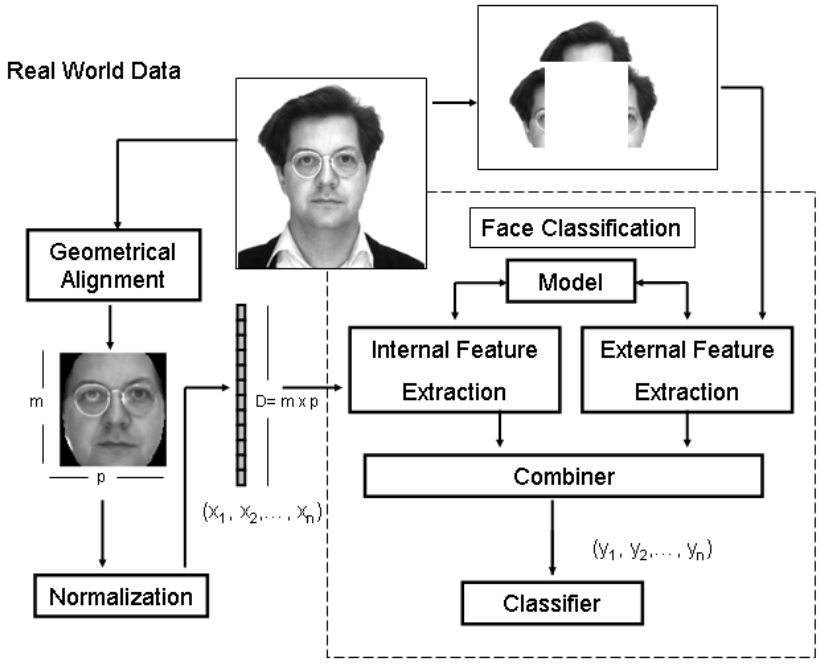


Figure 1.2: Face classification scheme followed in this work. Face images are normalized and converted to a D-dimensional data vector. A feature extraction is performed according to a model on the internal and external part of the faces, and the sample is finally classified according to the joint feature set.

to identify a face as familiar and to determine its gender is completely independent. This fact suggest that there is an independence between these two face classification tasks in the human visual system, as if they were separate modules.

In this thesis we will deal with computational face classification systems, which is far from being a solved problem. The first question to be answered is which will be the scheme used in an automatic face classification problem. As in the human visual system, the starting point is a set of features observed from the real world. Only the interesting parts from the observations are captured and processed in the classification stage, where the final decision about the observed sample is taken. The extraction of the best features for classification is performed according to a model of the problem previously learned. Figure 1.2 illustrates the whole process. According to the global scheme proposed, the immediate question to answer is: which is the most suitable feature extraction technique to classify face images? Depending on the application and the classifier used, the results can vary considerably. In this introductory chapter a brief overview of some common feature extraction techniques on face classification problems will be performed. In the next chapters, a new feature extraction technique specially suitable for high dimensional subspaces will be introduced.

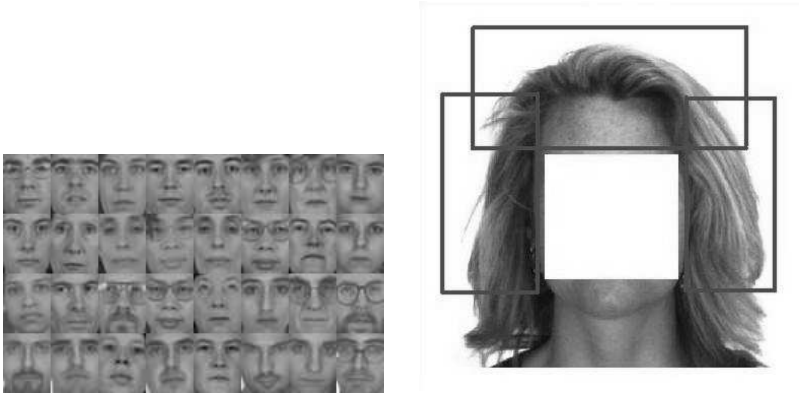
Figure 1.3: The Presidential Illusion [154, 155]. In both cases the internal features have been artificially altered to highlight the importance of the external information for face classification. At first sight, four different persons are wrongly distinguished from the image due to the external information.

The most common approach to feature extraction for face classification is to use the internal information of each face image. Nevertheless, it can be shown that external information present on face images can also help in many classification tasks, such as in gender or face recognition. In Figure 1.3 an example of the influence of the external features on face classification is shown, "the Presidential Illusion" [154, 155]. The internal features correspond to only one person on each picture, while the external features remain unchanged. Given the popularity of the subjects, at first sight, four different people are distinguished from the pictures.

External features can be defined as the part of a face image that contains the head, ears, hair and chin. In Figure 1.4 an example of external an internal information from face images is shown. As can be seen, in the internal case, is relatively straightforward to perform a previous normalization on the face images given that the position of the eyes can be used to put in correspondence faces acquired in different conditions (scale, rotations), and apply direct bottom-up classification schemes. Perhaps the main reason to justify the small use of external information in the most used face classification applications is the lack of alignment and its extreme diverse nature. Nevertheless, it has been proven that external features can be useful for classification tasks. I. Jarudi and P.Sinha [68] showed that the external features can be even more important than the internal when dealing with low resolution images. In the second part of this thesis, a complete scheme to extract the external features of face images will be introduced. Also a study of the importance of the added external information to the classic internal approach will be presented.

1.1 Face Classification

Face classification in visual pattern recognition can be defined as the problem of assigning some predefined labels to an image or subpart of the image that contains one or more faces. Actually, face classification can be divided into different subproblems depending on the concrete goal and the semantics of the class labels:



(a) (b)

Figure 1.4: (a) Internal part of face images. (b) Example of the zones where external features are extracted

- **Face detection**, where the goal is to detect the presence of faces in natural images, and accurately locate their position in complex uncontrolled environments. According to Yang et al.[97] face detection schemes can be classified in 4 different categories, although some methods can belong to more than one category:

    1. Knowledge-based methods, where some rules or common knowledge about face images and relationships between features are encoded. Kotropoulos and Pitas followed this approach using projection profiles to locate the face [32].
    2. Feature invariant approaches, where the idea is to detect the facial features first, such as eyes, mouth, eye-brows and group them into candidate faces [88].
    3. Template matching methods, where there is a predefined face pattern that is correlated with the image. Point Distribution Models (PDM) have been used for this purpose [83]
    4. Appearance-based methods, where the goal is to train a classifier that learns the features of the faces from a training set with face and non face images. Many classic techniques such as Principal Component Analysis [76], Gaussian mixture models [110], Neural Networks [164], Hidden Markov Models [144], Support Vector Machines [117], and Probabilistic Models [110] have been applied in this approach.

- **Face recognition**, where the goal is to assign a label from a predefined set to a located face image. Many methods of face recognition have also been proposed. Basically they can be divided into holistic template matching based methods, geometrical local feature based methods, and hybrid schemes [188]:

  - The holistic methods use the whole image as a raw input to the learning process. Examples of this techniques are Principal Component Analysis [76], Independent Component Analysis [112], or Support Vector Machines [91] applied to face recognition.
  - In the feature based schemes some structural features are extracted, eyes, mouth, and their local appearance, position or relative relationship are used for training the classifier. The most successful technique is the Elastic Bunch Graph Matching [184] where they use Gabor wavelets to extract the basic features for the graph matching scheme.
  - Hybrid methods try to use the best of the holistic and feature-based approaches combining local features and the whole face to recognize. An example of hybrid methods is the use of eigenfeatures [124], which extends the idea of eigenfaces to specific regions of the face such as mouth, nose, or eyes.

  Among the holistic methods, appearance-based methods are the most successful. They are commonly implemented following these steps:

  1. Image Preprocessing [101], where usually a illumination correction is performed, followed by the localization of some parts of the face for geometrical alignment that makes the feature-based approaches more accurate.
  2. Feature Extraction. Dimensionality reduction techniques have shown important advantages in some pattern recognition tasks and face processing is not an exception. Principal Component Analysis is perhaps one of the most spread dimensionality reduction techniques ([76] and [173]) in face classification.
  3. Feature Classification. Once the proper features are extracted, any classifier can be applied.

  Most of these methods have been successfully used in artificial environments, but do not perform well in many real world situations as several independent tests have documented [128].

- **Face verification**, where the identity of the subject is given, and the problem is to ensure its truthfulness. Face verification can be seen as a subtype of face recognition, where the amount of information available is greater. Nevertheless the accuracy required on face verification applications (such as secure identification for access control) is larger than in face recognition problems.

- **Gender recognition**, where male or female label is assigned to each face image. Humans are able to distinguish the gender from face images with high accuracy. In fact, some psychological studies [3] have shown that we are able to achieve accuracies close to 96%, using images where the additional information of the hair has been eliminated. Different approaches can be found in the literature, which can be divided in two groups: geometry-based and appearance-based classification techniques.

  - In the first case, a set of features extracted from each image is used to train a classifier, for example some kind of distance (between eyes, eyebrows, etc..), or size (face, mouth and nose size,etc...). Brunelli et al. [27] used a set of 16 geometric features per image to train two hyper basis function networks, and achieved accuracies of 79% in a database composed of 168 training images. In another work of Burton [3], discriminant analysis was used over a set of 73 features (such as distances between key points, rations and angles formed by the key points, etc..), achieving a 85% of accuracy.

  - In the appearance-based models the classifier is trained using the whole image instead of using some geometric extracted features. In an experiment performed in Burton et al. ([24]), human subjects were asked to identify the gender of a set of pictures of faces and a set of 3-D laser-scanned representations of the same faces. The results showed that it was more difficult to discriminate between classes in the 3-D images, what suggest that features like global skin texture are very important in the gender recognition process. In a similar way another experiment was performed using the face pictures and the inverted pictures. The accuracy in the inverted pictures decreased significantly. Perhaps the most representative appearance-based method is the eigenface approach, Abdi et al. [60] trained a perceptron classifier using PCA-based features of the input images, achieving a performance of the 91.8%. Cottrell et al. [35] used a two layer neural network approach, where each face image was compressed in the first layer of the network, and classified in the second layer. They obtained an accuracy of 63% using only 64 training images. In a similar work of Golomb et al. [6] a system named SEXNET was used with a 91.9% accuracy. They used a neural network with 40 units to encode (compress) the 900 dimensional face image, and then they used two layers of 40 hidden units to classify the encodings. Tamura et al. [142] also used a neural network to identify sex achieving accuracies close to 90% even using reduced 8×8 central face images. Gutta et al. [141] proposed an hybrid approach using RBF networks and inductive decision trees achieving an accuracy of 96%. And Moghaddam et al. [5] obtained the best performance on gender recognition achieving 96.6% on a large face database (1755 faces), using SVM with RBF kernels.

- **Ethnicity recognition**, where the goal is to identify the human race of subjects.

In this thesis, we will develop different techniques to solve some of these problems, although we will often refer to the general concept of face classification instead of distinguishing an specific one. The concrete applications will be useful to benchmark the proposed algorithms.

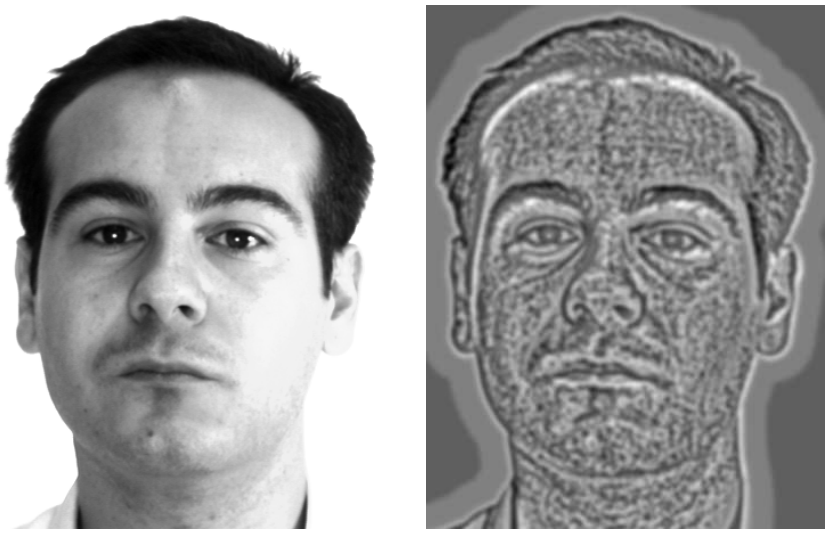
1.2 Feature Extraction

Each object that needs to be classified can be described by a set of characteristics called features. Features can be numerical (pressure, weight, speed, ...) or symbolic (material, profession, eye color, ...). In statistical pattern recognition we will only deal with numeric features. We consider each object $\mathbf{x}$ as a feature vector $\mathbf{x} = [x_1, \ldots, x_D]^T$ in a D-dimensional space, which is composed by the gray-level intensity of each pixel. Actually, the feature extraction process begins in the CCD of the acquisition devices. The measurements performed by sensor are quantized to a specific domain, and converted to pixel values.

Usually, in face classification schemes, a previous normalization step is needed to improve the accuracy. Faces acquired in natural environments suffer from problems of registration, partial illumination and gesture effects. In this thesis, a face alignment procedure has been performed in all the experiments:

- First the central position of each eye is selected, and images are rotated and scaled according to the inter-eye distance.
- A illumination correction is performed on each sample. Different normalization schemes have been used. The most simple approach is to remove the mean of each sample (related to global illumination), and divide by the variance on each pixel. More powerful normalization tools based on ridges and valleys have been developed [129]. In figure 1.5 and example of this local light normalization is shown.
- Finally the region of interest of each image is cropped according to the eye position in such a way that distance between eyes remains stable within subjects, and each thumbnail is reshaped to the final feature vector.

From now on, we will consider that faces have been properly acquired, normalized, and aligned constituting a data set $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]$, where each feature vector $\mathbf{x}$ belongs to $\Re^D$. Usually feature vectors that encode raw pixels from face images lay in high dimensional subspaces. This fact implies a tremendous increase in the computational cost to process each feature vector. Besides high dimensional spaces suffer from a problem known as *the curse of dimensionality* [130], which exponentially relates the amount of observations needed to model a single object with the dimensionality of the feature vectors representing it. This turns out to need a huge amount of data samples to estimate a good functional to identify it. Also, high dimensional spaces suffer from undesirable properties ([175, 51]) such as that almost every point lies on the distribution boundary (almost each point is closer to an edge than to another point), or that large neighborhoods are needed to enclose a small fraction of data points. On the other hand, it has been observed that there exists data redundancy in high dimensional vectors. Moreover, depending on the application, an efficient feature extraction can be beneficial for classification purposes, achieving a more accurate parameter estimation. Different dimensionality reduction techniques have been proposed in the literature to eliminate data redundancy. When the goal



(a) Original image. (b) Filtered image.

Figure 1.5: (a) Original image with partial illumination on the left side. (b) Illumination correction using gaussian filtering to extract the ridges and valleys from face images.

of the learning technique is to discover the structure of the data vectors without any prior information on class membership, it is known as *unsupervised* feature extraction. On the other hand, in *supervised* feature extraction, the algorithm uses the labels of the data to learn the new features. Therefore, for each original feature vector $\mathbf{x}_i$ there is a class label $c_j \in \{1, \ldots, K\}\ \forall j = 1, \ldots N$ associated. Supervised techniques can also learn class-invariant characteristics, or maximize some separability criteria, that improve the results of a predefined classifier.

Algorithms for feature extraction algorithms can be classified into linear and non linear. In this section a brief overview of feature extraction techniques and its application to face classification will be performed.

1.2.1 Unsupervised Linear Feature Extraction

Linear feature extraction techniques, in the most general formulation, can be expressed in terms of a linear projection from the original D-dimensional subspace to an M-dimensional one (usually M < D).

$$\mathbf{s} = \mathbf{A}\mathbf{x} \tag{1.1}$$

where $\mathbf{A}$ is an $M \times D$ projection matrix, and $\mathbf{s}$ the new extracted feature vector. According to the literature, the dimensionality reduction techniques can be classified in two categories: feature selection and feature extraction. In the feature selection

[69, 120] approach only a subset from the original feature vector is preserved. In feature extraction a projection matrix is used to combine the original features into the extracted. In this thesis, we will consider the feature selection as a special case of linear feature extraction where the selected features have coefficient 1 in the projection matrix $\mathbf{A}$, and 0 in the other features.

Principal Component Analysis

The basic idea of Principal Component Analysis (PCA) was first introduced by Pearson [73] in 1901, and it's also known as Karhunen-Loeve expansion and Hotelling Transform [63]. The first practical application of the algorithm appeared 3 years later, when Spearman [161] argued that a general factor of intelligence of its students could be obtained by a single linear combination of their school performance rankings. Later, in 1990, one of the first applications using PCA in face classification was performed by Kirby [76]. One year later, in 1991, Turk and Pentland introduced the notion of eigenfaces for classification, they used PCA to build a set of bases, and represented each face as a linear combinations of these bases.

The goal of any linear transformation applied to feature extraction is to find the set of vectors that span a new reduced subspace minimizing some specific criteria. Principal Component Analysis minimizes the mean square error (MSE) between the original and the projected data points. In addition, it has been shown that the linear transformation preserves the maximum variance in the projected space.

Suppose that we have the matrix $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]$ where each column is a D-dimensional training sample $\mathbf{x}$. We look for the linear transformation $\mathbf{A}$ that minimizes the mean square error criteria defined as [132]:

$$J = \sum_{n=1}^{N} \|\mathbf{x_n} - (\mathbf{A}\mathbf{x_n})\mathbf{A}^T\|^2 \tag{1.2}$$

To illustratively see how this projection can be obtained, an example in a 1-dimensional space will be shown. Without loss of generality, we suppose that the training column vectors $\mathbf{x}$ have mean 0. If it is not the case, new vectors $\widehat{\mathbf{x}}$ can be computed, and used instead:

$$\widehat{\mathbf{x}}_n = \boldsymbol{x_n} - m \tag{1.3}$$

where $\mathbf{m}$ is the mean

$$\mathbf{m} = \frac{1}{n}\sum_{n=1}^{N} \mathbf{x_n} \tag{1.4}$$

Supposing that only the best representation of each data point in a single line must be found, the goal will be to find the unitary vector $\mathbf{e}$ that defines the direction of that line.

$$\boldsymbol{x} = c\boldsymbol{e} \tag{1.5}$$

where $c_0, c_1, \dots c_N$ is the set of scalar coefficients in the direction of $\mathbf{e}$. According to the mean square error measure, the criteria to be minimized is:

$$J(c_0, c_1, \dots c_N, \mathbf{e}) = \sum_{n=1}^{N} \|\mathbf{x}_n - c_n\mathbf{e}\|^2 = \sum_{n=1}^{N} c_n^2\|\mathbf{e}\|^2 - 2\sum_{n=1}^{N} c_n\mathbf{e}\mathbf{x}_n + \sum_{n=1}^{N} \mathbf{x}_n^2 \quad (1.6)$$

The value of the scalar coefficients c_k can be calculated by differentiating J with respect to each coefficient, and equaling it to zero. As can be seen each coefficient will be:

$$c_n = \mathbf{e}\mathbf{x}_n \quad (1.7)$$

The next step is to calculate the direction $\mathbf{e}$ that bests reconstructs the data. Substituting the coefficients c_n for it's value from 1.7 in 1.6

$$J(\mathbf{e}) = \sum_{n=1}^{N} c_n^2 - 2\sum_{n=1}^{N} c_n^2 + \sum_{n=1}^{N} \mathbf{x}_n^2 = -\sum_{n=1}^{N} (\mathbf{e}\mathbf{x}_n)^2 + \sum_{n=1}^{N} \mathbf{x}_n^2 = -\sum_{n=1}^{N} \mathbf{e}^T\mathbf{x}_n\mathbf{x}_n^T\mathbf{e} + \sum_{n=1}^{N} \mathbf{x}_n^2 \quad (1.8)$$

and defining the scatter matrix $\mathbf{S_c}$ as:

$$\mathbf{S_c} = \sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^T \quad (1.9)$$

the eq. 1.8 becomes:

$$J(\mathbf{e}) = -\mathbf{e}^T\mathbf{S_c}\mathbf{e} + \sum_{n=1}^{N} x_n^2 \quad (1.10)$$

The vector $\mathbf{e}$ which minimizes eq. 1.10 is the vector that maximizes $-\mathbf{e}^T\mathbf{S_c}\mathbf{e}$. To find it the method of Lagrange multipliers can be used, subject to the constraint $\|\mathbf{e}\| = 1$. Using λ as a Lagrange multiplier, the equation to differentiate with respect to $\mathbf{e}$ is:

$$u = \mathbf{e}^T\mathbf{S_c}\mathbf{e} - \lambda(\mathbf{e}^T\mathbf{e} - 1) \quad (1.11)$$

$$\frac{\partial u}{\partial \mathbf{e}} = 2\mathbf{S_c}\mathbf{e} - 2\lambda\mathbf{e} \quad (1.12)$$

And after making it equal to zero to reach the maximum, we obtain $\mathbf{S_c}\mathbf{e} = \lambda\mathbf{e}$. The vector $\mathbf{e}$ that fulfils this equation must be an eigenvector of the scatter matrix $\mathbf{S_c}$. In fact we can see that $\mathbf{e}$ must be the eigenvector with largest eigenvalue λ, because the maximum of $\mathbf{e}^T\mathbf{S_c}\mathbf{e} = \lambda\mathbf{e}^T\mathbf{e} = \lambda$ is needed.

Using a similar process we can extend it to find M-dimensional spaces. In general, the PCA projection matrix from a D to an M-dimensional space can be found by estimating the M eigenvectors with largest eigenvalue of the covariance matrix of the data. The set of coefficients $\mathbf{c}$ are called the *principal components*.

Another interpretation of PCA can be considered in terms of finding the directions of maximum variance. The eigenvectors of the covariance matrix define the set of the orthogonal axis with maximum variance, while the eigenvalues define the variances

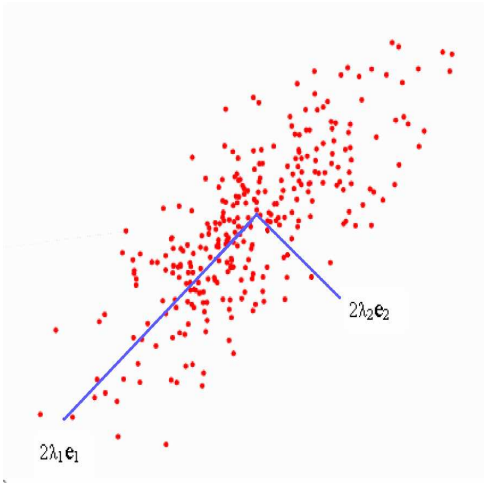


Figure 1.6: PCA axis for an artificial 2D-Gaussian data set.

along this axis. This fact is illustrated in figure 1.6, where data sampled from a 2-dimensional Gaussian is plotted, the axis of the distribution coincide with the PCA eigenvectors, and the variance is directly related to the eigenvalues.

In all the feature extraction techniques there is a loss of information. To measure the amount of variance preserved using only the M first PCA components, the ratio between the preserved variance and the original can be computed by adding the first M eigenvalues and dividing it by the trace of the scatter matrix:

$$R = 100 * \frac{\sum_{d=1}^{D} \lambda_m}{trace(\mathbf{S_c})} \tag{1.13}$$

Usually, before performing the feature extraction a percentage of the variance to be preserved is defined, and the eigenvalues of the scatter matrix $\mathbf{S_c}$ are ordered $\lambda_1 > \lambda_2 > \ldots \lambda_n$, and selected according to this percentage.

Although there is a loss of information using PCA, the technique has been proven to be effective in face classification, often improving the results of the classifiers in the original spaces. This is due to the fact that PCA reduces the dimensionality by eliminating the directions of small variance, which are commonly associated to noise. Therefore, PCA eliminates the noisy components of data in the dimensionality reduction process.

Another important characteristic of PCA projection is that the features extracted are always mutually uncorrelated, and in the case of gaussian data the features are independent. An application of this fact is the data *whitening*, and it is often used as a preprocessing step in many classification tasks to provide invariance to scale in the features. Given the matrices $\mathbf{V} = diag\{\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D}\}$ with the M largest eigenvalues, and $\mathbf{A}$ the PCA linear projection, the data vector x is whitened performing the

(a) (b)

Figure 1.7: (a) Orignal images with lateral illumination (b) Normalized images using the mutual information.

transform:

$$y = V^{\frac{-1}{2}} A(\mathbf{x} - \overline{\mathbf{x}}) \tag{1.14}$$

where $\overline{\mathbf{x}}$ is the sample mean of the vectors $\mathbf{x}_i$. Projecting the data using the matrix $V^{\frac{-1}{2}}\mathbf{A}$, the dimensionality is reduced and also the data become uncorrelated and have unit variance.

Principal Component Analysis is possibly one of the most used techniques in feature extraction, and multiple improvements over the classic algorithm have appeared, such as weighted PCA [157], probabilistic PCA [170], kernel PCA [108], or Robust PCA [82].

Bart and Ullman [8] used the Principal Component Analysis algorithm for a new class-based image normalization method. In their approach they use the mutual information to identify and select the most relevant components instead of taking into account the correspondent eigenvalues. They defined the mutual information between each class and the feature strength as:

$$I_\theta(C, F) = \sum_{c=1}^{K} p(C = c, F = f) log \frac{p(C = c, F = f)}{p(C = c)p(F = f)} \tag{1.15}$$

where the strength is a discrete representation of each feature from the PCA algorithm using the threshold θ, and c is the class of each sample. The algorithm discards the non-informative components, those that provide less information about the class membership of the sample, preserving the ones that have higher mutual information. Figure 1.7 shows an illumination normalization example extracted from [8], with two samples from the ARFace database with strong lateral illumination. The normalized images are obtained reconstructing using only 82 preserved components (with larger mutual information). As can be seen the normalization obtained is very accurate.

Eigenfaces example

Face recognition applications are an important field where PCA projections have been specially used. The first application for characterization of human faces was

done in [76], and one year later Turk and Pentland (see [173]) developed a real-time system to locate, track and recognize human faces. They extracted the most significant features in terms of variation, which were called *eigenfaces*. Each image was represented as a weighted sum of this set of bases, and only these weights were compared to recognize each face.

In Figure 1.8 the first 100 eigenfaces of a set of 500 face images extracted from the AR Face database [100] are shown. Using only 100 eigenfaces a 91.83 % of the input variance was preserved. It has been shown that the first eigenfaces are strongly related to illumination and low frequencies, while the larger ones are related to high frequencies. As can be seen in Figure 1.9, the most part of the variance is preserved using a small subset of the first eigenfaces. We also show how using only the first 100 bases, we are able to reconstruct a very similar face to the original one. The figure shows also how the reconstruction is affected when the dimensionality of the subspace is reduced. Using only 10 or 5 eigenfaces, the recognition of the person becomes unfeasible.

In a more recent application to face recognition, Moghaddam et al. [111, 110], used the PCA algorithm to divide the $\Re^D$ space into two complementary subspaces using the basis defined by the M eigenvectors of largest eigenvalue, and the residual of the PCA expansion. They defined a similarity measure based on the Bayes rule on intra-personal and extra-personal variations, achieving a 10% gain in performance in the 1996 FERET competition.

In this thesis the basic PCA algorithm has often been used as a benchmark to compare with the feature extraction techniques proposed.

Independent Component Analysis

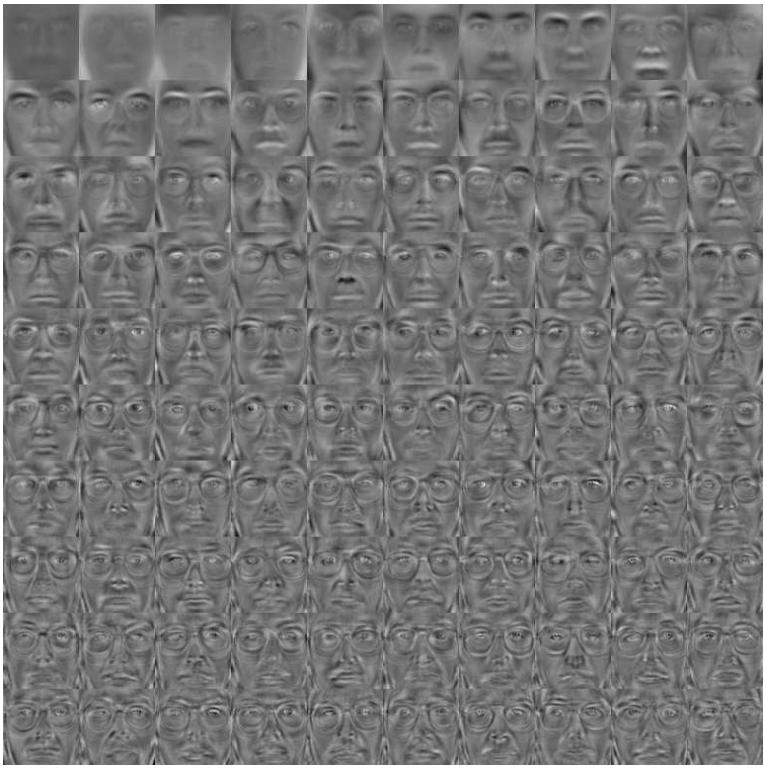
The main limitation of Principal Component Analysis lies on the gaussianity assumption made on the distribution of the input data. PCA computes the axis of maximum variance of the input data taking into account only the covariance matrix of the training vectors, remaining completely blind to high order statistics. Independent component analysis (ICA) overcomes this drawback by finding the axis where the projected vectors are statistically independent ([65]).

Independent component analysis is a technique originally developed to solve the problem of separating a set of signals which have been mixed in a linear way (Blind Source Separation). The most relevant assumption made by ICA to solve it is that signals are produced by mixing independent sources. The general formulation is:

$$\mathbf{x} = \mathbf{B}\mathbf{s} \tag{1.16}$$

where $\mathbf{x}$ is the observed random vector, $\mathbf{s}$ are the latent components, and $\mathbf{B}$ is the mixing matrix, which are both completely unknown and must be estimated assuming only statistical independence on the mixture components. Usually the mixture matrix is assumed to be square, although it is not strictly necessary. Different approaches can be followed to obtain the solution to the ICA problem: maximum likelihood

(a)



(b)

Figure 1.8: (a) The mean face image of a training set made up of 500 face images. (b) The 100 first eigenfaces of the same training set.

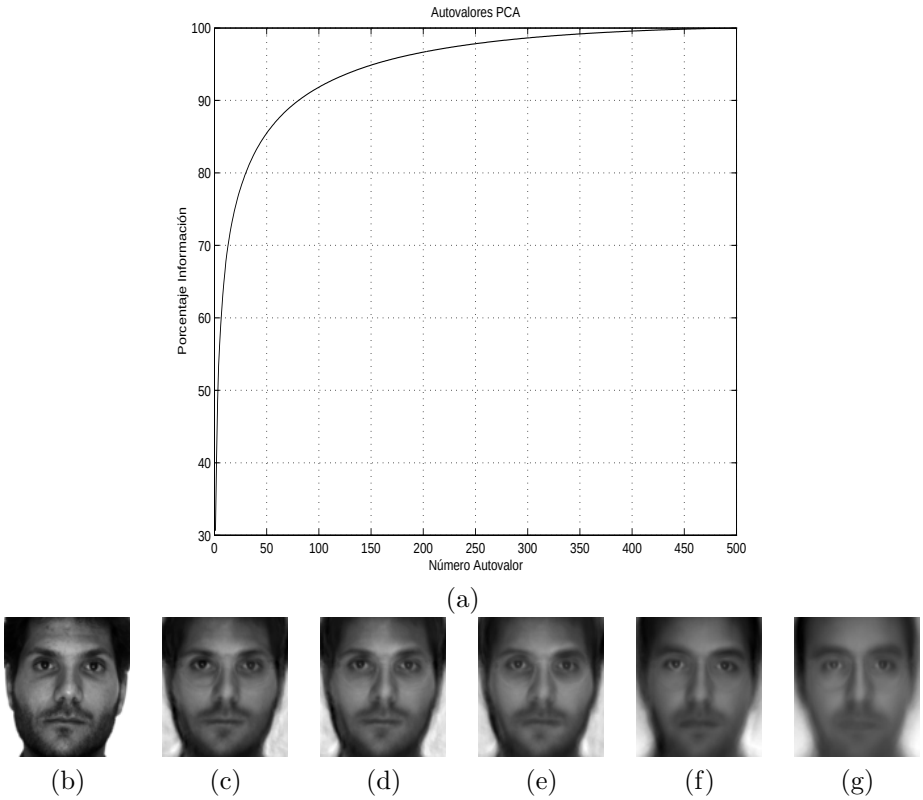


Figure 1.9: (a) Percentage of variance preserved (b) Example of face used in the data set (c) Result of reconstructing this face using the first 100 coefficients obtained from the PCA projection (d) The same as (c) but using only 75 coefficients (e) 50 coefficients (f) 10 coefficients (g) and only 5 coefficients.

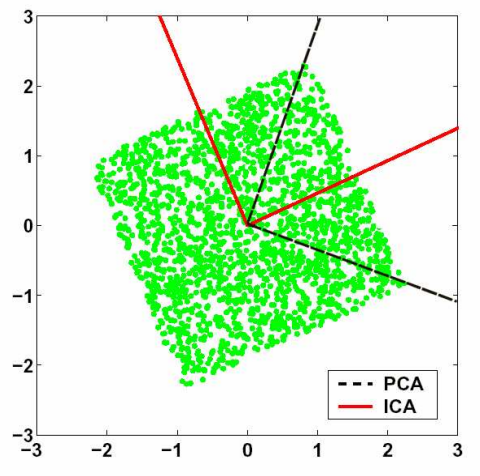


Figure 1.10: Example of the axis found using PCA and ICA in a uniformly distributed data

estimation of the parameters (see chapter 9 of [65] for further details), by maximizing the non-gaussianity of the independent components [65], or by minimizing the mutual information [33]. In Figure 1.10, we can see the axis found by ICA and PCA, in a 2-dimensional uniformly distributed data [37]. PCA founds the axis that maximize the variance, but the projections in those axis are still mixed, on the other hand ICA is able to find the projections of the independent components.

Feature extraction using ICA has been successfully applied in face classification tasks. Bartlett et al.[9] combined two different ICA architectures for a face recognition problem, obtaining better performances than using PCA. In addition, as there are different algorithms for solving the ICA problem, different results can be obtained depending on the option chosen. Draper et al.[26] successfully used the ICA algorithm to face classification tasks, they compared the FastICA [66] and the Infomax [10] algorithms with PCA in a face identification and a facial expression recognition problem using the FERET database, and they concluded that the best accuracies in face identification are obtained using FastICA, while Infomax algorithm performed better for facial action recognition.

Non negative matrix factorization

The PCA transformation is only constrained by minimizing the reconstruction error, and it obtains global holistic bases to represent objects. Lee and Seung [86] designed a new algorithm called non Negative Matrix Factorization (NMF), based on adding a positivity constraint. Both the basis matrix $\mathbf{B}$ and the coefficients $\mathbf{S}$ of the factorization are forced to be non negative. As in computer vision we usually work with positive data descriptions (pixel values) it seems plausible to use the positive data instead of PCA or ICA representations. Moreover, the NMF finds a parts-

based representation of the objects that shares similarity with biological systems [93, 121, 182]. This fact implies important advantages when dealing with occlusions and perturbations that affect to specific locations of the objects (local changes in illumination).

NMF also tries to minimize the mean square error criteria as PCA, but the key point in the non negative matrix factorization is the use of the non negativity constrains applied to the set of bases and to the coefficients that represent each vector in the reduced space. This non negativity constraint provides sparse bases due to the fact that the weighted combination of the bases is always additive, so bases can not have big portions active because once a big portion of an image is used as a base, it can not be removed (due to the constraint of additivity).

The NMF algorithm takes a matrix $\mathbf{X}$ of N D-dimensional vectors (all of them nonnegative). The goal is to find a good approximation of $\mathbf{X}$ as:

$$\mathbf{X}_{ij} \approx (\mathbf{B} * \mathbf{S})_{ij} = \sum_{d=1}^{D} \mathbf{B}_{id}\mathbf{S}_{dj} \tag{1.17}$$

where the $D \times M$ matrix $\mathbf{B}$ contains the set of bases, and the $M \times N$ $\mathbf{S}$ matrix are the weights corresponding on each base per each vector. The estimation of the matrix factorization is performed optimizing an objective ([87]) function defined as:

$$F = \sum_{i=1}^{N}\sum_{j=1}^{M}[\mathbf{X}_{ij} log(\mathbf{BS})_{ij} - (\mathbf{BS})_{ij}] \tag{1.18}$$

Which can be viewed as the likelihood of generating the images $\mathbf{X}$ from the bases $\mathbf{B}$ and the weights $\mathbf{S}$. But due to the non negativity constraint, it is not possible to find the analytical solution to this expression. A gradient ascent can be used to solve it iteratively. The optimization of this function can be achieved using the iterative rules:

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \sum_{d} \frac{\mathbf{X}_{id}}{(\mathbf{BS})_{id}}\mathbf{S}_{jd} \tag{1.19}$$

$$\mathbf{B}_{ij} \leftarrow \frac{\mathbf{B}_{ij}}{\sum_k B_{kj}} \tag{1.20}$$

$$\mathbf{S}_{jd} \leftarrow \mathbf{S}_{jd} \sum_{i} B_{ij} \frac{\mathbf{X}_{id}}{(\mathbf{BS})_{id}} \tag{1.21}$$

One problem that arises from the non negative nature of the technique is the projection of the new unseen samples. The NMF algorithm uses additions of non negative bases, so the problem of projecting new unseen vectors must be faced carefully. In the general linear dimensionality reduction approach, each time that a new unseen vector must be projected we perform:

$$\mathbf{S} = \mathbf{AX} \tag{1.22}$$

where $\mathbf{A}$ in the NMF case is the inverse matrix of the NMF bases $\mathbf{B}$. Nevertheless, the resulting projection matrix computed in this way can have negative values, infringing

the non negativity constraint. To overcome this drawback the projected vectors are found by running a few steps of the iterative algorithm, using as a input matrix $\mathbf{X}$ the new unseen vectors, and fixing the bases to the ones found in the training step (only the new weight coefficients are learned). The resulting weights are always non negative, and the algorithm converges after a few iterations, usually 50-100 depending on the dimensionality.

In most of the cases the additive parts based bases found by the NMF algorithm are not necessary localized. Stan Li et al. [90] proposed a variation of the algorithm called local non negative matrix Factorization, for learning more localized parts based bases. To reach this goal, three new constraints are added to the non negativity constraint of classic NMF.

First the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are defined as $\boldsymbol{U} = \mathbf{B}^T\mathbf{B}$ and $\boldsymbol{V} = \mathbf{S}\mathbf{S}^T$ both $M \times M$, and the new constraints added are:

1. The number of bases required to present $\mathbf{X}$ should be minimized. The bases should not be further decomposed in more components. This can be achieved by minimizing $\sum_i \boldsymbol{U}_{ii}$.
2. It's necessary minimize the redundancy between bases, this can be achieved by minimizing $\sum_{i \neq j} \boldsymbol{U}_{ij}$ making the basis as orthogonal as possible.
3. The components giving the most important information should be favored. This can be achieved by maximizing $\sum_i \boldsymbol{V}_{ii}$.

The final update rules using the new restrictions become (see [90] for more details):

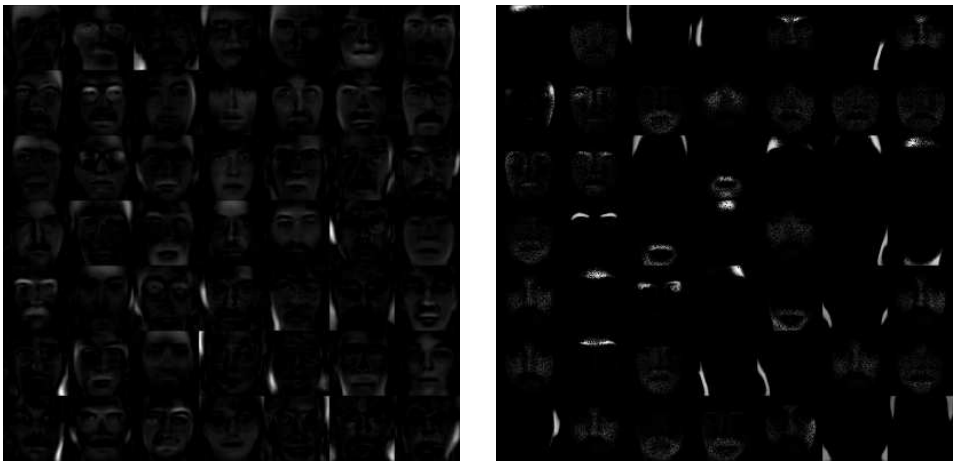
$$\mathbf{B}_{ij} \leftarrow \frac{\mathbf{B}_{ij} \sum_n \mathbf{X}_{in} \frac{\mathbf{S}_{jn}}{\sum_k \mathbf{B}_{ik}\mathbf{S}_{kn}}}{\sum_n \mathbf{S}_{jn}} \tag{1.23}$$

$$\mathbf{B}_{ij} \leftarrow \frac{\mathbf{B}_{ij}}{\sum_k B_{kj}} \tag{1.24}$$

$$\mathbf{S}_{ij} \leftarrow \sqrt{\mathbf{S}_{ij} \sum_d \mathbf{X}_{dj} \frac{\mathbf{B}_{di}}{\sum_k (\mathbf{B}_{dk}\mathbf{S}_{kj})}} \tag{1.25}$$

In Figure 1.11 an example of bases found using 500 face images is shown. LNMF is able to find more localized bases. We can see how some bases capture specific characteristics of faces, such as eyebrows, mouth, chin. This fact is very useful when dealing with occlusions. In general, NMF based algorithms perform worse than PCA and the other techniques used in classification experiments, but usually it is possible to obtain the bests results in occlusion problems, because they only affect some specific bases due to its sparse nature.

On the other hand, the reconstruction error of NMF is close to the one obtained using PCA, but using the LNMF algorithm, the reconstruction results are very poor.



(a) (b)

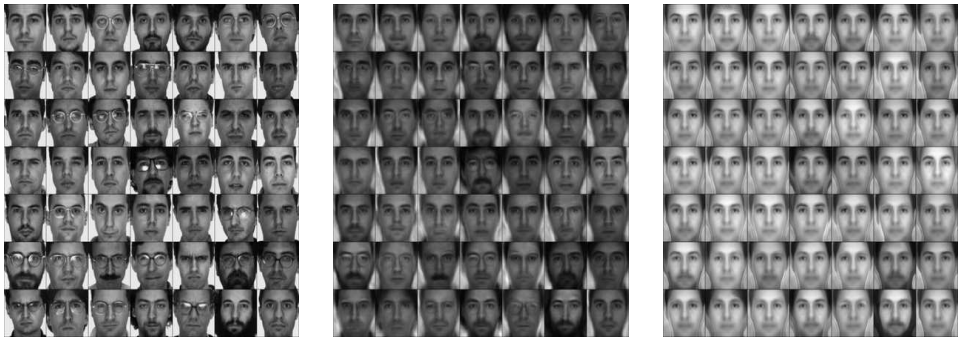
Figure 1.11: Example of 49 bases found on a 500 faces image data set. (a) Bases found using NMF (b) Bases found using LNMF. In the second case the bases vectors are more localized.

LNMF provides worse reconstructed faces, even using more iterations in the convergence process. In Figure 1.12 this problem is illustrated, 500 faces were used to learn 49 NMF and 49 local NMF bases, a large amount of the discriminant visual features is lost using the LNMF algorithm.

The NMF algorithm has been used in face recognition [86] due to the robustness against partial occlusions. Chen et al. [186], used the algorithm for feature extraction in a face detection application using Adaboost. Guillamet et al. [42, 58] introduced a weighted version of the algorithm to focus the calculation on some specific samples using a set of weights. Later in this thesis, in chapter 2, we show a face detection scheme that uses the weighted NMF algorithm combined with boosting, in order to reach an adaptive feature extraction that improves the results obtained using a fixed feature extraction.

1.2.2 Supervised Linear Feature Extraction

The feature extraction techniques previously presented, perform a linear transformation that minimizes a reconstruction criteria, under some constraints. But the best features to reconstruct the original samples are not always the best features for classification. In Figure 1.13 a toy example in a 2-dimensional space is shown. Projecting the data on the axis of maximum variance is not the best feature extraction option for classification, while the second direction separates perfectly the data. In visual object recognition, when often data samples lie on high dimensional subspaces, the discriminant features can be a small subset of the original ones. For example, suppose a manuscript digit recognition problem, where the digits "0" and "8" must



(a) (b) (c)

Figure 1.12: Example of the reconstruction error using LNMF and NMF. (a) Some faces of the original 500 data set.(b) Faces reconstructed from the weights and the 49 bases found using NMF. (c) Faces reconstructed from the weights and the 49 bases found using LNMF.

be classified. The best way to separate both classes should be to learn features based on the small portion of the center of each digit from the training samples, but unsupervised techniques would not give too much importance to this discriminative fact. Feature extraction techniques can benefit from prior knowledge of class membership of the training samples. Alternative approaches for supervised linear feature extraction have been proposed based upon different criteria and assumptions made on the training data. In this section a brief review of the discriminant analysis techniques used in this thesis will be performed. First the classic Fisher Linear Discriminant Analysis technique (FLD) will be introduced, to show later how its nonparametric extension [53] can solve the main drawbacks of FLD:

- Gaussian assumption on the class distribution of the input data.
- And a limitation of the final dimension in the projected subspaces.

Then a modification of the classic NDA proposed by Bressan ([23] and [103]) which improves the nearest neighbor classification will be shown. Finally, a recent discriminant analysis technique based on the Chernoff criterion will be explained.

