Universitat Politècnica de València

Departamento de Comunicaciones

Tesis Doctoral

Contributions on Automatic Recognition of Faces using Local Texture Features

Presentada por:
David Monzó Ferrer
Dirigida por:
Dr. Alberto Albiol Colomer

Valencia, 2012.

A mi familia, por su apoyo incondicional

Siempre fuiste mi espejo; para verme sólo tenía que mirarte.

Julio Cortázar

Acknowledgments

This work is not only the result of all my efforts but a consequence of the many supports and useful advices that I received.

To my parents and my brothers. The pillar of my education and unconditional support.

To my friends. They patiently released me from the hardest moments and continuously encouraged me to go on.

To all the people of the old and new Clementine. We shared the most time of this Thesis between hours of work and laughs.

To the people of Visual. In their eyes, they always saw me as a doctor.

To Alberto, Antonio and Jorge. They offered me the opportunity of developing this work and pushed me into the field of Computer Vision.

Resumen

Uno de los temas más destacados del área de visión artificial se deriva del análisis facial automático. En particular, la detección precisa de caras humanas y el análisis biométrico de las mismas son problemas que han generado especial interés debido a la gran cantidad de aplicaciones que actualmente hacen uso de estos mecanismos.

En esta Tesis Doctoral se analizan por separado los problemas relacionados con detección precisa de caras basada en la localización de los ojos y el reconocimiento facial a partir de la extracción de características locales de textura. Los algoritmos desarrollados abordan el problema de la extracción de la identidad a partir de una imagen de cara (en vista frontal o semi-frontal), para escenarios parcialmente controlados. El objetivo es desarrollar algoritmos robustos y que puedan incorporarse fácilmente a aplicaciones reales, tales como seguridad avanzada en banca o la definición de estrategias comerciales aplicadas al sector de *retail*.

Respecto a la extracción de texturas locales, se ha realizado un análisis exhaustivo de los descriptores más extendidos; se ha puesto especial énfasis en el estudio de los Histogramas de Gradientes Orientados (*HOG features*). En representaciones normalizadas de la cara, estos descriptores ofrecen información discriminativa de los elementos faciales (ojos, boca, etc.), siendo robustas a variaciones en la iluminación y pequeños desplazamientos.

Se han elegido diferentes algoritmos de clasificación para realizar la detección y el reconocimiento de caras, todos basados en una estrategia de sistemas supervisados. En particular, para la localización de ojos se ha utilizado clasificadores boosting y Máquinas de Soporte Vectorial (SVM) sobre descriptores HOG. En el caso de reconocimiento de caras, se ha desarrollado un nuevo algoritmo, HOG-EBGM (HOG sobre Elastic Bunch Graph Matching). Dada la imagen de una cara, el esquema seguido por este algoritmo se puede resumir en pocos pasos: en una primera etapa se extrae un grafo facial ubicando automáticamente los puntos más significativos de la cara; de cada uno de estos puntos, se extrae un descriptor local HOG y se concatenan. Finalmente, el vector de características biométricas pasa por una etapa de reducción de dimensionalidad. Al vector resultante se le aplica una clasificación basada en vecino más próximo (*Nearest Neighbor*) para asignarle una etiqueta (identidad de la persona).

Usando las base de datos de FRGC, se consiguió localizar ojos con una precisión del 92.3% con un error menor al 5% de la distancia inter-ocular, mejorando los resultados obtenidos por otros autores marcados como referentes. En reconocimiento de caras, usando la base de datos de FERET se ha demostrado que el uso de descriptores locales HOG proporciona mayor información bio-

metrica que otros descriptores clásicos como los coeficientes Gabor (en algunos casos mejorando hasta un 40% la tasa de reconocimiento). En la localización de puntos faciales característicos, el uso nuestro propio algoritmo proporcionó resultados comparables con el uso de otros grafos como Active Appearance Models (AAM). Por último, se ha demostrado que la inclusión de información de color en los descriptores HOG añade información útil para el reconocimiento, mejorando en la base de datos de FRGC hasta un 11% la tasa de reconocimiento frente a los descriptores trabajando con la intensidad de las imágenes.