Fisher Discriminant Analysis

The goal of discriminant analysis is to find the features that best separate the different classes maximizing the criterion $\mathcal{J}$:

$$\mathcal{J} = tr(\boldsymbol{S}_B \boldsymbol{S}_W) \tag{1.26}$$

where the matrices $\mathbf{S}_B$ and $\mathbf{S}_W$, generally represent the scatter of sample vectors between different classes and within a class respectively. It has been shown [39, 52]

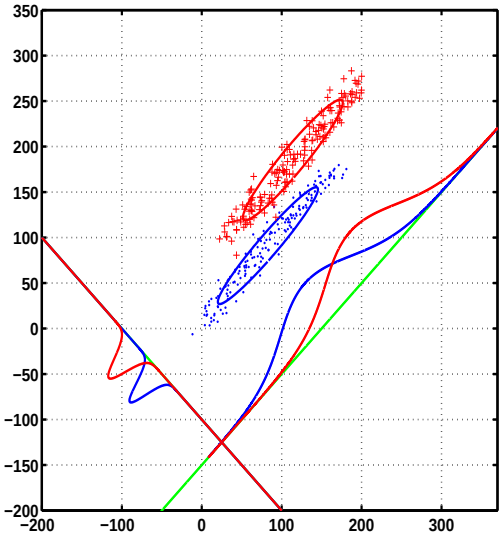


Figure 1.13: Toy example where two classes are plotted. The axis of maximum variance are not the axis most suitable for classification.

that the $M \times D$ linear transform that satisfies:

$$\hat{\mathbf{A}} = \arg \max_{\mathbf{A}^T \mathbf{S}_W \mathbf{A} = \mathbf{I}} tr(\mathbf{A}^T \mathbf{S}_B \mathbf{A}) \tag{1.27}$$

optimizes the separability measure $\mathcal{J}$. This problem has an analytical solution based on the eigenvectors of the scatter matrices. The algorithm presented in table 1.1 obtains this solution [52]. The most widely spread approach for defining the within and between class scatter matrices is the one that makes use of only up to second order statistics of the data. This was done in a classic paper by Fisher [48] and the technique is referred to as Fisher Discriminant Analysis (FLD). In FLD the within class scatter matrix is usually computed as a weighted sum of the class-conditional sample covariance matrices. If equiprobable priors are assumed for classes c_k, $k = 1, \ldots, K$ then

$$\mathbf{S}_W = \frac{1}{K} \sum_{k=1}^{K} \mathbf{\Sigma}_k \tag{1.28}$$

where $\mathbf{\Sigma}_k$ is the class-conditional covariance matrix, estimated from the sample set. The between class-scatter matrix is defined as,

$$\mathbf{S}_B = \frac{1}{K} \sum_{k=1}^{K} (\mu_k - \mu_0)(\mu_k - \mu_0)^T \tag{1.29}$$

where μ_k is the class-conditional sample mean and μ_0 is the unconditional (global) sample mean.

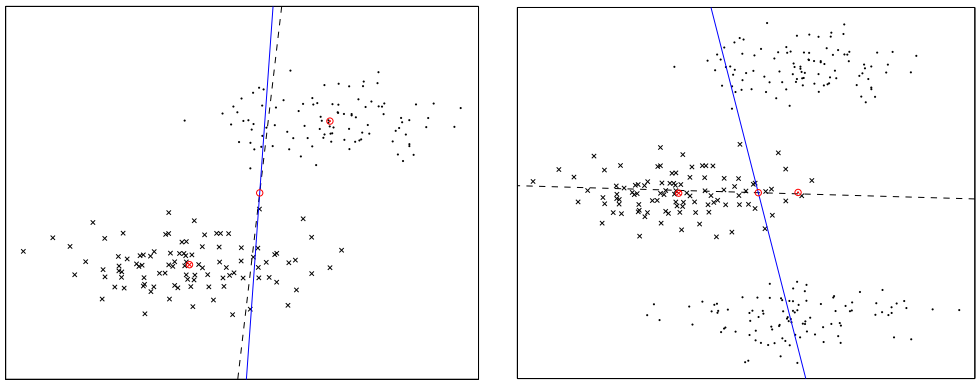
Table 1.1: General algorithm for solving the discriminability optimization problem stated in equation (1.27) given the scatter matrices $\mathbf{S}_B$ and $\mathbf{S}_W$.

1. Given $\mathbf{X}$ the matrix containing data samples placed as N D-dimensional columns, $\mathbf{S}_W$ the within class scatter matrix, and M maximum dimension of discriminant space,
2. Compute eigenvectors and eigenvalues for $\mathbf{S}_W$. Make $\mathbf{\Phi}$ the matrix with the eigenvectors placed as columns and $\mathbf{\Lambda}$ the diagonal matrix with only the nonzero eigenvalues in the diagonal. M_W is the number of non-zero eigenvalues.
3. Whiten the data with respect to $\mathbf{S}_W$, to obtain M_W dimensional whitened data,
$$\mathbf{Z} = \mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^T\mathbf{X}$$
4. Compute $\mathbf{S}_B$ on the whitened data.
5. Compute eigenvectors and eigenvalues for $\mathbf{S}_B$ and make $\mathbf{\Psi}$ the matrix with the eigenvectors placed as columns and sorted by decreasing eigenvalue.
6. Preserve only the first $M_B = \min\{M_W, M, \mathsf{rank}(\mathbf{S}_B)\}$ columns, $\mathbf{\Psi}_M = \{\psi_1, \ldots, \psi_{M_W}\}$ (those corresponding to the M_B largest eigenvalues).
7. The resulting optimal transformation is $\hat{\mathbf{A}} = {\mathbf{\Psi_M}}^T\mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^T$ and the projected data, $\mathbf{Y} = \hat{\mathbf{A}}\mathbf{X} = \mathbf{\Psi}_M^T\mathbf{Z}$

Notice the rank of $\mathbf{S}_B$ is $K-1$, so the number of extracted features is, at most, one less than the number of classes. The problem can be artificially solved increasing the number of classes using clustering algorithms. Also notice the parametric nature of the scatter matrix. The solution provided by FLD is blind beyond second-order statistics. So we cannot expect the FLD method to accurately indicate which features should be extracted to preserve any complex classification structure, for example in multimodal class distributions, where a Gaussian is assumed for the whole class.

Nonparametric Discriminant Analysis

In [53] Fukunaga and Mantock present a nonparametric method for discriminant analysis (NDA) in an attempt to overcome the limitations of FLD. In nonparametric discriminant analysis the between-class scatter $\mathbf{S}_B$ is of nonparametric nature. This scatter matrix is generally full rank, thus loosening the bound on extracted feature dimensionality. Also, there is no gaussian assumption on the class distribution of the input data, only some gaussian behavior is assumed in the distribution of the distances between points of different classes, which is a more relaxed condition. Now this technique will be briefly described, and it's extensively detailed in [52].



(a) Two Gaussian classes (b) One Gaussian and one bimodal class

Figure 1.14: First directions of NDA (solid line) and FLD (dashed line) projections, for two artificial datasets. Observe the results in the right-hand figure, where the FLD assumptions are not met.

In NDA, the between-class scatter matrix is obtained as an average of N local covariance matrices, one for each point in the data set. This is done as follows. Let $\mathbf{x}$ be a data point in $\mathbf{X}$ with class label c_j. Denote by $x^{\text{different}}$ the subset of the k nearest neighbors of $\mathbf{x}$ among the data points in $\mathbf{X}$ with class labels different from c_j. We calculate the "local" between-class matrix for $\mathbf{x}$ as

$$\Delta_B^{\mathbf{x}} = \frac{1}{k-1} \sum_{\mathbf{z} \in x^{\text{different}}} (\mathbf{z} - \mathbf{x})(\mathbf{z} - \mathbf{x})^T \tag{1.30}$$

The estimate of the between-class scatter matrix $\mathbf{S}_B$ is found as the average of the local matrices

$$\mathbf{S}_B = \frac{1}{N} \sum_{\mathbf{z} \in X} \Delta_B^{\mathbf{z}} \tag{1.31}$$

Using $k = 1$, the subset $x^{\text{different}}$ contains only one element, $\mathbf{z}_{\mathbf{x}}^{\text{different}}$, and

$$\mathbf{S}_B = \frac{1}{N} \sum_{\mathbf{x} \in X} (\mathbf{x} - \mathbf{z}_{\mathbf{x}}^{\text{different}})(\mathbf{x} - \mathbf{z}_{\mathbf{x}}^{\text{different}})^T. \tag{1.32}$$

A parametric form is chosen for the within-class scatter matrix $\mathbf{S}_W$, defined as in 1.28. Figure (1.14) illustrates the differences between NDA and FLD in two artificial datasets, one with Gaussian classes where results are similar, and one where FLD assumptions are not met. For the second case, the bimodality of one of the classes displaces the class mean introducing errors in the estimate of the parametric version of $\mathbf{S}_B$. The nonparametric version is not affected by this situation, and classes are more separable in the projection found.

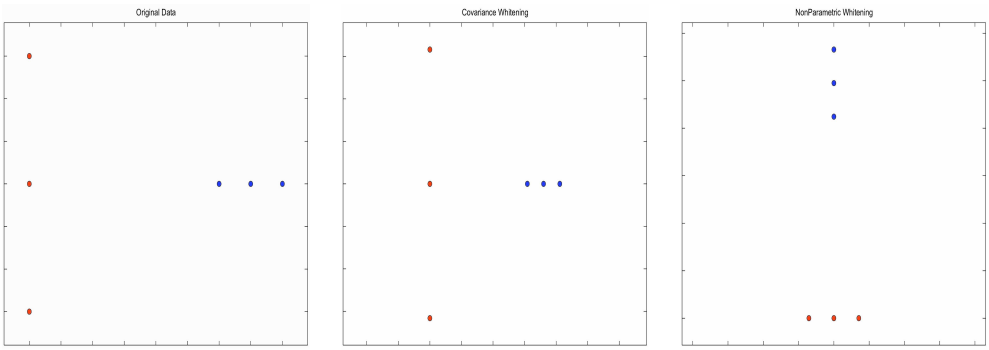


Figure 1.15: Example where the whitening has been performed using a parametric form of the within class scatter matric (b) and a non parametric one (c) using a toy data set (a) composed by 2 classes

M. Bressan and Vitrià [22] introduced also a non parametric form of the within-class scatter matrix $\mathbf{S}_W$, which is expected to provide features which work well with the nearest neighbor classifier. They propose to use

$$\mathbf{S}_W = \frac{1}{N} \sum_{\mathbf{z} \in X} \Delta_W^{\mathbf{z}} \tag{1.33}$$

where $\Delta_W^{\mathbf{x}}$ is calculated from the set of k nearest neighbors of $\mathbf{x}$, x^{same}, from the same class label c_j.

$$\Delta_W^{\mathbf{x}} = \frac{1}{k-1} \sum_{\mathbf{z} \in x^{\text{same}}} (\mathbf{z} - \mathbf{x})(\mathbf{z} - \mathbf{x})^T \tag{1.34}$$

For $k = 1$,

$$\mathbf{S}_W = \frac{1}{N} \sum_{\mathbf{x} \in X} (\mathbf{x} - \mathbf{z}_{\mathbf{x}}^{\text{same}})(\mathbf{x} - \mathbf{z}_{\mathbf{x}}^{\text{same}})^T. \tag{1.35}$$

In this thesis we use the local approximations of both $\mathbf{S}_B$ and $\mathbf{S}_W$, as in [22].The influence of the new within-class scatter matrix can be seen in the whitening step. The non parametric form of $\mathbf{S}_W$, normalizes the data according to the distances of each point to the nearest neighbors of the same class as can be seen in Figure 1.15. There is no gaussian assumption in the distribution of the points of the same class, in fact, the assumption now is in the distribution of the distances between the nearest neighbors of points of the same class, which is less restrictive.

Heteroscedastic LDA (Chernoff)

Recently Loog and Duin [95] extended the LDA criterion for the case of Gaussian classes with different covariance matrices (heteroscedastic data). They replace

the criterion (1.26) with the Chernoff criterion which is expected to be superior for heteroscedastic data. Their experiments show encouraging results.

Let $\mathbf{m}_1$ and $\mathbf{m}_2$ be the estimated means of the two classes, p_1 and p_2 be the estimated prior probabilities ($p_2 = 1 - p_1$), and $\mathbf{S}_1$ and $\mathbf{S}_2$ be the respective covariance matrices. For simplicity of notation, denote by $\mathbf{S}$ the within-class covariance matrix $\mathbf{S}_W$ calculated as $\mathbf{S} = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$. The criterion matrix is again $\mathcal{M} = \mathbf{S}_W^{-1}\mathbf{S}_B$ where $\mathbf{S}_B$ is calculated as

$$\begin{aligned}\mathbf{S}_B &= p_1\, p_2\, \mathbf{S}^{\frac{1}{2}} \left(\mathbf{S}^{-\frac{1}{2}}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{S}^{-\frac{1}{2}} \right. \\ &\quad \left. - \frac{1}{p_2} \log\left(\mathbf{S}^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}^{-\frac{1}{2}}\right) - \frac{1}{p_1} \log\left(\mathbf{S}^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}^{-\frac{1}{2}}\right) \right) \mathbf{S}^{\frac{1}{2}} \end{aligned} \tag{1.36}$$

Here a function f (logarithm or power) of a matrix $\mathbf{A}$ is calculated in the following way. Let $\mathbf{VDV}^{-1}$ be the eigenvalue decomposition of $\mathbf{A}$, i.e., $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix with the eigenvalues on the leading diagonal. The function is applied to the eigenvalues and the results are placed at the leading diagonal of a diagonal matrix, denoted $f(\mathbf{D})$. Then $f(\mathbf{A}) = Vf(\mathbf{D})\mathbf{V}^{-1}$.

For equal covariance matrices $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}$, the Chernoff matrix $\mathbf{S}_B$ in (1.36) is the same as the matrix $\mathbf{S}_B$ of FLD.

Figure 1.16 shows four two-class problems and the projections obtained through FLD, NDA and Chernoff. In subplots (a) and (c) both classes have Gaussian distribution, with the same covariance matrices. The Chernoff and FLD projections overlap, and NDA also yields a similar projection. On the other hand, in (b) and (d) only one of the two classes is Gaussian. The bimodality of one of the classes displaces the class mean introducing errors in the estimate of the parametric version of $\mathbf{S}_B$ in FLD. NDA and Chernoff are not affected by this for small noise levels. Also notice that Chernoff seems to tolerate noise better than NDA. Subplot (d) shows that the projection selected by NDA is affected by noise on the y-axis while the Chernoff projection is not.

In face classification, many variants of linear discriminant analysis have appeared. Swets and Weng [165] compared the performance of principal component analysis and FLD in a face recognition. Belhumeur et al. [119], proposed a system called *fisherfaces*, to find the invariant features against strong changes in the illumination in a face recognition problem. In [104] the modified NDA technique combined with the nearest neighbor classifier has been used in a gender recognition experiment.

1.2.3 Nonlinear Feature Extraction

Some data sets lie on high dimensional non linear manifolds that can not be properly learned using linear projections. For example: let's suppose that we have a face, and a camera recording the face and moving itself around the x axis. The result will be a large amount of frames with the face of the person captured under different points of view. Each face image will be a high dimensional data point, but in fact,

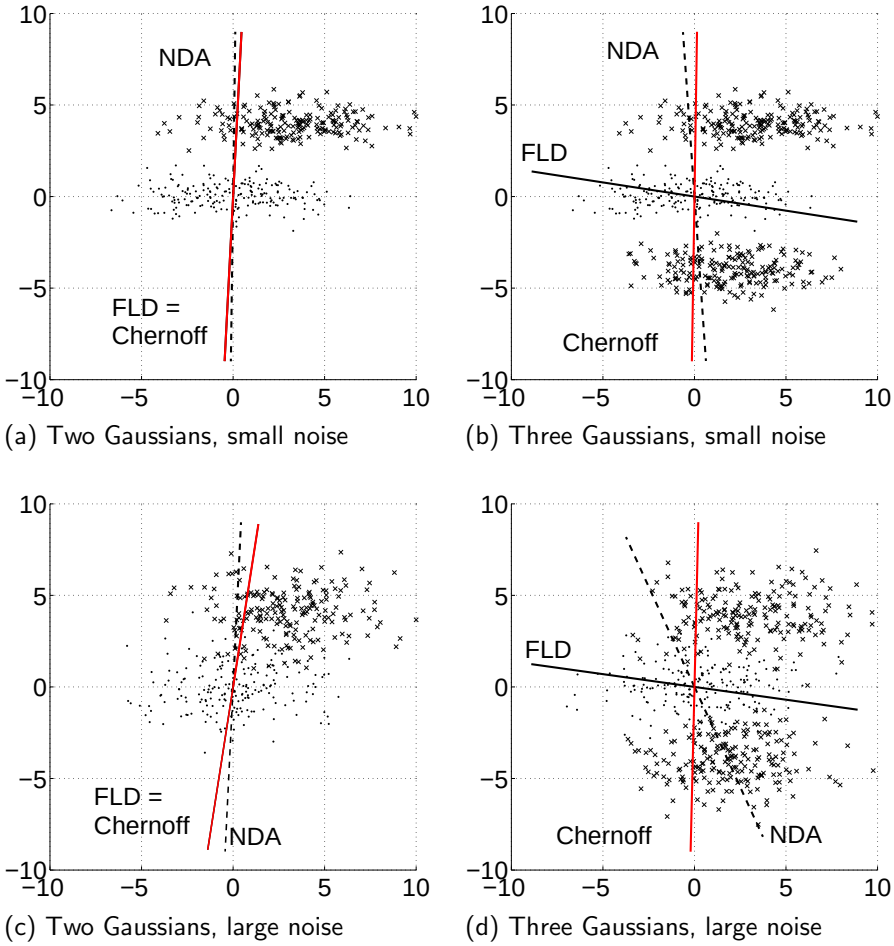


Figure 1.16: Examples of FLD, NDA and Chernoff for Gaussian ((a) and (c)) and non-Gaussian ((b) and (d)) classes for two levels of noise on the y-axis.

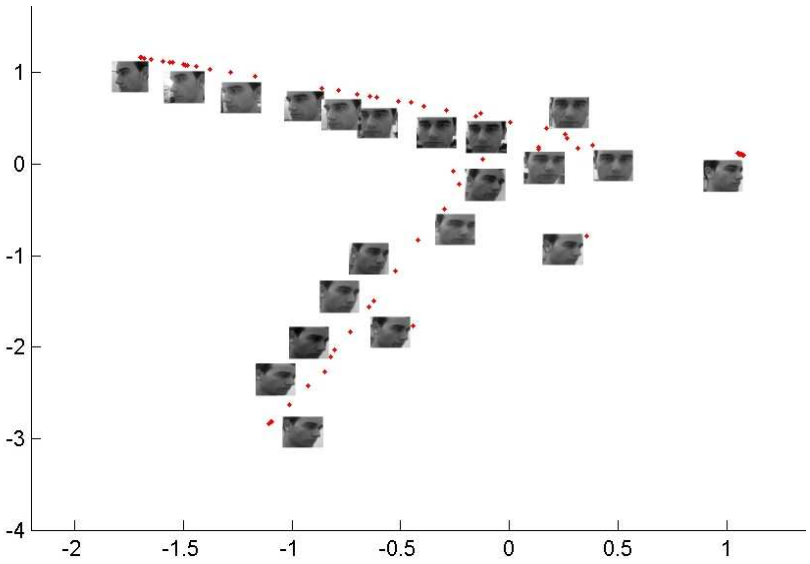


Figure 1.17: Example of the 2-dimensional plot made using a non linear dimensionality reduction technique (the locally linear embedding) on a set of face images captured under different points of view of the camera. It can be observed a relationship between the face orientation and the situation of each point in the 2D mapping.

we could consider that the data lies in very low dimensional subspace, because if we have the first face picture, there is only one degree of freedom in the generation of the other face images (the camera movement around the x axis). So the goal will be to try to find the nonlinear projections that can provide us the underlying manifold of the data. Two important non linear techniques have recently appeared to overcome this problem: Isomap and Locally Linear Embedding (LLE). Both techniques try to find the underlying non linear structure of the data, preserving the neighborhood of each data sample. In Figure 1.17, the example of a 2-dimensional LLE embedding on the face images is shown. The original image is plotted near each point, to show the relationship between face orientation and the new axis found.

In this section the Multidimensional Scaling (MDS) algorithm will be introduced, as a necessary step in the Isomap technique, that introduces the nonlinear behavior. Then the LLE algorithm will be shown, emphasizing the improvements made in the projections of unseen vectors and the supervised extensions of the algorithm.

Multidimensional Scaling

Multidimensional scaling is a non linear dimensionality reduction technique focused on displaying the structure of distances intra data in a low dimensional plot [113]. It has its origins in psychometrics where a similar idea was used to understand the way how similarities are perceived in subjects such as distances between colors

or olfaction senses. Also it has been used to obtain 2D plots of abstract psychologically related concepts [67], such as to see a spatially distribution of people working on different job. MDS has also been applied to sociology, econometrics and political sciences. The term MDS was first proposed by Torgerson [172], and later used in works from Kruskal [77], Young and Guttman [113].

The main goal of MDS is to represent a set of N D-dimensional input data points in a low dimensional space using some dissimilarity measure between each data point. The algorithm can be divided in two steps:

1. First the dissimilarity measure between each data point is computed. Usually, the most used measure is the Euclidean distance, but sometimes it does not capture the real dissimilarities between high dimensional points, for example in cases where there are different scales per each axis or when dealing with discrete or specific symbolic data. So other different distances can be used such as Mahalanobis distance, or more complex measures to model de intrinsic geometry of nonlinear manifolds.

2. Compute the MDS using the matrix NxN which contains the dissimilarity measure between each pair of points.

Depending on the nature of the dissimilarities the computation of the coordinates in the low dimensional space in the second step can be performed analytically or optimizing a loss function. When the dissimilarities between each pair of points are proportional to its distances, the MDS dimensionality reduction is called metric MDS, and has an analytic solution known as classical scaling ([171]), which is shown in table 1.2. Otherwise the embedding is called non-metric MDS, and it must be solved minimizing a loss function [113, 78].

In the Figure 1.18 we show an example of the projected points belonging to 2 classes (800 different numbers extracted from the MNIST database [84]) to a 2-dimensional space using the MDS projection. As it is shown, the resulting points are not completely separable in this low dimensional space.

Isomap

The isomap algorithm, proposed by Tenenbaum et al.([168]), tries to find the underlaying non linear manifold of the input data, using as a base method the classic MDS algorithm, but changing the dissimilarity measure to a very nonlinear one: geodesic distances. Once this geodesic distances are computed a MDS step is performed, and the resulting embedding is able to encode the real degrees of variance of the input data.

A geodesic distance between two data points is defined using the shortest path between both points. This path is constructed using a graph where each node is a data sample, and there is an edge with the Euclidean distance associated between each data point and its nearest neighbors. The geodesic distance is defined as the sum

Table 1.2: MDS classic algorithm

The MDS algorithm takes as inputs a matrix $\boldsymbol{X} = [X_1, \ldots, X_N]$ which are N D-dimensional points, and the matrix $\boldsymbol{D}$ with the dissimilarities between points.

1. Compute the matrix $\mathbf{A}$ as $A = -\frac{1}{2}d_{ij}^2$, where d_{ij} is the dissimilarity between the points i and j.
2. A new matrix $\mathbf{B}$ is computed as $\mathbf{B} = \mathbf{HAH}$, where the matrix $\mathbf{H}$ is defined as $\mathbf{H} = \boldsymbol{Id} - \frac{1}{N}\boldsymbol{O}$, $\mathbf{O}$ is a $N \times N$ ones matrix, and $\mathbf{Id}$ is the $N \times N$ identity matrix.
3. Compute the matrix $\mathbf{V}$ with the N eigenvectors of $\mathbf{B}$ with their corresponding eigenvalues $\mathbf{\Lambda} = \lambda_1, \ldots, \lambda_N$, so $\mathbf{B} = \mathbf{V\Lambda V}^T$.
4. It can be shown ([113]) that there are $N-D$ zero eigenvalues in Λ. The D remaining eigenvectors must be ordered according to their eigenvalue, obtaining the matrix $\mathbf{\Lambda_r} = \lambda_1, \ldots, \lambda_D$ and $\mathbf{V_r} = [v_1, \ldots, v_D]$.
5. The final coordinates can be computed as:

$$\mathbf{X} = \mathbf{V_r}\Lambda_r^{\frac{1}{2}} \tag{1.37}$$

of the distances associated to the edges that connect the shortest path between each point. Therefore, geodesic distances are better adapted to the underlaying manifold, in comparison with Euclidean distance, because it must border the real structure of the data instead of going straight across the manifold structure. In Figure 1.19 an example of geodesic compared to Euclidean distance between two points in a parabolic data set is shown. The geodesic distance models better the underlaying manifold of the data, therefore it seems more suitable to use it in the classic MDS algorithm. In the Figure 1.20 the same example using the MNIST handwritten digits used in Figure 1.18 is plotted. It can be seen how the final projections using this real data set are quite similar, although isomap improves separability between the two classes.

Locally Linear Embedding

In a similar way as isomap, the locally Linear Embedding algorithm (LLE) [166] finds a non linear mapping from a high dimensional space to a low dimensional one. To reach this objective, LLE takes into account the restriction that neighborhood points in the high dimensional space must remain in the same neighborhood in the low dimensional space, and placed in a similar relative spatial situation. The algorithm does not change the local structure of the nearest neighbors of each point.

LLE can be considered a locally linear method, because it takes into account the

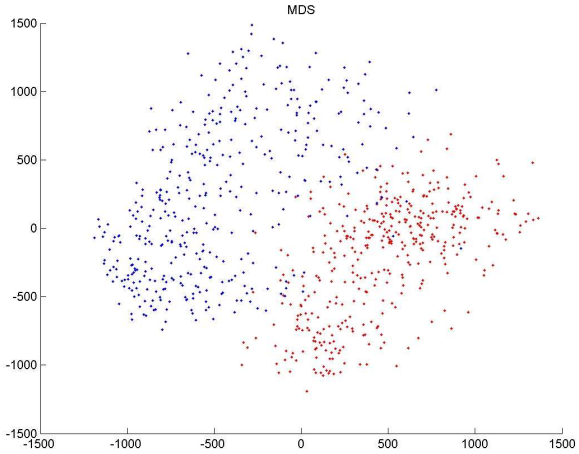


Figure 1.18: A 2-dimensional MDS projection example using Euclidean distance of a reduced data set comprising 800 characters of the MNIST database which correspond to 2 different data classes (numbers 4 and 7).

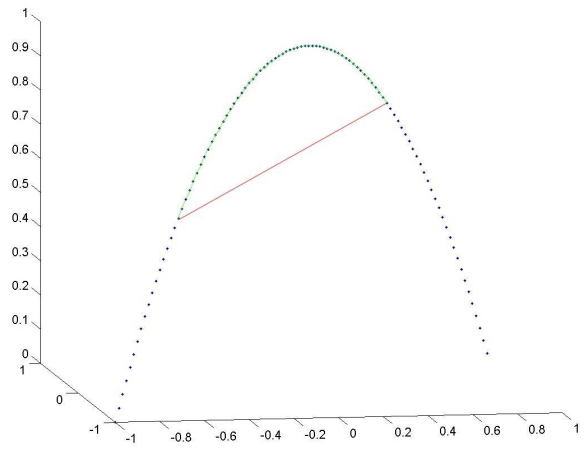


Figure 1.19: Example of data points generated in a 3-dimensional space according to the equation $(x, y, 1 - x^2)$. It can be seen the difference between considering the classic Euclidean distances (path in blue) and the geodesic distances (path in green), which can be used to capture the underlaying manifold of the data.

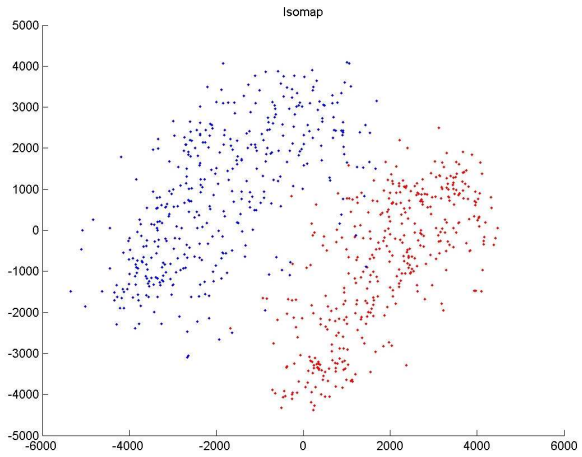


Figure 1.20: Isomap embedding of the same data set used in figure 1.18

local geometry of each point, but only the K nearest neighbors for each point are used, making the algorithm globally nonlinear, and enabling to capture the geometric properties of nonlinear manifolds. This characteristic also allows to work with very sparse matrices making the algorithm computationally efficient, and allowing to work with more points than other techniques (such as isomap).

The algorithm takes as inputs N D-dimensional training points $\mathbf{x}_{1,\dots,N}$ and can be divided in 3 steps:

1. In the first step, the K nearest neighbors of each point are found. The most common used distance is the Euclidean. Nevertheless, other distance metrics can be used, and the selection of the neighbors can be not homogeneous. For example the points within an envelope of certain radius can be selected as candidate neighbors.

2. In the second step the local geometry of the input data is captured, using a set of coefficients $\mathbf{W}$ per each point, corresponding to the weights that best reconstruct the vector from its K nearest neighbors (usually using the Euclidean distance). So the weights W_{nk} must minimize the error reconstruction equation:

$$\varepsilon(\mathbf{W}) = \sum_{n=1}^{N} |\mathbf{x}_n - \sum_{k=1}^{K} W_{nk}\mathbf{x}_{n_k}|^2 \tag{1.38}$$

   where $\mathbf{x}_{n_k}$ indicates the K nearest neighbors of the sample $\mathbf{x}_n$. To find the weights that minimize this equation, a least-squares problem must be solved. See [74] for more details.

3. In the last step the coordinates of each point in the low dimensional space

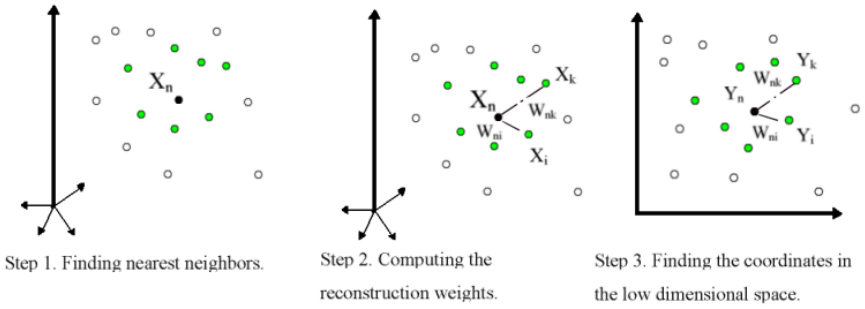


Figure 1.21: Schematic view of the locally linear embedding algorithm, where the three important steps are shown: the location of the nearest neighbors of each point, the computation of the weights that encode its local properties, and the low dimensional embedding.

$M \ll D$ are computed as the vectors $\mathbf{y}_n$ that best minimize the equation:

$$\theta(\mathbf{y}) = \sum |\mathbf{y}_i - \sum W_{nk}\mathbf{y}_{n_k}|^2 \tag{1.39}$$

The weights found during the previous stage are constant, and the goal is to find the low dimensional outputs $\mathbf{y}_n$ that best reconstruct each vector using its K nearest neighbors and the weights of the second step, which capture the local geometric properties of each point in the original space. So the equation to find the output vectors is independent of the input vectors $\mathbf{x}$ in the final step. To efficiently find the vectors $\boldsymbol{y}_n$ an eigenvector problem must be solved. A new sparse matrix $\boldsymbol{M}$ is defined as:

$$M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_{k=1}^{K} W_{ki}W_{kj} \tag{1.40}$$

It can be proved that the output vectors $\boldsymbol{y}_n$ are the $M+1$ eigenvectors of the matrix $\mathbf{M}$ associated to the lowest eigenvalues (see [74, 166] for more details).

An important drawback of the LLE technique is the difficulty to project new unseen vectors $\mathbf{u}$ to the low dimensional subspace. Parametric and non parametric models have been used to solve it (see [74, 166]). In [103], we proposed the use of a multi layer perceptron neural network approach to learn the nonlinear mapping and solve the regression problem. The idea is to run the LLE algorithm with a set $\mathbf{X}$ of N training vectors (so the network must have N inputs), in order to obtain their projection $\mathbf{Y}$ in the low dimensional space. Then we train the MLP using $\mathbf{X}$ as inputs and $\mathbf{Y}$

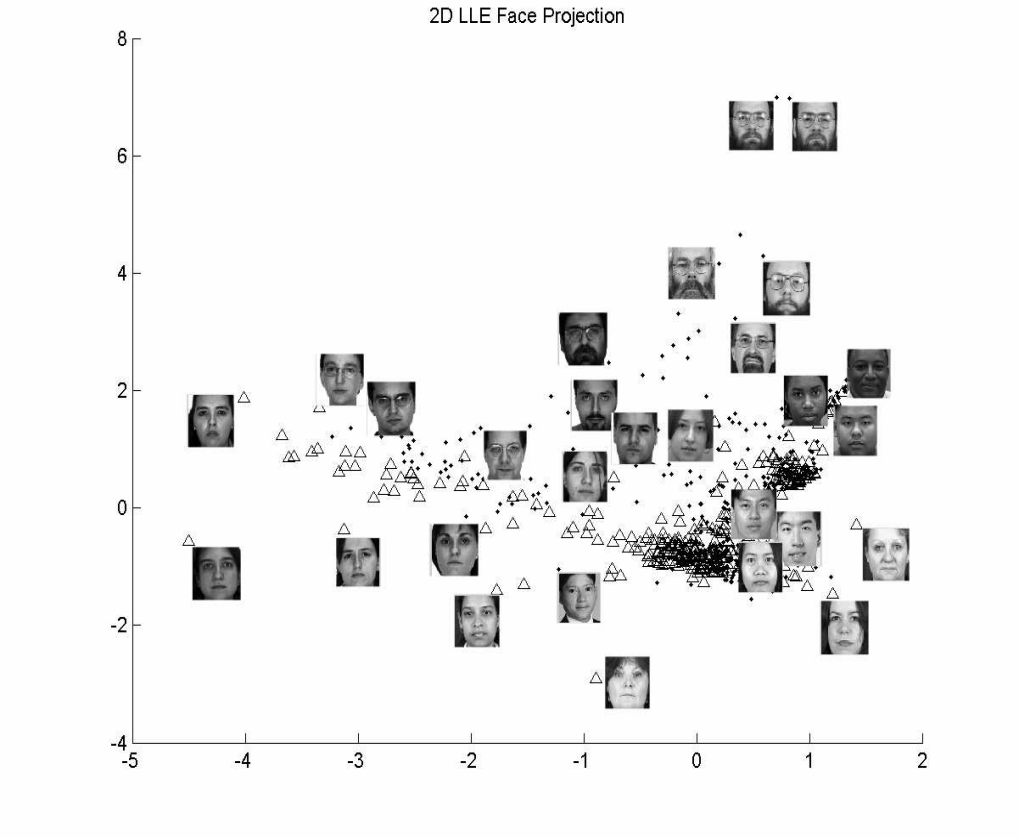


Figure 1.22: 2-dimensional reduction of faces using LLE. Original faces are plotted near each reduced point. Triangles stand for female subjects and dots for male subjects. As can be observed some characteristics as global illumination are captured by LLE embedding. On the other hand other features such as beard (on the up-right corner), or ethnicity are also captured in the spatial distribution of the faces.

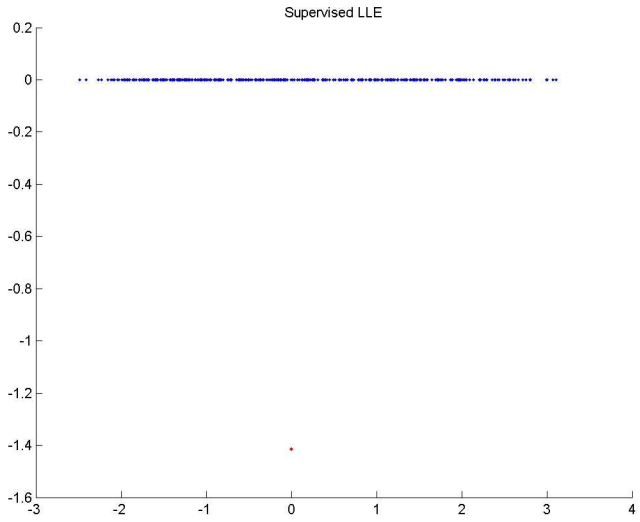
as desired outputs. As a previous step all the vectors must be normalized to the range [-1,1]. The intrinsic characteristics of the neural network can depend on the problem, but in our experiments made with face images we have realized that a MLP with 2 hidden layers (with 15 and 10 neurons each one) is enough to capture the nonlinear dimensionality reduction. The projection of new input vectors in the low dimensional space is then straightforward, running a forward step in the MLP to obtain the reduced vector. Another approach to project the new unseen vectors is to use the same principle used in the standard training algorithm, but for individual points ([38]). Given an unseen sample $\mathbf{u}$, its K nearest neighbors from the training set are found. Then the reconstruction weights for the point are computed (minimizing the equation 1.38). The final coordinates $\mathbf{y}_u$, are computed as:

$$\mathbf{y}_u = \sum_{i=1}^{K} \mathbf{W}_i Y_{N(i)}) \tag{1.41}$$

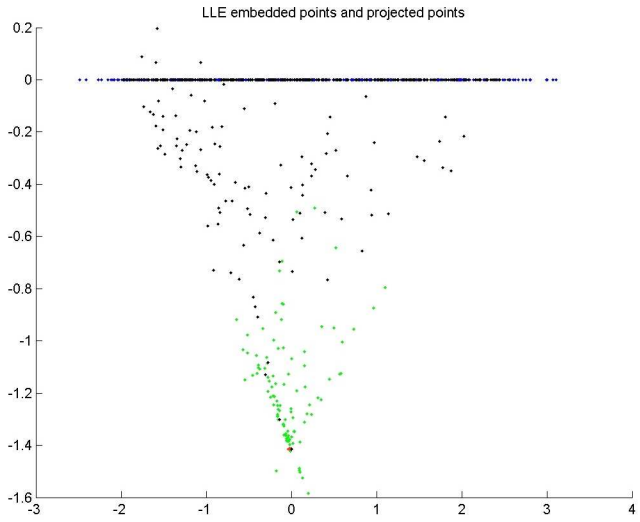
where $\mathbf{Y}_{N(i)}$ are the coordinates of the nearest neighbor i in the reduced space corresponding to the nearest neighbor i in the high dimensional space. In experiments performed with face images, we have seen that the clouds of points representing each class in a 2-dimensional subspace remain stable after projection new unseen vectors. Okun et al.[116] developed a modification of the LLE algorithm by adding to the algorithm information of class membership of the training points, called supervised locally linear embedding (SLLE). The difference between the supervised algorithm and the original one relies on the selection of the neighbors, in the first step. While LLE looks for the K nearest neighbors of each point, supervised LLE looks for the nearest neighbors that belong to the same class of the point. So the weights computed in the second step encode the best way to reconstruct each point from its nearest neighbors of the same class. The rest of the algorithm remains the same as the original LLE. In the Figure 1.23 it can be seen how the supervised locally linear embedding allows a separation of the 100% of the training data points, concentring them in to distant clouds. If new test points are projected, this perfect separation is not possible, although there is still a very good separation between classes.

1.3 Statistical Face Classification

Feature extraction techniques provide an appropriate starting point for the second step: classification of the feature vectors or assigning a label to each new unseen example. In face classification a large amount of classifiers have been used in the literature, with different degree of success depending on the application. In this section a brief overview of some classification schemes used in this thesis will be performed.



(a)



(b)

Figure 1.23: (a) 2-Dimensional embedding using supervised LLE of the same data set used in figure 1.18. (b) The projection of new unseen test vectors, black points are the projections of vectors of the same class as the blue ones, while green points belong to the same class as red points. The points remain separable, but the clouds are not as far as the clouds used in the training.

1.3.1 Bayes Decision Theory

Theoretically, the best classifier must be the one that minimizes the number of possible mislabelings, and this classifier is known as the optimal Bayes classifier. The classification is performed as follows: given a new feature vector $\mathbf{x}$, and the set of classes $\{c_1, \ldots, c_K\}$, the *posterior* or *a posteriori* probability $P(C|\mathbf{x})$ is computed for each class c_j. Then the label assigned to the vector is the one that has maximum probability [132]. This is known as the Maximum a Posteriori decision rule (MAP):

$$c_{MAP} = argmax_{i=1\ldots K} P(c_i|\mathbf{x}) \tag{1.42}$$

The $P(c_i|\mathbf{x})$ are unknown, but can be computed using the Bayes rule:

$$P(c|\mathbf{x}) = \frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})} \tag{1.43}$$

The denominator of 1.43 does not depend on the class label, so the class labels can be computed directly from:

$$c_{MAP} = argmax_{i=1\ldots K} P(\mathbf{x}|c_i)P(c_i) \tag{1.44}$$

where $P(c_i)$ is the *prior probability*, and $p(\mathbf{x}|c_i)$ is the *class conditional probability*. The prior probabilities are usually calculated using some a priori knowledge about the frequency of the classes, while class conditional probabilities are directly estimated from a training set. When there is no knowledge about the class priors and are all considered equally probable, then eq. 1.44 becomes:

$$c_{MAP} = argmax_{i=1\ldots L} P(\mathbf{x}|c_i) \tag{1.45}$$

and it is called Maximum Likelihood (ML) decision rule, and it is directly related to Mahalanobis distance metric to the class means when gaussian distributions are assumed.

Although the Bayesian classifier is considered theoretically the best possible classifier, it has an important drawback, it assumes that the class conditional probabilities are known or can be perfectly estimated from the training set, and usually this assumption does not hold in high dimensional data.

Probably the most spread density function to estimate the conditional probabilities is the normal distribution, denoted by $p(\mathbf{x}|c_i) = N(\mu_i, \Sigma_i)$:

$$p(\mathbf{x}|c_i) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\Sigma_i|}} exp[-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)] \tag{1.46}$$

where $\mu_i \in \Re^D$ is the mean of the vectors of the class i and Σ_i is the $D \times D$ covariance matrix. The choice of the normal distribution to model the class densities is made assuming that there is a ideal prototype object (the mean), and the rest are small "distortions" of it. The parameters of the distribution can be estimated using the

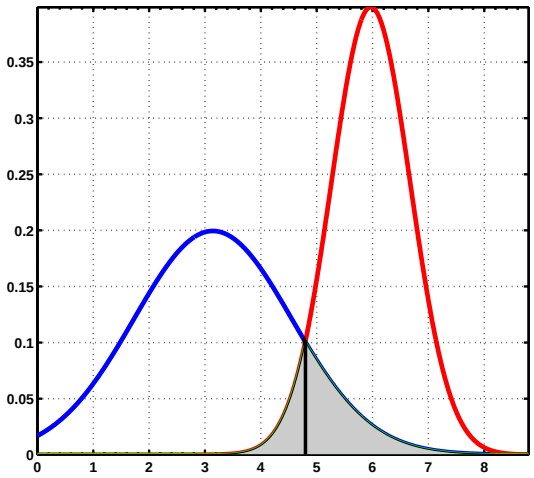


Figure 1.24: Example of Bayes error for two normal distributions. The optimal separability is marked with the black vertical line, and the error region is plotted in gray.

maximum likelihood technique [132], it can be shown that μ_i and Σ_i are represented by the population mean $\widehat{\mu}$ and sample covariance matrix $\widehat{\Sigma}$ respectively.

$$\widehat{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{1.47}$$

$$\widehat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}-\widehat{\mu})(\mathbf{x}-\widehat{\mu})^T \tag{1.48}$$

The error of a any classification rule is defined as the probability of assigning a certain sample to the wrong class. The error of the Bayes classifier C^* is defined as [96]:

$$P_e(C^*) = 1 - \sum_{c_i=1}^{c_K}\int_{\Re_i^*} p(c_i)p(\mathbf{x}|c_i)d\mathbf{x} \tag{1.49}$$

where $\Re_i^*$ is the classification region for the class i. In figure 1.24 an example with two normal densities is plotted. Given the optimal decision boundary (in black), the Bayes error is the area defined in gray. In practice, the Bayes probabilities are almost never known, so it is not possible to analytically calculate the error in real problems. Nevertheless, the Bayes error can be useful when designing new classifiers, to evaluate its expected performance. An interesting consideration to take into account is the behavior of the classification error when a feature extraction step is previously performed. It has been shown that for any transformation $\mathbf{s} = \mathbf{A}\mathbf{x}$, the Bayes error will

be greater or equal than with the original data [22]. This result can seem contradictory with the introduction of feature extraction in pattern recognition problems, given that even the best feature extraction algorithm will not improve the final error, and often will make it worse. But this affirmation is only true if we know the probability distributions on the original space, what never holds in practice, so the parameters of the distributions must be estimated from training samples. As face classification always implies to work with high dimensional samples, an accurate estimation of the distributions becomes a difficult problem (*curse of dimensionality*). Therefore, in this context, the feature extraction allows a better estimation of the distributions on the reduced space, and improves the global performance of the classifier. Also feature extraction can focus the attention on class discriminability, obtaining more accurate specific classifiers.

Parameter Estimation

The class-conditional distributions of eq.1.44 are in general unknown, so they must be estimated from the training data. Different likelihood estimation methods have been defined. In a first approximation the estimation methods can be divided in parametric and non parametric [12].

Parametric methods are based on assuming a certain distribution function defined by a set of parameters. A typical example of parametric techniques is the assumption of gaussian distribution. To estimate the parameters of the distribution given the training data $\mathbf{X}$ the maximum likelihood (ML) algorithm is used [132]. The ML technique defines the likelihood function depending on the set parameters $\theta = \theta_1, \theta_2, \ldots, \theta_n$. Therefore, the goal is to find the parameters θ_{ML} that maximize:

$$p(\mathbf{X}|\theta) = p(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}|\theta) \tag{1.50}$$

Usually independence assumption between the measurements $\mathbf{x_i}$ is assumed, and the log likelihood is used instead of 1.50, obtaining:

$$ln(p(\mathbf{X}|\theta)) = ln(\prod_{n=1}^{N} p(\mathbf{x_n}|\theta)) = \sum_{n=1}^{N} ln(p(\mathbf{x_n}|\theta)) \tag{1.51}$$

The parameters needed can be found by differentiating the log likelihood function and solving the equations $\frac{\partial}{\partial \theta_i} ln(p(\mathbf{X}|\theta)) = 0$. Frequently the solutions must be found using numeric procedures such as Newton's method or the Expectation Maximization algorithm [2]. The main drawback of parametric methods is that often data samples do not perfectly fit on predefined likelihood functions, even when a perfect estimation is performed.

In nonparametric methods, the density function is not specified in advance, and only the training data is used to estimate the class-distributions. One of the most simple methods is to use the frequential probability of each training sample. A quantization is performed on each feature, dividing the range in bins. Then the probability is calculated depending on the number of points that fall in each bin. Given the bin

B_i, with volume V_i, N the total number of samples, and N_i the number of samples falling on each bin, the probability of the value $\mathbf{x}$ to fall in B_i is:

$$p(\mathbf{x}) = \frac{N_i}{V_i N} \tag{1.52}$$

The main problem of this approach is that the total number of bins grows exponentially with the dimensionality of the data, so the number of training samples needed to model the probabilities grows also exponentially (*curse of dimensionality*).

Another non parametric approximation is the *Parzen window* approach [122]. This method sums the contribution of a predefined kernel function $H(\mathbf{x}, \mathbf{x_n}, h)$ on each sample $\mathbf{x_n}$, where h defines the width of the neighborhood, and V the volume.

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{V(h, D)} H(\mathbf{x}, \mathbf{x_n}, h) \tag{1.53}$$

Different kernel functions can be used, such as Gaussian or hypercube (Parzen) kernels. The advantage of kernel methods, is the smoothness with respect to the direct bin approach, where there appear discontinuities on the points close to the boundaries of the bins.

Another different approach is not to fix the volume of the bins, and fix the number of samples falling on this bin, it is known as the k-*nearest neighbor approach* (K-NN). This method can be considered a likelihood estimator and a classifier itself, and it will be explained in a further section.

1.3.2 Linear Discriminant Classifier

The Linear Discriminant Classifier (LDC) is a parametric technique derived from the Bayes classifier assuming normally distributed classes with equal covariance matrices. The function $f_i(\mathbf{x})$ is defined for each class:

$$f_i(\mathbf{x}) = log(P(c_i)p(\mathbf{x}|c_i)),\ i = 1, \ldots, K \tag{1.54}$$

where $p(c_i)$ and $p(\mathbf{x}|c_i)$ are prior and the class conditional probabilities. Assuming that the classes are normally distributed, with class mean μ_i and the same covariance matrix Σ, $p(\mathbf{x}|c_i) = N(\mu_i, \Sigma)$ we obtain [96]:

$$f_i(\mathbf{x}) = log(P(c_i)) - \frac{1}{2}\mu_i^T \Sigma^{-1} \mu_i + \mu_i^T \Sigma^{-1} \mathbf{x} = w_{i0} + \mathbf{W}_i^T \mathbf{x} \tag{1.55}$$

where f_i is the discriminant function. The parameters μ_i and Σ_i are estimated from the training data, and the data samples will not be normally distributed in practice, so the final error will differ from the optimal Bayesian error.

1.3.3 Quadratic Classifier

The Quadratic classifier (QDC) assumes classes normally distributed, but the covariance matrix is different on each class. Substituting in eq.1.54 using the class

covariance Σ_i, the final discriminative function obtained is [96]:

$$f_i(\mathbf{x}) = w_{i0} + \mathbf{w_i^T x} + \mathbf{x}^T W_i \mathbf{x} \tag{1.56}$$

where

$$w_{i0} = log(P(c_i)) - \frac{1}{2}\mu_i^T \Sigma_i^{-1} \mu_i - \frac{1}{2} log(|\Sigma_i|) \tag{1.57}$$

$$\mathbf{w_i} = \Sigma_i^{-1} \mu_i \tag{1.58}$$

$$\mathbf{W_i} = -\frac{1}{2}\Sigma_i^{-1} \tag{1.59}$$

And the parameters Σ_i and μ_i are directly estimated from the training samples. Sometimes the inversion of the covariance matrices Σ_i poses a problem when the matrices are singular. This situation happens when the number of training samples is smaller than the data dimensionality, and can be solved applying *regularization*. The method consists in averaging the covariance matrices for each class with the global covariance matrix.

$$\widetilde{\Sigma}_i(\alpha) = (1-\alpha)\Sigma_i + \alpha\Sigma \tag{1.60}$$

When $\alpha = 1$, the QDC classifier becomes the LDC, and $\lambda = 0$ means that no regularization has been used.

1.3.4 Naive Bayes Classifier

The naive Bayes classifier [132] is a modification of the optimal Bayes classifier where statistical independence is assumed on the features, so the conditional density can be marginalized as the product of the unidimensional densities of each feature:

$$P(\mathbf{x}|c_i) = \prod_{d=1}^{D} p(x_d|c_i) \forall i = 1 \ldots L. \tag{1.61}$$

Then the MAP rule is followed using the $P(\mathbf{x}|c_i)$ calculated using the naive assumption. Although the independence assumption is not realistic on the most cases, the naive approach has been used in many applications of text classification [114, 106] and information retrieval [89], and real world problems such as statistical diagnosis [61]. Domingos and Pazzani [44] performed a complete study using 28 databases from the UCI repository, where they showed that naive Bayes classifier outperform the Bayesian classifier with full density estimation, among other classifiers.

When the independence assumption holds, the naive Bayes approach is the best classifier we can get for linearly separable data. But even when no independence assumptions can be performed on the data, some preprocessing on the feature vectors can be performed to improve the results. Bressan et al. [41, 21] introduced an algorithm based on projecting the data using ICA before applying the naive Bayes decision rule. In the projected space, the assumptions made in the naive model are more realistic, and the performance of the classifier is improved.

1.3.5 *k*-Nearest Neighbor Classifier

The k-nearest neighbor classifier is based on selecting a set of labelled prototypes for each class, and the classification of new samples is performed using a measure of similarity between the new example and the prototypes. Given a set of $\mathbf{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_v}\}$ labelled prototypes and their labels $l(\mathbf{p_i})$, the k prototypes which are closer to a new input vector $\mathbf{x}$ according to the similarity measure are selected, an the most represented label among the k nearest neighbors is assigned to the sample $\mathbf{x}$. If the method is viewed under the likelihood estimation point of view, the pdf estimated is:

$$p(\mathbf{x}) = \frac{k}{NV} \tag{1.62}$$

where k is the number of prototypes used, N the number of samples and V is the volume that contains the k nearest neighbors. Considering N_{vi} the number of elements in V of the set P that belong to the class c_i, we obtain a class-conditional density:

$$p(\mathbf{x}|c_i) = \frac{N_{vi}}{N_i V} \tag{1.63}$$

where N_i is the total number of elements from class c_i. Applying the posterior probabilities, we obtain:

$$p(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i)p(c_i)}{p(\mathbf{x})} = \frac{\frac{N_{vi}}{N_i V}\frac{N_i}{N}}{\frac{k}{NV}} \approx \frac{N_{vi}}{k} \tag{1.64}$$

So the minimum Bayesian error is achieved by assigning to x the class with maximum 1.64. Therefore the class assigned is the one that has more representants in the volume defined by the k nearest neighbors. Two important results can be derived, first the k-nn error is optimal (equal to the Bayesian error) when $k \to \infty$, $V \to 0$ nd $\frac{k}{N} \to 0$. And second, when the 1-NN rule is used, the error is upper bounded by twice the Bayes error $P_{1-nn} \leq 2P_e$ [36].

The k-nearest neighbor classifier is one of the most simple classifiers, and it has been proven to be very robust in many pattern classification tasks. When the number of training samples is large, the computational needs of the algorithm can increase considerably. Prototype selection algorithms have been proven to be very efficient in these cases, drastically reducing the sample size, with small impact on the final accuracies.

1.3.6 Support Vector Machines

Support Vector Machines (SVM) have been used on many face classification problems achieving interesting results: Moghaddam and Yang applied SVM to gender recognition [5], they used their approach on low resolution images, where hair information has been eliminated. Their experimental results with SVM outperform the classic pattern recognition techniques.

In addition, Osuna et. al successfully used SVM on a face detection problem [117], in their work, they introduced a decomposition algorithm specially suitable for dealing with large sets of data, given that face detection usually involves a large set of face candidates. They iteratively decompose the problem in subproblems and combine them adjusting the margins. With this approach they could solve a face detection problem composed by more than 50000 data points.

The roots of SVM are located in the Statistical Learning Theory developed by Vladimir Vapnik at AT&T Bell Laboratories. SVMs are based on the principle of structural risk minimization, and can be applied both to classification and regression purposes. The main characteristic of SVM technique is the fact that it generalizes well when classifying new unseen data. In this section the basis of simple linear SVM will be shown, then a kernel extension of the method where data is previously projected to a high dimensional space will also be exposed. For more details on the algorithm, see [109, 31, 160, 28].

Given a set of training samples $\mathbf{x}_j$ with labels $c_j \in \{1, -1\}$ the SVM method tries to find the hyperplane that best separates the data points and maximizes the distance between points from both classes to the hyperplane (the margin). The discriminant hyperplane is defined by:

$$f(\mathbf{x}_i) = \mathbf{w}\mathbf{x}_i + w_0 \tag{1.65}$$

where $\mathbf{w}$ and w_0 are the parameters of the hyperplane. A new unseen sample will be classified depending on:

$$(\mathbf{w}\mathbf{x}_i) + w_0 \geq +1 \qquad when \qquad c_i = 1. \tag{1.66}$$

$$(\mathbf{w}\mathbf{x}_i) + w_0 \leq -1 \qquad when \qquad c_i = -1. \tag{1.67}$$

which can be merged on:

$$c_i[(\mathbf{w}\mathbf{x}_i) + w_0] \leq 1 \qquad i = 1, \ldots, N \tag{1.68}$$

The minimum distance between the hyperplane and the closest point is called margin, so the optimal hyperplane will be the one that maximizes the margin γ with the closest point. This property is strongly related to the generalization capabilities of the algorithm, the largest the margin, the most separable will be both classes.

The optimal hyperplane is defined by a set of vectors, called support vectors. The number of vectors necessary to find the hyperplane is an indicator of the class separability of the problem. Vapnik showed that the generalization error is related to the number of support vectors. Moreover, Vapnik [176, 177] introduced an upper bound on the generalization error, although this bound is not easily computable in practice, it offers interesting insights for new theoretical developments ([109]). Figure 1.25 shows a synthetic example of two separable classes, where the most important parameters are shown.

Analytically, to find the hyperplane that maximizes the margin, the following expression must be minimized [31]:

$$min \quad \frac{1}{2}||\mathbf{w}||^2 \tag{1.69}$$

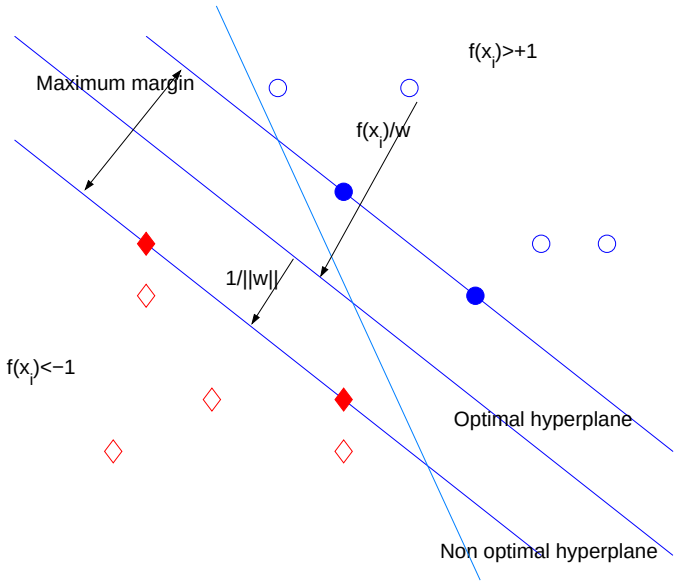


Figure 1.25: Synthetic example of two separable classes. The central line defines the optimal hyperplane defined by the support vectors sown in filled markers. Also the maximum margin is illustrated.

for both $\mathbf{w}$ and w_0, subject to

$$c_i[(\mathbf{w}\mathbf{x}_i) + w_0] \leq 1 \qquad i = 1, \ldots, N \tag{1.70}$$

To find the analytic solution to this problem quadratic programming techniques can be applied. Nevertheless, when the dimensionality of the data is moderately large, the problem must be formulated using the Lagrangian dual form [163].

$$max(min\ L(\mathbf{w}, w_0, \alpha)) \tag{1.71}$$

where

$$L(\mathbf{w}, w_0, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i (c_i[(\mathbf{w}\mathbf{x}_i) + w_0] - 1) \tag{1.72}$$

The dual formulation can be performed following the Kuhn-Tucker theorem, given that both the cost functions and constrictions are convex functions (see [31] for more details).

Sometimes it is not possible to find the optimal hyperplane, because the class distribution does not allow full separability (there are some points that do not fulfil eq.1.70). To solve this problem a set of positive margin variables ξ_i $(i = 1, \ldots, N)$ are defined to model the distance to the limit of the margin of each class for each non separable point:

$$c_i[(\mathbf{w}\mathbf{x}_i) + w_0] \leq 1 - \xi_i \tag{1.73}$$

Therefore, the minimization problem becomes:

$$min \quad \frac{C}{N}\sum_{i=1}^{N}\xi_i\frac{1}{2}||\mathbf{w}||^2 \tag{1.74}$$

for both $\mathbf{w}$ and w_0, subject to the restriction of eq.1.73, and C is a large enough constant defined by the user. The hyperplane solution is called soft margin hyperplane. As in the previous case, the solution for high dimensional subspaces can be found using the dual formulation [175, 31].

Kernel SVM

Using this formulation, the optimal hyperplane is defined on the original space, so only linearly separable problems can be solved. One solution to extend the SVM algorithm is to map the training data to a new feature space $\phi : \mathcal{X} \longrightarrow \mathcal{F}$, using a mapping function ("kernel trick"). The functions interesting are those which correspond to a dot product on the feature space $\mathcal{F}$ [160]:

$$k(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}) \cdot \phi(\mathbf{y})) \tag{1.75}$$

where k($\mathbf{x}$,$\mathbf{y}$) is the mapping that depends on the user choice. The kernels are constructed satisfying the Mercer theorem [131, 176, 34]. The kernel function can be linear and non linear. The most commonly used are:

- **Linear Kernel**: It is equivalent to the linear SVM as explained above.

$$k(x, y) = x \cdot y \tag{1.76}$$

- **Polynomial Kernel**: with parameter d, the degree of the polynomial, and r is a parameter defined by the user depending on the application.

$$k(x, y) = (x \cdot y + r)^d \tag{1.77}$$

- **Radial Basis Functions (RBF)**: with parameter σ defining the radius.

$$k(x, y) = exp(\frac{\|x - y\|^2}{\sigma^2}) \tag{1.78}$$

- **Sigmoid Kernels**: With parameter θ

$$k(x, y) = \tanh((x \cdot y) + \theta) \tag{1.79}$$

- **Other kernels**: Some other kernels also have been used in pattern recognition, such as neural networks based kernels, spline based kernels, and kernels based on the Fourier transform.

Figure 1.26 shows an example where after projecting the data in a higher dimensional space it becomes linearly separable. The main problem using nonlinear SVM is the proper selection of the parameters of the kernel. The accuracies obtained are very sensible to these parameters and usually a trial and error method is the best way of tuning the algorithm.

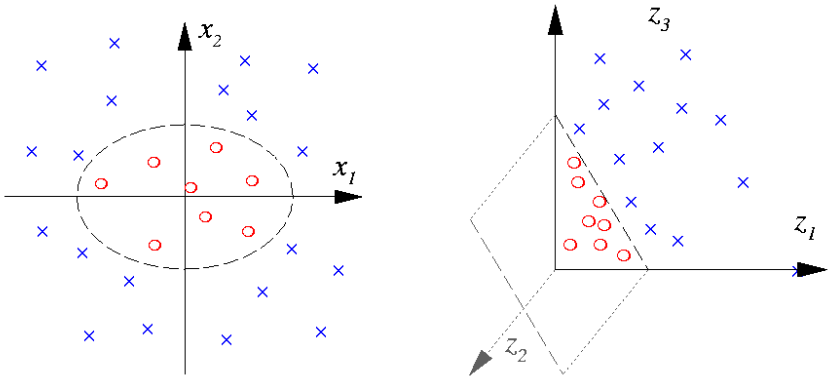


Figure 1.26: Example extracted from [109]. Data is projected to a 3D space using the mapping $(x_1, x_2) \rightarrow (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2))$, where it becomes linearly separable.

1.3.7 Classifier Combination

In this section a brief overview of the classifier combination techniques will be presented, although in the next chapter the different strategies will be explained in depth. By classifier combination we denote the fact of training multiple classifiers to obtain a more powerful decision rule. A simple taxonomy of the building ensembles of classifiers can be established depending on the approach followed on the individual classifiers [96]:

1. Use different combining rules.
2. Combine classifiers of different nature.
3. Use different feature sets on the classifiers.
4. Use different data subsets for each classifier.

In this work we will focus on the last proposal, where methods such as boosting and bagging have been successfully applied in many pattern recognition tasks. Another popular combining technique, belonging to the third taxonomy is the random space methods (RSM).

Bagging

Bagging was proposed by L.Breiman [19], and it is based on randomly selecting samples from the training set. A classifier is trained on each subset, and the classification results are combined using some rule such as majority voting, or averaging the results. Bagging has been proven to be efficient on small training sets and in

presence of outliers, that can be isolated in some subsets, having less influence in the final voting results.

Bagging was successfully used on a gender recognition problem [104] using frontal face images from the AR Face database [100] and XM2VTS [105]. Different feature extraction techniques were used and the NN rule was applied for classification. The bagged combination of classifiers using NDA feature extraction outperformed the other techniques considerably (see Table 1.3). Also the improvement of using the bagged NDA algorithm against a single NDA feature extraction should be noticed.

Table 1.3: Gender recognition accuracies on the AR Face and XM2VTS databases using different feature extraction techniques.

Algorithm	**Accuracy**
NN	86.28
PCA	86.57
SVM	90.95
FLD	81.30
LLE	76.27
SLLE	87.12
NDA	90.56
Bagged NDA	91.76

Boosting

Boosting methods are inspired on an algorithm called $Hedge(\beta)$ [50]. According to Freund and Schapire [49], the base algorithm consists in sequentially training a set of weak classifiers, and the final decision rule is constructed as a linear combination of the results of the partial classifiers. In the first step the weak classifier is learned using a training set. According to the classification performance, a set of weights for each learning sample are adjusted. In the next step the new classifier is trained taking into account the weights of the previous step, giving more importance to the misclassified samples. At each step the classifiers are more focused on the most difficult samples, and the final combination is hoped to achieve 0 training error quickly.

Two different versions of boosting have been proposed in the literature: with resampling and reweighting, depending on the how the information encoded on the weights is used on the training set. In the next chapter the different implementations of the boosting models will be explained in detail, as it is the base of the feature extraction algorithm proposed in this thesis.

In face classification problems, one of the most successful applications of boosting was performed by Viola and Jones [181, 180], were they used the Adaboost algorithm on a large subset of features to solve a face detection problem.

Random Subspace Methods

The RSM technique is a combining method [62] where the original feature space is sampled and only $M < D$ features are used for training the classifiers. An ensemble of classifiers is built using the M-dimensional data set, and the final decision rule is a simple majority voting. RSM is specially suitable for problems where the number of samples available is relatively smaller than the dimensionality of the data, or in problems where there is a strong redundancy on the feature set.

1.4 Example: Online Face Detection and Classification Application

In this section a brief description of a complete face classification system that illustrates some of the introductory methods exposed on this chapter will be explained. The final face recognition application was running on the principal stairs of the Computer Vision Center for 6 weeks [102], and consist of:

- A face detector based on boosting of naive Bayesian classifiers.
- A feature extraction step performed using the NDA algorithm
- A face recognition phase using the nearest neighbor classifier with the extracted features.

Figure 1.27 illustrates the main steps of the global application.

1.4.1 Face detector

The face detector uses a boosting cascade of naive Bayes (NB) classifiers. In a first step a feature extraction based on ridges and valleys (see [1] for more details) is performed on the image. This representation has been shown to be more robust against changes in illumination [129]. A threshold on the resulting filtered images is applied, assigning the value 1 when a pixel is on a ridge, -1 when a pixel is situated on a valley, and 0 otherwise, Figure 1.4.1 shows some filtered images from faces and non faces. To convert this ternary representation into a binary one, the filtered image is separated into two representations where we put to 1 the pixels where there is a ridge/valley and 0 otherwise. Both representations have been vectorized and concatenated, so at the end the filtered image becomes a binary vector with double dimensionality.

From each original video frame image of 576×768 pixels, the set of candidate sub images to contain a face is generated as follows: the image is first resized to 288×384 to avoid the effect of interlacing, then a set of sliding windows of 32×24 pixels is generated for each frame, in such a way that all the possible sub-windows from the

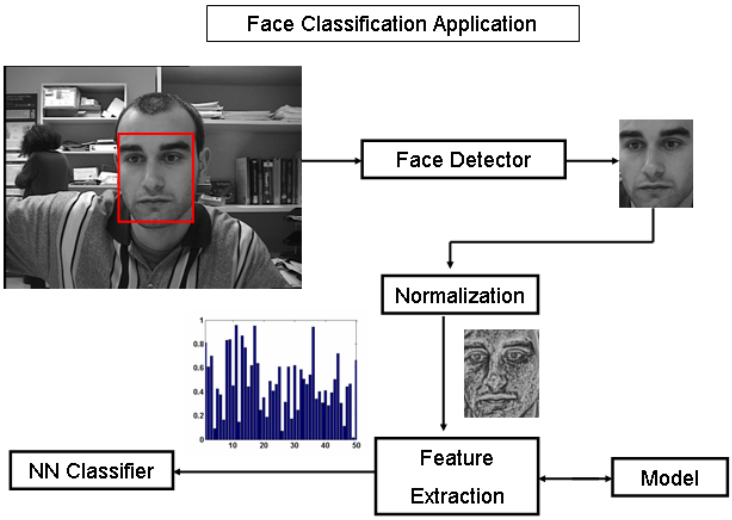


Figure 1.27: Scheme of the global application, where first a face detection is applied, then a feature extraction process over a normalized face image is performed, and finally the classification is done using the Nearest Neighbor classifier.

image are obtained. The center of the sliding window is considered every two pixels. The process is repeated at 4 different scales, each time the image is re-scaled by a factor of 1.25. Four scales have been shown to be enough for our application, detecting faces up to 64×64 pixels. Each sub image is classified as face or non face according the cascade described below.

As a face image can appear in more than one sliding window or in multiple scales, a merging step has been added to avoid multiple detections of the same face. We compute the overlap among each sliding window with a detected face in close positions (overlapping more than 80% of their surface). We keep the one that is closer to the mean of the overlapping windows (taking into account the center).

The proposed classifier for the sub images extracted makes use of the Adaboost algorithm, where the chosen weak classifier is the naive Bayes. In particular we have assumed a *Bernoulli* distribution on the data [20], given that only the binary filtered samples are used. For each target image $\mathbf{x}_i$ we decide that it is a face following the MAP rule $p(\mathbf{x}_i|C_{Faces}) > p(\mathbf{x}_i|C_{NonFaces})$. To estimate the conditional probabilities for the faces we use:

$$p(\mathbf{x}|C_{Faces}) = \prod_{d=1}^{D} p_d^{x_d} q_d^{1-x_d} \tag{1.80}$$

where x_d is d-th pixel of the image and p_d is the probability of finding a 1 in the pixel

Figure 1.28: Example of ridges and valleys detection for a subset of face and non face images.

d, and q_d the probability of finding a 0 in the pixel d ($p_d = 1 - q_d$). This probabilities p and q can be estimated directly from the training samples by finding the frequencies of the ones and zeros of the face vectors. In a similar way the conditional probabilities for the non faces are:

$$p(\mathbf{x}|C_{NonFaces}) = \prod_{d=1}^{D} \widetilde{p}_d^{\mathbf{x}_d} \widetilde{q}_d^{1-\mathbf{x}_d} \tag{1.81}$$

where $\widetilde{p}$ and $\widetilde{q}$ are obtained as p and q but using the non face instead of the face samples.

Face detection requires high detection rates (higher than 90%) and very low false positive rates (lower than 0.00001%) in order to be useful, given that there is a large amount of sub images in any video frame to be analyzed. In order to get this kind of false positive rates Viola and Jones [181] proposed the use of boosted classifiers in a cascade architecture where each classifier was specialized in classifying a specific subset of non faces while they keep high detection rates. We have also followed a similar approach using the NB as a base classifier.

A single NB classifier was trained using 6500 images, with 1500 faces extracted from different public face databases (we have used the XM2VTS [105] and the first image of each subject from the AR Face database [100]) and 5000 non face images. The classification results using a large set of 28000 images (26000 non faces and 2000 faces taken from the same databases) show a global error performance of 2.45% and only a 0.83% false positive rate in the first cascade level. This classification results were still poor for a face detection task, so a full cascade consisting of 32 additional levels (as described in [180]) were added. In the first level less than 1% of the images are considered faces, discarding the rest. Each new level is a more specialized classifier. The final detection rate is close to the 94%, while we obtain one false positive every 100000 samples.

In figure 1.29, we show some examples of images wrongly classified by the detector. Some twisted and blurry faces are confused, and also some natural images which present a structure very similar to a face are also misclassified. Also we have tested the face detector using the CMU test database [139] (with 483 labelled faces), obtaining 94.2% of detection rate and just 82 false positives. Notice that these results are similar to the most common techniques used in the state of the art, although our detector is

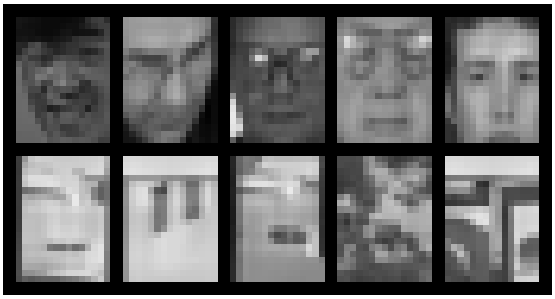


Figure 1.29: Examples of false negative and false positive images obtained in the boosting scheme. The structure of the false positives is similar to face images (with valleys in the eyes and nose zones).

more simple, which allows us to use it in a real environment. A complete comparative can be found in [97], where it can be seen that the best technique achieves 98% of detection rate, but also obtains a large amount of false detections (12758). Other techniques such Fisher Linear Discriminant achieve just 74 false detections, although the detection rate decreases (93.6%). Our purpose achieves detection rates close to the other techniques, keeping a reasonable false detection rate, and allowing a fast implementation.

1.4.2 Face Recognition in an Uncontrolled Environment

The final application consist in a face recognition scheme in a non controlled environment. A Sony EVI D-31 camera was installed looking at a staircase in the Computer Vision Center. This camera was connected to a VCR and four hours of recordings at peek hours were gathered each day, for a total lapse of 6 weeks. The face detector was applied to these videos and the detected images saved and manually labelled. From the approximately 80 different people detected in all tapes, only those 47 with more than 30 detected faces were included in the gallery. The total number of faces for these 47 subjects was 4176, approximately 88 faces per subject.

The modified NDA representation was used for real-time recognition in an uncontrolled environment. The real time requisite has justified the election of a feature extraction algorithm (to reduce the amount of data storage) in the verification step. Also the face detector is restricted to faces with small rotations (less than 10 degrees), although the set up of the real application allows the capture of enough frontal views of each person to train the NDA.

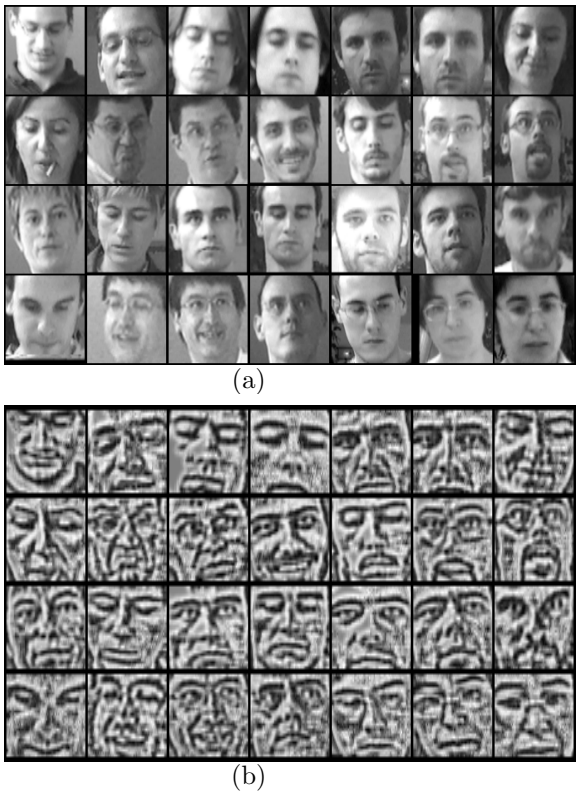
Feature extraction and face recognition

For the recognition engine we considered a scheme based on a linear projection using the NDA algorithm followed by a nearest neighbor classification. Recognition was performed on the 32×32 faces in the original frame. This is illustrated in Figure

1.30. In Figure 1.30 (a) some detected faces from a video sequence are shown. Figure 1.30 (b) shows these same faces after the normalization.

As a previous step, we have computed a Principal Component Analysis projection matrix using a huge face data set, and the data vectors have been projected to a 224-dimensional subspace that preserves approximately the 97% of the variance, prior to learning the NDA representation. Then 128 NDA components were preserved. Classification was performed using the 5 nearest neighbors average voting.

All the parameters from this scheme (PCA and NDA dimensionality, number of nearest neighbors and classifier combination policy) were set by cross-validating the training set.



(a) (b)

Figure 1.30: Example of some faces used in the face recognition experiment, before and after normalization.

Experimental results

To evaluate the performance of our classifier, a 10-fold cross validation on the faces was used. For this experiment, classification accuracy was **96.83**%. We also did

a supervised cross-validation so no samples from the same day were at the same time in training and test sets. Results were very similar, yielding a **95.9**% accuracy.

Recognition was also evaluated online, recording the recognition results and video to manually evaluate the online classification accuracy. Recognition rate for approximately 2000 test images belonging to the 47 subjects in the gallery was **92.2**%. In this experiment, we also observed that the classifier would greatly benefit from temporal integration which, at the moment, was not implemented. The frame rate of the application with all three engines working and this gallery of 47 subjects was approximately 15 fps. Also prototype selection techniques applied to the NN classifier could be applied to speed up the system. Figure 1.31 shows two frames taken directly from the working application. A first frame illustrates the environment in which the experiment took place, and the second frame illustrates the recognizer at work.

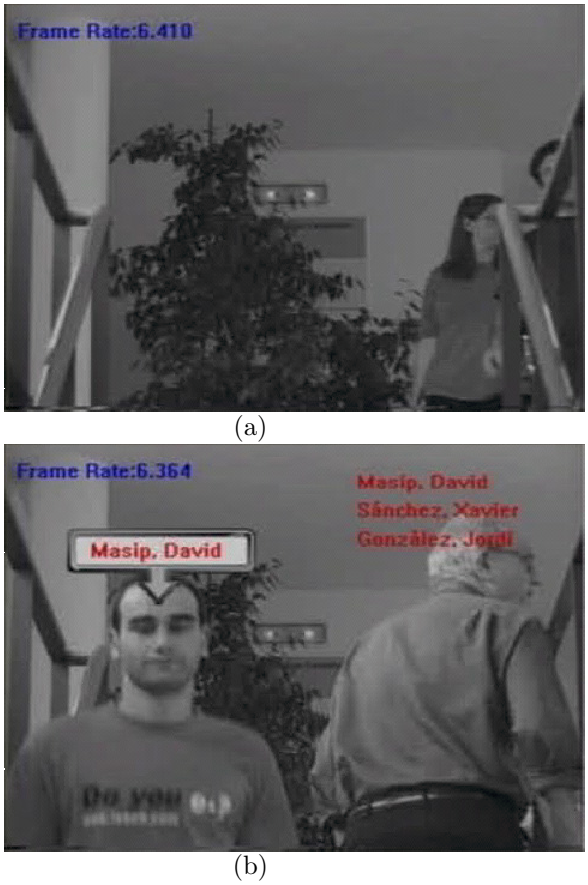


Figure 1.31: Frames extracted from the human ID at distance application in the Computer Vision Center.

1.5 Conclusions

In this introductory chapter an overview of the face classification problematic has been presented. The global scheme for face classification is divided in two parts: feature extraction and classification.

The feature extraction can be performed taking into account two independent face parts: the external and the internal information. A brief overview of the classic internal feature extraction techniques has been performed, leaving the external feature extraction for further chapters. Feature extraction yields important advantages on face classification tasks:

- Feature extraction allows to learn invariant characteristics to separate data into classes, highlighting discriminative information, and obtaining representations more suitable for classification, reducing the noise present in any natural image.

- As it has been shown, the problem of *the curse of dimensionality* complicates the estimation of the class densities for training the classifier. Feature extraction eliminates redundancy on the data, reducing the dimensionality and improving the parameter estimation.

- A most compact representation of the data is obtained, reducing the dimensionality of the original images we also reduce the storage and computational needs.

In the second part of this introduction an overview of different classifiers has been performed. We use as a base classifier the optimal Bayes classifier. It has been shown that the Bayes classifier minimizes the probability of missclassification, in this terms it is considerated theoretically the best classification method. The results of any classifier are upper bounded by the bayesian, nevertheless, the estimation of the class conditional distributions is complicated in high dimensional subspaces (as it is the case of face images) unless the proper feature extraction is performed. Depending on the assumptions performed on the data, different classifiers can be derived from the original Bayes decision theory, such as the linear, quadratic, or the nearest neighbor classifier.

Moreover, a brief overview of more recent classification methods based on combining weak classifiers has been performed. This methods are the base of the feature extraction technique proposed on this thesis. Also an example of a real face classification application is shown, where an efficient real time face detector based on the Adaboost algorithm is used, combined with a NDA-based feature extractor and the nearest neighbor classification rule. Despite of its simplicity, the performance of the whole system is close to the state of the art methods.

1.6 Contributions and Outline of the Thesis

The main contributions of this thesis can be divided in two parts: (i) feature extraction using the internal features of face images, (ii) feature extraction of external features and a framework for integrating internal and external information in face classification problems.

- In the first part we propose different feature extraction techniques based on applying dimensionality reduction on the vectors that contain the features representing each face (typically the grey scale values of the image pixels). Usually, prior to the feature extraction, face images are put into correspondence taking into account the center pixel of each eye, and normalized to mitigate the effects of changes in illumination. Some of the most used statistical pattern recognition techniques for feature extraction applied to face classification are introduced in this chapter. Also a brief review of some of the existing classification techniques in the literature is shown.

  In chapter 2 we summarize different classifier combination schemes that underlie the main purpose of this work for feature extraction: Bagging algorithm is introduced, and the Adaboost algorithm is discussed to develop the internal feature extraction exposed in chapter 3. We propose a novel linear feature extraction algorithm based on combining different projection vectors which are selected using Adaboost. Although the technique was initially designed to solve a gender recognition problem, we show in chapter 3 an evaluation using general databases from the UCI ([13]), concluding that the algorithm can outperform other linear discriminant techniques provided that the dimensionality of the original data set is large enough.

- In chapter 4 we show a framework for extracting external features from face images. We make use of an existing patch-based segmentation algorithm [15] to build a model of the external features of face images. The main difficulty of extracting external features is to put in correspondence the same parts between different subjects. Sizes, location and scale can hugely vary depending on the subject, making the normalization used in the internal features completely insufficient for the external case. In order to build a feature vector to apply standard classification techniques we use a model based on small fragments extracted from a predefined set of training people. Given a general enough fragment model, we can approximate each person external features by a linear combination of the fragments from the model. Two different algorithms have been used in our approach to obtain the final coefficients that are used as external features: the normalized correlation and the Non-negative Matrix Factorization algorithm.

  In the same chapter we propose methods for combining the external and internal information for face classification, showing that there can be an improvement in the classification accuracies with respect to the ones obtained using the classic pattern recognition techniques on the internal part of faces.

Finally, chapter 5 summarizes the main contributions of this thesis, and shows the future lines of this work.

Chapter 2

Classifier Combination Algorithms

2.1 Introduction

Usually, in problems where data lays on high dimensional subspaces or when the amount of samples available is reduced, it is difficult to built a good single classifier, because the estimation of the parameters is poorly performed. Different approaches can be followed to improve the performance of a weak classifier. In this chapter, combination strategies will be studied for this purpose. In combination strategies many weak classifiers are constructed instead of a single one, and a more powerful decision rule is constructed by combining them. The term "weak classifier" has been used to refer low complexity classifiers, unstable classifiers, or just badly performing classifiers [159]. Along this thesis, we will consider that a classifier is weak when it has poor performance, but with accuracy strictly higher than 0.5 for the binary case (in general $1/K$ for K-class problems).

Multiple methods have been proposed for combining weak classifiers, here we will describe some of the most popular: bagging and boosting. In this chapter the bagging technique will be introduced, and the boosting methods will be explained in depth. The Adaboost algorithm will be presented, given that it is the base algorithm of the feature extraction technique proposed in the next chapter. Moreover, a modification of the original Adaboost algorithm will be shown, where the classifier generation is performed jointly with the feature extraction. We will show 2 experiments dealing with visual data where the proposed technique yields lower classification errors, using less boosting steps.

2.2 Bagging

The term bagging was firstly introduced by L.Breiman [19] as an acronym of Boostrap AGGregatING. The idea is to generate random boostrap replicates from

the training set, construct a classifier on each subset, and then combine them using the simple majority vote rule. The complete algorithm is shown in table 2.1. The weak classifier used in bagging should be sensible to the training samples, in such a way that small variations on the training set should produce large changes on the classifier obtained. Otherwise the diversity of the classifiers will be low, and the combined results will not outperform the individual weak classifiers. Bagging is a

Table 2.1: Bagging algorithm.

1. In the training process.
   For t = 1,2, ... T do:
   - Build a random subset $\mathbf{X}_t$ taking randomly selected samples from the original training set.
   - Train a classifier C_t using the subset $\mathbf{X}_t$.
   - Add the classifier C_t to the ensemble.
2. In classification process.
   Build the final decision rule by combining the results of the classifiers. The class with maximum number of votes is chosen for each sample.

parallel algorithm in the training and exploitation phases, given that the results from the previous steps do not influence the next ones. Moreover, the algorithm is specially useful in problems with misleading examples, that appear only on a few subsets and reduce its influence on the final classification rule.

Bagging has often been used with decision trees, so majority voting is usually the most used combination rule. Nevertheless, when applying statistical classifiers other combining rules can be used [158]:

- Classify according to the posteriori probabilities using the output of the discriminant function.

$$P(c_j|\mathbf{x}) = sigmoid(C_t(\mathbf{x})) = \frac{1}{1 + exp(-C_t(\mathbf{x}))} \tag{2.1}$$

  Two approaches can be followed using the posteriori probabilities: take the class from the classifier with maximum probability, or take the mean of the probabilities of the T classifiers.

- Combine using the average of the outputs of the base classifiers:

$$R(\mathbf{x}) = \frac{1}{T} \sum_{t}^{T} C_t(\mathbf{x}) \tag{2.2}$$

  where $C_t(x)$ is the output of the discriminant function of the classifier at the step t.

Skurichina [158] proposed a modified bagging algorithm called "nice" bagging, where only the nice base classifiers where averaged. The nice classifiers are those which have training error lower than performing the classification on the whole set (not bootstrapped). The nice bagging classifier achieves, in general, just slightly better results than ordinary bagging, although it is a more stable bagging version.

2.3 Boosting

Boosting has been traditionally used as a way to produce an accurate classifier by combining weak and inaccurate classifiers [96] trained in a serialized way. Taking as input a set of samples represented by a finite set of features, the algorithm incrementally builds the final classifier by adding at each step a new weak classifier. This process is leaded increasing the importance of the misclassified samples at previous steps. A set of weights are adjusted for this purpose according to the classification results. The algorithm is repeated a fixed amount of times, and the final decision rule is constructed weighting the weak classifiers at each step.

2.3.1 Adaboost

Adaboost (ADAptive BOOSTing) algorithm has been considered the best method [147, 49] for boosting. Boosting algorithms are inspired by a learning algorithm called Hedge(β) [50]. This algorithm uses an ensemble of classifiers $\mathcal{C} = \{C_1, ..., C_T\}$ to decide the label of each sample $\mathbf{z}_j$. The importance or classification ability of each classifier on the ensemble is not known a priori, therefore the algorithm finds a set of weights $\mathbf{w} = [w_1, \ldots, w_T]$ that encode the importance of a classifier given their classification performance. The algorithm runs N steps, using one training sample at each step. The weights are usually uniformly initialized, and are adjusted according to the classification results of the previous step. Weights from classifiers that made a correct prediction on the previous step are increased, as the algorithm tries to improve the global prediction. The algorithm returns a weight distribution for the classifiers that minimizes the cumulative loss of the predictions. In table 2.2 the complete algorithm is shown. Further details on the algorithm can be found at [50, 96].

Adaboost algorithm also runs a predefined number of steps to sequentially train a weak classifier C_i at each step, yielding an ensemble $\mathcal{C} = \{C_1, ..., C_T\}$ as output. In Adaboost, the weights encode the importance of each training sample (instead of the classifiers), and at each round a classifier is trained focusing on the most "difficult" training samples (those misclassified in previous steps). Contrary to the Hedge(β) algorithm the parameter β is not fixed, but rather changes at each iteration according to the current error ε_t. Given the more refined analysis and choice of β in the Adaboost algorithm, the error bound on ε (error of the combined classifiers) obtained is significantly superior [50].

In the literature, two different approaches can be found depending on how the weights are used in the generation of the next step classifier:

Table 2.2: Hedge(β) algorithm.

Given:

- The matrix $\{\mathbf{X}\} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ of data samples
- $\mathcal{C} = \{C_1, \ldots, C_T\}$ the T classifiers for the ensemble.

1. Initialize the parameters
   - Pick $\beta \in [0, 1]$
   - Set the weights $w^1 = [w_1, \ldots, w_T], w_i^1 \in [0, 1], \sum_{i=1}^{T} w_i^1 = 1$.
   - Set $\Lambda = 0$ (cumulative loss)
   - Set $\lambda_i = 0, i = 1, \ldots, T$ (individual loss)
2. For every $\mathbf{x}_j, j = 1, \ldots, N$
   - Calculate the distribution by

$$p_i^j = \frac{w_i^j}{\sum_{k=1}^{T} w_k^j}, i = 1, \ldots, T. \tag{2.3}$$

   - Find the T individual losses ($l_i^j = 1$ if C_i misclassifies $\mathbf{x}_j$ and l_i^j=0 if C_i classifies $\mathbf{x}_j$ correctly, $i = 1, \ldots, T$).
   - Update the cumulative loss

$$\Lambda \leftarrow \Lambda + \sum_{i=1}^{T} p_i^j l_i^j \tag{2.4}$$

   - Update the individual losses

$$\lambda_i \leftarrow \lambda_i + l_i^j \tag{2.5}$$

   - Update the weights

$$w_i^{j+1} = w_i^j \beta_{l_i^j} \tag{2.6}$$

3. Calculate the return Λ, λ_i, and p_i^{N+1},$i = 1, \ldots, T$.

- Reweighting using weak classifiers that directly accommodate weights.
- Resampling the training set according to the distribution defined by the weights [18]. Therefore the most difficult samples should appear with more likelihood in the next classifier training. This variant is known as arcing, acronym of "adaptive resampling and combining" or arc-fs (in honor to Freund and Schapire). The combination of the classifiers in arcing is usually performed assigning to the object the class with maximum number of votes, instead of a weighted majority voting.

In table 2.3 the general Adaboost algorithm with resampling for the multiclass case is shown. In addition, some strategies have been developed to avoid solving directly the multiclass problem using Adaboost. Perhaps one of the most important

approaches is the use of error output correcting codes (ECOC). In this strategy, the multiclass problem is converted to multiple two class problems by grouping the classes. A binary codeword is assigned to each class, where 1 indicates that the class belongs to the subgroup and 0 otherwise (the number of bits of the codeword is equal to the number of subgroups made). The classification is performed by finding the codeword with lower Hamming distance between the classifier outputs (one for each subgroup) and the possible codewords. Detailed implementations of multiclass extensions of Adaboost can be found in [146, 43]

Table 2.3: Adaboost.M1 algorithm.

Given the training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$

1. Initialize the weight vector $\mathbf{w^1} = [w_1, \ldots, w_N]$ to $w_i^1 \in [0,1]$ and $\sum_{i=1}^N w_i^1 = 1$(usually $w_j^1 = \frac{1}{N}$)
2. For $t = 1, 2, \ldots T$ do:
   (a) Resample the training data $\mathbf{X}$ obtaining a subset $\mathbf{Z}$.
   (b) Build a classifier C_t using the training set $\mathbf{Z}$.
   (c) Calculate the weight ensemble error:
   $$\varepsilon_t = \sum_{i=1}^{N} w_i^t l_t^i \tag{2.7}$$
   where $l_t^i = 1$ if C_t misclassifies $\mathbf{z}_i$, and $l_t^i = 0$ otherwise.
   (d) If $\varepsilon_t = 0$ or $\varepsilon_t \geq 0.5$ reinitialize the weights.
   (e) Else, calculate β_t as:
   $$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}, \quad where \;\; \varepsilon_t \in [0, 0.5] \tag{2.8}$$
   (f) Update the individual weights
   $$w_i^{t+1} = \frac{w_i^t \beta_t^{(1-l_t^i)}}{\sum_{k=1}^N w_k^t \beta_t^{(1-l_t^k)}} \tag{2.9}$$
3. Return as an output the classifiers $C_1, \ldots, C_T$ and the $\beta_1, \ldots, \beta_T$ The classification of new samples is performed by computing the support for each class c_i, and taking the class with maximum support:
   $$\phi_c(\mathbf{x}_i) = \sum_{C_t(\mathbf{x})=c_i} ln(\frac{1}{\beta_t}) \tag{2.10}$$

Adaboost algorithm can reach fastly training error close to 0. In this context, Freund and Schapire [50] introduced a theorem that sets an upper bound on the

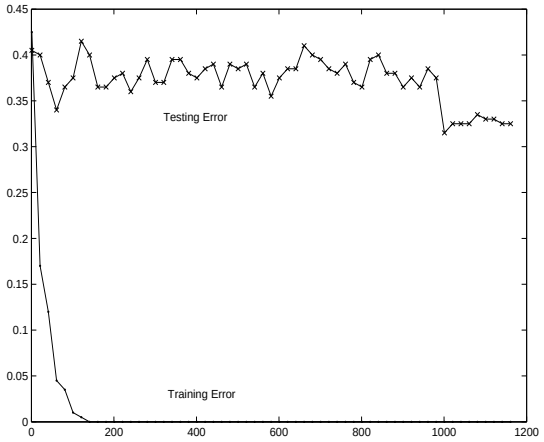


Figure 2.1: Training and testing errors using Adaboost on a two Gaussian classes problem.

training error using Adaboost:

$$\varepsilon < 2^T \prod_{i=1}^{T} \sqrt{\varepsilon_i (1 - \varepsilon_i)} \tag{2.11}$$

Where T is the number of weak classifiers on the ensemble, ε its training error, and $\varepsilon_i, i = 1, \ldots, T$ the weighted training errors of the weak classifiers at each step. According to this theorem, as Adaboost weak classifiers have error strictly $\varepsilon_i < 0.5$, the error ε decreases as the number of weak classifiers T is increased. The proof of this theorem for the two-class case is shown in the appendix B, in addition in [96, 50] the proof for the multi-class case can be found.