Para evaluar el sistema totalmente automático de HOG-EBGM para reconocimiento de caras, se ha participado en el concurso internacional MOBIO sobre el desarrollo de nuevos algoritmos. MOBIO proporcionó una base de datos en formato de video grabado en escenarios realistas con un dispositivo móvil. Este concurso nos ha aportado un excelente contexto para comparar nuestra solución con la de otros participantes. En la evaluación de los resultados del concurso, HOG-EBGM se posicionó como la cuarta mejor solución.

Abstract

One of the most prominent topics today in the field of computer vision is that of facial analysis. In particular, the detection and location of human faces in images and the biometric analysis of them are topics that have raised great interest due to the number of industrial applications that make use of them.

This doctoral dissertation carries out an independent study of the problems derived from two topics: face detection with eye location and face recognition using a local texture feature-based approach. The algorithms developed are focused on overcoming the problem of extracting the identity from a face image (in frontal or semi-frontal views) in semi-controlled scenarios. The goal was to develop robust algorithms readily applicable to real applications, such as advanced banking security and the definition of marketing strategies based on client statistics.

Regarding the extraction of local textures, an in-depth study is performed on some of the most extended features, taking into special consideration the Histograms of Oriented Gradients (HOG descriptors). Working with normalized face representations, these descriptors offer discriminative information about key facial landmarks (such as the eyes, the mouth, etc.), being robust to illumination variations and small displacements.

Various classification algorithms have been considered for face detection and recognition, all following a supervised learning strategy. Specifically, some boosting and Support Vector Machines (SVM) classifiers have been used to classify local textures extracted from the eyes (i.e. HOG descriptors), for eye location. In the case of face recognition, a novel feature-based algorithm has been developed, HOG-EBGM (HOG on Elastic Bunch Graph Matching). Given a face, the main steps of HOG-EBGM can be summarized in the following: first, a facial graph is automatically built, locating some facial keypoints; from each of these points, a HOG local descriptor is extracted and all of them concatenated; the final biometric face vector is obtained applying dimensionality reduction techniques; finally, the samples are matched to a database using a *Nearest Neighbor* approach.

Performing on the database of FRGC, the eyes were localized with a precision of 92.3% with an error lower than 5% of the inter-ocular distance, overpassing the results obtained by some referent authors. Regarding face recognition, using the FERET database, it has been proved that our use of HOG local descriptors provides more biometric information than other classical descriptors, such as Gabor coefficients (improving the recognition rate up to a 40% in some cases). Using the HOG-EBGM algorithm for the localization of facial landmarks produced simmilar results to other extended algorithms, such as the

Active Appearance Models (AAM). Finally, the experiments have shown that the inclusion of color cues in HOG features provides with more information useful for face recognition, improving the recognition rates when using FRGC up to a 11% compared to the use of the descriptors with gray-scale images.

To evaluate the automatized HOG-EBGM for face recognition we also participated in the international MOBIO contest. MOBIO provided a database of video samples in realistic scenarios (recorded with a mobile device), and offered an excellent context to compare our solutions with those of the different participants. In the evalution of the MOBIO results, HOG-EBGM ranked as the fourth best solution among all participants.

Resum

Un tema destacat del camp de la visió artificial és el derivat de l’anàlisi facial automàtica. En particular, la detecció precisa de cares i l’anàlisi biométrica de les mateixes, són problemes que han generat especial interès a causa de la gran quantitat d’aplicacions que actualment fan ús d’aquests mecanismes.

En aquesta tesi doctoral s’analitzen per separat els problemes derivats de dos temes: la detecció precisa de cares basada en la localització dels ulls, i el reconeixement facial a partir de l’extracció de caractarístiques locals de textura. Els algorismes desenvolupats se centren en resoldre el problema de l’extracció de la identitat a partir d’una imatge de cara (en vista frontal o quasi-frontal), per escenaris parcialment controlats. L’objectiu d’aquest treball és desenvolupar algorismes robusts i que puguen incorporar-se fàcilment en aplicacions reals, tals com a seguretat avançada en banca o la definició d’estratègies comercials aplicades al sector de ’*retail*’.