Furthermore, some experiments performed with Adaboost showed a surprising and unexpected phenomena: the testing error does not increase when the number of classifiers used on the combination becomes larger [145]. Moreover, the testing error decreases even after the training error becomes 0. This fact, contradicts the Occam's razor principle, one of the most important in machine learning theory, which states that to reach low testing errors, the classifier must be as "simple" as possible. In theory, the difference between training and testing errors should increase with the complexity of the classifier. In Figure 2.1 the training and testing errors using Adaboost as a function of the number of weak classifiers used on a two class Gaussian problem in a 30-dimensional space are shown. The covariance matrices for the classes are the identity matrices. The mean for the first class is $[0, 0, \ldots, 0]^T$, and the mean for the second class is $[0.2, 0, 2, \ldots, 0.2]^T$. Augmenting the complexity of the classifier (adding more boosting steps) does not produce increasing the testing error. Contrary, after reaching 0 training error the testing error continues decreasing.

The explanations for this phenomenon use the margin theory. The margin term comes from statistical learning methods [176], where algorithms such as support vector

machines using kernel functions try to maximize the margin to obtain classifiers with low generalization error [34]. Both boosting and SVM find a linear classifier on a high dimensional space, although they are computationally different. While SVM apply kernels to compute the classifiers on the whole high dimensional spaces, boosting explores one coordinate at each time.

Using a confidence measure of the classification $\phi(\mathbf{x})$, the margin can be defined as the difference between the weight confidence assigned to a correct label and the maximal weight assigned to any single incorrect label:

$$ma(\mathbf{x}) = \phi_c(\mathbf{x}) - max_{i \neq c}\{\phi(\mathbf{x})\} \tag{2.12}$$

where c is the known class of $\mathbf{x}$ and $\sum_{i=1}^{K} \phi(\mathbf{x}) = 1$. The margin is a number in the range $[-1, 1]$, that is positive if the object is correctly classified, and negative otherwise. Large values of margin are interpreted as very "confident" correct classifications, while small margins are interpreted as unstable classifications. Therefore, maximizing the margins should involve lower testing error.

Schapire et al. [145] also presented a theoretical upper bound on the testing error that do not depend on the number of classifiers used. The bound is improved when the margin is large on the training set, in essence the theorem states that for any $\delta > 0$ and $\theta > 0$ with probability at least $1 - \delta$ over the choice of the training set $\mathbf{X}$, any ensemble of classifiers satisfy:

$$P_{\mathcal{D}} \leq P(training\ margin \leq \theta) + O(\frac{1}{\sqrt{N}}(\frac{\log N \log \mid \mathcal{C} \mid}{\theta^2} + \frac{1}{\delta})^{\frac{1}{2}}) \tag{2.13}$$

where $P_{\mathcal{D}}$ is the probability that the ensemble makes an error labelling a sample $\mathbf{x}$, $\mathcal{C}$ is the finite set of base classifiers, $P(training\ margin \leq \theta)$ is the probability that the margin of a random point from the training set does not exceed θ, and N the number of samples.

2.4 Boosted Adaptive Features

In this chapter we propose to add another adaptive layer to the Adaboost algorithm. Instead of using a fixed set of features to represent each sample, we propose to readapt at each round the feature extraction to the most difficult samples. Using this approach the training and test errors decrease faster, and the accuracies of the ensemble are higher. To perform the adaptive feature extraction different techniques can be taken into account depending on the used criteria, optimization of the class separation, and the assumptions made on the data. In our case, we have chosen a variant of the non negative matrix factorization (NMF) algorithm as a feature extractor, given that we assume that our proposal will deal only with visual data in problems of object recognition. The NMF algorithm has been shown to yield a parts based representation of the data, what makes the technique specially suited for visual pattern recognition problems [86], where usually the high dimensional data vectors can be represented as a linear combination of a sparse set of basis. Nevertheless, the

NMF technique is not appropriate for low dimensional problems, where it is difficult to distinguish parts from objects. Other feature extraction algorithms could be used depending on the final application.

One of the most successful applications of Adaboost learning in face classification can be the object recognition scheme proposed by Viola and Jones [180]. In their first implementation they used Haar-like basis functions to extract simple features from face and non face images for a face detection problem. Later they extended their work to multi-view face detection [72]. In this work they realized that the set of filters used in the first approximation was not enough to detect non-frontal faces, so they extended it adding filters for diagonal structures. In both cases, given the training samples, the features extracted were fixed during the learning process. In this chapter we propose a method for embedding the feature extraction into the Adaboost algorithm, in such a way that at each time the feature extraction is adapted to the classification problem.

The origin of the features has often been neglected in classification. Usually features are assumed to be a fixed set that exists regardless of the objects to classify. Nevertheless some studies have shown that humans are able to learn new features to discriminate better new objects. Schyns and Rodet made an experiment using three categories of Martian cells [150], one of them characterized by the feature X, another one by the feature Y and the third by both XY. They experimented with people divided in two groups. The first group learned first how to discriminate the objects based on the features X and Y, and then learned the objects XY. The second group learned first the type of objects based on the XY features, and then the ones based on the features X and Y. The results of the experiment showed that the members of the second group learned three features and did not realize that the third one (XY) was a composition of X and Y, while the members of the first group were able to categorize all the examples using the features X and Y. This study emphasizes that new features are learned during the process and these resulting features are highly related to the process of recognition followed.

In fact, the use of a fixed set of features upper bounds the amount of objects to recognize to a finite set, the features and combinations between features [125]. It seems logic to think that we must be able to evolve our feature set depending on the recognition problems that we need to solve if we live in a changing environment.

2.4.1 Sparse Feature Extraction

In the original NMF algorithm [86], the projection matrix is found under positivity constraints. This approach usually leads sparse bases and coefficients, and makes the technique specially suitable for computer vision problems such as object recognition, where usually local changes on the illumination and partial occlusions can mislead the classifier.

In addition, some studies have found physiological evidences about a parts-based representation of data in the brain of mammals [121, 93]. Wachsmuth et al. [182]

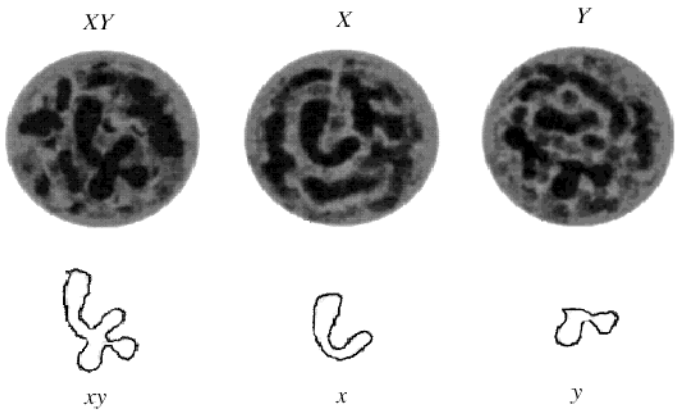


Figure 2.2: Examples of the 3 features that categorize the 3 classes X, Y and XY in the Schyns experiment [150].

investigate the response of cells from the temporal cortex of the macaque to the stimuli of a sight of a human body. The percentage of cells responding to the sight of an entire body (42%), to the sight of head alone (72%) and body alone (28%) shows that there is parts-based representation of complex objects in the brain. Moreover, they found that the majority (90%) of the cells respond selectively depending on the perspective view. It seems natural to think that natural evolution has discovered efficient cortical coding strategies for representing visual information. In this context, sparse representations have many advantages in terms of fault-tolerance and low-power consumption potential that can inspire computational systems [92] applied to cognitive systems and robotics.

2.4.2 Weighted Non Negative Matrix Factorization

When the NMF algorithm as shown in the introductory chapter is used in training sets that are far from a uniform distribution, often it appears a redundancy in the bases **B**. To solve this problem, Guillamet et al. [42, 40] introduced a weighted version of the NMF algorithm. The weights were used to give more importance to the samples that appear less frequently. It has been shown [59] that the weighted approach reduces the redundancy of the bases.

The formulation taking into account this weights adds a diagonal matrix $\mathbf{W}$ $N \times N$ in both sides of the factorization model:

$$\mathbf{XW} \approx (\mathbf{BSW}) \tag{2.14}$$

Where each element of the diagonal of $\mathbf{W}$ is the weight of its correspondent training sample. All the weights are normalized to sum 1. The addition of the weights modifies

the objective function to minimize. Two different objective functions have been used in the literature to find the iterative rules [86]:

- Minimizing the Euclidean distance between the vector samples from $\mathbf{X}$ and the approximation $\widehat{\mathbf{X}} = \mathbf{BS}$. The object function to minimize is:

$$\mathcal{D} = (\mathbf{B}, \mathbf{S}) = \sum_{j=1} \parallel \mathbf{X_j} - \mathbf{BS_j} \parallel = \sum_{i=1}^{D} \sum_{j=1}^{N} (X_{ij} - \sum_{l=1}^{M} B_{il} S_{lj}) \tag{2.15}$$

- Minimizing the Kullback-Leibler divergence measure $Div(\mathbf{X} \parallel \mathbf{BS})$. The iterative rules that solve this problem have been stated in the introductory chapter:

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \sum_{d} \frac{\mathbf{X}_{id}}{(\mathbf{BS})_{id}} \mathbf{S}_{jd} \tag{2.16}$$

$$\mathbf{B}_{ij} \leftarrow \frac{\mathbf{B}_{ij}}{\sum_k \mathbf{B}_{kj}} \tag{2.17}$$

$$\mathbf{S}_{jd} \leftarrow \mathbf{S}_{jd} \sum_{i} B_{ij} \frac{\mathbf{X}_{id}}{(\mathbf{BS})_{id}} \tag{2.18}$$

Using the last approach, and taking the diagonal $w_d = W_{dd}$, the objective function modified to accommodate weights becomes:

$$\mathcal{D} = (\mathbf{B}, \mathbf{S}) = \sum_{j=1}^{N} w_j \sum_{i=1}^{D} (X_{ij} \log (w_j (\mathbf{BS})_{ij} - (\mathbf{BS})_{ij}) \tag{2.19}$$

Considering $\mathbf{X}' = \mathbf{XW}$ $\mathbf{S}' = \mathbf{SW}$ and substituting in Eq.2.16-2.18, the following iterative rules are obtained for solving the WNMF factorization:

$$\mathbf{B_{ij}} \leftarrow \mathbf{B_{ij}} \sum_{\mathbf{d}} \frac{\mathbf{W_d X_{id}}}{(\mathbf{BS})_{\mathbf{id}}} \mathbf{S_{jd}} \tag{2.20}$$

$$\mathbf{B_{ij}} \leftarrow \frac{\mathbf{B_{ij}}}{\sum_{\mathbf{k}} \mathbf{B_{kj}}} \tag{2.21}$$

$$\mathbf{S_{jd}} \leftarrow \mathbf{S_{jd}} \sum_{\mathbf{i}} \mathbf{B_{ij}} \frac{\mathbf{X_{id}}}{(\mathbf{BS})_{\mathbf{id}}} \tag{2.22}$$

Notice that $\mathbf{S}$ remains the same, as the weight term appears only once in both in the numerator and denominator in the rule [59].

Table 2.4: Adaptive Adaboost algorithm with WNMF feature extraction.

Given the training samples $\mathbf{x}_1$, $\mathbf{x}_2$, $\ldots \mathbf{x}_N$

1. Initialize the weight vector to $1 \; \forall \; \mathbf{W}_{i=1,..N}$.
2. For $t = 1, 2, \ldots T$ do:
   (a) Resample the training data $\mathbf{X}$ according their actual weights, obtaining the samples $\mathbf{X_t}$.
   (b) Perform the WNMF feature extraction, obtaining the bases $\mathbf{B}$ and the coefficients $\mathbf{S_t}$ using the weights $\mathbf{W}$, and the samples $\mathbf{X_t}$.
   (c) Train a classifier $C_t(\mathbf{S_t})$.
   (d) Project the samples $\mathbf{X}$ using the bases $\mathbf{B}$, obtaining the coefficients $\mathbf{S}$.
   (e) Classify the samples $\mathbf{S}$ using C_t.
   (f) Compute the probability of the classification error taking into account the weights as follows:
   $$\varepsilon_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{W}_i^t \xi_i^t \tag{2.23}$$
   where
   $$\xi_i^t = \begin{cases} 1, & if \; \mathbf{x_i} \; was \; wrongly \; classified \; in \; the \; step \; s \\ 0, & otherwise \end{cases}$$
   (g) Compute ζ_t as :
   $$\zeta_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{2.24}$$
   (h) If $\varepsilon_t < 0.5$ set $\mathbf{W_i^{t+1}} = \mathbf{W_i^t} \exp(\zeta_\mathbf{t} \xi_\mathbf{i}^\mathbf{t})$ for each training vector i, and then normalize the weights in such a way that $\sum_{i=1}^{N} \mathbf{W_i^{t+1}} = \mathbf{n}$.
   Otherwise restart the algorithm
3. Finally the classifiers obtained are combined using a weighted majority voting using the coefficients ζ_t that encode a measure of the error in each step. The final decision rule for each new test vector is:
   $$O(\mathbf{x}) = \sum_t \zeta_t L_t > 0 \tag{2.25}$$
   where L is the label obtained in the classifier C_t, $L \in (-1, 1)$.

2.4.3 Boosting WNMF

In the modified version of the Adaboost, we do not assume that each sample is represented by a fixed set of features. Instead, we will use the best possible set of features to represent the samples at each step (according to the most difficult examples). We have modified the Adaboost algorithm to learn both the optimal combination of features and the weak classifiers. The resulting scheme is an open model that allows us to deal with problems where data can be variable such as the systematic presence of occlusions, changes in the objects appearance, or pose variations. The modified

algorithm includes the feature extraction into the Adaboost, using the WNMF technique. The table 2.4 summarizes the algorithm. The main difference with respect to Adaboost is the WNMF performed at 2.*b*. The WNMF is trained using the Adaboost weights of the selected samples. This focuses the feature extraction in the most difficult examples, which have larger Adaboost weight $\mathbf{W}$. The classifier at each step is trained using the proper feature extraction. The Adaboost base version for our proposal uses the resampling variant, and it is based on the two-class Adaboost described on [159]. The used weak classifier C_t is a single layer perceptron. Other weak classifiers could have been considered, such as a linear classifier or a single threshold.

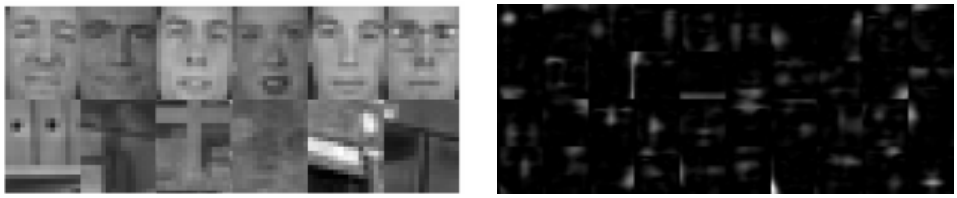
2.5 Experiments

The main goal of the method introduced in this chapter is focused on visual recognition problems, this fact justifies the election of the NMF algorithm for the feature extraction task. Two different experiments with visual data have been performed to show that including the feature extraction into the boosting scheme improves its accuracy: a face detection problem and a benchmark using the MNIST database.

2.5.1 Face Detection

For the face detection problem we have used a scheme similar to the one proposed in [181]. We have used 4000 images with faces extracted from two public databases (AR face [100] and XM2VTS [105]) and 25000 non face images extracted from natural images. The results shown are an average of a 5-fold cross validation, using a training set of 1000 faces and 4000 non faces randomly chosen from the original set. We have trained the adaptive Adaboost shown in 2.4 using similar settings as the ones described on the introductory chapter: sliding windows of 32×24 pixels have been generated from each image (labels for the face images where acquired manually), the center of the sliding window was moved 2 pixels each time. The classifier makes 300 rounds of boosting using the WNMF feature extraction, with 60 bases $\mathbf{B}$ at each round. With this set up, we have obtained 1.07% of faces wrongly classified (false negatives), and just 0.2% of non faces classified as a face (false positives). These results outperform the ones obtained on the face detection and classification example in the first chapter, where we obtained 2.45% of false negatives and 0.83% of false positives rates on each individual classifier. Nevertheless the computational needs of the adaptive approach make the algorithm unfeasible for a real time application. Projecting the data points using the WNMF algorithm takes more than one second, so a cascade of adaptive boosted classifiers can not be built for real time face detection as done following the Bayesian example. Figure 2.3 shows an example of faces an non face images used in the training and the bases found in one iteration of the WNMF algorithm.

The goal of the experiment was to show how an adaptive feature extraction in the boosting scheme can increase the final accuracy. For this purpose we have trained the



(a) (b)

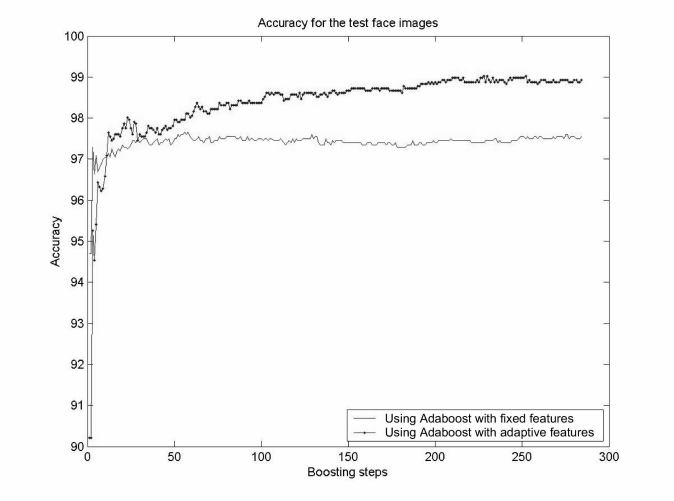
Figure 2.3: (a) Examples of face and non face images used in training. (b) Examples of sparse bases obtained using the weighted NMF algorithm.

same Adaboost scheme using a fixed set of features instead. Features for each sample where learned outside the algorithm (by applying classic NMF only once). The rest of the settings were the same as used in the adaptive experiment. In the figure 2.4 we plot the accuracies for the face and non-face images as a function of the boosting steps. Dotted lines show the accuracies using the adaptive scheme. As can be seen the best results are achieved using the adaptive scheme, from 30 boosting steps on.

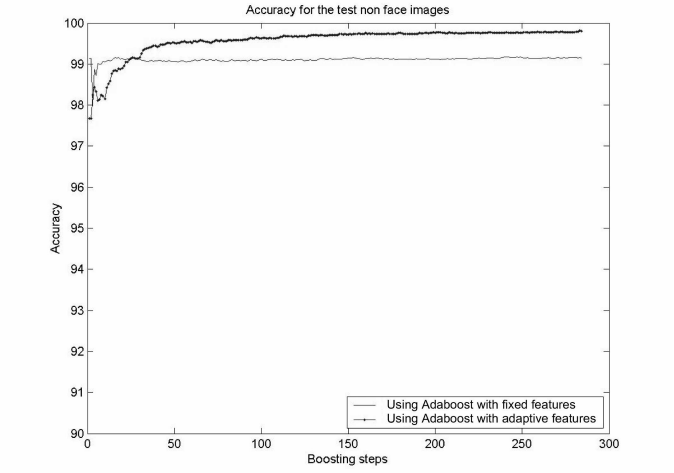
2.5.2 Digit Classification

In this experiment we have used the manuscript digits of the MNIST database [84]. We have trained a classifier for each of the 10 digits. Each training set was made using 1800 digits from one class and 1800 from the others. The rest of the MNIST digits were for testing. Figure 2.5 shows some examples of digits and 72 NMF bases generated using the training set. As can be seen bases are sparse an localized on specific parts of the digits. We have performed 5-fold cross validation and the results shown are an average of the accuracies of the 10 digits. As can be seen in the figure 2.6 the Adaptive approach reaches better accuracies than using fixed features even using a few Adaboost rounds.

In this experiment a new comparison using the adaptive approach with a simple NMF feature extraction technique has been added. The algorithm keeps adapting the feature set to the missclassified samples, given that the NMF is performed on data that has been resampled from the training set according to the Adaboost weights. But we do not use the WNMF approach for this case. It can be seen that when enough boosting steps are performed, both algorithms converge to the same accuracy. The difference between both approaches is that using the WNMF we achieve the accuracies faster, using less Adaboost iterations. This is due to the increase of the adaptability achieved using the weights inside the feature extraction algorithm.



(a)

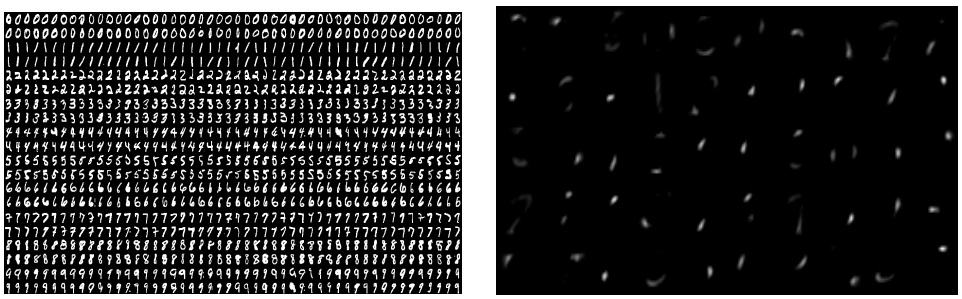


(b)

Figure 2.4: Comparison of the accuracies obtained as a function of the boosting steps performed for the face images (a) and non face images (b).

2.6 Conclusions

In this chapter a review of two successful classifier combination strategies has been performed: Bagging and Boosting. The Adaboost algorithm used in the feature extraction methodology proposed in the next chapter has been introduced. Adaboost has been shown to obtain an ensemble of classifiers that achieves a faster decrease of



(a) (b)

Figure 2.5: (a) Examples from different digits of the MNIST (b) Bases obtained using one training set.

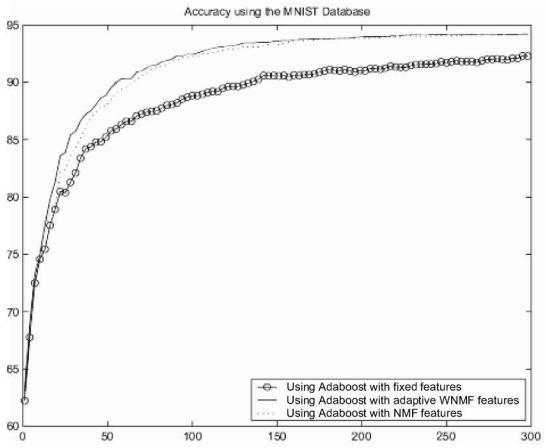


Figure 2.6: Comparison of the accuracies obtained as a function of the boosting steps performed for the MNIST database.

the training error. Actually, there is a theoretic bound on the training error, which should be reduced as more weak classifiers are added in the ensemble. Furthermore, there exist a theoretic bound on the testing error obtained from the margin theory.

A modification of the Adaboost algorithm where the feature extraction is introduced into the original algorithm has also been introduced. A weighted modification of the non negative matrix factorization algorithm has been used to perform the feature extraction focusing on the misclassified samples from previous steps. The experimental results on two independent data sets (a face detection and a manuscript digit recognition problems) show that the introduction of the feature extraction into the Adaboost algorithm achieves better final accuracies than classic Adaboost with fixed features.

The NMF algorithm has been proved to be specially robust in computer vision

problems where there are partial occlusions or local changes in the illumination, and has performed satisfactorily due to its straightforward integration into the boosting algorithm. Nevertheless, other more general feature extraction techniques could be analysed in future works, such as weighted Principal Component Analysis, or Independent Component Analysis. The main drawback of the NMF (and its weighted extension) is the time consuming projection step in the classification stage, given that the best algorithm known is iterative. The computationally intensive WNMF projection step has made the algorithm unviable for the real time face detection application, even when we have seen off line that accuracies obtained are sensible higher than the Bayesian model used.

Chapter 3

Internal Face Feature Extraction by Ensemble-based Methods

3.1 Introduction

The goal of feature extraction is to make evident interesting patterns from high dimensional data, in order to be used for classification of new unseen examples. Given their simplicity, linear transformations are the most used feature extraction techniques in practice. Different examples of linear transformations have been shown in previous chapters: Principal Component Analysis, Independent Component Analysis, Non Negative Matrix Factorization, as unsupervised methods, and the Linear Discriminant Methods as supervised. Both the supervised and unsupervised techniques shown make some kind of assumption on the data in the feature extraction task (positivity, specific class distribution, gaussianity,...). In this chapter, a new feature extraction technique for 2-class problems that makes no assumptions on the data distributions is proposed. The linear projection is incrementally found taking into account the classification errors of the training samples. This is achieved using an Adaboost-based algorithm that performs the selection of simple 1-Dimensional projections to build the final linear transformation.

In fact, the proposed feature extraction method is a general core algorithm from which a family of techniques can be derived depending on the generation of the individual 1-Dimensional projections. Three different extensions will be proposed in this chapter, the first one uses random projections, the second uses simple local deterministic projections, and the last one extends the fisher criterion to linear transformations of arbitrary dimension. In the next section an introduction to general feature extraction techniques using ensembles of classifiers will be performed. Then the detailed base algorithm will be exposed. In addition, a complete experimental test on the presented methods will be shown. The proposed methods are specially suitable for high dimensional subspaces (typical from face classification tasks) where often general

assumptions do not hold. Nevertheless, some experiments have been also performed on low dimensional data sets, extracted from the UCI repository.

3.2 Linear Feature Extraction

Feature extraction techniques try to find a subspace of dimensionality M in the original space of dimensionality D (usually $M < D$). In the case of linear transformations we have:

$$\mathbf{s} = \mathbf{A}\mathbf{x} \tag{3.1}$$

where $\mathbf{A}$ is the $M \times D$ projection matrix and $\mathbf{s}$ the extracted features. And if the projection matrix $\mathbf{A}$ is invertible, data can be reconstructed by:

$$\mathbf{x} = \mathbf{B}\mathbf{s} \tag{3.2}$$

where $\mathbf{B}=\mathbf{A}^{-1}$. The equation 3.2 can be viewed as the approximation of each original data vector by a weighted sum of the basis $\mathbf{B}$

In classifier combination schemes, usually the extracted features are used to construct multiple base classifiers to be used as an ensemble in classification tasks. We propose now to reverse the problem; that is, we will use the ensemble to extract features. The technique proposed is restricted to the two class problems, although we plan in future works an extension to the multi class case.

3.2.1 Previous Works on Feature Extraction Using Adaboost

The main contribution of this chapter is the use of Adaboost for linear feature extraction. Before explaining in detail the method proposed, a brief overview of some ensemble-based feature extraction algorithms used in the literature will be performed. Although our linear propose differs from the methods used in the past, some common ideas have been used from these previous works.

A feature selection method using Adaboost is proposed by Long and Vega in [94]. The authors use decision stumps as base classifiers so that each tree in the ensemble consists of a root and two leaves. The split at the root is done on a single feature, different for each classifier. At each step the selected feature is removed from the set of candidates for the next classifier. This method has been applied for microarray data where there is a large number of features and a small number of samples. The method can achieve two important goals, first the feature selection was performed on gene information, so finding the small subset of genes that classifies the data can inspire new research on the field, and also the diagnosis becomes cheaper as only a few measures must be performed.

Athitsos et al. [4], introduced the *BoostMap* multidimensional embedding in retrieval problems. They formulated the embedding process as a combination of simple 1-Dimensional embeddings that preserve information of proximity structure. They

transform the simple embeddings in classifier problems, and apply Adaboost for finding the best combination of the embeddings on the training data. Finally, data samples are classified using a weighted Manhattan distance obtained from the embedding.

Sirlantzis et al. [156] suggested a fusion scheme performing a 2 stage classification where first a n-tuple based classifier is used to extract intermediate features that are the input to train the second stage classifiers. The classification is performed combining the feature extraction resulting from the first stage. In a similar way, our proposal uses simple 1-Dimensional projections as individual classifiers to perform the feature extraction task. In a way, all ensemble methods can be viewed as "feature extractors". We can regard each classifier in the ensemble as a feature extractor, taking the original features as its inputs and producing a class label as the extracted feature. The combination of these extracted features can be perceived as the classifier in the new feature space.

3.3 Linear Feature Extraction using Adaboost

Assume that we use linear classifiers as the base classifiers in the ensemble to solve a two-class problem. Instead of thresholding the output of each classifier and taking the class label to be the output, we use the value computed as the linear combination to be our extracted feature. While any ensemble method can be applied for this feature extraction, here we chose Adaboost which has been declared to be "the best off-the-shelf" ensemble method [18].

As explained in the introductory chapter, linear discriminant analysis techniques tend to maximize a criteria under some assumptions on the data samples. Our proposed feature extraction technique makes no assumptions on the statistical distribution of the data, and it is not restricted to the case of orthogonal transformations. We propose to find the projection matrix incrementally, using a modified Adaboost algorithm. The original Adaboost algorithm [147, 49] (explained in detail in chapter 2) is based on incrementally building a set of classifiers that are combined in more powerful decision rule. At each boosting step a new classifier is generated, and the training samples are reweighted according to the classification results. The weights are used to generate the next step classifier. In our proposal at each boosting step we generate projection vectors (candidate linear features) and select the best one to be the extracted feature for this step. To evaluate a candidate feature, we build a classifier on it. The values of the feature are calculated for the training data and an optimal threshold is found minimizing the number of misclassified training samples. The weighted error is calculated using the weights for the data points at the present Adaboost step. The output of the algorithm is a projection matrix $\mathbf{W}$, that accumulates the different selected projections during the boosting process. Table 3.1 summarizes the generic algorithm.

Depending on the generation and the selection of the projections in 3(a) from table 3.1 at each boosting step, different methods can be derived from the general idea. In this thesis three variants of the algorithm are proposed.

Table 3.1: A generic learning algorithm for boosted discriminant projections.

1. Given are the matrix $\mathbf{X}$ containing data samples $\mathbf{x}_i$, and the vector $\mathbf{y}$ with the corresponding labels $y_i \in \{-1, 1\}$ $(i = 1 \dots N)$
2. Initialize a set of weights: $W_i(1) = \frac{1}{N}$.
3. For $t = 1 \dots M$:
   (a) Generate P projections from the original space to an $1-$dimensional subspace.
   (b) For $p = 1 \dots P$
      i. Project the training data into the $1-$dimensional space using projection p.
      ii. Learn the threshold that best separates the samples into two classes, thereby constructing hypothesis $h_{t,p}$ $(h_{t,p}(\mathbf{x}_i) \in \{-1, 1\}, \forall \mathbf{x}_i \in \mathbf{X})$. Denote by $l_{t,p}(\mathbf{x}_i)$ the loss incurred in labeling $\mathbf{x}_i$ by $h_{t,p}$. The loss is $l_{t,p}(\mathbf{x}_i) = 1$ if a misclassification occurs and $l_{t,p}(\mathbf{x}_i) = 0$, otherwise.
      iii. Compute the weighted error for the projection as:

$$Err_p = \sum_{i=1}^{N} W_i(t) l_{t,p}(\mathbf{x}_i). \tag{3.3}$$

   (c) Find the projection, m, with the minimum error, i.e., $Err_m = \min_{p=1}^{P} Err_p$. Classify the training set using m.
   (d) Calculate the weight for classifier t, β_t, as:

$$\beta_t = \frac{Err_m}{1 - Err_m} \tag{3.4}$$

   (e) Update the data weights:

$$W_i(t+1) = W_i(t) \beta_t^{(1 - l_{t,m}(\mathbf{x}_i))}, \quad i = 1, \dots, N. \tag{3.5}$$

   (f) Normalize the weights so that $W(t+1)$ is a distribution.

$$W_i(t+1) \leftarrow \frac{W_i(t+1)}{\sum_j W_j(t+1)} \tag{3.6}$$

   (g) Store m as the t-th projection in the projection matrix $\mathbf{W}$.
4. Output the projection matrix $\mathbf{W}$, built using the vectors selected at each boosting step as columns.

3.3.1 Random Boosted Discriminant Projections (RBDP)

Maybe the simplest way of generating 1−dimensional projections from the data is to randomly select pairs of points, one point from each class, and take the vector between the two as the candidate projection. It is hoped that the difference vector between points from different classes can have more discriminative information than using direct random vectors for the projection. Using a large enough set of candidate projections generated in this way, we can use Adaboost weights to select the projection with the smallest weighted error. In the general algorithm shown in table 3.1 the random generation explained above is implemented as 3(a). The parameters of the algorithm which need to be picked in advance are: M, the number of classifiers in the ensemble (desired number of projections); and P, the size of the "projection pool" (how many candidate-projections are generated at each step of Adaboost).

The calculation of the optimal threshold for projection p in 3.(b).(ii) follows the steps below. Let $\mathbf{p} = [p_1, p_2, \ldots, p_D]^T$ be the coefficient vector for projection p. First, calculate $y_i = \mathbf{p}^T\mathbf{x}_i$, $i = 1, \ldots, N$. Second, sort the y_i's. Third, calculate the classification error for all the values of the threshold in the middle between every two consecutive y_i's. Fourth, choose and retain the threshold with the minimum error.

Note that there is no need for calculating the final hypothesis in this Adaboost version because the information needed at the end is only the projection matrix $\mathbf{W}$.

3.3.2 Local Boosted Discriminant Projections (LBDP)

Another approach to feature extraction using Adaboost is to build a set of projections in a deterministic way. We implement 3(a) in Table 3.1 by the following steps:

1. For each point $\mathbf{x}_i$ find the nearest neighbor from the same class, $\mathbf{z}^{\text{same}}$, and the nearest neighbor from the opposite class, $\mathbf{z}^{\text{different}}$.

2. The points $\mathbf{x}_i$, $\mathbf{z}^{\text{same}}$ and $\mathbf{z}^{\text{different}}$ define a plane, γ, in the initial space of dimensionality D. We propose that the linear projection that we are looking for lies in γ. The transformation matrix $\mathbf{A}_{2\times D}$ is found using $\mathbf{x}_i$, $\mathbf{z}^{\text{same}}$ and $\mathbf{z}^{\text{different}}$. We construct vectors $\mathbf{v}$ and $\mathbf{w}$, $\mathbf{v}, \mathbf{w} \in \gamma$, as

$$\begin{aligned} \mathbf{v} &= [v_1, v_2]^T = \mathbf{x}_i - \mathbf{z}^{\text{same}} \\ \mathbf{w} &= [w_1, w_2]^T = \mathbf{x}_i - \mathbf{z}^{\text{different}}. \end{aligned} \tag{3.7}$$

   These vectors can be perceived as local descriptors for the within and between class distances. To illustrate the calculations, consider the following example for $D = 3$. Let $\mathbf{x}_i = [0, 0, 0]^T$, $\mathbf{z}^{\text{same}} = [1, 3, 6]^T$, and $\mathbf{z}^{\text{different}} = [5, 1, -2]^T$. The transformation matrix is

$$\mathbf{A} = \begin{bmatrix} 0.9129 & 0.1826 & -0.3651 \\ 0.1474 & 0.4423 & 0.8847 \end{bmatrix}.$$

The projections of the three points on the plane γ, as well as vectors $\mathbf{v}$ and $\mathbf{w}$ are shown in Figure 3.1.

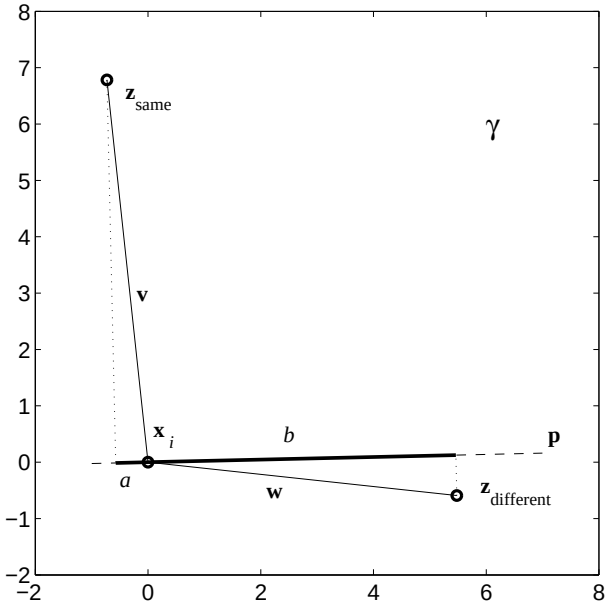


Figure 3.1: Projections of $\mathbf{x}_i$, $\mathbf{z}^{\text{same}}$ and $\mathbf{z}^{\text{different}}$ on γ. Vectors $\mathbf{v}$ and $\mathbf{w}$ are shown. The solid line indicates the direction of the optimal projection $\mathbf{p}$.

3. In the 2-D subspace, γ, we are looking for a direction vector $\mathbf{p}_i$ such that the projections of $\mathbf{x}_i$ and $\mathbf{z}^{\text{same}}$ on $\mathbf{p}_i$ are close to one another (distance a in Figure 3.1) while the projection of $\mathbf{z}^{\text{different}}$ on $\mathbf{p}_i$ is as far away from $\mathbf{x}_i$ as possible (distance b in Figure 3.1). Assuming that $\mathbf{p}_i$ has unit length, the distance between the projections of $\mathbf{x}_i$ and $\mathbf{z}^{\text{same}}$ on $\mathbf{p}_i$ is

$$|\mathbf{p}_i^T \mathbf{x}_i - \mathbf{p}_i^T \mathbf{z}^{\text{same}}| = |\mathbf{p}_i^T \mathbf{v}|. \tag{3.8}$$

Therefore a possible criterion function to maximize is

$$\max \left\{ \left(\mathbf{p}_i^T \mathbf{w}\right)^2 - \left(\mathbf{p}_i^T \mathbf{v}\right)^2 \right\}. \tag{3.9}$$

The four solutions of (3.9) for $\mathbf{p}_i = [p_1, p_2]^T$ are

$$\begin{aligned} p_1 &= \pm\sqrt{1-(p_2)^2}, \\ p_2 &= \sqrt{\frac{1}{2}\left(1 \pm \frac{w_2^2 - v_2^2 - w_1^2 + v_1^2}{\sqrt{4\left(w_1 w_2 - v_1 v_2\right)^2 + \left(w_2^2 - v_2^2 - w_1^2 + v_1^2\right)^2}}\right)}. \end{aligned} \tag{3.10}$$

For each $\mathbf{x}$ we take the solution that maximizes (3.9). Then we project back $\mathbf{p}_i$ in the original D-dimensional space using $\mathbf{A}^{-1}$. The analytic solutions of Equation 3.10 are obtained by deriving the objective function 3.9, in the Appendix C a complete proof of the solution can be found.

In Figure 3.2 we show an example on a 3-Dimensional space and the projection on the plane γ where the projection $\mathbf{p}$ that maximizes the separation is found.

Note that the pool of candidate projections for LBDP consists of N projections and can be calculated in advance, before running Adaboost.

3.3.3 Boosted Fisher Projections (BFP)

The third variant that we propose is to use more than two points to compute the projection at each boosting step. The P projections in 3.(a) are generated as follows:

1. Sample K points from each class according to the distribution defined by the Adaboost weights at the current step.
2. Perform Fisher Linear Discriminant Analysis (FLD) on the selected samples to obtain the projection that best separates the data.

We expect BFP to be more robust than RBDP because more points are involved in the calculation of each projection. Thus the projections in BFP are expected to be more accurate but more similar to one another compared to these in RBDP.

The classic Fisher Linear Discriminant Analysis technique has one important limitation: the final dimensionality is upper bounded by the number of classes. The BFP technique proposed overcomes this limitation obtaining projection matrices to arbitrary dimensionality, that can be even larger than the original space (dimensionality augmentation instead of dimensionality reduction).

We note that contrary to the previous feature extraction methods explained in chapter 1, we do not start with a criterion function to optimize but use heuristics which have proven to work for classifiers. It is not straightforward to find an explicit expression of the three criteria behind the three variants, especially when there is a random component involved.

Figure 3.3 shows 4 examples on two-class problems using the RBDP, LBDP and BFP methods, where there are gaussian and multimodal classes, and small/large noise presence. First, as it can be expected, the projections obtained by RBDP are almost arbitrary as seen in the four plots. This random behavior is due to the fact that only one single projection is computed, so the weights engine of the Adaboost has not taken part in the feature extraction. On the other hand, the BFP algorithm extracts a single feature using the classic FLD. If the number of points from each class participating in the calculation, K is chosen so that all points are used, BFP is exactly equivalent to FLD. In our example $K = 100$ which explains both the similarities and differences between BFP and FLD. Finally, LBDP finds a good projection in all four

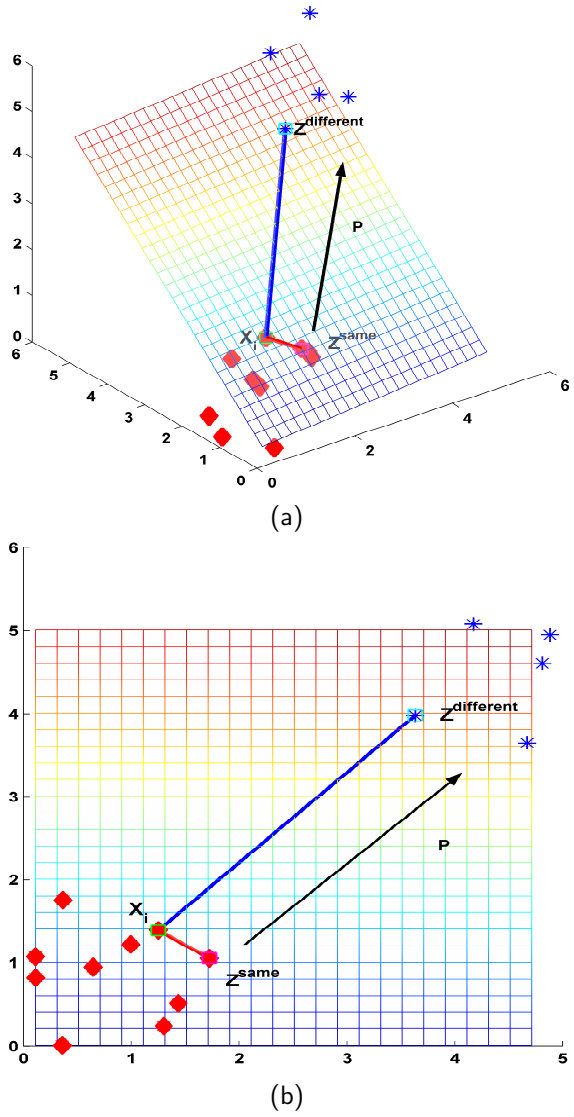


Figure 3.2: (a) 3-dimensional points $\mathbf{x}_i$, $\mathbf{z}^{\text{same}}$ and $\mathbf{z}^{\text{different}}$ and the plane where we restrict the solution. (b) 2-dimensional space where the directions $\mathbf{p}$ are found.

cases because it picks empirically the best projection out of N candidates. These findings are not unexpected as the strength of the proposed methods is supposed to come from applying Adaboost for constructing a collection of projections. This fact suggest that the techniques proposed are most suitable for high dimensional

subspaces, where the dimensionality reduction process can take several Adaboost rounds. In the experiments performed in the next section this fact will be empirically proved, specially for the case of visual problems such as face classification.

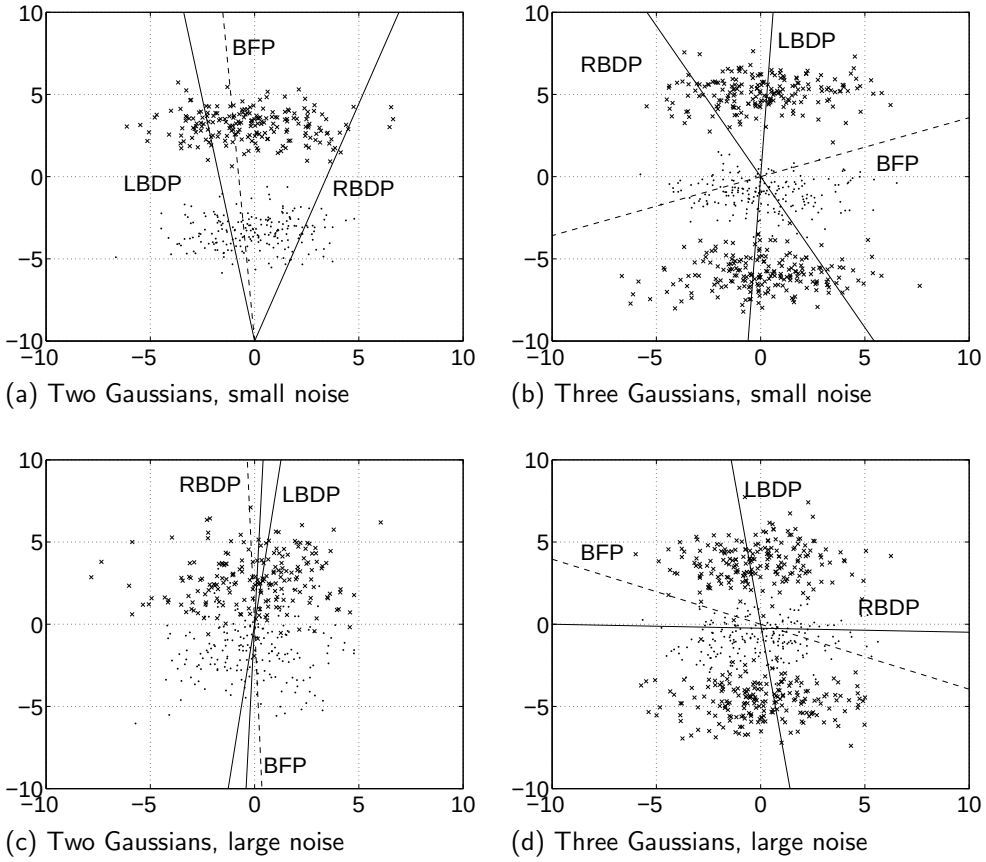


Figure 3.3: Examples of RBDP, LBDP and BFP for Gaussian ((a) and (c)) and non-Gaussian ((b) and (d)) classes for two levels of noise on the y-axis.