Respecte a l’extracció de textures locals, s’ha portat a terme una anàlisi exhaustiva d’alguns dels descriptors més estesos; especialment, s’han estudidat amb més deteniment els Histogrames de Gradients Orientats (*HOG features*). Fent servir representacions normalitzades de la cara, aquests descriptors ofereixen informació discriminativa dels elements facials (ulls, boca, etc.), mostrant robustesa a variacions en la iluminació i petits desplaçaments.

Així doncs, per realitzar la detecció i el reconeixement de cares s’han triat diferents algorismes de classificació han estat triats per realitzar la detecció i el reconeixement de cares, tots ells basats en una estratègia de sistemes supervisats. En particular, per la localització d’ulls s’han emprat classificadors boosting i SVM (Support Vector Machines) treballant amb descriptors HOG. En el cas de reconeixement de cares, s’ha desenvolupat un nou algorisme, l’HOG-EBGM (HOG on Elastic Bunch Graph Matching). Donada la imatge d’una cara, l’esquema seguit per aquest algorisme es pot resumir en diferents passos: de primer s’extrau un graf facial situant automàticament els punts més significatius de la cara; de cadascun d’aquests punts, s’extrau un descriptor local HOG i es concatenen. Finalment, el vector de característiques biométriques passa per una etapa de reducció de dimensionalitat. Al vector resultant se li aplica una classificació basada en veí més proper (*Nearest Neighbor*) per assignar-li una etiqueta (la identitat de la persona).

Usando las base de datos de FRGC, se consiguió localizar ojos con una precisión del 92.3% con un error menor al 5% de la distancia inter-ocular, mejorando los resultados obtenidos por otros autores marcados como referentes. En reconocimiento, usando la base de datos de FERET se ha demostrado que el uso de descriptores locales HOG proporciona mayor información biometrica que otros descriptores clásicos como los coeficientes Gabor (en algunos casos mejorando hasta un 40% la tasa de reconocimiento). En la localización de puntos faciales característicos, el uso nuestro propio algoritmo proporcionó resultados comparables comparables con el uso de otros grafos como Active Appearance Models (AAM). Por último, se ha demostrado que la inclusión de información de color en los descriptores HOG añade información útil para el reconocimiento, mejorando en la base de datos de FRGC hasta un 11% la tasa de reconocimiento frente a los descriptores trabajando con la intensidad de las imágenes.

Fent ús de la base de dades de FRGC, s'han localitzat ulls amb una precissió del 92.3%, amb un error menor al 5% de la distància inter-ocular, millorant els resultats obtinguts per altres autors considerats referents. En reconeixment de cares, fent ús de la base de dades de FERET s'ha demonstrat que l'utilització de descriptors locals HOG dona major informació biomètrica que altres descriptors més clàssics, com els coeficientes de Gabor (millorant en alguns casos fins al 40% la tasa de reconeixement). En la localització de punts característics facials, fent ús del nostre algoritme va donar resultats comparables als obtinguts amb altres algoritmes, com l'Active Appearance Models (AAM). Per ùltim, s'ha demonstrat que la inclusió d'informació de color en els descriptors HOG afegeix útil per al reconeixement, millorant en la base de dades FRGC fins a un 11% la tasa de reconeixment comparat en l'ús de descriptors treballant en imatges de intensitat.

Per avaluar el sistema totalment automàtic de HOG-EBGM de reconeixement de cares, s'ha participat en el concurs internacional MOBIO sobre el desenvolupament de nous algorismes. El MOBIO ens ha proporcionat una base de dades en format de video gravat en escenaris realistes amb un dispositiu mòbil. Aquest concurs ens ha proporcionat un excel·lent context per comparar la nostra solució amb la d'altres participants. En la avaluació dels resultats del concurs, HOG-EBGM es va possicionar com a la quarta millor solució.

Contents

Chapter 1

Introduction to the Facial Analysis Problem

1.1 Introduction

Most biological mechanisms are continuously repeated even without noticing. This is the case of the brain processes associated to the sensing and understanding the world that surrounds us. The information perceived through our senses is internally processed to give meaning to a complex context. This way, we are able to detect all kind of objects and also to recognize different instances of them, even on adverse external conditions. Although we are familiar with the object analysis mechanisms from the biological perspective at a subconscious level, it is still a central problem to develop them in computer vision.

Facial analysis is a subcase of the more general *object analysis*, in which the understanding processes are focused on a specific object class, human faces. This issue is of high relevance for the human reasoning and therefore the brain has developed specific cells to perform it. Each human brain is specifically trained in detecting and recognizing great quantities of different individuals, performing these tasks in very efficient ways. However, computer vision solutions to the facial analysis problem are far away from the results obtained by the brain, both in accuracy and speed.

Two facial analysis topics, *face detection* and *face recognition*, have attracted the attention of the scientific community, and are still target of many studies.

This thesis intends an approach to the facial analysis problem in computer vision, providing a joint face detection and face recognition approach. For that purpose, this work relies on the use of local texture features to extract descriptive information of the facial elements.

The primary goals of this work are to address two different problems:

1. **To go beyond the current state of the art face detection methods, producing new ones whit a higher level of location accuracy**: to achieve this goal, an accurate approach based on the automatic detection of the eyes is studied.

2. **Study and design face recognition methods to deal with partially uncontrolled scenarios**: the input of such systems will be automatically

linked to the output of the face detection with eye location method.

Finally, let's notice that the human brain is a blank page when we are born. Only through a thorough learning process, our brain cells are trained to detect and recognize different objects and faces by extension. The learning process takes months and even years for the baby, as it needs time to gather a significant quantity of models.

The analogue way of emulating the learning process of the brain in computer vision is using *supervised algorithms.* In machine learning, supervised algorithms are tasks that use datasets of labelled samples to train a system. During the training, the system learns the properties of the elements that have to be detected or recognized.

In this work, supervised algorithms are used in both, eye location and face recognition tasks. In the first case, the models represent *eyes* as opposed to *non-eyes* –being the latter the facial elements that are not eyes. In the case of face recognition, two kind of training models can be used. One, to describe the different facial elements, so that they can be located. Another, to learn how to deal with the variations between different individuals. These are called the *inter-personal* variations.

The rest of this chapter is organized as follows: it first starts reviewing some neurological facts in relation with facial analysis; next it goes with a definition of the problems of face detection and face recognition. After this, the chapter describes the philosophy of supervised learning algorithms and finally an outline of the organization of this thesis work is presented.

1.2 Neurological base of the Facial Analysis

Face detection and face recognition have been two topics widely studied by neuroscientists from a biological point of view. Physiological researches [102] have indicated that in the human brain we posses specific cells for facial analysis. These cells are placed in the *inferotemporal* cortex and also spread over the frontal right hemisphere.

Engineers have found that the prior knowledge acquired on psychophysics and neurophysiologic may sometimes become relevant when trying to implement an automatic face detection or recognition system. From the study carried by Zhang *et al.* [126] some interesting points related on this prior knowledge can be highlighted:

- Human recognition system does not only use visual perception. Instead, it uses a broad spectrum of *stimuli*, specially those that come from auditory and olfactory senses, besides the visual sense.

- The process of face perception combines both holistic and feature analysis. The adult brain starts with an holistic approach, followed by a refinement carried out taking on account the individual facial features [75]. In the case of children, their brain pay attention mainly to isolated features of the face, as they still have not learned to focus on big objects [88, 26].

- Spatial frequency analysis plays an important role in face detection. Low frequencies contribute to detect global features while high frequencies contribute to differentiate the finer details. Also the localization of features, like borders and edges, has an important contribution in the process.

Thanks to these mechanisms, the human brain can detect and recognize faces even if they appear in different locations, with different sizes, rotations or poses. Also the brain can deal with other external factors such as marked illumination variances or even partial occlusions.

On the contrary, two factors may contribute to a deterioration of the human face perception: the fatigue and dealing with an elevated order of faces to be recognized. These two factors can easily be overcome by computers.

In this thesis, the biological base of some of the facial analysis algorithms used is remarked, specially in relation with the extraction of texture features.

1.3 Facial Analysis in Computer Vision

The topic of facial analysis in Computer Vision is still an open issue: the approaches that have been developed produce results not comparable to the ones obtained by humans.

This section addresses the problems that arise when performing automatic face detection and face recognition algorithms. It is highly important to learn from the difficulties that are associated to the facial analysis, as they constitute the base from which the systems are designed. To achieve this goal, first a more formal definition on face detection and face recognition is given and then the section tackles the main challenges that are derived from them.