3.4 Experiments

The main goal of this section is to evaluate the performance of the feature extraction techniques proposed, and find in which cases are most suitable than classic discriminant analysis techniques. The experiments are performed on standard pattern recognition databases extracted from the UCI repository, synthetic data and face data. We compare the performance of the proposed feature extraction techniques with the methods based on classic discriminant analysis described in the introductory chapter.

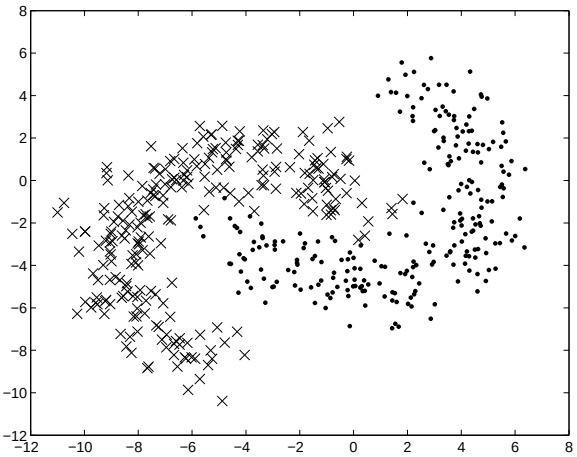


Figure 3.4: Scatter of a two dimensional projection of the 8-D banana shaped dataset.

Other comparative studies between classic discriminant analysis techniques can be found in [107]. Therefore, in our experimental study we compare Fisher discriminant analysis (FLD), the nonparametric discriminant analysis (NDA), the discriminant analysis using the Chernoff criterion (Chernoff), and the three techniques proposed here: the random boosted discriminant projections (RBDP), the local boosted discriminant projections (LBDP), and the boosted Fisher projections (BFP).

We have tested the six methods with 11 data sets, 8 of them taken from the UCI machine learning repository [13]. We also have generated two synthetic data sets in a similar way as done in [158]:

- *8-D Banana shape*. In the space of the first two features, the two classes are uniformly distributed along two concentric arcs with radii $r_1 = 0.125$ and $r_2 = 0.375$, respectively. Gaussian noise with unit variance is added to each class. The remaining eight features have Gaussian distribution with mean 0 and variance 0.1. Figure 3.4 shows an scatter plot of the first two features.
- *500-D Gaussian data*: This set consists of two Gaussian classes. The covariance matrices for the classes are the identity matrices. The mean for the first class is $[0, 0, \ldots, 0]^T$, and the mean for the second class is $[0.1, 0, 1, \ldots, 0.1]^T$.

The last data set is extracted from the AR Face database [100], and the goal is to solve a gender recognition problem, showing that Adaboost-based feature extraction techniques are most suitable for face classification given that faces are usually represented as high dimensional vectors. We have taken 500 examples from male and female face images, each image represented as a 2964-dimensional vector. Figure 3.5 shows and example with male and female images used in the experiment.

In Table 3.2 the most important characteristics of the 11 data sets are explained in

Figure 3.5: Example of face images taken from the AR Face database for the gender recognition problem.

detail. To perform the experiments some functions from the PRTOOLS 3.1.7 toolbox [140] have been used. Our comparative study uses four different classifiers in the space of the extracted features:

- Linear classifier, which assumes normal distribution of the classes and equal covariance matrices. To avoid computational problems due to the appearance of nearly singular covariance matrices (when the number of training examples is smaller than the data dimensionality), we have set the regularization parameter to 0.5.
- Quadratic classifier, also assuming normal distributions but with arbitrary covariance matrices. The regularization parameter has been set to 0.5.
- The 1-nearest neighbor classifier.
- Support Vector Machines classifier (SVM), using radial basis functions (with parameters $\sigma = 1$, and cost $C = 1$).

3.4.1 Experimental Protocol

For each data set we performed one hundred times the experiment described below:

1. First the data set is randomly split into training data and testing data, using 90% of the samples for training and the rest for testing.
2. Following [95], we transform the data using PCA. We compute a PCA projection matrix using the training samples. We select the eigenvectors corresponding to eigenvalues larger than 10^{-7}. The training and testing vectors are projected using the same PCA projection matrix.
3. Next we find a projection matrix using each of the six feature extraction algorithms.

Table 3.2: The 11 data sets used in the experiments. We show the database name, dimensionality D, the number of features preserved after the PCA was performed, the total number of samples available (after removing the samples with missing values), and the Sparseness of the data (number of data points/dimensionality). The two last data sets are separated to indicate that they have considerably larger dimensionality.

Database	Label	Feat.	F. PCA	Samples	Sparseness
BUPA Liver disorder	(a)	6	6	345	57.50
Wisconsin diagnostic breast cancer	(b)	30	7	569	81.28
$8-$D Banana shaped data	(c)	8	8	500	62.50
Wisconsin breast cancer	(d)	9	9	666	74.00
Cleveland heart disease	(e)	13	13	297	22.84
German database	(f)	24	24	1000	41.67
Ionosphere database	(g)	34	33	351	10.63
Sonar signals database	(h)	60	59	208	3.52
SPECTF heart	(i)	44	44	349	7.93
$500-$D Gaussian	(j)	500	449	500	1.11
AR Face database	(k)	2964	453	500	1.10

Table 3.3: Maximum number of different projections, M, for the six compared feature extraction techniques (D is the initial dimensionality of the original data, N_1 and N_2 are the sample sizes for the two classes ($N = N_1 + N_2$) and K is the number of samples from each class used for a projection construction in BFP).

Method	FLD	NDA	Chernoff	RBDP	LBDP	BFP
Maximum M	1	D	D	$N_1 N_2$	N	$\binom{N_1}{K}\binom{N_2}{K}$

4. For each feature extraction algorithm we train the four classifiers (linear, quadratic, nearest neighbor and SVM) on the space of extracted features. We store the accuracies on the training and the testing sets for each possible dimension up to $M = 40$. For example, FLD allows for one feature only whereas in RBDP there can be $N_1 N_2$ different projections, where N_1 and N_2 are the number of samples from classes C_1 and C_2, respectively ($N = N_1 + N_2$). Table 3.3 shows the maximum number of different projections that each of the six compared techniques can extract.

The results from the experiments are shown in Tables 3.4-3.7. These are computed as follows: for each feature extraction algorithm and each classifier, we find the number of extracted features M^{tr} for which the training error is minimum. Then using this dimensionality we take its corresponding error on the testing results. The final error rates and optimal dimensionality shown in the tables are the means of the one hundred runs.

Table 3.4: MCE results with the Linear classifier

DB	FLD	NDA	Chernoff
(a)	**0.317*** $\pm 1.6^{-2}$ (1.0)	0.355 $\pm 1.8^{-2}$ (4.8)	**0.349** $\pm 1.6^{-2}$ (5.1)
(b)	**0.052*** $\pm 0.6^{-2}$ (1.0)	0.093 $\pm 0.7^{-2}$ (3.7)	0.189 $\pm 1.0^{-2}$ (6.2)
(c)	**0.154** $\pm 0.9^{-2}$ (1.0)	**0.156** $\pm 0.9^{-2}$ (3.9)	0.512 $\pm 1.8^{-2}$ (2.9)
(d)	**0.041** $\pm 0.5^{-2}$ (1.0)	**0.044** $\pm 0.5^{-2}$ (3.6)	0.061 $\pm 1.4^{-2}$ (6.0)
(e)	**0.167*** $\pm 1.3^{-2}$ (1.0)	0.239 $\pm 1.6^{-2}$ (5.6)	0.386 $\pm 2.9^{-2}$ (5.2)
(f)	**0.131** $\pm 1.1^{-2}$ (1.0)	**0.136** $\pm 1.0^{-2}$ (19.4)	0.291 $\pm 1.9^{-2}$ (2.0)
(g)	**0.235*** $\pm 0.8^{-2}$ (1.0)	0.277 $\pm 0.8^{-2}$ (18.5)	0.312 $\pm 1.1^{-2}$ (1.0)
(h)	**0.251** $\pm 1.9^{-2}$ (1.0)	**0.260** $\pm 1.9^{-2}$ (30.8)	0.474 $\pm 1.9^{-2}$ (1.3)
(i)	0.234 $\pm 1.3^{-2}$ (1.0)	0.240 $\pm 1.4^{-2}$ (35.1)	0.302 $\pm 1.9^{-2}$ (4.5)
(j)	0.491 $\pm 1.2^{-2}$ (1.0)	0.402 $\pm 1.3^{-2}$ (36.0)	0.542 $\pm 1.2^{-2}$ (6.0)
(k)	0.448 $\pm 1.4^{-2}$ (1.0)	0.085 $\pm 0.8^{-2}$ (24.3)	0.518 $\pm 1.6^{-2}$ (5.9)
Rank	2.73	3.82	5.91

DB	RBDP	LBDP	BFP
(a)	**0.343** $\pm 1.5^{-2}$ (4.5)	**0.341** $\pm 1.6^{-2}$ (4.8)	**0.334** $\pm 1.7^{-2}$ (3.5)
(b)	0.117 $\pm 0.7^{-2}$ (4.7)	0.114 $\pm 0.8^{-2}$ (3.2)	**0.059** $\pm 0.6^{-2}$ (3.4)
(c)	**0.152** $\pm 0.9^{-2}$ (1.3)	**0.151*** $\pm 0.9^{-2}$ (5.1)	**0.162** $\pm 1.0^{-2}$ (4.2)
(d)	**0.040** $\pm 0.5^{-2}$ (5.5)	**0.040** $\pm 0.5^{-2}$ (4.6)	**0.040*** $\pm 0.5^{-2}$ (4.0)
(e)	0.293 $\pm 1.5^{-2}$ (5.5)	0.292 $\pm 1.6^{-2}$ (6.3)	**0.173** $\pm 1.4^{-2}$ (6.2)
(f)	**0.141** $\pm 1.2^{-2}$ (21.0)	**0.141** $\pm 1.1^{-2}$ (19.7)	**0.129*** $\pm 1.1^{-2}$ (19.9)
(g)	0.294 $\pm 0.9^{-2}$ (5.5)	0.294 $\pm 0.9^{-2}$ (13.7)	0.272 $\pm 1.0^{-2}$ (2.8)
(h)	**0.244** $\pm 1.7^{-2}$ (23.9)	**0.237*** $\pm 1.7^{-2}$ (29.0)	0.292 $\pm 2.0^{-2}$ (9.2)
(i)	**0.201*** $\pm 1.4^{-2}$ (16.3)	**0.206** $\pm 1.4^{-2}$ (24.3)	**0.225** $\pm 1.5^{-2}$ (23.3)
(j)	**0.297** $\pm 1.1^{-2}$ (37.8)	0.306 $\pm 1.2^{-2}$ (36.8)	**0.278*** $\pm 1.2^{-2}$ (30.7)
(k)	0.097 $\pm 0.9^{-2}$ (28.8)	0.091 $\pm 0.8^{-2}$ (28.5)	**0.025*** $\pm 0.4^{-2}$ (6.4)
Rank	3.36	2.91	2.27

Table 3.5: MCE results with the Quadratic classifier

DB	FLD	NDA	Chernoff
(a)	**0.363*** $\pm 1.6^{-2}$ (1.0)	0.427 $\pm 1.6^{-2}$ (3.5)	**0.385** $\pm 1.8^{-2}$ (3.1)
(b)	**0.055*** $\pm 0.6^{-2}$ (1.0)	0.099 $\pm 0.6^{-2}$ (2.5)	0.158 $\pm 0.9^{-2}$ (4.8)
(c)	**0.154** $\pm 0.9^{-2}$ (1.0)	**0.156** $\pm 0.9^{-2}$ (3.7)	0.503 $\pm 1.7^{-2}$ (3.1)
(d)	**0.030** $\pm 0.4^{-2}$ (1.0)	**0.033** $\pm 0.4^{-2}$ (2.0)	0.047 $\pm 0.7^{-2}$ (3.2)
(e)	**0.168*** $\pm 1.3^{-2}$ (1.0)	0.230 $\pm 1.5^{-2}$ (11.4)	0.378 $\pm 2.8^{-2}$ (5.3)
(f)	0.130 $\pm 1.1^{-2}$ (1.0)	0.091 $\pm 0.9^{-2}$ (15.2)	0.263 $\pm 3.4^{-2}$ (4.6)
(g)	**0.234*** $\pm 0.8^{-2}$ (1.0)	0.284 $\pm 0.8^{-2}$ (21.5)	0.328 $\pm 1.2^{-2}$ (2.3)
(h)	0.251 $\pm 1.9^{-2}$ (1.0)	0.210 $\pm 1.7^{-2}$ (30.9)	0.505 $\pm 2.1^{-2}$ (5.2)
(i)	0.254 $\pm 1.3^{-2}$ (1.0)	**0.220** $\pm 1.5^{-2}$ (31.4)	**0.214** $\pm 3.1^{-2}$ (8.3)
(j)	0.491 $\pm 1.2^{-2}$ (1.0)	0.480 $\pm 1.4^{-2}$ (25.1)	0.506 $\pm 1.4^{-2}$ (17.2)
(k)	0.448 $\pm 1.4^{-2}$ (1.0)	0.049 $\pm 0.6^{-2}$ (30.7)	0.428 $\pm 1.6^{-2}$ (18.0)
Rank	3.00	3.82	5.27

DB	RBDP	LBDP	BFP
(a)	0.399 $\pm 1.8^{-2}$ (1.6)	0.399 $\pm 1.9^{-2}$ (1.5)	**0.364** $\pm 1.6^{-2}$ (2.0)
(b)	0.074 $\pm 0.5^{-2}$ (2.6)	0.068 $\pm 0.6^{-2}$ (1.9)	**0.057** $\pm 0.5^{-2}$ (2.2)
(c)	**0.149*** $\pm 0.9^{-2}$ (1.5)	**0.149** $\pm 0.9^{-2}$ (5.3)	**0.164** $\pm 1.0^{-2}$ (4.0)
(d)	**0.029*** $\pm 0.4^{-2}$ (1.1)	**0.031** $\pm 0.4^{-2}$ (2.5)	**0.031** $\pm 0.4^{-2}$ (2.6)
(e)	0.315 $\pm 1.6^{-2}$ (5.0)	0.313 $\pm 1.6^{-2}$ (6.5)	**0.181** $\pm 1.5^{-2}$ (8.2)
(f)	0.084 $\pm 0.9^{-2}$ (15.7)	**0.061*** $\pm 0.8^{-2}$ (23.1)	**0.068** $\pm 0.9^{-2}$ (7.8)
(g)	0.299 $\pm 0.8^{-2}$ (8.1)	0.291 $\pm 0.8^{-2}$ (11.4)	0.272 $\pm 0.9^{-2}$ (3.5)
(h)	**0.204** $\pm 1.8^{-2}$ (22.0)	**0.172*** $\pm 1.6^{-2}$ (32.3)	0.300 $\pm 2.0^{-2}$ (7.7)
(i)	0.265 $\pm 1.5^{-2}$ (2.0)	0.291 $\pm 1.5^{-2}$ (3.0)	**0.214*** $\pm 1.3^{-2}$ (9.3)
(j)	0.314 $\pm 1.2^{-2}$ (38.8)	0.362 $\pm 1.3^{-2}$ (38.5)	**0.278*** $\pm 1.2^{-2}$ (27.0)
(k)	0.092 $\pm 0.9^{-2}$ (31.5)	0.083 $\pm 0.8^{-2}$ (27.6)	**0.025*** $\pm 0.4^{-2}$ (5.7)
Rank	3.36	3.09	2.45

Table 3.6: MCE results with the Nearest Neighbor classifier

DB	FLD	NDA	Chernoff
(a)	0.414 $\pm 1.8^{-2}$ (1.0)	**0.370*** $\pm 1.7^{-2}$ (4.0)	0.442 $\pm 1.5^{-2}$ (1.0)
(b)	0.082 $\pm 0.7^{-2}$ (1.0)	**0.045*** $\pm 0.5^{-2}$ (5.9)	0.392 $\pm 1.3^{-2}$ (1.0)
(c)	0.192 $\pm 1.0^{-2}$ (1.0)	**0.023*** $\pm 0.4^{-2}$ (4.2)	0.498 $\pm 1.3^{-2}$ (1.9)
(d)	**0.041*** $\pm 0.5^{-2}$ (1.0)	**0.045** $\pm 0.5^{-2}$ (4.5)	0.074 $\pm 1.1^{-2}$ (1.0)
(e)	**0.229*** $\pm 1.6^{-2}$ (1.0)	0.285 $\pm 1.6^{-2}$ (7.7)	0.452 $\pm 2.0^{-2}$ (1.7)
(f)	0.170 $\pm 1.2^{-2}$ (1.0)	**0.123** $\pm 1.1^{-2}$ (11.5)	0.385 $\pm 2.0^{-2}$ (7.9)
(g)	**0.314*** $\pm 0.9^{-2}$ (1.0)	**0.315** $\pm 0.9^{-2}$ (12.2)	0.399 $\pm 1.1^{-2}$ (2.9)
(h)	0.267 $\pm 2.0^{-2}$ (1.0)	**0.175** $\pm 1.4^{-2}$ (18.9)	0.483 $\pm 2.3^{-2}$ (5.2)
(i)	0.143 $\pm 1.3^{-2}$ (1.0)	**0.130** $\pm 1.1^{-2}$ (21.0)	0.185 $\pm 2.8^{-2}$ (4.8)
(j)	0.419 $\pm 1.2^{-2}$ (1.0)	0.469 $\pm 1.3^{-2}$ (4.8)	0.517 $\pm 1.3^{-2}$ (27.2)
(k)	0.033 $\pm 0.5^{-2}$ (1.0)	0.040 $\pm 0.6^{-2}$ (9.3)	0.431 $\pm 1.4^{-2}$ (16.5)
Rank	3.27	2.64	6.00

DB	RBDP	LBDP	BFP
(a)	**0.384** $\pm 1.5^{-2}$ (3.2)	**0.397** $\pm 1.5^{-2}$ (2.0)	**0.372** $\pm 1.6^{-2}$ (4.3)
(b)	0.091 $\pm 0.7^{-2}$ (4.0)	0.081 $\pm 0.6^{-2}$ (1.7)	0.070 $\pm 0.6^{-2}$ (4.9)
(c)	**0.030** $\pm 0.5^{-2}$ (5.2)	**0.029** $\pm 0.4^{-2}$ (4.2)	0.152 $\pm 1.2^{-2}$ (5.7)
(d)	**0.041** $\pm 0.5^{-2}$ (3.7)	**0.048** $\pm 0.5^{-2}$ (1.5)	**0.044** $\pm 0.5^{-2}$ (4.1)
(e)	0.417 $\pm 1.5^{-2}$ (3.2)	0.397 $\pm 1.6^{-2}$ (1.2)	**0.245** $\pm 1.5^{-2}$ (4.7)
(f)	0.116 $\pm 1.1^{-2}$ (11.4)	**0.125** $\pm 1.0^{-2}$ (11.6)	**0.105*** $\pm 1.0^{-2}$ (6.1)
(g)	0.394 $\pm 1.0^{-2}$ (7.4)	0.386 $\pm 1.1^{-2}$ (1.5)	**0.314** $\pm 0.9^{-2}$ (14.3)
(h)	**0.180** $\pm 1.6^{-2}$ (15.3)	**0.161*** $\pm 1.5^{-2}$ (23.8)	0.365 $\pm 2.1^{-2}$ (4.5)
(i)	**0.114** $\pm 1.2^{-2}$ (8.3)	**0.108*** $\pm 1.2^{-2}$ (3.9)	0.134 $\pm 1.1^{-2}$ (8.8)
(j)	0.399 $\pm 1.3^{-2}$ (25.5)	0.451 $\pm 1.5^{-2}$ (26.1)	**0.289*** $\pm 1.1^{-2}$ (34.9)
(k)	0.057 $\pm 0.7^{-2}$ (28.2)	0.040 $\pm 0.6^{-2}$ (30.4)	**0.022*** $\pm 0.4^{-2}$ (6.8)
Rank	3.36	3.27	2.45

Table 3.7: MCE results with the Support Vector Machines Classifier

DB	FLD	NDA	Chernoff
(a)	0.324 $\pm 1.6^{-2}$ (1.0)	**0.273** $\pm 1.3^{-2}$ (3.1)	0.344 $\pm 1.4^{-2}$ (2.2)
(b)	0.056 $\pm 0.6^{-2}$ (1.0)	**0.041** $\pm 0.5^{-2}$ (3.7)	0.068 $\pm 0.6^{-2}$ (3.9)
(c)	0.152 $\pm 1.0^{-2}$ (1.0)	**0.024** $\pm 0.4^{-2}$ (2.3)	0.208 $\pm 1.4^{-2}$ (7.7)
(d)	**0.028** $\pm 0.4^{-2}$ (1.0)	**0.027** $\pm 0.4^{-2}$ (1.3)	**0.034** $\pm 0.5^{-2}$ (2.6)
(e)	**0.165** $\pm 1.4^{-2}$ (1.0)	0.221 $\pm 1.4^{-2}$ (2.6)	0.231 $\pm 1.6^{-2}$ (8.1)
(f)	**0.225*** $\pm 0.7^{-2}$ (1.0)	0.272 $\pm 0.8^{-2}$ (3.2)	0.256 $\pm 0.8^{-2}$ (15.1)
(g)	0.130 $\pm 1.1^{-2}$ (1.0)	0.097 $\pm 0.9^{-2}$ (4.2)	**0.049*** $\pm 0.6^{-2}$ (17.3)
(h)	0.268 $\pm 1.6^{-2}$ (1.0)	0.226 $\pm 1.7^{-2}$ (2.1)	0.470 $\pm 2.2^{-2}$ (1.0)
(i)	0.248 $\pm 2.2^{-2}$ (1.0)	**0.057** $\pm 1.2^{-2}$ (5.5)	**0.057** $\pm 3.1^{-2}$ (4.6)
(j)	0.505 $\pm 1.2^{-2}$ (1.0)	0.435 $\pm 1.0^{-2}$ (5.7)	0.423 $\pm 1.8^{-2}$ (7.4)
(k)	0.142 $\pm 1.6^{-2}$ (1.0)	0.139 $\pm 0.8^{-2}$ (23.2)	0.122 $\pm 1.4^{-2}$ (12.3)
Rank	4.82	3.09	3.36

DB	RBDP	LBDP	BFP
(a)	0.383 $\pm 1.4^{-2}$ (1.4)	0.340 $\pm 1.4^{-2}$ (1.8)	**0.271*** $\pm 1.4^{-2}$ (3.2)
(b)	0.370 $\pm 1.3^{-2}$ (1.0)	0.088 $\pm 1.4^{-2}$ (1.0)	**0.038*** $\pm 0.5^{-2}$ (2.9)
(c)	0.132 $\pm 1.2^{-2}$ (1.5)	**0.020*** $\pm 0.4^{-2}$ (2.3)	0.106 $\pm 0.8^{-2}$ (4.7)
(d)	0.042 $\pm 0.5^{-2}$ (1.0)	**0.028** $\pm 0.4^{-2}$ (1.7)	**0.025*** $\pm 0.4^{-2}$ (1.8)
(e)	0.448 $\pm 1.8^{-2}$ (1.4)	0.354 $\pm 1.6^{-2}$ (1.5)	**0.145*** $\pm 1.3^{-2}$ (2.5)
(f)	0.294 $\pm 0.9^{-2}$ (1.9)	0.278 $\pm 0.9^{-2}$ (2.7)	0.248 $\pm 1.0^{-2}$ (3.1)
(g)	0.088 $\pm 0.8^{-2}$ (3.7)	**0.054** $\pm 0.6^{-2}$ (9.0)	0.082 $\pm 0.9^{-2}$ (2.9)
(h)	**0.105** $\pm 1.3^{-2}$ (12.3)	**0.101*** $\pm 1.3^{-2}$ (15.0)	0.329 $\pm 1.6^{-2}$ (1.5)
(i)	**0.058** $\pm 1.1^{-2}$ (1.4)	**0.059** $\pm 1.1^{-2}$ (2.9)	**0.055*** $\pm 1.2^{-2}$ (4.9)
(j)	0.446 $\pm 1.1^{-2}$ (17.5)	**0.377*** $\pm 0.9^{-2}$ (23.1)	0.442 $\pm 0.9^{-2}$ (31.6)
(k)	**0.058*** $\pm 1.3^{-2}$ (17.8)	**0.064** $\pm 1.4^{-2}$ (33.0)	0.114 $\pm 1.3^{-2}$ (12.7)
Rank	3.27	3.73	2.73

3.4.2 Experimental Results

Tables 3.4-3.7 show the mean classification error (MCE) for each database, and the mean optimal dimensionality in the brackets. Also we have marked in bold and with an '*' the method that achieves the minimum MCE for each database. We have also computed the 95% confidence interval for each MCE value, and have marked in bold the MCE of the methods whose confidence intervals overlap with the interval for the best method.

Since the values of MCE are not directly comparable across the data sets and the classifiers, in order to have a measure of overall performance, we calculated the ranks for the 6 compared methods. Each row in Tables 3.4-3.7 is arranged in ascending order of MCE and ranks are assigned to the methods. The method with the smallest MCE obtains rank 1 (best) and the method with the largest MCE obtains rank 6 (worst) for the particular data set and classifier. For example, the first row in Table 3.4 corresponds to the BUPA liver disorder data classified in the space of extracted features by the linear discriminant classifier. FLD gets rank 1 (smallest MCE=0.317), LBDP gets rank 2, etc., and NDA gets rank 6. The ranks for each method were then averaged across the 11 data sets and are shown at the bottom of the respective table.

In general, for the four classifiers, we observed that as dimensionality of the data increases, the ensemble based methods perform better than the other methods. Among the three ensemble algorithms, BFP achieved best overall result for all four classifiers.

As expected, for the nearest neighbor classifier, NDA is either the best or not significantly different from the best method in 8 of the 11 databases. In fact, NDA is specially designed to achieve low errors using NN. However, the performance of NDA is considerably worse for the Linear and Quadratic Classifiers.

Two tendencies can be observed from Tables 3.4-3.7: when the dimensionality of the data is high, the methods based on Adaboost perform better in almost all databases and classifiers. However when original data is low dimensional, FLD often achieves the best result, despite using only a single dimension.

The MCEs are in generally lower using the SVM classifier. The Chernoff technique ranks much better with SVM than with any of the other three classifiers.

We found that the higher the dimensionality is, the clearer becomes the advantage of the ensemble feature extraction. This is demonstrated in data sets (j) and (k). The BFP is significantly better than all the other methods except for RBDP with the linear classifier (Table 3.4). The overall ranks suggest that BFP is the most successful one among the examined feature extraction techniques. We conjecture that the random component combined with the weighting mechanism of Adaboost are responsible for the good performance of the ensemble feature extraction methods in high dimensional spaces.

The second best method is FLD which shows that for many cases the simple classical methods might be the best solution. Disappointingly, Chernoff was the worst of the methods compared here. While it is optimal for heteroscedastic data, other data distributions appear to be a challenge for this method. In the reference where

Chernoff method was advocated [95], regularization via special parameter was not considered. The pre-processing through PCA ensures that the covariance matrices are not singular and thus no further regularization has been suggested. Here we used the Matlab implementation of Chernoff mapping found in PRTOOLS 3.1.7 [140]. This implementation does not provide for a regularization parameter either.

We assume that Adaboost based feature extraction techniques perform better on high dimensional databases. Nevertheless, the sparseness of the data could also be the reason of this good performance. To examine the relationship between the sparseness of the data and the feature extraction methods we plotted the mean classification error (MCE) against logarithm of the sparseness. Figure 3.6 shows the graphs for the two most successful feature extraction methods, Fisher's Linear Discriminant(dashed line) and the proposed variant of Adaboost feature extraction, BFP (solid line). The left subplot gives the results for the linear classifier and the right subplot gives the results with SVM.

The graphs show that sparseness is not strongly related to classification error nor is it a reliable guide as to which type of feature extraction method should be preferred. For data with large dimensionality (low value of the sparseness index), we found that the ensemble feature extraction gives lower classification error (subplot (a)). On the other hand, the features selected by BFP are more useful for SVM for data sets in the middle range of sparseness (subplot (b)). For large values of the sparseness index the results are almost identical for both classifiers. This gives us ground to propose that the simple Fisher's linear discriminant may be sufficient when the ratio data size to dimensionality is large. On the other hand, it has been shown that when the dimensionality of the data is large, Adaboost based feature extraction obtains the best performance.

BFP has two parameters to be tuned, the number of samples K, taken from each class at each step, and the number of projections P, from which we choose at each step. In all our experiments these parameters were set to $K = 100$, and $P = 1$. We believe that the proposed methods are not critically sensitive to the choice of these parameters. A sensitivity study with respect to K and P is a possible future direction, but it seems that these parameters should be tuned for the specific problem. Nevertheless, the methods proposed seem to be robust to a wide range of choice of these parameters.

In addition, an experimental study of the accuracy evolution as a function of the dimensionality reduction in the Adaboost-based feature extraction has been performed. We have used the LBDP technique in this experiment, given that it does not involve any random factor. Two typical visual databases have been used: a face database on a gender recognition problem, and the MNIST manuscript digit database, where two similar digits have been selected for the discriminative task (numbers 1 and 7).

The face database was taken from the AR Face and the XM2VS databases [105], an consists of 2500 images instead of only 500 (we just leave out images with strong occlusions). The resulting data set has strong changes in illumination making necessary a previous normalization with respect to the mean and variance. Figure 3.7 shows some samples of faces and manuscript digits used in the experiment. The ex-

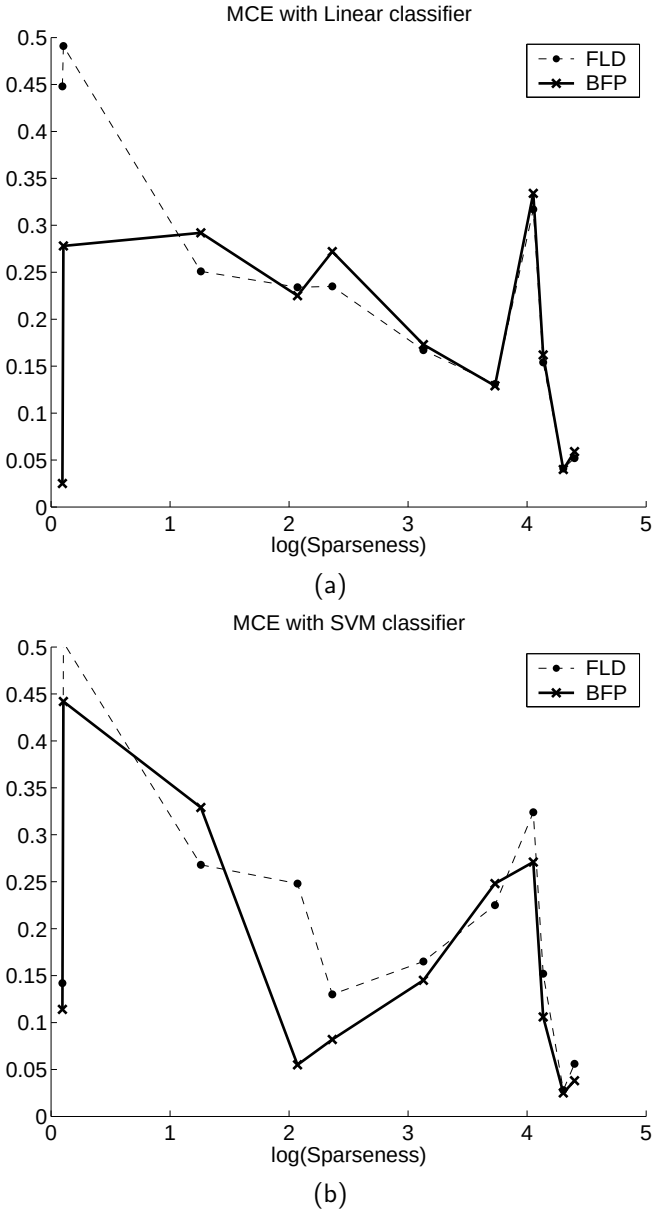
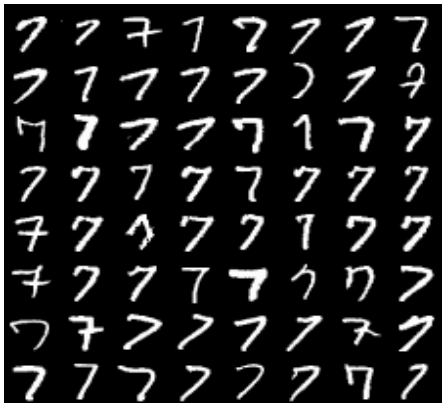


Figure 3.6: Classification error versus data sparseness for the best feature extraction methods: FLD and BFP (a) Using the linear classifier (b) Using the SVM classifier.

periments shown have been performed splitting the training set on 5 independent subsets and following a 5-fold cross-validation scheme. The plots shown are a mean of the 5 iterations.

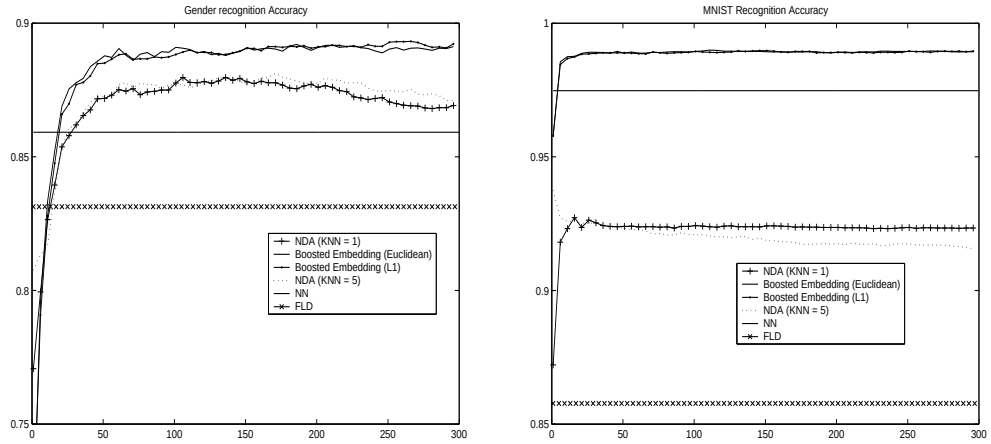
(a) Original faces and their normalized version



(b)Some digits of the MNIST database.

Figure 3.7: Examples of the male and female images used in the experiments, (a) taken from the AR face database, and the XM2VTS. The normalized version of the faces is also shown. (b) Also samples from the MNIST are plotted.

Figure 3.8 shows the accuracies obtained on both databases, using the NN classifier and two distance metrics (Euclidean and L1). The LBDP method is compared with classic Fisher Discriminant Analysis and the non parametric discriminant analysis (using 1 and 5 internal nearest neighbors to find the optimal projection). As it can be seen, LBDP technique outperforms the other techniques when more than 20 features are extracted, achieving a maximum accuracy of 89% in the face case and 98% on the digit classification case. In the plot, the results using the nearest neighbor on the original space are plotted as a horizontal line as a reference. Notice that in these cases feature extraction is justified given that we achieve better accuracies, working on low dimensional subspaces.



(a) Gender recognition accuracies. (b)MNIST databases recognition accuracies.

Figure 3.8: Accuracies obtained in the Gender Recognition and the MNIST tests as a function of the dimensionality reduction (number of features extracted using each method).

3.5 Conclusions and Future Directions

Three linear feature extraction methods based on Adaboost have been proposed. The main algorithm does not require making any assumption on the data distribution. At each boosting step we select from a pool of linear projections the one that minimizes the weighted error. Different algorithms can be derived from this idea depending on how the projections are selected within an Adaboost step.

Experiments were performed on 9 real and 2 artificial data sets. It seems that high dimensional data sets are the best target for the boosted techniques, compared to the three methods based on eigenvector decomposition. For this reason the feature extraction techniques proposed seem to be specially suitable for face classification tasks, where often we deal with high dimensional data.

The method developed in this thesis is now restricted to two class problems, and has been successfully applied to gender recognition. Other applications could be found, such as face verification, where usually the amount of data available from the same subject is too small to be modelled by classic parametric techniques.

Although there are specific methods to automatically convert two class classifiers in multi class classifiers, one important extension of the methods proposed could be its application to the multiclass case. There is a variety of possibilities for such extensions. These include (but are not limited to)

- Using a straightforward extension of Adaboost such as Adaboost.M1
- Using a variant of Adaboost based on error correcting codes (ECOC) preserving the most important projections obtained for each two-class classifier.

Along with the possibilities coming from Adaboost, there are multiple choices to be made arising from the specifics of the projection generation in 3.(a) in Table 3.1. For example, in LBDP, we need to select one point from the same class as $\mathbf{x}_i$, and one point from the "opposite" class. When there are more than one opposite classes, different paths can be followed:

- use the closest point from any other opposite class,
- take the mean of the nearest neighbors of the opposite classes, or
- just randomly select a class and take the closest vector from this class.

The range of options is large and it can be an interesting future research direction as there is no obvious guide to see what is best.

Different choices of individual $1-$dimensional projections at each step can lead to different unexplored variants of the method with different accuracy. An interesting open question is developing a guideline towards a more systematic choice of projection generation at step 3.(a) in Table 3.1. Along this thesis different options have been analysed. The influence of the diversity in the projections found has been studied for this purpose. A version of the algorithm proposed where the projection was selected maximizing the diversity between the current feature extraction and the classification results using the new added projection has been developed. The final projection matrix found did not improve the results obtained using "classic" Adaboost projection selection according to the classification error. Therefore, finding a measure of the quality of the projections added at each boosting step is another interesting future direction.

An important advantage of the techniques proposed is that the upper bound on the final dimensionality from classic FLD is solved. Moreover, the BFP method could extract more features than the present on the original data set, achieving dimensionality augmentation instead of reduction. A carefully study on the classification performance using dimensionality augmentation is another feature direction that arises from the methods suggested.

Chapter 4

External Face Feature Extraction

4.1 Introduction

In previous chapters feature extraction techniques applied to internal face feature extraction have been discussed. Nevertheless, the internal information is not the only source of information available from face images, although it is the most used on computational face classification systems.

Traditionally, classification techniques using the internal part of face images have been seen as a non intrusive method for face verification and identification, substituting the use of passwords (data that a subject knows) for some measure of the characteristics of the subject (data that is part from the subject). Face verification on biometrics is a subject of continuous development, and the recognition rates are increasing as new research is done. However, the performance of current verification systems is close to 90% [128], what seems to be still far from classic non biometric systems of identification. In addition, most of the tests are performed in controlled environments, where usually images are taken indoor, and the number of subjects to verify is limited. Therefore, recognition from outdoor imagery remains a challenge research, and applications to general secure identification systems in uncontrolled environments are still evolving.

Usually, as face classification methods found in the literature deal with security applications, they focus the attention of their algorithms in the internal features of facial images, such as mouth, eyes and nose. External features (hair, forehead, chin, ears) have often been ignored. This fact has been justified because external features present a lack of temporal stability and can be easily changed, and applications related to security using biometrics need to focus on features difficult to imitate. Nevertheless, as technology evolves, new electronic devices are developed, which are everyday computationally more powerful, and which can include small cameras. The capabilities of the new embedded systems can be augmented with facial classification technologies not oriented to security. Applications such as user profiling, intelligent environments,

Figure 4.1: Internal and external features of a face. In both cases the information is useful for recognizing the subject at first sight.

reactive publicity, where the user does not make efforts to mislead the classifier are emerging. In this context, non security applications can take benefit from external information for face classification. Figure 4.1 shows a simple example of the importance of external information on face classification, depending on the subject, the face recognition becomes easier looking at the external information than using the internal one.

Formally we could define the external information of face images as the region that covers up to the hair of the subject (including ears, forehead, and chin), omitting the eyes mouth and nose. Nevertheless, in this thesis only the lateral and upper regions of the face will be used, rejecting the information located in the chin. Figure 4.2 shows an example with the zones where the external information is extracted.

The use of internal information from faces has been deeply studied, and there exist a plethora of normalization algorithms to extract internal information. Usually the most used method is to align each face image according to the inter eye distance, obtaining a final D-dimensional vector where each value is related to a pixel from the internal region. In the use of the external information, two problems related to the feature extraction immediately arise:

- External information does not have the same size in different subjects, given that the hair volume can differ considerably between subjects. Pixel values at certain position do not mean the same depending on the sample.
- There is a lack of alignment on the features, given that there are no points of reference between samples from different subjects, or even between the same subject with different hairstyle.

Commonly the extraction of internal information is faced using bottom-up techniques. In the case of external features, this strategy is not suitable due to the problems mentioned above. We propose to follow a top-down procedure to extract external

Figure 4.2: Example of the three face zones where we extract relevant external face features. Also, the internal and original image are shown

information instead. We have followed a segmentation-based [16] algorithm to build a model of the external information completely off line. Then each face image is reconstructed according to the model and some predefined restrictions, yielding a final feature vector encoding the external information in a non topographic way, perfectly aligned between samples, and independent of the size of the external information.

In this chapter the global framework to extract external information from face images will be presented, solving the alignment and extraction problems. In the next section the use of external feature extraction is introduced reviewing some previous works on the literature. Then the segmentation based method is explained in detail. Finally some experiments using different face databases are shown, dealing with different problems of face classification (gender, verification,...). Also simple combination strategies between internal and external features are shown. Finally, the chapter is concluded with some future works that can improve the proposed method.

4.2 Previous Works on External Feature Extraction

Jarudi and Sinha [68] showed that the external features of face images (defined as hair and jaw-line) can act as an important cue to the identity judgments in the human visual system. In their experiments with thirty subjects ranging in age from 18 to 38 they obtained a 40% accuracy in face recognition using only external features. Also, they introduced the well known "Presidential illusion" images [154, 155] (shown in the introductory chapter) to illustrate that the contribution of external features is extremely important in face recognition, and this importance depends on different factors, such as the prior knowledge from the subject to identify. The internal features of both pair of faces are exactly the same, while human visual system easily identifies the two persons at first sight. We have also build a similar visual illusion using two

Renaissance portraits (see Figure 4.3), where at first sight two different women can be distinguished. Nevertheless, in both portraits the internal features are the same. Bruce et al. [25] studied this fact, concluding that external features can be even more important than internal ones for the recognition of unfamiliar faces. On the other hand, according to Ellis et al. [47] the role of internal features increases when the degree of familiarity with the person is increased. In addition, it has been shown [68] that the importance of internal and external features differs depending on the resolution of the image. External information is more effective than internal in low resolution images. Figure 4.4 shows an example of an image acquired by a security camera, where the low resolution of the internal features makes unviable the recognition process. Studies performed by Pascalis et al. [123] on 4-day born infants show that external features play an important role in face recognition. New born babies are able to recognize the face of their mother in normal conditions. Nevertheless, in their study they show that babies were unable to recognize their mother when she was wearing a scarf covering the hair. In addition, Campbell et al. [29] showed that the use of external features differs from adults and children. Adults tend to use eyes, nose and mouth to identify people, while children tend to use more hair, jaw and ears. They found that the shift from external to internal features is produced when the child is 10-11 years old, and is clearly noticeable in children above 15 years old.

In this chapter the contribution of the external features applied to face classification will be showed in three independent problems:

- A gender recognition problem.
- A face recognition problem.
- A face verification problem.

4.3 Extraction of External Information

The main difficulty when extracting external face information is the impossibility of applying classical feature extraction techniques given the lack of alignment of the input images. In the internal feature extraction, face images can be aligned according to the eye positions, and samples are scaled to have the same sizes. Each internal face image is represented as a D-dimensional feature vector, where each coordinate corresponds to the same facial characteristic. Therefore, direct bottom-up algorithms can be applied to the feature extraction process, as stated in the previous chapters.

Nevertheless, the diverse nature of the external information makes this approach unfeasible. Hair volume differs between subjects, so direct transformations such as PCA or FLD can not be directly applied. In this chapter a top-down algorithm based on constructing a global model for the external information of face images will be presented. The final features used for classification will be extracted according to the selected model.

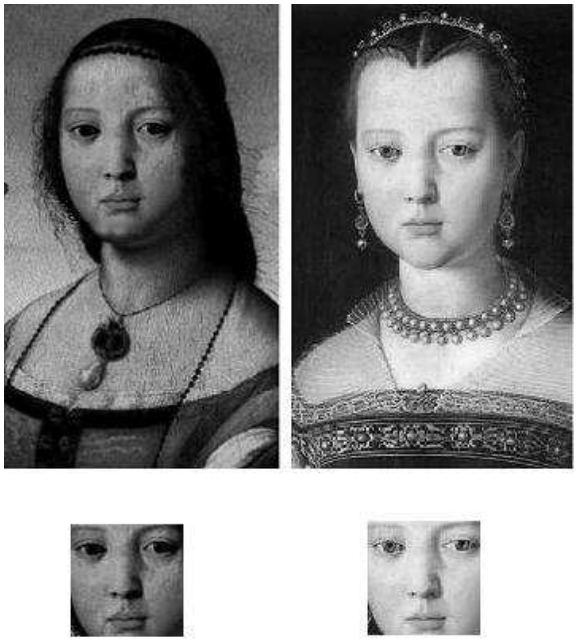


Figure 4.3: Example of two Renaissance portraits, where the internal features from the first princess have been substituted by the internal features of the second one. At first sight two different women are distinguished.