1.3.1 Face Detection: Definition and Applications

Face detection is known as the process to extract all faces in an image, regardless from the scenario where they are located. The result in this process is a bounding box which roughly locates the face. Additionally, some information related to the bounding box is also provided, such as the coordinates of its central point or its extent.

The influence of the scenarios on face detection algorithms has been studied in depth. Torres *et al.* [105] concluded that in general, scenarios can be classified into two large groups:

- *Simple scenarios*: constituted by images with a controlled number of faces –usually a single face per image– and semi-controlled conditions. Usually, faces are presented in frontal view, without occlusions and good illumination.

  Also in this group, the background is mainly static, usually with plain colors that facilitates the differentiation between the faces and their surroundings.

  The detection of faces in this group tends to be simple for static images and video sequences. Most of the public face datasets present people in *simple scenarios.* Figure 1.1 shows an example of this, extracted from the FERET database [93].

- *Complex scenarios*: constituted by difficult images to perform reliable detections. This is the case of images with partially or totally occluded faces,

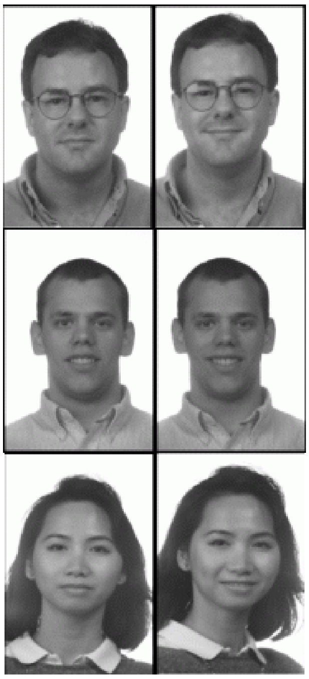


Figure 1.1: Face images from FERET database in *simple scenario.*

several face orientations (i.e. profiles, mid-profiles, etc.), with dynamic backgrounds or with sharp shadows and illumination variances, among others.

Face detection systems working with *difficult scenarios* are more sensitive to make errors, not only when finding out the position of the faces but also when tracking them along video sequences. Robust methods become necessary in these cases, sometimes reinforced with additional image *preprocessing* steps.

Figure 1.2 shows a few examples from *Labeled Faces in the Wild* [45], one of the few datasets that present people in *complex scenarios.*

Regarding its applications, several fields benefit from face detection. Next, some examples are summarized:

- **Face Recognition prior step:** a proper face detection stage is considered key to obtain good recognition performances. This issue is directly related to the purposes of the current work.

- **Video Indexing:** it is one of the features included in MPEG-7. A great quantity of details may be specified along with the video information, such as the number of individuals on each frame, in order to allow fast and efficient searching for material that is of interest to the user. In that sense, face detection has become a powerful tool for people indexing [5].

- **Entertainment:** more and more, the video-game industry allows the users to interact with the system by automatically detecting the face of

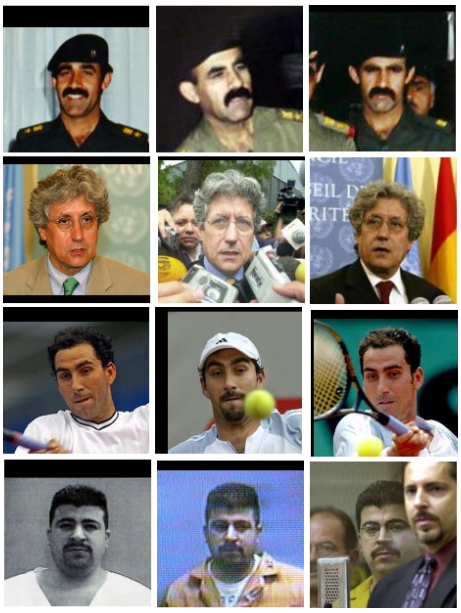


Figure 1.2: Face images from *Labeled Faces in the Wild* database in *complex scenario.*

the player [115]. In that sense, video-games like *FIFA Soccer 12* let the player upload a face picture, which is detected and analyzed to create a soccer player avatar from it.