Figure 4.4: An example of an image where the internal features are acquired in low resolution. The image corresponds to a video sequence captured in Maine airport, and the subject is Mohammad Atta [68]. Internal features are not robust enough in low resolution images.

The developed method to obtain the model of the external features is based on a segmentation algorithm proposed by Borenstein et al. [16, 14]. The idea is to extract some object parts from a large set of object examples and use a selection of these parts as a model. We call this set of fragments the *Building Blocks* of our object. Then,

given a new unseen image, we find the subset of these parts that best reconstructs it and we use this representation to classify.

There are three different steps that should be distinguished in the original algorithm:

- Learning from the training examples the optimal set of fragments that constitute the model. This step is performed off line, and it is usually the most computationally intensive.
- Reconstruct a new unseen object image according to the fragments from the model that best fit with the image.
- Encode the object appearance using the selected fragments yielding a new aligned feature vector.

In the next sections the three steps of this approach will be developed in detail for the case of external face image information.

4.3.1 Learning the Building Blocks Model

The learning model algorithm receives as an input a training set C with representative face images (with visible and diverse enough external features). Using this training set, all possible sub images from each sample of predefined sizes are generated. For computational optimization, the sub images are taken from specific zones of the image, assuming that the face detector yields the coordinates of the internal features it is straightforward to find the surrounding areas where external information is located. A large enough margin is left between the coordinates of the eyes and the end of the external features region. Each sub image will be initially a candidate fragment F_i for the final model.

Given C and a large set of non face images $\overline{C}$ the goal is to find the fragments more representative of the external information of faces. In order to select which fragments will constitute the best model, the following selection criterion is applied: those fragments that can be found with high probability in face images but with low probability in non face images will be selected as building blocks. To determine wether a given fragment F_i is similar to a part p of an image I, we need to define a criterion of matching. As done in [7], we use the normalized cross-correlation as measure:

$$NCC(p, F_i) = \frac{\frac{1}{N}\sum_{\mathbf{x},\mathbf{y}}(p(x,y) - \overline{p})(F_i(x,y) - \overline{F_i})}{\sigma_p \sigma_{F_i}} \tag{4.1}$$

where N is the number of pixels in F_i, $\overline{p}$ and $\overline{F_i}$ are the means of p and F_i respectively, and σ_p and σ_{F_i} are their standard deviations.

Table 4.1: Building Blocks learning algorithm.

The algorithm takes as input:

- The face images set C,
- The set $\overline{C}$ of non face images
- The possible sizes of the fragments to analyze $Si \in \{S_1 \dots S_s\}$,
- The maximum number of fragments K that will be considered as building blocks, and
- The predefined threshold of false positives α.

1. For each fragment size S_i
   - Extract all the possible sub images F_i of size S_i from the set C using a sliding window procedure.
   - Add each sub image to the candidate fragments set.
   - Calculate and store the normalized correlation between each candidate fragment F_i and each image from C and $\overline{C}$.
2. Compute the threshold θ_i for each fragment F_i that allows at most an α false positive ratio from the training set, $p(NCC_i(\overline{C}) > \theta_i) \leq \alpha$.
3. Compute the probability (frequency) of each fragment to describe elements from class C using the threshold θ_i, $p(NCC_i(C) > \theta_i)$.
4. Select the K fragments with highest value $p(NCC_i(C) > \theta_i)$.

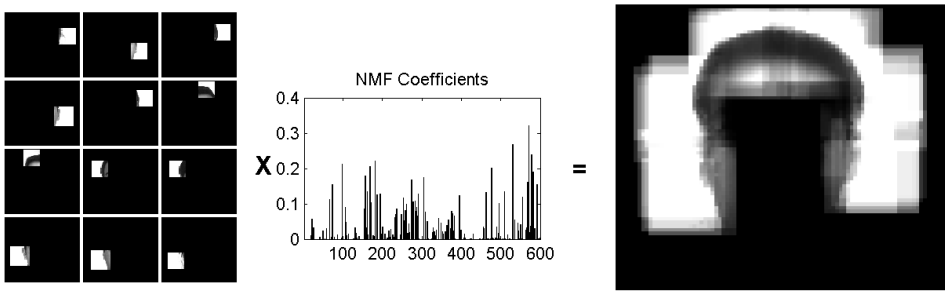
For each fragment F_i the maximum values of the normalized cross-correlation between F_i and each possible sub image p of $I \in C$, $NCC_i(C)$, and in $\overline{C}$, $NCC_i(\overline{C})$, are computed. Given the number of false positives α that can be tolerated for a fragment in $\overline{C}$ we can compute a threshold value θ_i in order to assure that $p(NCC_i(\overline{C}) > \theta_i) \leq \alpha$. This value can be used for determining if a given fragment is present in an unseen image.

Finally, the K fragments with highest $p(NCC_i(C) > \theta_i)$ are selected, therefore the fragments with highest probability to appear in the face set, and not to appear in the non face set [16]. The complete algorithm is detailed in Table 4.1.

Ullman et al. [151, 143, 174] proposed a variation of the fragment selection for specific classification purposes. They used the mutual information between the fragments and the classes to select the optimal set of fragments, realizing empirically that the most informative fragments are typically those which are of intermediate size. In [179] Vidal-Naquet and Ullman introduced the segmentation algorithm for object recognition, by performing linear classification on the informative features extracted using the mutual information criterion. Their comparative study shows that features extracted using the building blocks model can outperform the classic generic feature extraction methods (wavelet features were used in their comparison).

As we need the model to extract the best external features possible to posterior

(a)



(b)

Figure 4.5: Example of reconstruction of the external information using the linear combination defined by the fragments basis. First the original image is shown, (a) from the zones marked we find the Fragments from the model that best fit on the image, (b) a reconstruction using the NMF algorithm is performed achieving the coefficients that weight the importance of each fragment from the model. Finally the resulting image from the linear combination of the fragments is shown.

classification, a geometrical constraint about the location of the fragments to ensure enough diversity on the fragments has been heuristically imposed in the selection process. Otherwise, the selected fragments could be concentrated in a small region of the face, achieving a poor global reconstruction. More concretely, we have divided the fragments of the model in three categories: fragments belonging to the frontal part, from the left side, and from the right side.

The selection according the best probabilities is performed inside these categories, assuring that there will be the same amount of fragments in the model from each of these parts. Figure 4.5 shows an example of the three regions where fragments are located (a) to obtain the external fragments that best reconstruct the image (b).

4.3.2 Extraction of the External Features from Unseen Images

Once the fragment-based model of the external features for the face case is learned, and supposing that the face detector has located a face in an image (internal features), we can extract the external face features by covering the surrounding of the face area with the set of building blocks. To achieve this goal a function $NC(I, F_i)$ is defined as the pixel coordinates where the maximum $NCC(p, F_i)$ for all the possible sub images $p \in I$ is reached. Therefore, for each building block the place where the normalized cross-correlation value is maximum is computed, and then, the optimal covering is defined as an additive composition of the fragments that yields an optimal reconstruction of the surroundings of the detected internal face features.

To find the optimal cover, the original top-down segmentation method [16] uses an iterative algorithm based on optimizing a combined weighted criterion that considers:

- The quality and reliability of the fragments, which are measured as the combination of the factor matching s_i, defined as a weighted sum of the maximum normalized correlation and a measure of edge matching (to avoid background noise) [17], and the probability of finding a fragment on an object and not anywhere else:
$$\sum_i = s_i \frac{p(NCC_i(C) > \theta_i|C)}{p(NCC_i(C) > \theta_i|NC)} \tag{4.2}$$
- The consistency of the cover, defined as a measure of overlapping between the fragments. The final fragment selection must consistently cover the surface of the object, avoiding only local coverings. For this purpose the consistency term penalizes the overlapping between each new fragment and the current cover.

The iterative algorithm improves the criterion at each step adding new fragments to the cover, and there is a trade off between the quality of the fragment added and the amount of new surface covered. The algorithm typically converges after a small number of iterations.

In the method proposed in this thesis, there is an extra alignment requirement on the feature extraction, given that our purpose is not completely focused on the segmentation. The main goal is to perform a feature extraction from external features that can be used as a feature vector in traditional face classifiers, even if the whole region is not properly covered by the fragment set (the segmentation is not the final goal).

In our case the external feature extraction is performed following three steps:

- Given a new unseen image $\mathbf{x}$, the normalized correlation between the fragments composing the model and the area of the image that surrounds a face are computed. Also the position of the maximum correlation $NC(I, F_i)$ is stored for each fragment.
- Using the optimal position for each fragment, a set of basis vectors $\mathbf{B}$ are constructed as follows: for each fragment an image of the same size as the original

image is generated with the fragment set at the optimal position (obtained in the first step), and the rest of the pixels set to 0.

- Given $\mathbf{B}$, we find the coefficients $\mathbf{S}$ that best approximate the linear transform:

$$\mathbf{x} \simeq \mathbf{BS} \tag{4.3}$$

To calculate the set of coefficients $\mathbf{S}$ we use a variant of the Non Negative Matrix Factorization (NMF) algorithm [86]. The NMF algorithm has been designed to simultaneously compute $\mathbf{B}$ and $\mathbf{S}$, but in our case we fix $\mathbf{B}$, and the only unknown variable is $\mathbf{S}$. We have chosen the NMF algorithm because it fulfills the three constraints inherent to this problem:

- The combination of coefficients must be additive, given that each fragment contributes to the reconstruction of the external features.
- The reconstruction error of the image external features using the fragments of the model must be minimized, and it has been shown that NMF minimizes the reconstruction error [87].
- The fragment set is diverse, given that is extracted from different subjects in order to model the variability of the problem. Therefore, only a small part of the fragments from the general model can be useful to reconstruct a specific face. This fact implies that the coefficients $\mathbf{S}$ must be sparse, and only a small part of the fragments of the model should be activated for each face.

We have followed an implementation of the NMF algorithm similar as done in [64], fixing the bases matrix $\mathbf{B}$. This implementation has the advantage that the sparseness coefficient can be adjusted, in order to allow or restrict the amount of fragments that take part on the reconstruction. In our experiments the sparseness coefficient was fixed to 0.7. Figures 4.5 (b) and 4.6 illustrate the reconstruction of the external features from the image given the fragment basis. In Figure 4.6 (b), the reconstructed external features are shown with the original internal features inserted (in the last picture). Notice that the NMF algorithm recovers a good approximation of the external features from a set of 600 fragments where only a small part are active (see the sparse bar plot of the coefficients in part (a)). Also, it must be pointed out that the image $\mathbf{X}$ to reconstruct is not exactly the original face image (see the second picture in Figure 4.6(b)), the central part of the face has been set to 0. The main reason to this alteration of the original face image is to focus the NMF algorithm to optimize only the external part of the faces. As can be observed from the algorithm proposed, the basis $\mathbf{B}$ are fixed and placed all on the optimal external position of the faces, so it is hoped that there will be no samples in $\mathbf{B}$ with active pixels in the central part. Fixing the central part of $\mathbf{X}$ to 0, improves the iterative gradient scaling of the NMF algorithm, achieving faster convergence. Typically, with this modification the algorithm converges in less than 50 steps, given that only the coefficients $\mathbf{S}$ must be adjusted. Figure 4.7 shows the NMF reconstruction error as a function of the iterations, and as can be seen the error is stabilized after 50 iterations. The same

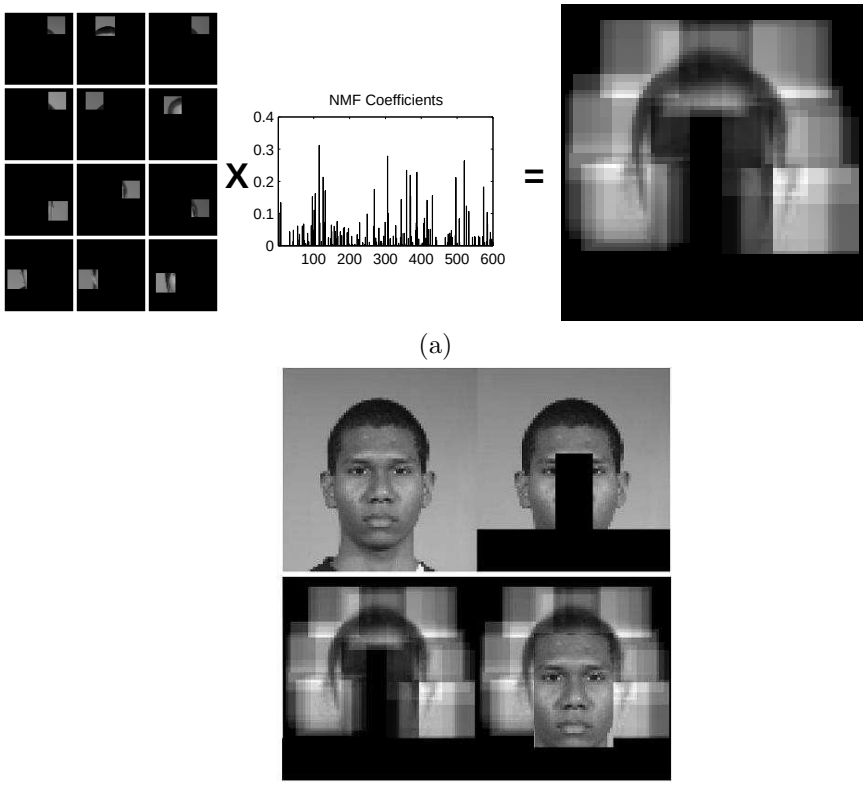


Figure 4.6: Example of reconstruction of the external information using the linear combination defined by the building blocks basis. (a) First the reconstruction process using the NMF algorithm is shown: the corresponding basis obtained from the Building Blocks set, a graphic with the weights of each fragment from the model and the resulting image from the linear combination of the building blocks. In (b) there is the original image, the external face feature zones of the image used for the NMF algorithm, the obtained reconstruction and finally an image where the original internal features of the face have been added to the reconstruction.

study should be performed depending on the database where the experiments are performed. Nevertheless, our empiric studies using two different facial databases show that 30-50 iterations are enough.

4.4 Face Classification Using External Features: Experiments

The main question that naturally arises after the external feature extraction methods has been proposed is: Are the external features useful for face classification prob-

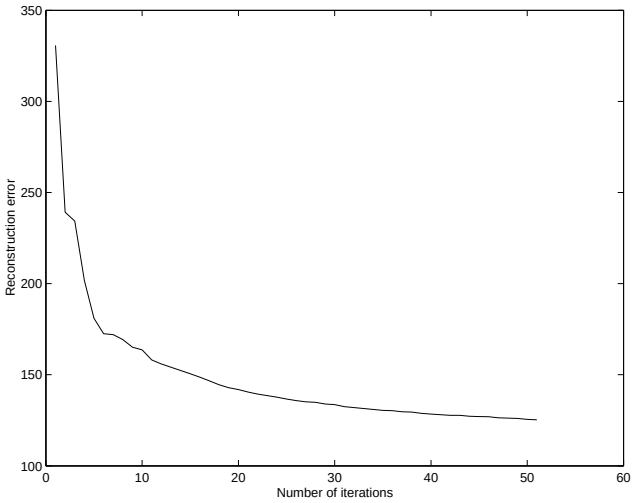


Figure 4.7: Plot of the reconstruction error of the NMF projection as a function of the number of iterations.

lems? As it has been shown in the introductory section, humans use the external information to classify, and in some circumstances the external features are even more important than the internal ones. Nevertheless the computational model proposed needs a validation using public face databases. In this section the external feature extraction method proposed is used on three face classification problems: Gender Recognition, Face Recognition and Face Verification. Two different face databases have been used for this purpose: the AR Face Database, and the IEEE Face Recognition Grand Challenge Workshop face database, FRGC, (see appendix D.3 for a more detailed description of the databases).

4.4.1 Gender Recognition

In this experiment the performance of the external feature extraction is analysed on a gender recognition experiment designed for the FRGC database. As there is no baseline comparison on the FRGC still image database, we have selected five classification algorithms in our study:

- Support Vector Machines Classifier (SVM). An implementation extracted from the OSU SVM Classifier Matlab Toolbox has been used [1], setting the parameters $\sigma = 1$ and cost function $C = 1$. The best result on gender recognition up to our knowledge was obtained using SVM. Moghaddam et al. [5] achieved a 96.6% recognition rate using a large face database (1755 faces of the FERET face database), and applying SVM with Radial Basis Functions.
- The 1-nearest neighbor classifier (NN).

[1] The toolbox can be downloaded from $http://www.ece.osu.edu/\sim maj/osu\_svm/$

- The linear classifier (Lin) assuming normal distribution of the classes and equal covariance matrices. A regularization parameter $r = 0.5$ has been added to avoid numeric problems when the inverse of a covariance matrix that is nearly singular must be computed.
- The quadratic classifier (Quad) with also a regularization parameter $r = 0.5$.
- The maximum entropy classifier (ME), based on the implementation of Nigam et al. [75]. The constraint functions have been designed as follows:

$$f_{i,c}(w, \widetilde{c}) = \begin{cases} w_i & \text{if } \widetilde{c} = c \\ 0 & \text{if } \widetilde{c} \neq c \end{cases}$$

  $\forall i = 1 : K$, where $c, \widetilde{c} \in C = \{male, female\}$ and w_i is the i-th coordinate of the encoded face w.

Experimental protocol

In the first step, a generic model to represent external features from face images is constructed by following these steps:

- A set of 80 face images from the FRGC database has been used (40 male and 40 female images) to extract the fragments. These images have not been considered anymore in the experiment to ensure that the reconstruction of an image never makes use of fragments extracted from itself (or from the same person).
- 100 natural images (with no faces) extracted from the web have been selected for the $\overline{C}$ set.
- Since the coordinates of the eyes were known, we have automatically extracted the set of fragments from each image to construct the set of candidate fragments.
- We run the selection algorithm explained in the previous section, using the following parameters $\alpha = 0.1$ and $K = 600$.

All the images of the experiment have been previously resized to a common size. The distance between the eyes in the final images was 16 pixels and the total size of the images was 81×77.

The accuracies of the method have been computed in all cases as the mean of 100 repetitions (using a cross-validation strategy) of the following experimental protocol:

- The general FRGC experimental data set proposed on this experiment has been split in a training set containing the 90% of the samples and a test set with the remaining 10% (2400 images and 240 images, respectively). Samples from the same person appear only in one data set, to avoid face recognition instead of gender and the presence of male and female samples on each set has been balanced.

Figure 4.8: Examples of non valid images, where external features cannot be reliably extracted.

Table 4.2: Results achieved in the Test set (percentage), using the Maximum Entropy classifier, Support Vector Machines, Nearest Neighbor, Linear and Quadratic classifier. The 95% confidence intervals for each method are also provided.

Algorithm	ME	SVM	NN	Lin.	Quad.
Accuracy	83.24	94.19	92.83	88.75	88.32
Interval	±0.43	±0.27	±0.26	±0.37	±0.38

- 50 iterations of the NMF algorithm are performed yielding the set of coefficients used for classification.

It should be pointed out that the FRGC experiment description includes a larger set of face images. Nevertheless, some of the original samples have not enough space between the eyes and the top of the images. This fact avoids an automatic processing of the feature extraction algorithm, and usually no external features can be extracted from the most important regions of these images. In addition, there are some subjects with important occlusions that can not be addressed by the algorithm. Figure 4.8 shows some examples of images that have been manually discarded.

Experimental results

The mean results after 100 rounds of classification of the 5 classifiers are shown in table 4.2. The 95% confidence interval is also shown for each classifier. The best classification accuracy is obtained using SVM, achieving a 94.19%. Nearest neighbor is the second best technique (92.83). The confidence intervals from linear and quadratic classifiers overlap, showing no real differences in performance. These results, when compared to the best (up to our knowledge) gender classification algorithms using internal face features demonstrate that external face features can be reliably used for classifying face images. Figure 4.9 shows some examples of misclassified faces.

Figure 4.9: Some examples of misclassified faces in the gender recognition problem.

4.4.2 Face Recognition

In the face recognition experiment the same subset of the FRGC database has been used, composed by a set of samples from 275 different subjects having uniform (grey) background. The external features have also been extracted using the scheme presented in this chapter, using the same subset of training samples (40 male and 40 female images that do not appear any more in the classification tests), and images were previously aligned according to the inter-eye distance (16 pixels). The parameters α and K have been also set to 0.1 and 600 respectively.

The face recognition experiment is organized as follows:

- For each subject, we have randomly selected 10 images, generating the training set.
- The testing set is build using the remaining images from each subject.
- The experiment is repeated 100 times, and at each round the random split is performed.
- The nearest neighbor has been used to classify the samples. Nevertheless, as the amount of classes is high, the second, third, fourth, and fifth nearest class are computed, considering in this last case that a vector is correctly classified if it has the correct label in one of the five nearest classes.

The mean accuracy of the 100 rounds is shown on Table 4.3. As it can be observed there is a great improvement considering the 2 nearest classes, while increasing more the tolerance does not modify significantly the accuracies.

Table 4.3: Recognition rates (and confidence intervals) obtained by the Nearest Neighbor classifier (NN) applied to the external information encoded on the NMF coefficients.

	1 NClass	2 NClass	3 NClass	4 NClass	5 NClass
Accuracy NN	43.3	66.32	68.31	69.3	71.9
Interval	±0.22	±0.31	±0.30	±0.26	±0.25

4.4.3 Face Verification

Finally, the last face classification experiment is a face verification problem. Contrary to the face recognition case, the number of classes in face verification is limited to 2. Therefore the problem is specially suitable for adding an extra discriminant layer to the method. In this experiment, we have also added a discriminative feature extraction step on the NMF coefficients encoding the external information. The Adaboost-based feature extraction technique presented in chapter 3 has been used for this purpose. Although the technique has been validated on a large database set in previous chapters, only an application to gender recognition has been shown. In special, the RBDP technique has been used. The choice of these technique over the BFP (more accurate in the experiments performed on the general databases) is justified given that the amount of samples to model each person is extremely reduced, and the scatter matrices for the intra class variability in BFP would be poorly estimated. The RBDP does not need any assumption and can be used in data sets of arbitrary size.

The experiment has been performed following the settings shown in the previous gender and face recognition problems. A total set of 600 NMF coefficients encoding the external information of each image (previously preprocessed as explained) are used as the input to the discriminant feature extractor. The experiment has been repeated 150 times, and each time a different person has been verified according to the following protocol:

- At each round a subject is randomly selected from the data set.
- Then 10 face images from the person are also randomly selected to be the training set.
- The non used faces from the same person and the remaining data set are used for testing.
- The nearest neighbors between the testing and the training vectors are computed. Also the 5 nearest classes are computed.

Table 4.4 shows the mean accuracies obtained from the 150 rounds. The results shown are the direct nearest neighbor on the 600 NMF features, and applying the Adaboost-based feature extraction to the NMF coefficients. The optimal dimensionality for the discriminant feature extraction is also shown. As can be seen, the feature extraction improves significantly the direct nearest neighbor on the original space.

Table 4.4: In the first line there are the verification results obtained by the Nearest Neighbor classifier (NN) applied to the NMF codification of the external features, considering from the first to the fifth nearest class. Second line shows the best accuracy obtained using the Nearest Neighbor classifier on the extracted features (Boosted FE). The last line specifies the corresponding optimal dimensionality.

	1 NClass	2 NClass	3 NClass	4 NClass	5 NClass
Rate NN	43.3	53.3	66.0	69.3	74.7
Rate Boosted FE	56.0	66.7	71.6	73.8	76.6
Dim	315	302	220	180	387

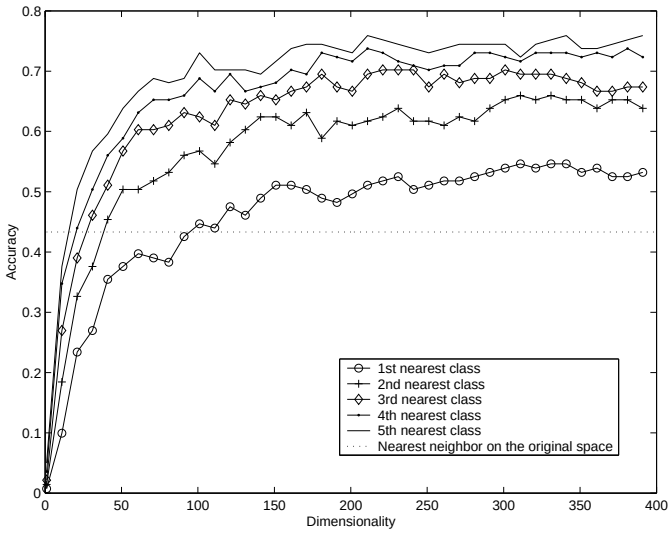


Figure 4.10: Mean accuracy as a function of the extracted features for the face verification case. The obtained result using directly the NN classifier is also indicated.

This fact is specially noticeable when the nearest or the second nearest class are used. As more error tolerance is allowed (considering correctly verified faces up to the 5 nearest classes) the advantages of the discriminant algorithm are reduced. Also it should be noticed that in addition to the increase in the accuracies, the discriminant feature extraction achieves an important dimensionality reduction, which is computationally important when dealing with the nearest neighbor classifier.

Figure 4.10 shows the accuracies as a function of the dimensionality reduction in the discriminant feature extraction step. The straight line shows the accuracy of the NN on the original space. As can be seen, in subspaces of dimensionality larger than 100 the accuracies obtained using the RBDP algorithm outperform the NN in the original space.

4.5 Combining the Internal and External Information

In the previous section, the main question was wether the computational external feature extraction scheme proposed in this thesis can be useful for face classification. The experiments performed on face classification show that external features encode enough information. Nevertheless, the accuracies obtained are still slightly lower than classification techniques using only internal information. This fact seems reasonable, given that external appearance is more variable than the internal one. However, the natural continuation of the proposed method could be its use in global face classification scheme together with the internal information, in such a way that now the question to answer is: Can the addition of the external features improve the accuracies obtained using only internal information?

Figure 4.11: Samples with male and female images taken from the AR Face database. Internal and external regions are marked.

In this section, a scheme for combining internal and external face information is presented. Moreover, some experiments to show wether there is an improvement adding the external information on a gender recognition problem will be shown. For this purpose, the AR Face database has been used. Images from the database have been aligned, and resized according to the inter-eye distance as performed in previous experiments. The central part of 36×33 pixels from each image has been used as internal feature set and the external features have been extracted following the NMF scheme presented in this chapter. Figure 4.11 shows some samples from the AR Face database, with the external and internal zones marked.

Once we have obtained the internal (pixel values) and external features (NMF coefficients) for each image, we need to combine this information to perform the classification. Perhaps the easiest combination rule is concatenating the features originated in the internal and external feature extraction. Chang et al [30] used this method to combine information from face and ear images on a biometric subject identification problem. Then the standard classification algorithms can be applied over the joint feature set. In the experimental section, the NN classifier has been used for this purpose.

Another possibility is to consider the joint feature set as a source information vector, treating each feature (internal and external) as an knowledge source. The maximum entropy (ME) principle can be used for this purpose. In the previous section, a classification algorithm using the ME has been used for classification, however the capabilities of the algorithm include also a natural model to combine information from different sources. Moreover, the algorithm is specially suitable for combining information sources of different nature, as it is the case of external and external features of face images. In the next section the ME algorithm used as a combination rule in the experiments will be explained in detail, and finally a comparison of the performance of the internal and the combination of internal and external features of face images will be shown,

4.5.1 Maximum Entropy

The maximum entropy principle used in this thesis is based on a work by Rosenfeld [135] where a single model is constructed capturing all the information of different knowledge sources in statistical language modelling problems. Each feature is treated as a source of information that originates a constraint, and the intersection of all the constraints can yield a probability function consistent with all the information sources. To find the probability distribution function, only one restriction is imposed, the function chosen must be the one with highest entropy, so given the knowledge functions, no other additional assumptions are performed on the data [70, 79]. Nigam et al. [75] exemplify the ME principle in a text classification problem as follows:

"...consider a four-way text classification task where we are told only that on average 40% *of documents with the word "professor" in them are in the faculty class. Intuitively, when given a document with "professor" in it, we would say it has a* 40% *chance of being a faculty document, and a* 20% *chance for each of the other three classes. If a document does not have "professor" we would guess the uniform class distribution,* 25% *each..."*

Then main problem is to estimate the probability distribution subject to the constrains given by the labelled data sources, which characterize the class expectations. To solve it we followed the approach of [75],where the probability function is defined by an exponential family. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ be the learning data set, and $\mathbf{x}_t = (d_1, ..., d_D)$ is an encoded face obtained using both internal and external features. Let be also $C = \{0, 1\}$ the two possible classes of gender. The goal is to learn the conditional distribution $P(c|\mathbf{x})$, $\forall c \in C$.

The constraint function is defined as follows:

$$f_{i,c}(\mathbf{x}, c) = \begin{cases} d_i & \text{if } c = c \\ 0 & \text{if } c \neq c \end{cases}$$

$\forall i = 1 : D$ and $c \in C$, where d_i is the i-th coordinate of the encoded face $\mathbf{x}$. Thus, it is stipulated that the learned conditional distribution $P(c|\mathbf{x})$ must have the property:

$$\frac{1}{N}\sum_{\mathbf{x},c} f_{i,c}(\mathbf{x}, c(\mathbf{x})) = \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{c} P(c|\mathbf{x}) f_{i,c}(\mathbf{x}, c) \tag{4.4}$$

and if equiprobability for the distribution $P(d)$ is assumed, then:

$$\frac{1}{N}\sum_{\mathbf{x},c} f_{i,c}(\mathbf{x}, c(\mathbf{x})) = \frac{1}{N}\sum_{\mathbf{x}} \sum_{c} P(c|\mathbf{x}) f_{i,c}(\mathbf{x}, c) \tag{4.5}$$

It is guaranteed that a unique distribution exists that has maximum entropy. Moreover, it can be shown that the distribution is always of the exponential form:

$$P(c|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\sum_{i,c} \lambda_{i,c} f_{i,c}(\mathbf{x}, c)) \tag{4.6}$$

where $\lambda_{i,c}$ are the parameters to be estimated and $Z(\mathbf{x})$ is the normalizing factor to ensure a probability distribution. So the problem is reduced to find the optimal parameters $\lambda_{i,c}$ given the training feature vectors. The improved iterative scaling algorithm (IIS) finds this solution performing a hillclimbing given the constraints [75].

Improved Iterative Scaling Algorithm

The IIS algorithm [162, 11] performs a hillclimbing in the parameter log likelihood space. Given the model defined by Λ (the $\lambda_{i,c}$), the log likelihood is defined as:

$$l(\Lambda|\mathbf{X}) = log \prod_{\mathbf{x}\in\mathbf{X}} P_{\Lambda}(c(\mathbf{x})|\mathbf{x}) = \sum_{\mathbf{x}\in\mathbf{X}} \sum_{i} \lambda_{i,c} f_{i,c}(\mathbf{x}, c(\mathbf{x})) - \sum_{\mathbf{x}\in\mathbf{X}} log \sum_{c} exp \sum_{i} \lambda_{i,c} f_{i,c}(\mathbf{x}, c) \tag{4.7}$$

The idea is to iteratively improve the parameters λ by setting $\Lambda = \Lambda + \Delta$ in such a way that the likelihood is improved $l(\Lambda + \Delta|\mathbf{X}) - l(\Lambda|\mathbf{X}) > 0$. A function B that bounds this expression is defined as:

$$B = 1 + \sum_{\mathbf{x}\in\mathbf{X}} (\sum_{i} \delta_{i,c} f_{i,c}(\mathbf{x}, c(\mathbf{x}))) - \sum_{c} P_{\Lambda}(c|\mathbf{x}) exp(f^{\sharp}(\mathbf{x}, c)\delta_{i,c} \sum_{i} \frac{f_{i,c}(\mathbf{x}, c)}{f^{\sharp}(\mathbf{x}, c)}) \tag{4.8}$$

where $\delta_{i,c}$ are the increments added to λ_i at each iteration, and $f^{\sharp} = \sum_i f_{i,c}(\mathbf{x}, c)$. And differentiating B to find maxima:

$$\frac{\partial B}{\partial \delta_{i,c}} = \sum_{\mathbf{x}\in\mathbf{X}} (f_{i,c}(\mathbf{x}, c(\mathbf{x})) - \sum_{c} P_{\Lambda}(c|\mathbf{x}) f_{i,c}(\mathbf{x}, c) exp(\delta_{i,c} f^{\sharp}(\mathbf{x}, c))) \tag{4.9}$$

Equaling the derivative to 0 and solving the equation, we can obtain the increments $\delta_{i,c}$ that should be added to the parameters Λ at each iteration. The complete algorithm is shown in table 4.5.

Table 4.5: Improved Iterative Scaling Algorithm.

The algorithm takes as input the training vectors $\mathbf{X}$, the correspondent classes c_i $(i = 1 : N)$, and the feature functions $f_{i,c}$.

- Initialize the parameters $\lambda_{i,c}$ to 0.
- Iterate:
  - Calculate the expected classes given the current set of parameters $\lambda_{i,c}$ (using equation 4.6).
  - For each parameter $\lambda_{i,c}$
    - Set $\frac{\partial B}{\partial \delta_{i,c}} = 0$
    - Solve it for $\delta_{i,c}$
    - Set $\lambda_{i,c} = \lambda_{i,c} + \delta_{i,c}$

The algorithm yields the adjusted set of parameters $\lambda_{i,c}$ that can be used to predict the label of a new unseen feature vector.

The algorithm iteratively finds the solutions, and although it converges fastly, typically in our experiments we have added a fixed maximum number of iterations (set to 50).

4.5.2 Experiments

In this section we evaluate the combination of internal and external features using the publicly available AR Face database [100] on a gender recognition problem. A set of 2210 images has been selected from the database, discarding subjects with missing images and balancing the presence of male an females. Data has been randomly split in a training and a testing set, in such a way that the 90% of the data has been used for training and the 10% for testing. The splitting has been performed taking into account the person identity, so all samples from the same person must be in only one set to avoid person recognition instead of gender recognition. As in previous experiments, the process has been repeated 100 times, and the results shown in Tables 4.6 and 4.7 are the mean accuracy of the 100 iterations. The 95% confidence intervals are also computed.

As a prior step to the tests, images from the database were aligned according to the eye position, and sub sampled to 134×139 pixels. The internal features were selected by taking the 36×33 central part of the sampled images, keeping the center pixel of each eye aligned. A previous mean-variance normalization was performed to reduce the effects of global illumination. To extract the external features we have randomly chosen 20 images from each class (using only frontal faces), and 40 natural images with no faces. With these images we have learned the model fragments as described in this chapter. Several parameters were fixed empirically cross validating the training set at this step. We fixed the threshold $\alpha = 0.1$, and we took fragments of

Table 4.6: Results achieved using only internal, external, and a combination of internal-external features.

	NN	ME
Internal Features	81.5 ± 0.3	80.4 ± 0.63
External Features	64.4 ± 0.4	82.8 ± 0.57
Combination	87.8 ± 0.4	84.8 ± 1.0

size 18×18 and 22×22. In a previous processing step, fragments with less than 10% of its surface corresponding to a face were discarded. Then, only 600 fragments were preserved after the learning process, according to their best probability. In addition, the number of fragments belonging to each contour part has been again balanced (head and both laterals). Two classifiers have been used in our tests, the nearest neighbor (NN) classifier using Euclidean distance, and the maximum entropy (ME). The experiments were performed on the data sets with only the internal features (1188 pixel values), using only the external features (600 NMF coefficients), and the join feature set (1788 internal and external features). In Table 4.6 we show the accuracies obtained with both classifiers.

We have chosen the SVM classifier as detailed in [5] to be the base classifier to compare our proposal, as it is considered one of the best techniques of the state of the art up to our knowledge (they achieve a 96.6% of correct classification using the internal information of samples from the FERET database). The SVM with RBF kernel on our full data (with occlusions and strong changes in illumination) yield a 85.5% classification rate while nearest neighbor classifier achieves 81.5%.

The NN rule with a combination of external and internal information improves considerably the results using only internal features, achieving the best results 87.5%. It can be seen that the accuracy of the NN applied to the external information alone is relatively poor, nevertheless combining both internal and external information the results are improved, so there is not a complete overlap on the modelling capacity of both techniques.

Using the ME approach the accuracies obtained are inferior to the SVM. Nevertheless, it must be pointed out that ME algorithm achieves interesting results using only external features, suggesting that they can be a good source for extra information on the combined scheme.

To see the importance of the internal and external information in detail, a second experiment has been performed (shown in Table 4.7). In this experiment the subsets from the AR Face database are tested separately. In the first row the image type is indicated (from A01-A13), and a sample from the same person is plotted, so we test both classifiers on the 13 variants of the face database (with gesture, occlusions and illumination). We show the accuracies of each case and the confidence intervals below. The best accuracy is marked with an '*', and the methods whose confidence intervals overlap with the best results are shown in boldface. As can be seen, in almost all the cases the best accuracies are obtained using combining internal and external features. Notice that the extra information of the external features is specially important on

the data sets with occlusions (sets A08-A13), where sunglasses and scarf degrade the internal information considerably.

Also, as the size of the subsets is smaller than in the first experiment, the performance of the NN classifier is slightly worse than ME, being ranked in 4 of the 13 sets as the best classifier. In addition, as can be observed in only 3 data sets the internal features are slightly better than using the combined feature set, this sets are characterized by strong lateral illumination. This suggests that the results could be improved using some kind of normalization in the building block model.

Table 4.7: Accuracies achieved using internal, external and combined features with respect to the AR face image type using the NN and ME classifiers. The confidence intervals are shown under each result.

ME	 AR01	 AR02	 AR03	 AR04	 AR05	 AR06	 AR07	 AR08	 AR09	 AR10	 AR11	 AR12	 AR13
Int	83.7	**84.4**	85.6	**82.8**	84.3	**92.3**∗	**91.5**∗	**87.5**	88.3	**89.8**∗	57.3	59.3	**72.5**
	±0.6	±0.6	±0.5	±0.6	±0.7	±0.5	±0.5	±0.4	±0.5	±0.4	±0.9	±1.1	±1.0
Ext	82.7	81.8	87.2	79.6	83.6	81.1	78.3	85.3	83.8	79.8	**69.1**	68.5	**71.1**
	±1.4	±1.2	±1.3	±1.2	±1.2	±1.1	±1.1	±1.6	±1.6	±1.6	±1.4	±1.8	±1.4
Comb	**88.5**∗	**86.7**∗	**91.0**∗	**85.4**	**86.8**	**87.3**	**85.3**	**89.9**∗	**90.9**∗	**88.7**	**72.1**∗	69.7	**72.0**
	±1.2	±2.2	±2.4	±2.8	±1.2	±2.2	±2.0	±2.1	±1.5	±1.5	±3.0	±1.9	±2.2
NN													
Int	82.8	81.6	80.4	**82.1**	**85.1**	89.6	89.0	86.1	87.6	**88.1**	**67.5**	69.9	**70.6**
	±1.9	±2.1	±1.9	±2.2	±1.9	±1.6	±1.8	±1.6	±1.6	±1.8	±2.7	±2.4	±2.4
Ext	64.5	66.2	65.5	62.9	66.4	65.6	64.1	68.9	65.8	68.0	57.1	58.1	63.1
	±2.4	±2.6	±2.5	±2.8	±2.6	±2.3	±2.3	±2.6	±2.4	±2.6	±2.3	±2.3	±2.3
Comb	85.3	**85.4**	83.7	**85.5**∗	**87.5**∗	**90.9**	**91.0**	**87.9**	**89.7**	**89.2**	**71.4**	**74.9**∗	**73.4**∗
	±1.9	±2.0	±2.4	±2.3	±1.9	±1.9	±1.8	±1.7	±1.6	±1.7	±2.3	±2.2	±2.4

4.6 Summary and Conclusions

This chapter is focused on developing feature extraction algorithms for face classification. In previous chapters a family of feature extraction techniques has been proposed, always dealing with internal information of face images. In this chapter we have shown that some face classification problems can take benefit of the external features, specially in non controlled environments where partial occlusions caused by objects or changes in pose often appear. This fact has been previously highlighted by several psychological studies, therefore, we propose a computational framework to use the external features of faces in face classification. We have adapted a top down segmentation based algorithm for extracting external features from face images and a natural way of obtaining the sparse set of coefficients that encode the importance of each fragment from the model. The resulting scheme solves the main problems of external feature extraction:

- It generates an aligned feature set, in such a way that direct classic classification algorithms can be applied to the registered features.
- It deals in a natural way with the diversity inherent to the external information of faces (specially in hair zones)

We have tested this technique on two standard benchmark face databases, in three different face classification problems: gender recognition, face recognition and face verification. In a first attempt, the goal was to prove that our computational model to extract external information encodes useful information for classifying faces. We showed that accuracies close to the state of art methods can be obtained using the NMF coefficients that constitute our extracted features. The second step has been to show that there is an improvement in the accuracies with respect to using only internal information. Two different approaches have been proposed to combine internal and external feature extraction: use a concatenation of both feature sets and apply classic classifiers, or use the Maximum Entropy principle to derive a classifier using the joint features as a set of constraints. An empirical study of both cases in a database with occlusions and strong changes in illumination shows that external information contribute significantly to the recognition.

Nevertheless, there is still some future work to be done. The experiments in the gender recognition problem show that the fragment model could benefit from using some kind of normalization on the fragments in the model generation. In particular, problems modelling local illumination of lateral external zones have been detected. Techniques based on ridges and valleys detection could be useful for data normalization prior to the fragment extraction. Also, the construction of the building blocs model could be improved. Although the cross correlation seems to be a good similarity measure, more sophisticate matching methods should be included to be more robust to non uniform cluttered backgrounds. The selection process could also be improved, by now the fragments with larger probability to appear in face images and not to appear on non face image are selected, some measure of diversity of the frag-

ment set could be added to model a larger rank of hairstyles. Mutual Information based techniques are being studied for this purpose.

Moreover other combination techniques could be applied to the internal and external features. For example Adaboost for feature selection [180] could be studied when the number of features become larger. The use of ensembles on the joint feature set seems to be a natural continuation of the combination of internal and external information.

The computational resources needed are also a drawback, although the most computational time consuming part is the generation of the model. Nevertheless, this step must be performed once, off line and can be easy parallelized. The great computational resources needed are due to the exhaustive search of fragments performed to generate the model. Several optimizations can be performed at this stage: (i) In our case, we propose to take benefit of face detection schemes, which can allow us to know in advance the approximate position of each part of the face, therefore, the correlations in the search of the optimal position for each fragment are reduced to smaller regions of the image (left, right an upper parts). (ii) Also the sliding windows step used to extract all the possible sub images could be increased, in order to take fragments each 2 or 3 pixels (having less overlapping between fragments).

The exploitation time of the algorithm once the fragment model is constructed, is not considerably high. The whole testing algorithm takes 2 seconds per image to be classified on a Pentium 4-2.4Ghz, using Matlab 6.0 program on Windows-NT being the most time consuming part the correlation of each fragment from the model with the image and the NMF projection. Both steps could be seriously optimized in C code, allowing the algorithm to be run in nearly realtime.

Chapter 5

Concluding Remarks

5.1 Conclusions

This thesis is focused on designing specific feature extraction techniques for face classification. More concretely, only linear feature extraction techniques are studied in depth along this work. Feature extraction has been studied in previous works as a way to solve the main problems when dealing with face classification:

- Face images are treated as high dimensional vectors. This fact implies the *curse of dimensionality* problem, that complicates the parameter estimation of the classifiers. Using feature extraction usually a dimensionality reduction is also achieved improving the classifier estimation.
- Usually face images have noise, which can mislead the classifier. Feature extraction can eliminate noise and redundancy on the data, reducing also the storage and computational needs.
- Faces acquired in natural environments can suffer from changes in the illumination. Feature extraction can learn invariant characteristics between samples of the same subject, obtaining more robust classifiers.

Most of the feature extraction techniques found in the literature are based on making some kind of assumption on the input data: Principal Component Analysis (PCA) performs gaussian assumptions on data distribution, NMF assumes positivity constraints and Fisher Linear Discriminant Analysis also performs class gaussian assumptions. In this thesis we have tried to relate the feature extraction process with the classifier used. The essential algorithm used is the Adaboost. Although it was originally designed to be a classifier, we have adapted it to perform the feature extraction process.

The Adaboost algorithm has been extensively used as a way to combine weak classifiers in a more powerful decision rule. It performs a set of rounds, where each time

a classifier is built. A set of weights is adjusted according to the classification results of the training samples, in such a way that the classifier learnt in the next round is more focused on the miss classified samples. As it has been defined in the original algorithm, the learning process of the classifier uses a fixed set of features corresponding to each sample. In this thesis we have introduced the feature extraction process inside the Adaboost algorithm, therefore at each step samples are represented by a different set of features, which are extracted focusing on the most difficult samples. The following contributions have been performed:

- We have designed an Adaboost algorithm where a feature extraction process is performed at each boosting step. The WNMF (Weighted Non Negative Matrix Factorization) algorithm has been used for this purpose. The weight vector encodes the most difficult samples where the feature extraction will be focused, and are shared with the Adaboost algorithm.
- We analyse our purpose in two different problems, a face detection and a manuscript digit recognition problem. The accuracies on both problems are better using our adaptive scheme, and are achieved in less boosting steps.

The main contribution of this thesis is a discriminant feature extraction method for two-class problems also based on the Adaboost algorithm. Once we have seen the usefulness of the Adaboost for classifying using the features extracted, we propose to reverse the problem, we use the Adaboost to extract the features. The method finds a set of simple 1-Dimensional projections at each step, selecting the one that achieves better classification rates on the training data. At each boosting step the weights are adjusted encoding the most difficult samples to classify, leading the extraction of the next 1-D projection. The final projection matrix is build incrementally concatenating the 1-D projections obtained at each step. In fact, we have built a family of feature extractors based on the Adaboost algorithm, depending on how the 1-D linear projections are constructed:

- **Random Boosted Discriminant Projections**: The most straightforward approach to build the 1-D dimensional projection vectors was to randomly choose pairs of points from each class, and use the vectors between each pair as a candidate projection. The best projection is selected at each step from a pool of candidates according to the classification results of the training samples.
- **Local Boosted Discriminant Projections**: In this case we build the projections in a deterministic way. Each point and its nearest neighbors from the same and opposite class define a plane where we can find a projection where the distance to the opposite class is maximized and the distance to the same class minimized. The obtained projections model the local structure of the training data, and are selected within the Adaboost to built the final projection matrix.
- **Boosted Fisher Projections**: In the last variant proposed we take more than two points to find the linear projection at each step, a training subset is resampled according to the Adaboost weights, and the projection is learned using the

classic Fisher Discriminant Analysis algorithm. As in the previous approaches the projections are also selected within the Adaboost algorithm. This feature extraction method proposed is similar to the original FLD, nevertheless the main drawback for two class problems of the classic technique is solved given that the final dimensionality is not upper bounded by the number of classes.

Once we introduced the Adaboost-based linear feature extraction techniques, we have performed a complete experimental study using standard machine learning databases. We can conclude from the experiments that the technique is specially suitable for high dimensional data sets, where outperforms the classic discriminant algorithms. More concretely, we have applied the technique to a gender recognition problem, achieving the best accuracies even in low dimensional subspaces. Nevertheless the methods proposed can be used in any pattern recognition problem where features from high dimensional samples must be extracted.

On the other hand we do not restrict ourselves to the classic approach finding features for face classification only from the internal part of face images. In the last chapter of this thesis we have proposed a new computational scheme to extract features from the external part of the face (head, hair, ears), and we show its utility in face classification problems.

Psychological studies show that external features from face images play an important role in the human visual system. Automatic Face Classification applications could take benefit of this fact. Nevertheless, dealing with external features poses a new problematic, completely different to the internal one. Some of the drawbacks to be solved when using external features from face images are:

- External features are very diverse between different subjects, and even between the same subject with different hairstyle. Moreover, there is a huge variation on the number of pixels (or initial features) covering the external features between subjects.
- There is a lack of alignment of the pixel values given that no points of reference can be used (such as eyes on the internal information). The same pixel position does not mean the same in different subjects.

An important contribution of this thesis is the introduction of a methodology for extracting external features from face images for classification purposes, solving the main problems cited above. The main contributions can be summarized as:

- Use a top down algorithm to construct a model of the external features of face images. We have proposed to adapt a segmentation algorithm, based on building a set of fragments which can reconstruct new unseen external parts of faces. Fragments are automatically extracted from training faces, and should model the space of the external information of faces. The more diverse are the training samples that are used to build the model, the better reconstruction of new images is achieved.

- Apply the model learned to face images, obtaining a representation suitable for classification. Each fragment can be seen as a base of the external features, in such a way that each face (only its external part) can be reconstructed as an additive combination of the base fragments. To reach it, we have used the NMF algorithm, obtaining the set of weights that encode the external information of faces images, as a weighted additive combination of the base fragments. The use of the NMF approach on the constructed fragment-bases has two important advantages:

  - The resulting features are aligned, given that each feature will be the weight of the fragment on the reconstruction of the external information. Moreover there will be the same number of features for each sample (the number of selected fragments defines the length of the feature vector).
  - The NMF algorithm yields an sparse representation of data, so only a few fragments are used in a reconstruction. This fact allows to reconstruct each external region using only a small set of fragments (those really useful), while the fragments that are not similar to the specific face remain inactive.

Also we have introduced the idea of combining both internal and external information for face classification, in order to achieve better accuracies with the joint strategy. Two basic methods have been proposed:

- Concatenate the feature vectors and apply classic classifiers to the joint feature set. A previous feature selection/extraction could be followed on the joint feature set, such as the Adaboost based feature extraction proposed on this work. This approach has been followed on a face verification problem, achieving interesting results.

- Use the Maximum Entropy principle, specially suitable for classifying data from different sources.

An experimental study of the use of external features has also been presented. First we have shown that external features can be useful for classification by themselves. Then we have evaluated the performance of the joint feature set (both internal and external features), improving the separate accuracies, specially when occlusions are present in the images.

5.2 Future Work

This thesis has opened issues in the field of feature extraction for face classification, that can be subject of further research. Among others, it seems interesting to explore the following main future research directions:

5.2.1 Linear Feature Extraction Using Adaboost

In chapter 3, we showed that the Adaboost-based feature extraction is useful on high dimensional subspaces. The family of methods proposed has been tested on different databases showing promising results, nevertheless, the general method can still be extended in several directions.

- The most important extension of the method is its use on multi class problems. The range of options is large and it can be an interesting future research direction as there is no obvious guide to see what is best. Here we suggest two options to perform this extension, although multiple sub paths can be followed in both cases:
  - Directly use the Adaboost.M1 extension for the multi class case. This option would imply the redefinition of the 1-dimensional projections. In the two class case we build a projection vector using the class sample and its nearest neighbor from the opposite class, when more classes are considered, several options can be followed as "opposite class": (i) Take the nearest point class as the "opposite" (ii) Take the class that has the nearest mean (iii) Randomly select the nearest class. All this possible choices may have advantages an drawbacks, so an empiric testing as the one made in chapter 3 should be performed in order to choose the proper multi class extension.
  - A more straightforward approach could be followed by merging the feature extraction and classification steps. There exist techniques to extend two class classifiers to multi class, for example applying a pairwise rule, or using Error Correcting Codes (ECOC). Another possibility is to apply the ECOC strategy and then keep a projection matrix build using the principal projection vectors found in each dichotomy.
- In addition, we have considered to include a diversity measure in the choice of the projections at each step. Even when there is a random component in the generation of the 1-D projections, it can be possible to obtain certain redundancy in the projection matrix. As has been shown [96, 80] diverse base classifiers favor the ensemble accuracies, although it is not clear which diversity measure is most suitable in our case. We suggest to average the classification results with some diversity measure considering the results of previous projections, and select the 1-D projection according to this weighted criterion at each step. An empiric evaluation of different diversity measures could be performed to see wether there is an improvement on the classification results.
- In this thesis three different approaches for finding the simple 1-D projections have been proposed, use random projections, local discriminant projections, and global Fisher projections. Nevertheless, finding the best way of generating the single projections at each step is not a solved problem. Actually, there are multiple choices for this task that could improve the results obtained in this thesis. In particular, we plan as a feature work introduce the margin criterion in the 1-D projection generation. In a similar way as done in the LBDP method,

we could find the projection that given a data point, maximizes the margin between this point and its opposite class.

5.2.2 Extraction of External Features from Face Images

The model proposed for external feature extraction has reached certain success modelling both laterals and upper head from face images, nevertheless there are still several open issues to be solved, moreover, the work performed on this thesis could be considered quite preliminary, in the sense that many improvements can be done.

- Psychological experiments performed on humans show that the chin region has great interest in face classification. Moreover, in the specific case of gender recognition, it seems logic to suppose that adding chin fragments to the model would improve considerable the accuracies, given the presence of beard in male images. An extension of the proposed framework to model chin regions would improve the capabilities of the scheme. This extension requires absolutely different choices on the similarity measures used, given that cross correlation lacks of the precision necessary to locate and align the fragments at their most suitable position. Moreover, it seems difficult to segment the internal chin region from the background (neck or clothing). A priory information from the face detector (such as eye position) could be used for this purpose, followed by a ridges and valleys detection (characteristics from this region).

- The building blocks construction directly takes fragments from face images. We believe that a more neutral blocks representation could be obtained if some kind of filtering were applied to original training images. For example, we could perform an anisotropic diffusion on the training set before extracting the fragment, in order to obtain a database less sensible to noise.

- The matching criterion used (normalized cross correlation) could be improved, in order to make it more robust against local changes in illumination. A weighted matching measure using both the correlation and some measure of edges (based on firsts derivatives) would benefit the fragment detection and location [16].

- The fragment selection criterion proposed takes into account only the likelihoods of appearance of the fragment on the face and non face sets. Nevertheless, posterior classification stages would benefit from a more diverse fragment set. For example, fragments of upper-lateral parts of the head are very common (specially in male images), and have large likelihood to be selected, yielding a redundant fragment set. Instead of selecting the most probable fragments, we could select the fragments that maximize some measure of mutual information within the model.

- On the other hand, in certain problems it could be interesting to introduce a discriminant criterion on the fragments selection. Instead of selecting the fragments that best model the face (set C) external features against non faces (set $\overline{C}$), we could use two different sets according to the problem. For example

in gender recognition, a set C with male and $\overline{C}$ with female images could be used to learn the fragments, therefore the final building blocks set would model the most common external characteristics presents in males and not in females. This approach could be only useful in two class problems where classes are known a priori, renouncing to model generic faces.

- Although the method proposed for external feature extraction seems to be successful in the experiments performed, the combining issue is still not completely solved. By now, the direct concatenation of features (internal and external) is used, as a previous step for the classifier (using NN or a more sophisticated Maximum Entropy algorithm). We believe that the combination rule could be improved by:
  - Applying and intermediate feature selection/extraction on the joint feature set, in order to add some discriminant criterion on the final feature set used by the classifier. Experiments performed on face verification suggest that a previous linear combination of the internal and external features increases the classification rates.
  - Considering the use of classifier ensembles to combine internal and external information, in such a way that the combined feature extraction and the classification steps could be merged in the same method. The most straightforward way to use ensembles to combine both feature sets is to apply the Adaboost algorithm considering each feature independently and train a simple classifier (decision stumps) at each boosting step on each one. The Adaboost could be used both for feature combining, selection ([180]) and classification.

Appendix A

Notation

In general, the following notation has been used: Bold face are used for matrices and vectors, subscripts indicate number within a vector. In the following table the most commonly used notation used in this thesis is shown.

D:	Dimensionality of the original space
N:	Number of vectors
$\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]$:	Data set $\mathbf{X}$ $D \times N$
x_i:	i-th value of the vector $\mathbf{x}$
K:	number of classes
c_j:	class of the j-th element $c_j \in \{1, \ldots, L\} \forall j = 1, \ldots N$
M:	dimensionality of the reduced space
V:	Diagonal matrix diag$(sqrt(\lambda_1), \ldots, sqrt(\lambda_D))$
S:	$M \times N$ extracted features matrix
A:	$M \times D$ linear projection matrix
B:	$D \times M$ basis matrix
z^{same}:	Nearest neighbor of the same class
$z^{different}$:	Nearest neighbor from another class
S_W:	Within class scatter matrix
S_B:	Between class scatter matrix
T:	Predefined number of the Adaboost steps
C_t :	Classifier at the step t of the Adaboost

Appendix B

Adaboost convergence proof

In this appendix the proof of the upper bound on the training error of the Adaboost algorithm enunciated in chapter 2 is reproduced [50]. The proof has been extracted from [96] changing only some notation aspects, and refers only to the two class case. The extension to the multi class case can be found in the provided reference.

Theorem Supposing a two class problem $\{c_1, c_2\}$, and denoting ε the ensemble training error and let ε_i $i = 1, \ldots, T$ be the weighted training errors of the classifiers in $\mathcal{C}$ then:

$$\varepsilon < 2^T \prod_{i=1}^{T} \sqrt{\varepsilon_i (1 - \varepsilon_i)} \tag{B.1}$$

Therefore, as $\varepsilon_i < 0.5$, the ensemble error decreases as the number of classifiers T is increased.

The following Lemma is needed in the proof:

Lemma Let $a \geq 0$ and $r \in [0, 1]$. Then

$$a^r \geq 1 - (1 - a)r \tag{B.2}$$

The proof of this Lemma can be found in [96].

Proof The proof of the theorem is divided in two parts. In part 1, the relationship between ε and β_i calculated within the Adaboost algorithm is shown, and in part 2, the upper bound is minimized by finding appropriate values of β_i. The theorem becomes proved when the optimal β_i is substituted in the bound in part 1.

Proof, Part 1. The weighted error of the first classifier is:

$$\varepsilon_1 = \sum_{j=1}^{N} w_j^1 l_j^1 \tag{B.3}$$

where l_j^1 is the classification result for an object $\mathbf{z}_j$ in the training set $\mathbf{Z}$ using the classifier C_1, $l_j^1 = 1$ if $\mathbf{z}_j$ is misclassified and 0 otherwise. The value of β_1 is calculated from ε_1. In this first part of the proof the goal is to derive β_i as a function of ε_i minimizing the ensemble error ε. The weights are adjusted for the second step

according to:

$$w_j^2 = \frac{w_j^1 \beta_1^{(1-l_j^1)}}{\sum_{k=1}^N w_k^1 \beta^{(1-l_k^1)}} \tag{B.4}$$

denoting the normalizing coefficient at step i by D_i

$$D_i = \sum_{k=1}^N w_k^i \beta_i^{(1-l_k^i)} \tag{B.5}$$

Then the second classifier is trained, and the new error ε_2 becomes:

$$\varepsilon_2 = \sum_{j=1}^N w_j^2 l_j^2 \tag{B.6}$$

β_2 is calculated and the new weights for the third step are

$$w_j^3 = \frac{w_j^2 \beta_2^{(1-l_j^2)}}{\sum_{k=1}^N w_k^2 \beta_2^{(1-l_k^2)}} = \frac{w_j^1 \beta_1^{(1-l_j^1)} \beta_2^{(1-l_j^2)}}{D_1 D_2} \tag{B.7}$$

We can generalize the formula for the weights depending on the boosting step as:

$$w_j^{t+1} = w_j^1 \prod_{i=1}^t \frac{\beta_i^{(1-l_j^i)}}{D_i} \tag{B.8}$$

Denote by $\mathbf{Z}^{(-)}$ the subset of elements of $\mathbf{Z}$ that are misclassified by the ensemble. The ensemble error, weighted by the initial data weights w_j^1 is

$$\varepsilon = \sum_{\mathbf{z}_j \in \mathbf{Z}^{(-)}} w_j^1 \tag{B.9}$$

Given that the w_i for the correctly classified samples is 0. If we assign equal initial weights of $1/N$ to the objects, ε is the proportion of misclassifications on $\mathbf{Z}$ made by the ensemble.

Since at each step, the sum of the weights in our algorithm equals one, we can write:

$$1 = \sum_{j=1}^N w_j^{T+1} \geq \sum_{\mathbf{z}_j \in \mathbf{Z}^{(-)}} w_j^{T+1} = \sum_{\mathbf{z}_j \in \mathbf{Z}^{(-)}} w_j^1 \prod_{i=1}^T \frac{\beta_i^{(1-l_j^i)}}{D_i} \tag{B.10}$$

For the ensemble to commit an error in the labelling of some $\mathbf{z}_j$, the sum of the weighted votes for the wrong class must be larger than the sum for the correct label. Recall that l_j^i is 1 if D_i misclassifies $\mathbf{z}_j$. Then

$$\sum_{i=1}^T l_j^i \ln(\frac{1}{\beta_i}) \geq \sum_{i=1}^T (1 - l_j^i) \ln(\frac{1}{\beta_i}) \tag{B.11}$$

Taking exponent on both sides.

$$\prod_{i=1}^{T} \beta_i^{-l_j^i} \geq \prod_{i=1}^{T} \beta_i^{-(1-l_j^i)} \tag{B.12}$$

Multiplying by $\prod_i \beta_i$ on both sides ($\prod_i \beta_i > 0$),

$$\prod_{i=1}^{T} \beta_i^{1-l_j^i} \geq \prod_{i=1}^{T} \beta_i \prod_{i=1}^{T} \beta_i^{-(1-l_j^i)} \tag{B.13}$$

since β_i is always positive

$$\prod_{i=1}^{T} \beta_i^{2(1-l_j^i)} \geq \prod_{i=1}^{T} \beta_i \tag{B.14}$$

Taking the square root on both sides

$$\prod_{i=1}^{T} \beta_i^{(1-l_j^i)} \geq \prod_{i=1}^{T} \beta_i^{1/2} \tag{B.15}$$

And from B.10,

$$1 \geq \sum_{\mathbf{z_j} \in \mathbf{z}^{(-)}} w_j^1 \prod_{i=1}^{T} \frac{\beta_i^{(1-l_j^i)}}{D_i} \geq (\sum_{\mathbf{z_j} \in \mathbf{z}^{(-)}} w_j^1) \prod_{i=1}^{T} \frac{\beta_i^{\frac{1}{2}}}{D_i} = \varepsilon \prod_{i=1}^{T} \frac{\beta_i^{\frac{1}{2}}}{D_i} \tag{B.16}$$

Solving for ε,

$$\varepsilon \leq \prod_{i=1}^{T} \frac{D_i}{\beta_i^{\frac{1}{2}}} \tag{B.17}$$

Using the Lemma B.2,

$$\begin{aligned} D_i &= \sum_{k=1}^{N} w_k^i \beta_i^{(1-l_k^i)} \leq \sum_{k=1}^{N} w_k^i (1 - (1-\beta_i)(1-l_k^i)) \\ &= \sum_{k=1}^{N} w_k^i (\beta_i + l_k^i - \beta_i l_k^i)) \\ &= \beta_i \sum_{k=1}^{N} w_k^i + \sum_{k=1}^{N} w_k^i l_k^i - \beta_i \sum_{k=1}^{N} w_k^i l_k^i \\ &= \beta_i + \varepsilon_i - \beta_i \varepsilon_i = 1 - (1-\beta_i)(1-\varepsilon_i) \end{aligned} \tag{B.18}$$

And combining B.17 and B.18,

$$\varepsilon \leq \prod_{i=1}^{T} \frac{1 - (1-\beta_i)(1-\varepsilon_i)}{\sqrt{\beta_i}} \tag{B.19}$$

Proof, Part 2. The goal is to find the values of β_i that minimize the bound of ε in Eq. B.19. Denoting the right side of Eq. B.19 by ε_{max}, and taking the first derivative with respect to β_i:

$$\frac{\partial \varepsilon_{max}}{\partial \beta_i} = \frac{\beta_i(1-\varepsilon_i) - \varepsilon_i}{2\beta_i\sqrt{\beta_i}} \times K \tag{B.20}$$

where K is a constant. Setting $\partial\varepsilon_{max}/\partial\beta_i = 0$, and solving for β_i we obtain:

$$\beta_i = \frac{\varepsilon_i}{1-\varepsilon_i} \tag{B.21}$$

The second derivative of ε_{max} at $\beta_i = \varepsilon_i/1-\varepsilon_i$ is

$$\frac{\partial^2 \varepsilon_{max}}{\partial^2 \beta_i} = (1-\varepsilon_i)\Big(\frac{\varepsilon_i}{1-\varepsilon_i}\Big)^{3/2} \times K \geq 0 \tag{B.22}$$

And it can be shown that the constant is positive at $\beta_i = \varepsilon_i/(1-\varepsilon_i)$, $i = 1, \ldots, T$. Therefore the solution for β_i is a maximum of ε_{max}. Finally the theorem is proven by substituting Eq.B.21 in B.19

$$\varepsilon < 2^T \prod_{i=1}^{T} \sqrt{\varepsilon_i(1-\varepsilon_i)} \tag{B.23}$$

This theorem proves that the Adaboost weighted training error decreases as new boosting steps are added. Nevertheless, it has been shown that Adaboost achieves very good generalization errors. Testing error usually keeps decreasing as new weak classifiers are added, even when the training error has been reduced to 0 [145].

Appendix C

Boosted Discriminant Projections: maximization criterion

In this appendix the obtention of the solution of find the optimal projection $\mathbf{p}_i$ on each step of the Local Boosted Discriminant Projections (LBDP) algorithm from chapter 3 is explained [1]. Given the data points $\mathbf{v}$ and $\mathbf{w}$ laying on the plane γ, we look for the direction $\mathbf{p}_i$ that maximizes:

$$\max \left\{ \left(\mathbf{p}_i^T \mathbf{w}\right)^2 - \left(\mathbf{p}_i^T \mathbf{v}\right)^2 \right\}. \tag{C.1}$$

The function has four possible solutions, that have the form:

$$\begin{aligned} p_1 &= \pm\sqrt{1-(p_2)^2}, \\ p_2 &= \sqrt{\frac{1}{2}\left(1 \pm \frac{w_2^2 - v_2^2 - w_1^2 + v_1^2}{\sqrt{4\left(w_1 w_2 - v_1 v_2\right)^2 + \left(w_2^2 - v_2^2 - w_1^2 + v_1^2\right)^2}}\right)}. \end{aligned} \tag{C.2}$$

The solution can be found taking the $\mathbf{p} = [p_1, p_2]$ that maximizes C.1

C.1 Proof

The criterion function to maximize is:

$$\mathcal{J} = \max \left\{ \left(\mathbf{p}_i^T \mathbf{w}\right)^2 - \left(\mathbf{p}_i^T \mathbf{v}\right)^2 \right\}. \tag{C.3}$$

subject to the restriction:

$$\|\mathbf{p}\| = 1 \tag{C.4}$$

[1] Author would like to thank Dra. Ludmila I. Kuncheva her help in building this proof.

developing the criterion:

$$\begin{aligned}
\mathcal{J} &= \max\left\{\left(\mathbf{p}_i^T\mathbf{w}\right)^2 - \left(\mathbf{p}_i^T\mathbf{v}\right)^2\right\} \\
&= (p_1w_1 + p_2w_2)^2 - (p_1v_1 + p_2v_2)^2 \\
&= p_1^2w_1^2 + 2p_1p_2w_1w_2 + p_2^2w_2^2 - p_1^2v_1^2 - 2p_1p_2v_1v_2 - p_2^2v_2^2 \\
&= p_1^2(w_1^2 - v_2^2) + 2p_1p_2(w_1w_2 - v_1v_2) + p_2^2(w_2^2 - v_2^2)
\end{aligned}$$

considering $p_1^2 + p_2^2 = 1 \Rightarrow p_1 = \sqrt{1 - p_2^2}$, then

$$\begin{aligned}
\mathcal{J} &= (1 - p_2^2)(w_1^2 - v_1^2) + 2p_2\sqrt{1 - p_2^2}(w_1w_2 - v_1v_2) + p_2^2(w_2^2 - v_2^2) \\
&= (w_1^2 - v_1^2) + p_2^2(w_2^2 + v_1^2 - w_1^2 - v_2^2) + 2p_2\sqrt{1 - p_2^2}(w_1w_2 - v_1v_2)
\end{aligned}$$

Differentiating the criterion $\mathcal{J}$:

$$\begin{aligned}
\frac{\partial\mathcal{J}}{\partial p_2} &= 2p_2(w_2^2 + v_1^2 - v_2^2 - w_1^2) + 2\sqrt{1 - p_2^2}(w_1w_2 - v_1v_2) + 2p_2^2(w_1w_2 - v_1v_2)\frac{-2p_2}{2\sqrt{1 - p_2^2}} \\
&= \left[2p_2\sqrt{1 - p_2^2}(w_2^2 + v_1^2 - v_2^2 - w_1^2) + 2(w_1w_2 - v_1v_2) - 4p_2^2(w_1w_2 - v_1v_2)\right]/\sqrt{1 - p_2^2}
\end{aligned}$$

And making $\frac{\partial\mathcal{J}}{\partial p_2} = 0$ we obtain:

$$\begin{aligned}
p_2\sqrt{1 - p_2^2}(w_2^2 + v_1^2 - v_2^2 - w_1^2) &= 2p_2^2(w_1w_2 - v_1v_2) - (w_1w_2 - v_1v_2) \\
&= (2p_2^2 - 1)(w_1w_2 - v_1v_2)
\end{aligned}$$

Taking the square of each side:

$$p_2^2(1 - p_2^2)(w_2^2 + v_1^2 - v_2^2 - w_1^2)^2 = (2p_2^2 - 1)^2(w_1w_2 - v_1v_2)^2$$

And renaming $(w_2^2 + v_1^2 - v_2^2 - w_1^2)^2 = A$, $(w_1w_2 - v_1v_2)^2 = B$ and $t = p_2^2$:

$$\begin{aligned}
t(1 - t)A &= (2t - 1)^2B \\
At - At^2 - 4Bt^2 + 4Bt - B &= 0 \\
(4B + A)t^2 - (4B + A)t + B &= 0
\end{aligned}$$

And solving the equation we obtain:

$$t_{1,2} = \frac{(4B+A) \pm \sqrt{(4B+A)^2 - 4B(4B+A)}}{2(4B+A)} \tag{C.5}$$

$$t_{1,2} = \frac{(4B+A) \pm \sqrt{(4B+A)A}}{2(4B+A)} \tag{C.6}$$

$$t_{1,2} = \frac{1}{2}\left(1 \pm \sqrt{\frac{A}{(4B+A)}}\right) \tag{C.7}$$

$$\tag{C.8}$$

And recovering $p_2 = \pm\sqrt{t}$, A, and B we obtain:

$$p_2 = \pm\sqrt{\frac{1}{2}\left(1 \pm \sqrt{\frac{A}{(4B+A)}}\right)} \tag{C.9}$$

$$p_2 = \sqrt{\frac{1}{2}\left(1 \pm \frac{w_2^2 + v_1^2 - v_2^2 - w_1^2}{\sqrt{(4(w_1w_2 - v_1v_2)^2 + (w_2^2 + v_1^2 - v_2^2 - w_1^2)^2)}}\right)} \tag{C.10}$$

$$\tag{C.11}$$

Appendix D

Databases

D.1 Face Databases

In this thesis different databases have been used to test the proposed methods, and compare them with other published works. The main part of the experiments has been performed on face databases. Nevertheless, the validation of the Adaboost-based feature extraction method required the use of general machine learning databases.

The main face databases used throughout the thesis are:

- **AR Face database**. Database composed of 26 samples from 126 distinct subjects (70 men and 56 women) [100]. Images were acquired in two different sessions in a 2-week interval, and 13 different acquisition conditions were imposed on each session, changing the illumination and the occlusions suffered by the subject.

- **FERET database**. The FERET database has been acquired under a program sponsored by the United States Department of Defense through the Defense Advanced Products Agency (DARPA). It contains a total of 14051 facial images collected from 994 subjects at various angles, over the course of 15 sessions between 1993 and 1996.

- **FRGC**. A subset of this database has been used in some of the experiments in chapter 4 dealing with external feature extraction. The database has been acquired to be the base experiment for: The IEEE Workshop on the Face Recognition Grand Challenge Experiments. The database consists of still high quality 36842 face images, and 4000 3D images.

- **XM2VTS**. Acquired in the University of Surrey, UK. It contains four recordings of 295 subjects taken over a period of four months [105].

- **CMU dataset**. Collected by the Face Detection Project in the Carnegie Mellon, is one of the standard benchmark databases for evaluating face detection algorithms [136].

- Faces from other databases have been collected to learn the generic PCA projection in the face classification application from chapter 1. These faces have been extracted from:
  - The ORL face database, acquired at the AT&T Laboratories, and consisting of ten images from 40 distinct subjects. Images were taken at different times, varying the lighting, facial expressions (open and closed eyes, smiling and serious), and facial details (glasses and no glasses). Images were taken against a dark homogeneous background with the subjects in an upright, frontal position.
  - CMU Pose Illumination Expression (PIE) database, acquired in the Carnegie Mellon University. It contains a total of 41,368 images taken from 68 individuals [153]. The subjects were imaged in the CMU 3D Room using a set of 13 synchronized high-quality color cameras and 21 flashes. The resulting images are 640x480 in size, with 24-bit color resolution. Each subject was recorded under the following conditions: expression (neutral face, smile, eyes closed), illumination (no flashes, one flash firing, room lights on/off), and talking.
  - The University of Oulu Physics-Based Face Database, consisting of 16 frontal views from 125 faces, acquired in different camera calibration and illumination conditions (Horizon, Incandescent, Fluorescent and Daylight illuminant).
  - The Yale Face Database, containing 165 grayscale images from 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink.

In addition, along this work we have generated our own face databases using a video grabber connected to a TV signal to train our face detector.

D.1.1 AR Face Database

This face database was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the Universitat Autonoma de Barcelona. It contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). The database is organized as follows:

- Faces were acquired in controlled conditions in two different sessions separated by 2 weeks.
- For each subject and each session 13 different images were acquired in different conditions: one frontal image, 3 images with facial expressions (smiling, anger and screaming), 3 images with different extra light conditions (left light on, right light on, and both lights on), 3 images with the subject wearing sunglasses (simple sunglasses, sunglasses with extra left light illumination, and sunglasses with extra right illumination), and 3 images wearing scarf (simple frontal scarf, wearing scarf and extra left illumination, and scarf and extra right illumination).

- No restrictions were imposed on wear, clothing, make-up, hair style and use of glasses.

(a) Frontal image

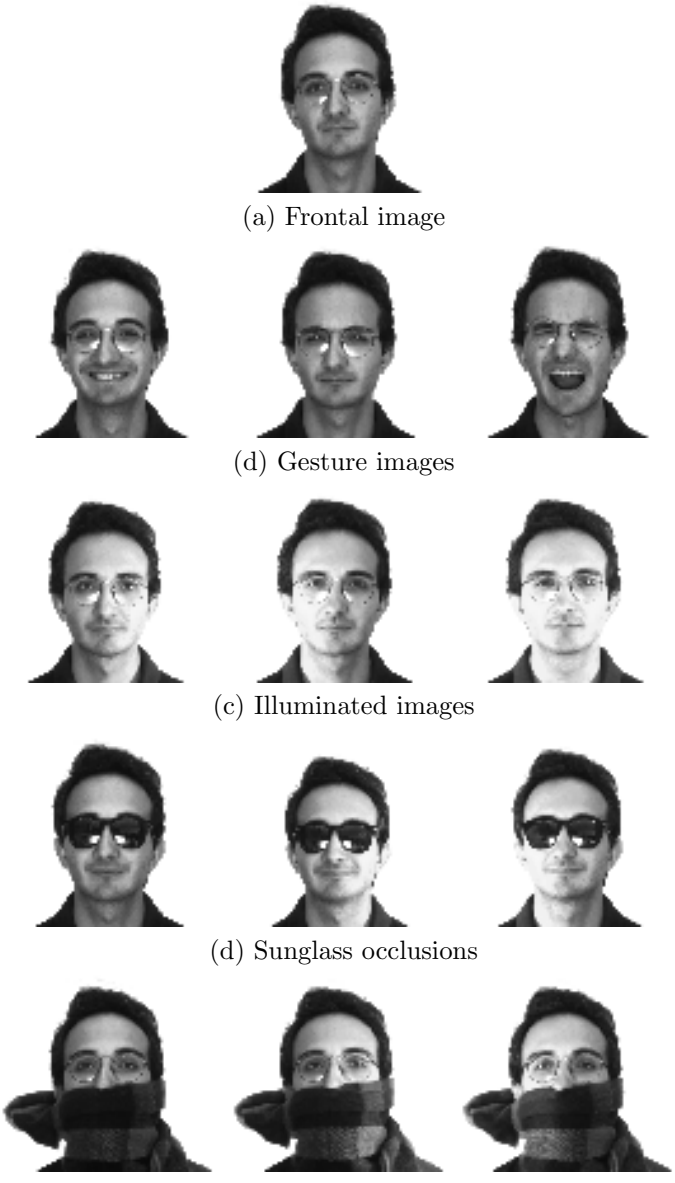
(d) Gesture images

(c) Illuminated images

(d) Sunglass occlusions

(e) Scarf occlusions

Figure D.1: Example of the 13 images taken from the same subject during the first session extracted from the AR Face Database.



(a) Frontal image

(d) Gesture images

(c) Illuminated images

(d) Sunglass occlusions

(e) Scarf occlusions

Figure D.2: Example of the 13 images taken from the same subject during the second session extracted from the AR Face Database. As it can be observed hair style changes from one session to other.

Figures D.1-D.2 show an example of a subject from the AR Face database in both sessions. The AR Face is one of the best databases to benchmark algorithms claimed to be robust to occlusions and changes in illumination, and also, given that there

is a delay of 14 days between each session, there is certain diversity in the hairstyle between samples of the same subject.

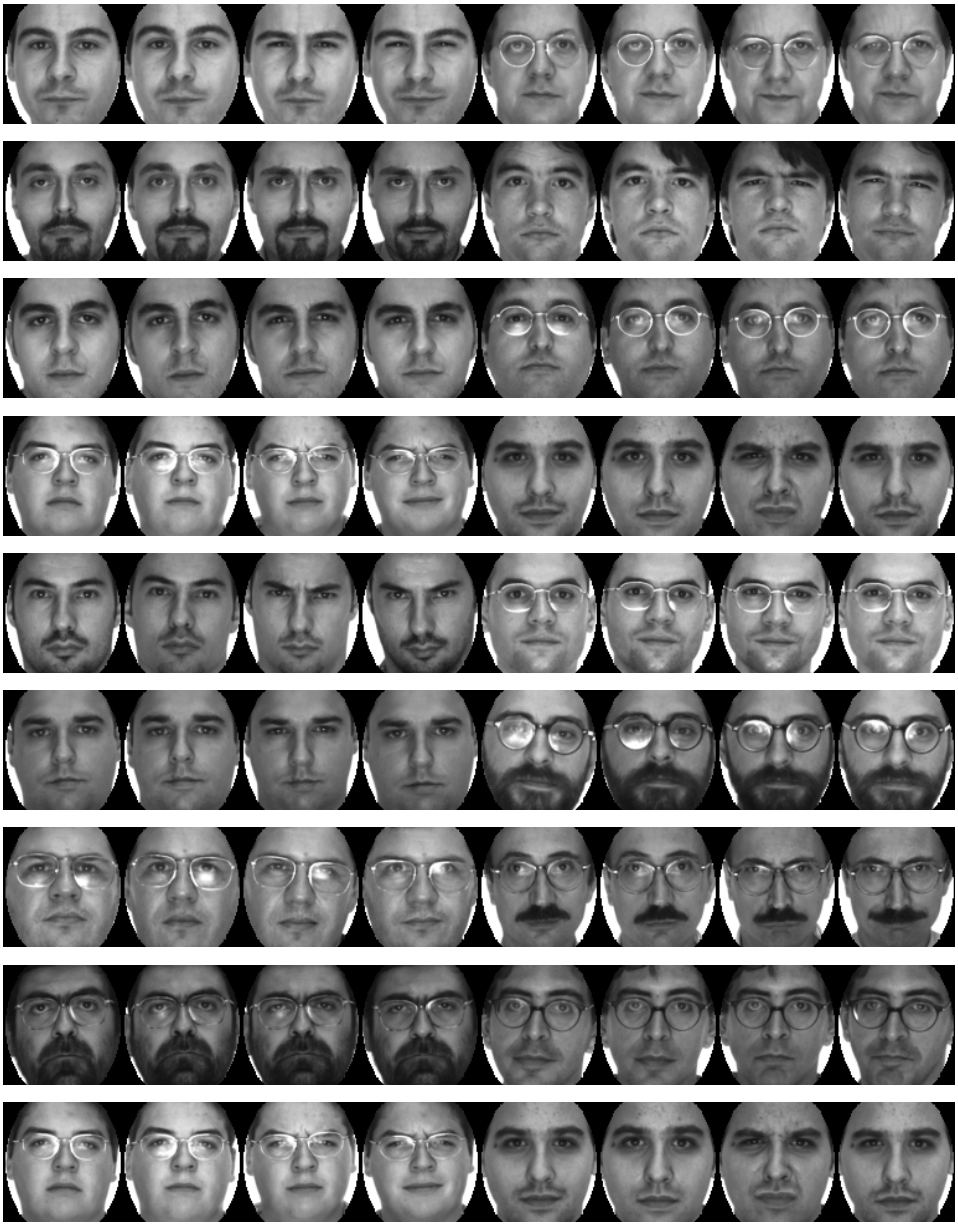


Figure D.3: Examples of male images from the AR Face database inscribed in an ellipse used in internal feature face classification.

Figure D.4: Examples of female images from the AR Face database inscribed in an ellipse used in internal feature face classification.

In the experiments performed using the internal features, usually a black ellipse has been added to the faces in order to mitigate the background biasing effects. Figures D.3-D.4 show examples of male and female images used for this purpose in this thesis.

D.1.2 FERET Database

The FERET database has been acquired under a program sponsored by the United States Department of Defense in the Department of Defense's Counterdrug Technology Development Program through the Defense Advanced Products Agency (DARPA) [127, 118, 134, 126], from 1993 to 1997. It was designed to develop automatic face recognition capabilities that could be employed to assist security, intelligence and law enforcement personnel in the performance of their duties. The acquisition was directed by Dr. Harry Wechsler at George Mason University.

It contains a total of 14051 facial images collected from 994 subjects at various angles, in 15 sessions, an stored as 8-bit grey images. In addition to the database, a protocol for evaluating face classification methods in frontal an different pose face sets is provided. Some of the samples from the FERET database were used on the Face Recognition Vendor Test 2000. We would like to thank to the NIST technical agent the chance to receive a copy of the FERET database for future research in our group [1]. Figure D.5 shows some samples of the FERET database.

D.1.3 Face Recognition Grand Challenge Database

The FRGC database is a large corpus of data and a set of challenge problems to be used as a benchmark for the special IEEE Workshop on Face Recognition Grand Challenge Experiments, that takes place in conjunction with the IEEE Conference in Computer Vision and Pattern Recognition 2005. The database consists of three-dimensional (3D) images, and high resolution controlled and uncontrolled still images. The data is divided into training and validation partitions, with the standard still-image training partition consisting of 12,800 images, and the validation partition consisting of 16,028 controlled still images, 8,014 uncontrolled stills, and 4,007 3D scans. The FRGC corpus supports a broad range of research in face recognition that includes developing high resolution still and 3D algorithms, comparison of human and machine performance, and also allows a XML based framework to design and reproduce new experiments on face recognition.

Figures D.6-D.7 show some samples extracted from the FRGC database. Notice that images are acquired with different global background illumination. In addition, there is a huge ethnicity variation between the database subjects. Images have been rotated in order to center and align the center pixels of the eyes.

D.1.4 XM2VTS

The XM2VTSDB set is a multi-modal face database consisting of still images, audio and video recordings. The XM2VTSDB contains four images of 295 volunteers taken over a period of four months. Each recording contains a speaking head shot and a rotating head shot. The database has been captured at the Centre for Vision, Speech and Signal Processing, University of Surrey. Figures D.8-D.9 show some examples of male and female subjects from the XM2VTS database.

[1] For all documents and papers that report on research that uses the Color FERET database, the use of the Color FERET database will be acknowledged as follows: "Portions of the research in this paper use the Color FERET database of facial images collected under the FERET program"

During each session frontal faces were acquired, and also a sequence showing the subject rotating the head from left to right, up to down, and returning to the center has been captured. Also a high-precision 3D model of the subjects head was built using an active stereo system provided by the Turing Institute.

The XM2VTS database defines also a standard protocol [105], called Lausane protocol, to report experiments dealing with face authentication. The database is divided in training, testing and evaluation sets. The exact composition of these sets is publicly known.

D.1.5 CMU dataset

This data set is an standard database to benchmark face detection algorithms [139]. Images have been collected at the Carnegie Mellon University (by Henry A. Rowley, Shumeet Baluja, and Takeo Kanade) [138, 137, 148, 149] and MIT (by Kah-Kay Sung and Tomaso Poggio). There are 130 images with 507 frontal faces (the location of the faces is provided with the database). Some examples of images from the CMU database are shown in Figure D.10.

D.2 MNIST

The MNIST database [84, 85] consists of 70000 images of handwritten digits, divided in a training set of 60000 examples and a testing set of 10000 examples. The digit images are part of a larger set available from NIST. The samples are black and white images size-normalized to 28×28 pixels. The database is suitable for researchers who want to test learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. Figures D.11-D.12 show some examples of the database.

D.3 UCI Repository

The UCI machine learning repository [13] is a set of databases compiled by Department of Information and Computer Sciences of the University of California, Irvine. Most of the databases of the repository have been donated by machine learning researchers for empirical analysis of new propsed algorithms. In this thesis the following databases have been used:

- *BUPA Liver disorder*: The database has been donated by Richard S. Forsyth from BUPA Medical Research Ltd. It consist of 345 instances with 6 numeric attributes and a class label (1 or 2).
- *Wisconsin diagnostic breast cancer*: This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [99, 185, 115, 98]. It has 569 samples with 30 features and a class label.
- *Wisconsin breast cancer* obtained from the same university. It consist of 666 samples with 9 features.

- *Cleveland heart disease*: This database contains data concerning heart-disease diagnosis donated from four locations by: Andras Janosi, from Hungarian Institute of Cardiology, Budapest, William Steinbrunn from University Hospital, Zurich, Switzerland, Matthias Pfisterer, University Hospital, Basel, Switzerland, and Robert Detrano V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. The original database consists of 297 samples with 76 attributes, although only 13 of them are used for ML experiments (also the class label is provided).

- *German database*: It consists of 1000 samples with 24 features, with credit data provided by Professor Dr.Hans Hofmann from the Hamburg University. Symbolic attributes have been converted to numeric ones previously by the Strathclyde University.

- *Ionosphere database*: This dataset was donated by Vince Sigillito [152] from the Applied Physics Laboratory, Johns Hopkins University. It consists of 351 samples with 34 attributes and the class label.

- *Sonar signals database*: This database was used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network [56]. It has 208 samples with 60 attributes and the class label.

- *SPECTF heart*: This database has been donated by Krzysztof J. Cios, Lukasz A. Kurgan from the University of Colorado at Denver [81], and describes diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. It consist of 349 samples with 44 attributes with the class label.

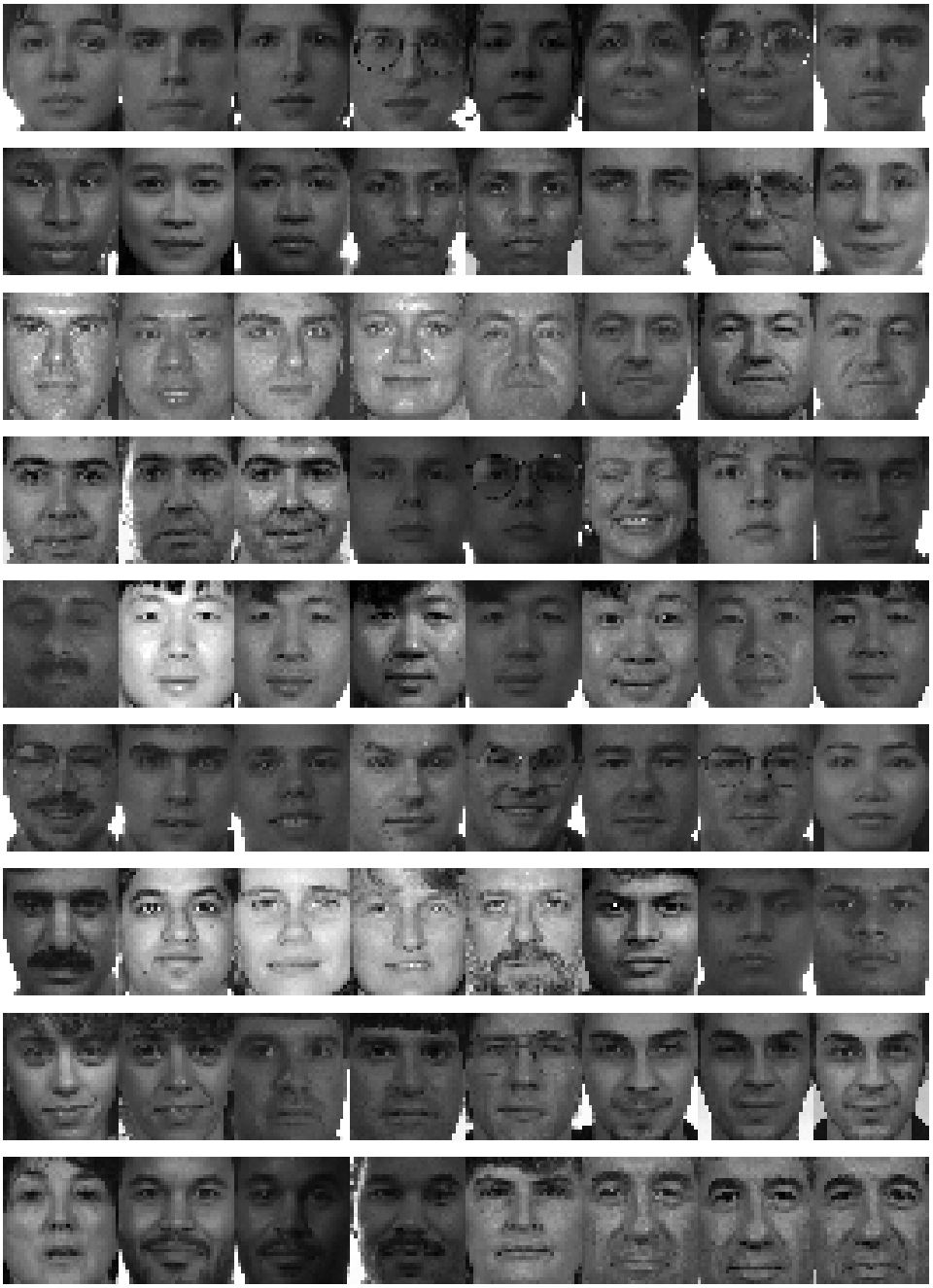


Figure D.5: Some thumbnails from the central part of frontal face examples from the FERET Database. Non uniform backgrounds and different global illumination conditions were present in the database.

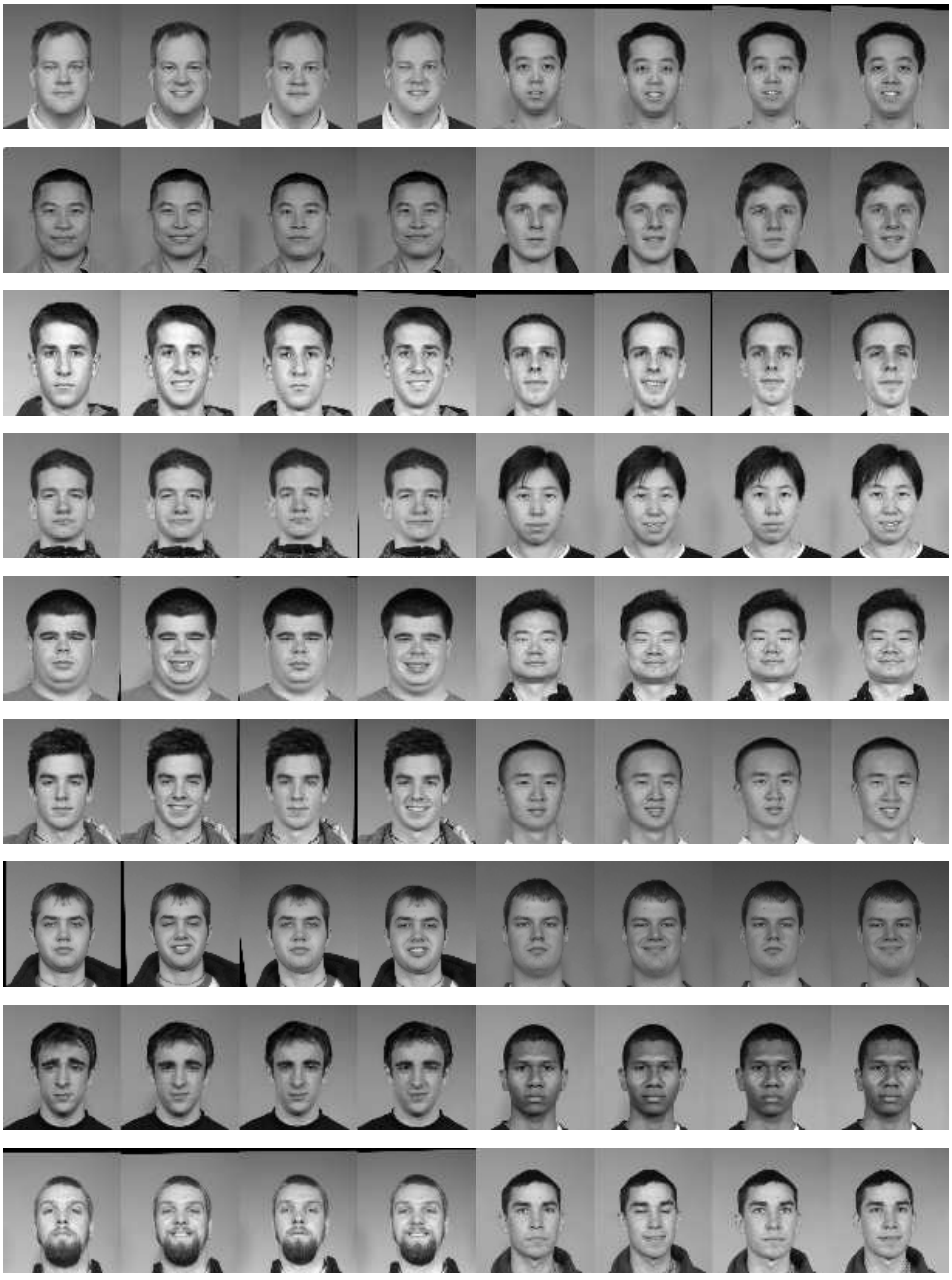


Figure D.6: Examples of male images from the FRGC face database.

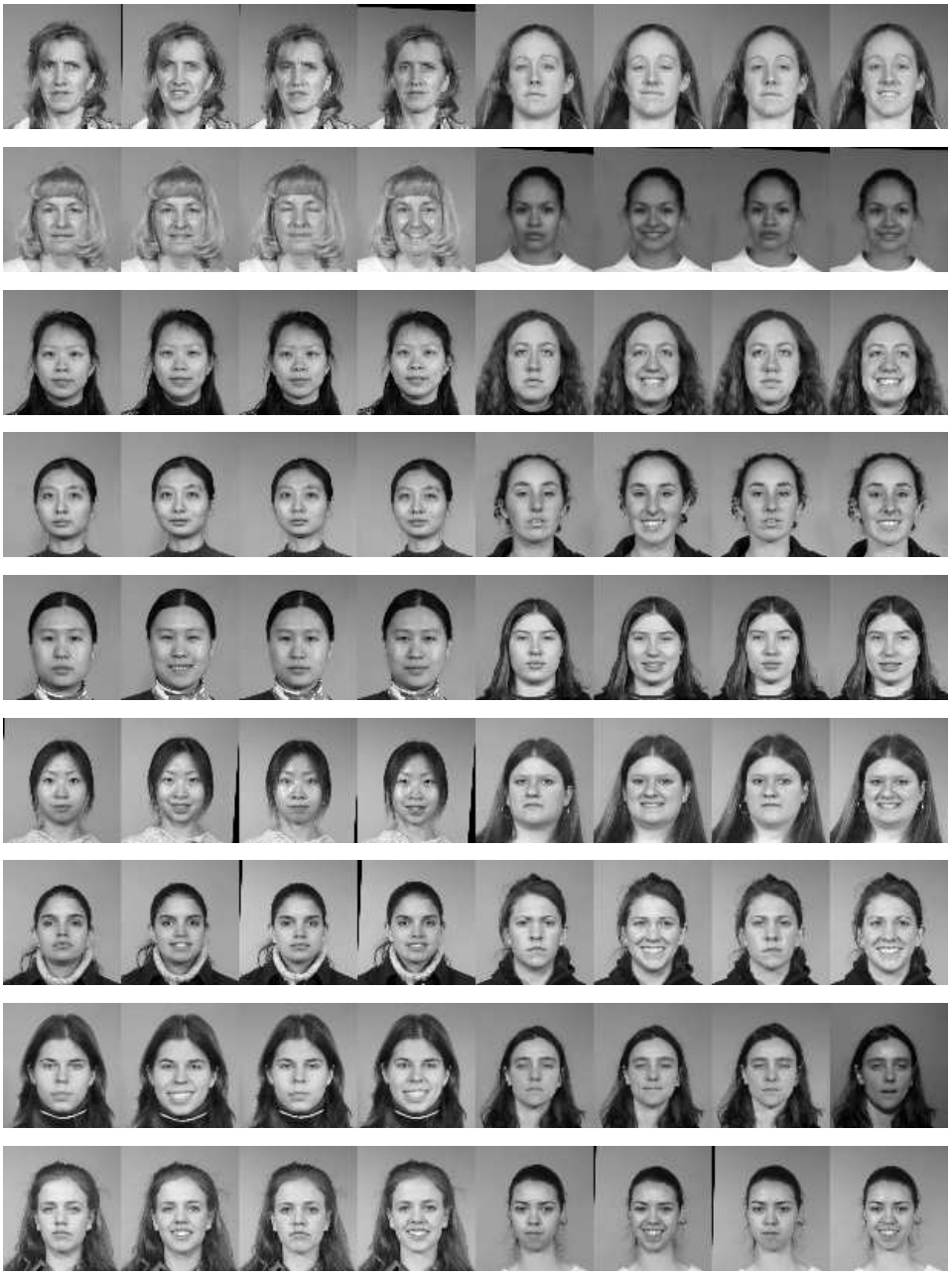


Figure D.7: Examples of female images from the FRGC face database.

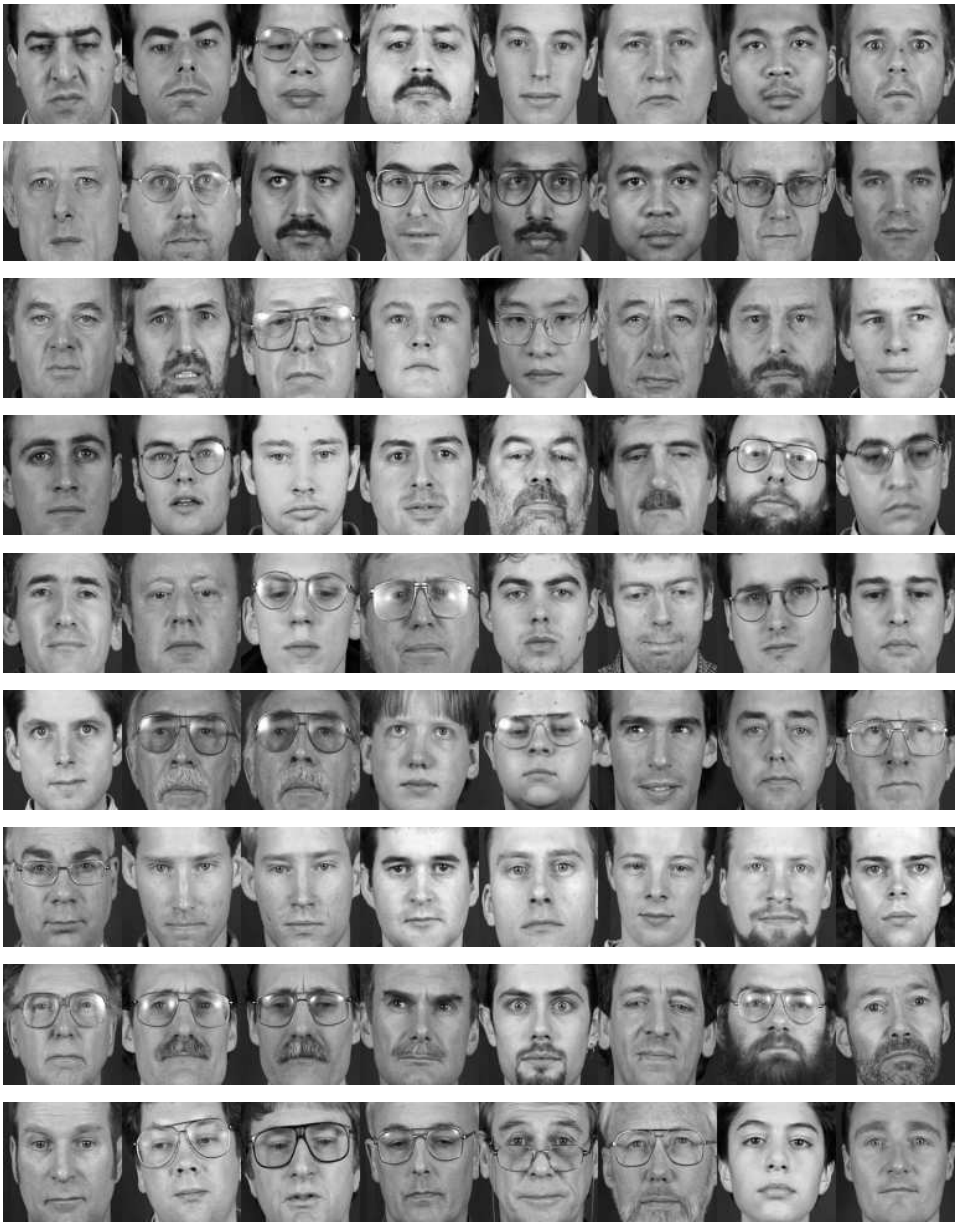


Figure D.8: Examples of male images from the XM2VTS face database.

Figure D.9: Examples of female images from the XM2VTS face database.

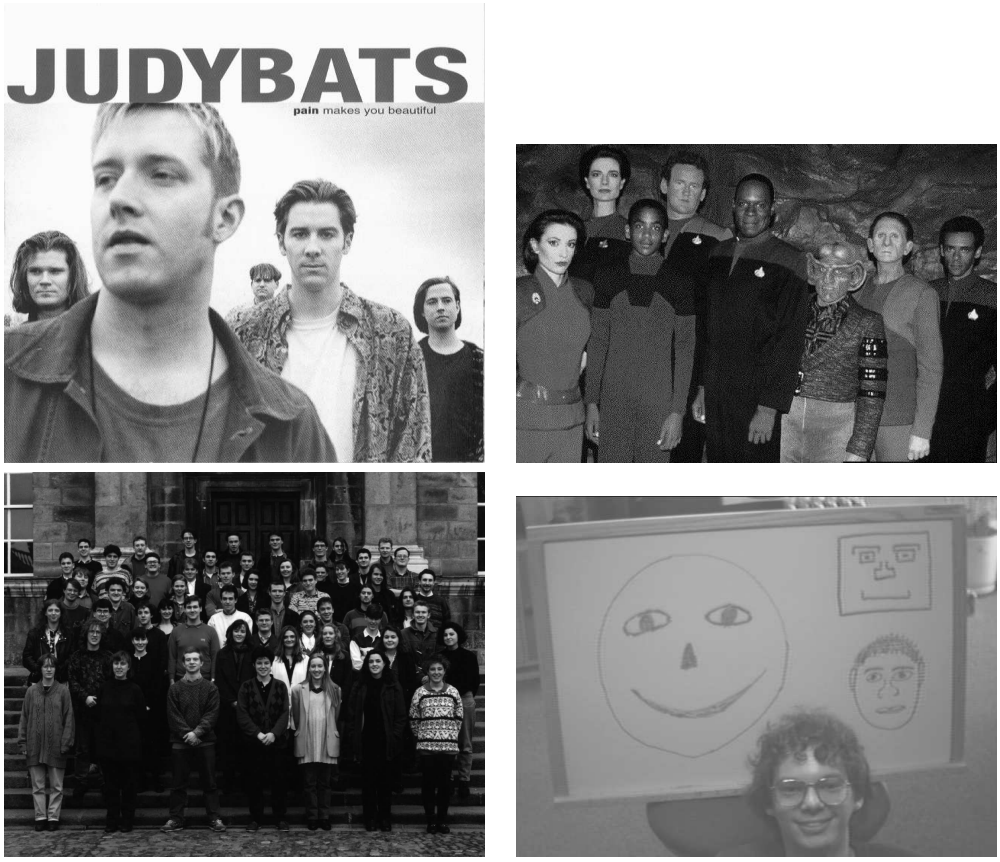


Figure D.10: Examples of images for face detection from the CMU database.

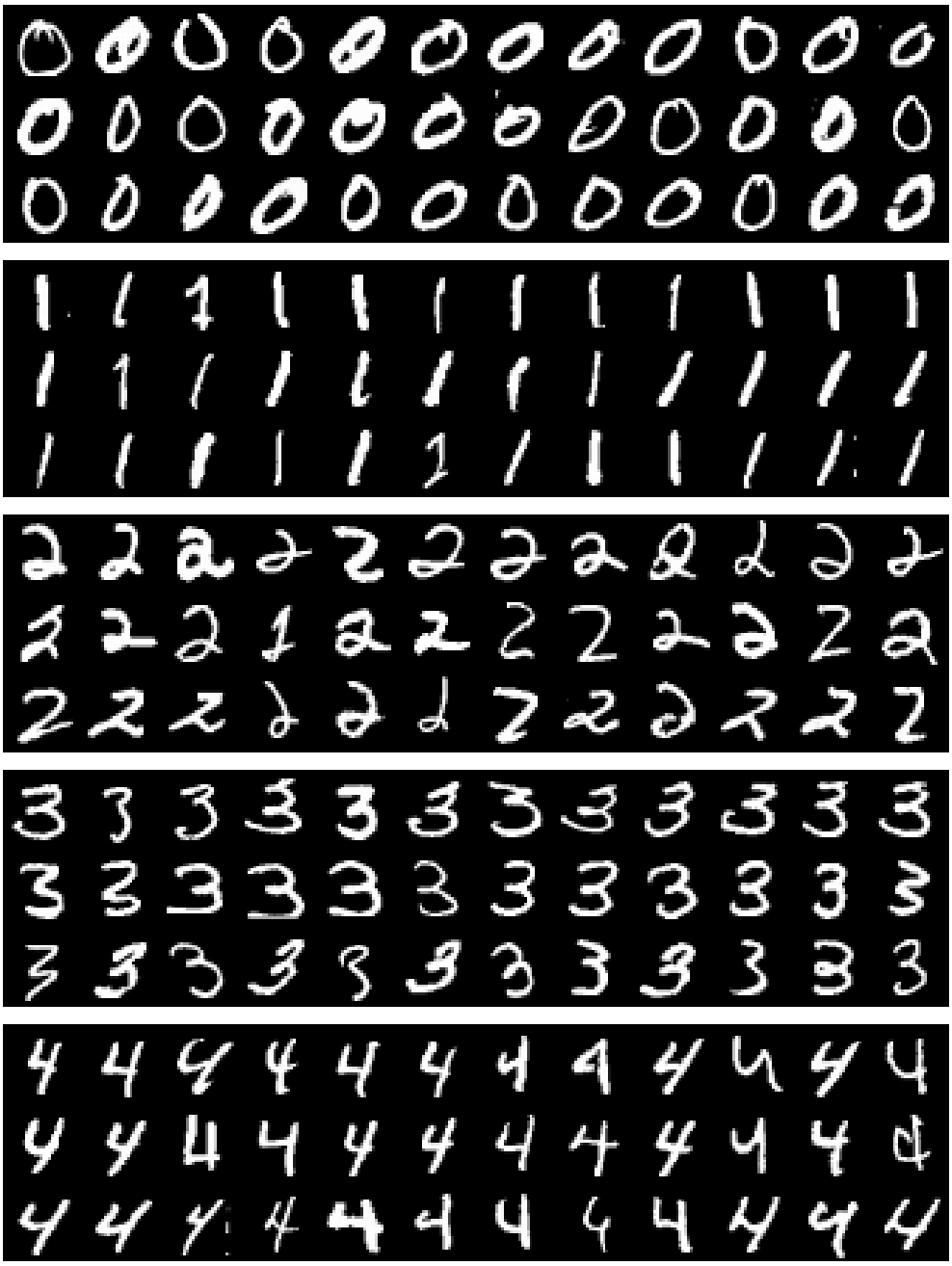


Figure D.11: Examples of some digits (0-4) from the MNIST database.

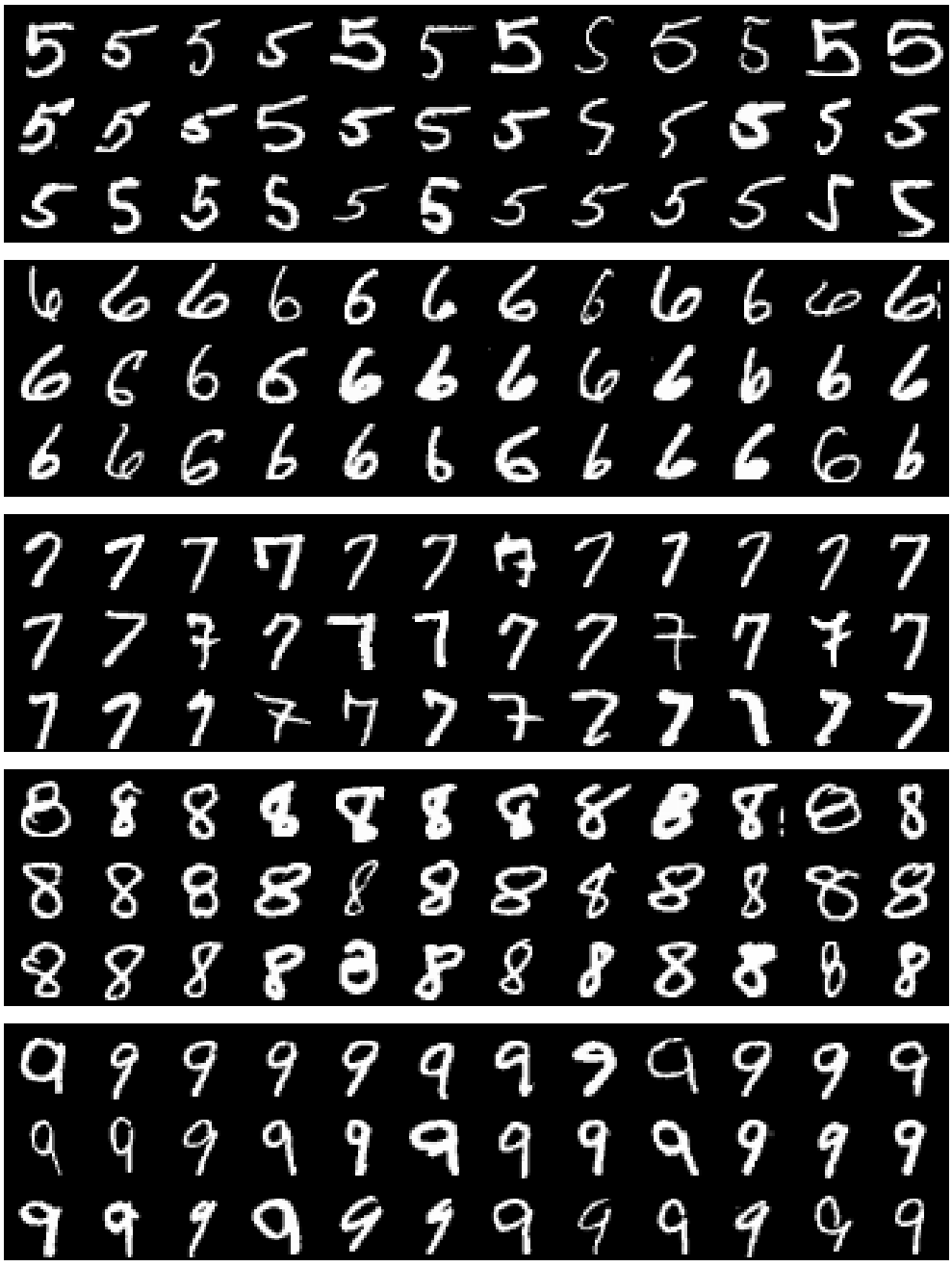


Figure D.12: Examples of some digits (5-9) from the MNIST database.

Appendix E

Publications

The basis and an overview of classic feature extraction algorithms have been previously analysed in my master's thesis:

- David Masip and Jordi Vitrià. Dimensionality Reduction Techniques Applied to Nearest Neighbor Classification. CVC Tech.Rep. #072. Centre de Visó per Computador. July 2003.

The first experimental applications of classic feature extraction techniques to face classification were published in:

- David Masip, Jordi Vitrià. An Experimental Comparision of Dimensionality Reduction for Face Verification Methods, 1st Iberian Conference on Pattern Recognition and Image Analysis (published in Lecture Notes in Computer Science 2652, F.J.Perales, A.Campilho, N.Prez, A.Sanfeliu (Eds.), Pattern Recognition and Image Analysis, June 2003, pp. 530-537). Springer-Verlag Berlin Heidelberg.

where different feature extraction (Linear and Non linear) were used in a face verification problem on natural and uncontrolled environments. The techniques were also applied to the field of gender recognition in:

- D.Masip, J.Vitrià. On the Feature Extraction for Gender Recognition Using the Nearest Neighbor Approach. International Journal of Intelligent Systems, vol. 20, n.5, pp. 561-576. 2005.

- D. Masip, J. Vitrià, On the Nearest Neighbor Approach for Gender Recognition. Catalan Conference on Artificial Intelligence, 2003.

- D.Masip, J.Vitrià. On feature extraction for gender recognition using the nearest neighbor approach, in I.Aguiló, Ll.Valverde, M.T.Escrig (eds), Artificial Intelligence Research and Development, IOS Press, Amsterdam, 2003, pp. 178-188.

The modification on the Adaboost algorithm from chapter 2 where a feature extraction step has been introduced inside the boosting has been published in:

- D.Masip, J.Vitrià. Object recognition using boosted adaptive features. Accepted for oral presentation at ECOVISION: Early Cognitive Vision Workshop, Sabhal Mor Ostaig, Scotland, 28.5. - 1.6.2004.

- D.Masip, J.Vitrià. Adaptive Feature Extraction for Adaboost Learning in Visual Pattern Recognition. Submitted to Pattern Recognition Letters.

The online face detection and Classification Application introduced on chapter 1 appears in:

- D.Masip, M.Bressan, J.Vitrià. Classifier combination applied to real time face detection and classification. AVR2004, Barcelona, February 3-4, 2004. Also in A.Grau, V.Puig, "Recerca en Automàtica, Visió i Robòtica". Any 2004, pp. 345-353, Ed. UPC, ISBN 84-7653-844-8, 2004.

- D.Masip, J.Vitrià. Real Time Face Detection and Verification for Uncontrolled Environments. Second COST 275 Workshop Biometrics on the Internet: Fundamentals, Advances and Applications, pp 55-58 . Vigo (Spain), March 25-26, 2004.

A comparison of the results of the face detection part using the Bayesian approach and the adaptive boosting for the face detection is also published in:

- D.Masip, M.Bressan, J.Vitrià. Feature extraction for real time face detection and classification. Accepted for publication in EURASIP Journal on Applied Signal Processing.

A first version of the application of the Adaboost algorithm to feature extraction task described in chapter 3, where the analytic method for obtaining discriminative projections is explained, was firstly published in:

- D.Masip, J.Vitrià. Boosted Discriminant Projections for Nearest Neighbor Classification. Accepted for publication in Pattern Recognition journal.

- D.Masip, J.Vitrià. Boosted Linear Projections for Discriminant Analysis. Catalan Conference on Artificial Intelligence, 2004. Recent Advances in Artificial Intelligence Research and Development, IOS Press, Amsterdam, 2004.

Later, during my stay in Bangor in collaboration with professor L.I. Kuncheva, the algorithm was evolved adding the random projections method. The final work include a description of the global family of methods for feature extraction using Adaboost, and experimental results using different standard pattern recognition databases and classifiers. This work is published in:

- David Masip, Ludmila I. Kuncheva and Jordi Vitrià, Ensemble-based method for linear feature extraction for two class problems, Accepted for publication in Pattern Analysis and Applications journal.

Finally, the use of external features for face classification as stated on chapter 4, has been published in:

- Àgata Lapedriza, David Masip, Jordi Vitrià. The Contribution of External Features to Face Recognition, 2nd Iberian Conference on Pattern Recognition and Image Analysis (published in Lecture Notes in Computer Science 3523, J.S. Marques et al. (Eds.), Pattern Recognition and Image Analysis, June 2005, pp. 537-544). Springer-Verlag Berlin Heidelberg.

- Àgata Lapedriza, David Masip, Jordi Vitrià. Gender Recognition Using Internal and External Features. Submitted to the tenth IEEE International Conference on Computer Vision (ICCV 2005) Beijing. China.

- Àgata Lapedriza, David Masip, Jordi Vitrià. Are External Face Features Useful for Automatic Face Classification? Accepted for oral presentation in the IEEE Workshop on Face Recognition Grand Challenge Experiments, in conjunction with Computer Vision and Pattern Recognition (CVPR 2005) San Diego.

- Àgata Lapedriza, David Masip, Jordi Vitrià. Experimental Study of the Usefulness of External Face Features for Face Classification. Accepted for publication in the Catalan Conference on Artificial Intelligence 2005. IOS Press series.

Bibliography

[1] A. Lopez, F. Lumbreras, J. Serrat, J. Villanueva. Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:327:335, 1999.

[2] N. M. Laird A. P. Dempster and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[3] V. Bruce A.M. Burton and N. Dench. What's the difference between men and women? evidence from facial measurement. *Perception*, 22:153:76, 1993.

[4] Vassilis Athitsos, Jonathan Alon, Stan Sclaroff, and George Kollios. Boostmap: A method for efficient approximate similarity rankings. In *CVPR (2)*, pages 268–275, 2004.

[5] Moghaddam B. and Yang. Learning gender with support faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):707–711, jan 2002.

[6] D.T. Lawrence B.A. Golomb and T.J. Sejnowski. Sexnet: A neural network identifies sex from human faces. *Advances in Neural Information Processing Systems*, page 572:577, 1991.

[7] Evgeniy Bart and Shimon Ullman. Class-based matching of object parts. In *Proceedings of CVPR Workshop on Image and Video Registration*, page 173, 2004.

[8] Evgeniy Bart and Shimon Ullman. Image normalization by mutual information. In *British Machine Vision Conference*, 2004.

[9] M. Bartlett, J. Movellan, and T. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13:1450–1464, 2002.

[10] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[11] Adam Berger. Convexity, maximum likelihood and all that. Carnegie Mellon University TR, 1998.

[12] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[13] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[14] Eran Borenstein, Eitan Sharon, and Shimon Ullman. Combining top-down and bottom-up segmentation. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 4*, page 46, Washington, DC, USA, 2004. IEEE Computer Society.

[15] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 109–124, London, UK, 2002. Springer-Verlag.

[16] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 109–124. Springer-Verlag, 2002.

[17] Eran Borenstein and Shimon Ullman. Learning to segment. In *ECCV (3)*, pages 315–328, 2004.

[18] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[19] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[20] Marco Bressan. *Statistical Independence for Classification of High Dimensional Data*. PhD thesis, Universitat Autonoma de Barcelona, 2003.

[21] Marco Bressan, David Guillamet, and Jordi Vitria. Using an ica representation of local color histograms for object recognition. *Pattern Recognition*, 36(3):691–701, 2003.

[22] Marco Bressan and Jordi Vitria. Nonparametric discriminant analysis and nearest neighbor classification. Technical report, Computer Vision Center.

[23] Marco Bressan and Jordi Vitria. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, nov 2003.

[24] V. Bruce, A.M. Burton, E. Hanna, P. Healey, O. Mason, A. Coombes, R. Fright, and A. Linney. Sex discrimination: how do we tell the difference between male and female faces. *Perception*, 22:131:52, 1993.

[25] V. Bruce, Z. Henderson, K. Greenwood, P.J.B. Hancock, A.M. Burton, and P. Miller. Verification of face identities from images captured on video. *Journal of Experimental Psychology: Applied.*, 5:339–360, 1999.

[26] Marian Stewart Bartlett Bruce A. Draper, Kyungim Baek and J. Ross Beveridge. Recognizing faces with pca and ica. *Comput. Vis. Image Underst.*, 91(1-2):115–137, 2003.

[27] R. Brunelli and T. Poggio. Hyperbf networks for gender classification. pages 311–314, 1992.

[28] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[29] R. Campbell, M. Coleman, J. Walker, J.P. Benson, S. Wallace, J.Michelotti, and S. Baron-Cohen. When does the inner-face advantage in familiar face recognition arise and why? *Visual Cognition*, 6:197–216, 1999.

[30] K. Chang, K.W. Bowyer, S. Sarkar, and B. Victor. Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions PAMI*, 25(9):1160–1165, September 2003.

[31] Pai-Hsuen Chen Chih-Jen. A tutorial on v-support vector machines, national taiwan university TR.

[32] C.Kotropoulos and I.Pitas. Rule-based face detection in frontal views. In *Procedings International Conference Accoustics, Speech, and Signal Processing*, volume 4, pages 2537–2540, 1997.

[33] Pierre Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314, 1994.

[34] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[35] G.W. Cottrell. Empath: Face, emotion, and gender recognition using holons. *Advances in Neural Information Processing Systems*, 3:564:571, 1991.

[36] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE, Transactions on Information Theory, IT-13*, 1:21–27, 1967.

[37] Dick de Ridder. Independent component analysis, Delft University of Technology TR.

[38] Dick de Ridder and Robert P.W. Duin. Locally linear embedding for classification, delft university of technology TR. Technical report, Delft University of Technology, 2002.

[39] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, UK, 1982.

[40] D.Guillamet and J.Vitria. Evaluation of distance metrics for recognition based on non-negative matrix factorization. *Pattern Recognition Letters*, 24(9):1599 – 1605, jun 2003.

[41] M.Bressan D.Guillamet and J.Vitria. Using an ica representation of high dimensional data for object recognition and classification. In *CVPR*, volume 1, pages 1004–1009, Kauai, Hawaii, December 2001.

[42] M.Bressan D.Guillamet and J.Vitria. Weighted non-negative matrix factorization for local representations. In *CVPR*, pages 942–947, Kauai, Hawaii, 2001.

[43] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[44] Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[45] A. Mike Burton Donald J. Morrison, Vicki Bruce. Covert face recognition in neurologically intact participants. *Psychological Research*, 63(2):83–94, 2000.

[46] H. Ellis. Theoretical aspects of face recogntion. *Perceiveing and remembering faces*, pages 171–197, 1981.

[47] H.D. Ellis, J.W. Shepherd, and G.M. Davies. Identification of familiar and unfamiliar faces from internal and external features: Some implications for theories of face recognition. *Perception*, 8:431–439, 1979.

[48] R. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[49] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[50] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[51] J.H. Friedman. An overview of predictive learning and function approximation. *V. Cherkassky, J.H. Friedman, H. Wechsler (Eds.), From Statistics to Neural Networks: Theory and Pattern Recognition Applications, Springer, Berlin*, pages 1–61, 1994.

[52] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, MA, second edition, 1990.

[53] K. Fukunaga and J. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):671–678, nov 1983.

[54] N. Towell G. Pike, R. Kemp and K. Phillips. Recognizing moving faces: The relative contribution of motion and perspective view information. *Visual Cognition*, 4:409–437, 1997.

[55] S.E.Brennan G. Rhodes and S.Carey. Identification and ratings of caricatures: implications for mental representation of faces. *Cognitive Psychology*, 19:473–497, 1987.

[56] R. P. Gorman and T. J. Sejnowski. Analysis of hidden unitsin a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[57] G.Rhodes. Face recognition and configural coding. *In Cognitive and Computational Aspects of Face Recognition*, pages 47–68, 1995.

[58] D. Guillamet, J. Vitria;, and B. Schiele. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recogn. Lett.*, 24(14):2447–2454, 2003.

[59] David Guillamet. *Statistical Local Appearance Models for Object Recognition.* PhD thesis, Universitat Autonoma de Barcelona, 2004.

[60] B. Edelman H. Abdi, D. Valentin and A. O'Toole. More about the difference between men and women: evidence from linear neural networks and the principal component approach. *Perception*, 24:539:562, 1995.

[61] J. Hilden. Statistical diagnosis based on conditional independence does not require it. *Comput Biol Med.*, 14(4):429–35, 1984.

[62] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.

[63] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[64] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, 2004.

[65] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley and Sons, 2001.

[66] Aapo Hyvarinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Process. Lett.*, 10(1):1–5, 1999.

[67] I. Borg and P. Groenen. *Modern Multidimensional Scaling, Theory and Applications.* Springer-Verlag, New York, 1997.

[68] Izzat N. Jarudi and Pawan Sinha. Relative contributions of internal and external features to face recognition. Technical report, Massachusetts Institute of technology, 2003.

[69] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):153–158, 1997.

[70] Edwin T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[71] R.A. Johnston and H.D. Ellis. The develpment of face recognition. *Cognitive and Computational Aspects of Face Recognition*, pages 1–23, 1995.

[72] M.J. Jones and P.Viola. Fast multi-view face detection. Technical Report 2003-96, Mitsubishi Electric research laboratories, July 2003.

[73] Pearson K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559:572, 1901.

[74] Saul Lawrence K. and Roweis Sam T. Think globally, fit locally: Unsupervised learning of nonlinear manifolds. Technical Report CIS-02-18, University of Pennsylvania, 2002.

[75] John Lafferty Kamal Nigam and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[76] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, Jan 1990.

[77] J. B. Kruskal and M Wish. *Multidimensional scaling*. Sage Publications, Beverly Hills, CA, 1978.

[78] J.B. Kruskal and J.D. Carroll. Geometrical models and badness-of-fit functions. *Multivariate Analysis II*, pages 103–110, 1969.

[79] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[80] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[81] L.A. Kurgan, , K.J. Cios, R. Tadeusiewicz, M. Ogiela, and L.S. Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, 23:149–169, Oct 2001.

[82] Fernando De la Torre and Michael Black. Robust principal component analysis for computer vision. In *Int. Conf. on Computer Vision, ICCV*, pages 362–369, Vancouver, 2001.

[83] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic face identification system using flexible appearance models. *IVC*, 13(5):393–401, June 1995.

[84] Y. LeCun. The mnist database of handwritten digits. The Courant Institute of Mathematical Sciences New York University.

[85] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[86] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, July 1999.

[87] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[88] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proceedings of the Fifth International Conference on Computer Vision*, page 637. IEEE Computer Society, 1995.

[89] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15. Springer-Verlag, 1998.

[90] S. Li, X. Hou, and H. Zhang. Learning spatially localized, parts-based representation. In *CVPR*, volume 1, pages 207–212, Kauai, Hawaii, Dec 2001.

[91] Zhifeng Li and Xiaoou Tang. Bayesian face recognition using support vector machine and face clustering. In *Computer Vision and Pattern Recognition 2004*, volume 2, pages 374–380, 2004.

[92] Marwan Jabri Li Yang. Sparse visual models for biologically inspired sensorimotor control. In *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 131–138, 2003.

[93] N.K. Logothetis and D.L. Sheinberg. Visual object recognition. *Anual Review of Neuroscience*, 19:577–621, 1996.

[94] Philip M. Long and Vinsensius Berlian Vega. Boosting and microarray data. *Machine Learning*, 52:31:44, July-August 2003.

[95] Marco Loog and Robert P.W. Duin. Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):732:739, June 2004.

[96] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, New Jersey, July 2004.

[97] M. Yang, D. Kriegman and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):34–58, jan 2002.

[98] K. P. Bennett & O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software 1*, pages 23–34, September 1992.

[99] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1:18, September 1990.

[100] A. Martinez and R. Benavente. The AR Face database. Technical Report 24, Computer Vision Center, june 1998.

[101] Aleix M. Martnez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):748–763, 2002.

[102] David Masip, Marco Bressan, and Jordi Vitria. Feature extraction methods for real time face detection and classification. *Eurasip Journal of Applied Signal Processing, TO APPEAR*.

[103] David Masip and Jordi Vitria. An experimental comparison of dimensionality reduction for face verification methods. *Lecture Notes in Computer Science*, 2652:530–537, June 2003.

[104] David Masip and Jordi Vitria. Feature extraction for nearest neighbor classification. application to gender recognition. *International Journal of Intelligent Systems*, 20(5):561–576, March 2005.

[105] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, Teewoon Tan, Hong Yan, F. Smeraldi, J. Bigun, N.Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the xm2vts database. In *ICPR '00: Proceedings of the International Conference on Pattern Recognition (ICPR'00)-Volume 4*, page 4858, Washington, DC, USA, Jul 1999. IEEE Computer Society.

[106] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. 1998.

[107] Geoffrey J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Jon Wiley and Sons, Inc, New York, 2004.

[108] D. Kriegman M.H. Yang, N. Ahuja. Face recognition using kernel eigenfaces. In *In Proceedings of Int. Conf. on Image Processing*, volume 1, pages 37–40, 2000.

[109] K.-R. Mller, S. Mika, G. Rtsch, K. Tsuda, and B. Schlkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[110] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[111] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *Proc. of Int'l Conf. on Automatic Face and Gesture Recognition (FG'98)*, pages 30–35, Nara, Japan, April 1998.

[112] H.M. Lades M.S. Bartlett and T.J. Sejnowski. Independent component representations for face recognition. In *in Proceedings of the SPIE: Conference on Human Vision and Electronic Imaging III*, volume 3299, pages 528–539, 1998.

[113] Antonie Naud. *Neural and statistical methods for the visualization of multidimensional data*. PhD thesis, University of Mikolaja Kopernika w Toruniu, 2001.

[114] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[115] R. Setiono O. L. Mangasarian and W.H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization, Thomas F. Coleman and Yuying Li, editors, SIAM Publications*, pages 22–30, September 1990.

[116] Kouropteva O Okun O and Pietikinen M. Supervised locally linear embedding algorithm. In *Proc. of the 10th Finnish Artificial Intelligence Conference*, Oulu, Finland, Dec 2001.

[117] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 130, Washington, DC, USA, 1997. IEEE Computer Society.

[118] P. J. Rauss P. J. Phillips and S. Z. Der. Feret (face recognition technology) recognition algorithm development and test results. Technical Report 995, Army Research Lab, October 1996.

[119] J. P. Hespanha P. N. Belhumeur and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul 1997.

[120] N.R. Pal and V.K. Eluri. Two efficient connectionist schemes for structure preserving dimensionality reduction. *IEEE Transactions on Neural Networks*, 9(6):1142–1154, 1998.

[121] E.S. Palmer. Hierarchical structure in perceptual representaion. *Cognitive Psychology*, 9:441–447, 1977.

[122] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[123] O. Pascalis, S. de Schonen, J. Morton, C. Deruelle, and M. Fabre-Grenet. Mother's face recognition by neonates: a replication and an extension. *Infant Behavior and Development*, 18:79–86, 1995.

[124] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[125] R. L. Goldstone P.G Schyns and J. Thibaut. The development of features in object concepts. *Behavioral and Brain Sciences*, 21:1–54, 1998.

[126] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing Journal*, 16(5):295–306, 1998.

[127] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[128] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E. Tabassi, and J.M. Bone. Frvt 2002: Evaluation report. Technical report, Mar 2003.

[129] Albert Pujol. *Contributions to Shape and Texture Face Similarity Measurement*. PhD thesis, Universitat Autonoma de Barcelona, 2001.

[130] R. Bellman. *Adaptive Control Process: A Guided Tour*. Princeton University Press, New Jersey, 1961.

[131] D. Hilbert R. Courant. *Methods of Mathematical Physics*. Springer-Verlag New York, Inc., Interscience Publishers, New York, 1953.

[132] R. Duda P.Hart and D. Stork. *Pattern Classification*. Jon Wiley and Sons, Inc, New York, 2nd edition, 2001.

[133] G. Rhodes. Superportraits:caricatures and recognition. *Essays in Cognitive Psychology, Psychology press*, 1996.

[134] S. A. Rizvi, P. J. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 48, Washington, DC, USA, 1998. IEEE Computer Society.

[135] Ronald Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, 1994.

[136] Henry Rowley. *Neural Network-Based Face Detection*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, May 1999.

[137] Henry Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. Technical Report CMU-CS-97-201, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, December 1997.

[138] Henry Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[139] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[140] R.P.W.Duin. PRTOOLS v3.17. Technical report, Delft University of Technology, 2004.

[141] H. Wechsler S. Gutta and P.J. Phillips. Gender and ethnic classification. In *Proceedings of the IEEE International Automatic Face and Gesture Recognition*, pages 194–199, 1998.

[142] H. Mitsumoto S. Tamura, H. Kawai. Male/female identification from 8×6 very low resolution face images by neural network. *Pattern Recognition*, 29:331:335, 1996.

[143] Erez Sali and Shimon Ullman. Combining class-specific fragments for object classification. In *Brittish Machine Vision Conference (BMVC)*, 1999.

[144] F.S. Samaria. *Face Recognition Using Hidden Markov Models*. PhD thesis, University of Cambridge, 1994.

[145] R. E. Schapire, Y. Freund, P. L. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5):322–330, 1997.

[146] Robert E. Schapire. Using output codes to boost multiclass learning problems. In *Proc. 14th International Conference on Machine Learning*, pages 313–321. Morgan Kaufmann, 1997.

[147] Robert E. Schapire. A brief introduction to boosting. In *IJCAI*, pages 1401–1406, 1999.

[148] Henry Schneiderman and Takeo Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98)*, pages 45–51, July 1998.

[149] Henry Schneiderman and Takeo Kanade. A statistical model for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2000.

[150] P.G. Schyns and L.Rodet. Categorization creates functional features. *J. Exp. Psychol: Learning, Memory and Cognition*, 23:681–696, 1997.

[151] Michel Vidal-Naquet Shimon Ullman and Erez Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, July 2002.

[152] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10:262–266, 1989.

[153] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database of human faces. Technical Report CMU-RI-TR-01-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2001.

[154] P. Sinha and T. Poggio. I think i know that face. *Nature*, 384(6608):404, 1996.

[155] P. Sinha and T. Poggio. United we stand: The role of head-structure in face recognition. *Perception*, 31(1):133, 2002.

[156] Konstantinos Sirlantzis, Sanaul Hoque, and Michael C. Fairhurst. Trainable multiple classifier schemes for handwritten character recognition. In *Multiple Classifier Systems*, pages 169–178, 2002.

[157] Danijel Skočaj and Aleš Leonardis. Weighted and robust incremental method for subspace learning. In *Ninth IEEE International Conference on Computer Vision ICCV 2003*, volume II, pages 1494–1501, October 2003.

[158] Marina Skurichina. *Stabilizing Weak Classifiers: Regularization and Combining Techniques in Discriminant Analysis*. PhD thesis, Delft University, 2001.

[159] Marina Skurichina and Robert P.W.Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis and Applications*, 5:121–135, 2002.

[160] A. Smola and B. Scholkopf. A tutorial on support vector regression. NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998, 1998.

[161] C. Spearman. General intelligence, objectively determined and measured. *Americal Journal of Psychology*, 15:201:293, 1904.

[162] Robert Mercer Stephen Della Pietra, Vincent Della Pietra and Salim Roukos. Adaptive language modeling using minimum discriminant estimation. In *In International Conference on Acoustics, Speech and Signal Processing*, pages 633–636, San Francisco, March 1992.

[163] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.

[164] Kah Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[165] Daniel L. Swets and John Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):831–836, 1996.

[166] Roweis Sam T. and Saul Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[167] P.Chiroro T. Valentine and R.Dixon. An account of the own-race bias and contact hypothesis based on a "face space" model of face recognition. *In Cognitive and Computational Aspects of Face Recognition*, pages 69–94, 1995.

[168] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[169] P. Thompson. Margaret thatcher: a new illusion. *Perception*, 9:483–484, 1980.

[170] M. Tipping and C. Bishop. Probabilistic principal component analysis. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September 1997.

[171] W.A. Torgerson. Multidimensional scaling 1.theory and method. *Psychometrika*, 17:401:419, 1952.

[172] W.A. Torgerson. *Theory and methods of scaling*. John Wiley and Sons, 1958.

[173] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, Mar 1991.

[174] Shimon Ullman, Erez Sali, and Michel Vidal-Naquet. A fragment-based approach to object representation and classification. In *4th International Workshop on Visual Form*, pages 85–102, May 2001.

[175] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory and Methods*. Wiley Interscience, 1998.

[176] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[177] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[178] A.Burton V.Bruce, P.Hancock. Human face perception and identification. *Face Recognition: From Theory to Applications (H.Wechsler, P.J.Phillips, V.Bruce, F.F.Soulie and T.S. Huang Editors)*, pages 51–72, 1998.

[179] Michel Vidal-Naquet and Shimon Ullman. Object recognition with informative features and linear classication. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 281, Washington, DC, USA, 2003. IEEE Computer Society.

[180] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–519, Kauai, Hawaii, Dec 2001.

[181] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, pages 137 – 154, 2002.

[182] E Wachsmuth, MW Oram, and DI Perrett. Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque. *Cereb. Cortex*, 4(5):509–522, 1994.

[183] Mark Weiser. The Computer for the 21st Century. *Scientific American Ubicomp*, 3:94–104, September 1991.

[184] Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[185] William H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences*, volume 87, pages 9193–9196, December 1990.

[186] S.Z. Li X.R. Chen, L. Gu and H.J. Zhang. Learning local features for face detection. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 19–26, Hawaii, December 2001.

[187] A. W. Young and V. Bruce. Perceptual categories and the computation of "grandmother". *European Journal of Cognitive Psychology*, pages 5–49, 1991.

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