- **People Counting:** a simple scenario for this application could be counting the number of customers entering a shop during the day by attaching a camera at the entrance [129].
- **Noise Elimination and Image Compression**: sometimes, when encoding an image it is useful to keep better quality when compressing the face area rather than on the background. For these cases, the detection of the face is a necessary first step [124].
- **Camera Autofocus**: Some recent digital cameras and mobile phones use face detection for autofocus [96], making sure that all faces in the image are properly exposed. Also, face detection has been used for selecting regions of interest in photo slideshows, using Ken Burns effects for panning and zooming [18].

1.3.2 Face Recognition: Definition and Applications

Face recognition is the process through which, given a face image, an identity label is automatically assigned to it after matching it against a database of known people. If no matching is possible, an *unknown person* label is given.

Face recognition is a complex task, as people can easily modify their facial appearance. One may change in a short period of time the color and length of the hair; people have also the possibility of wearing beard, moustache, glasses, cap or helmet. Also, as the people gets older their appearance varies considerably.

The classification of *simple* and *complex* scenarios exposed for face detection is also valid for face recognition. The simplest recognition methods are designed to detect individuals who usually look in frontal view without any elements occluding facial features. More complex systems deal with problems such as multiple-angle views, individuals wearing garments like a scarf, sunglasses, etc. Also, it is quite common to perform recognition tasks on single images, rather than videos. Usually, the techniques working with videos, the perform a low-level analysis frame by frame (or at least in some key images) and then the spatial and time redundancies of faces in the sequences are used to fuse all this information[1].

Finally, let us notice that face recognition is not only a theoretical methodology but its applications have spread over a number of different fields. Nowadays, it is considered to be one of the most prolific research fields in computer vision and so it has merged with advanced technologies. Some possible applications are proposed next:

- **Surveillance:** most face recognition applications are aimed to reinforce the security, mainly due to its versatility [23, 35]. The most common applications are advanced video surveillance or CCTV systems in public cluttered places where suspect people have to be localized, detected and tracked; other surveillance applications cover the issue of avoiding potential robberies (i.e. in banks or other commerces) by tracking suspect people.

- **Information security:** face recognition is a non-intrusive biometric, compared to other biometrics like the fingerprint analysis or the iris recognition [119]. Therefore, it is suitable for accessing personal data, such as Internet, bank accounts or medical records. Also, it is useful to validate control systems at the entrance of a restricted area by only showing the face to a camera.

- **Robotics:** as it has been studied in some works [76], face detection and recognition makes possible an improvement of the features in human machine interfaces (HMI), specially when working with robots. With a simple decision mechanism a robot can decide whether it is interacting with the right person or not.

- **Assisted Living:** the care of elder and impaired people can be made easier incorporating elements of computer vision. For example, a recognition system could be installed at the main entrance of a home to help the elder that is living alone to recognize incoming visitors.

Due to its great versatility and the currentness of the computer face analysis technologies, a number of enterprises[2] are specifically aimed to cover the necessities produced in this field, being an economically profitable area that should not be ignored.

[1] In Chapter 6 we develop our own face recognition algorithms oriented to work with single images, but in Chapter 7.1 we extend them to take advantage of the information in video sequencies

[2] Some of the most outstanding are *Argus Solutions*, *DigiSensory Technologies*, *Neurotechnolojiya* and *Cognitec Systems Gmbh*

1.3.3 Challenges of Facial Analysis

Even for the human brain and depending on the context, faces corresponding to the same individual may look completely different. Detect and recognize faces in complex scenarios depends on a high number of uncontrollable factors. These factors, can be related to the faces themselves (we call them *intrinsic* factors) or can come from external agents (we call them *extrinsic* factors).

Regarding the *intrinsic* factors, the main sources of variation are the following:

- **Interpersonal Variations**: although all faces are structured following a common biological pattern (of its elements and their distributions) the difference among faces of different people can be quite significant. These variances are mainly due to factors such as the identity, sex, ethnicity, skin or hair color. These variations can make people look very different from one another. Taking as many of these factors as possible into consideration leads to considerably enhance a recognition system.

- **Intrapersonal Variations**: these are the factors that make that even on the same scenario, a unique face of a person may be presented quite different aspects. We can summarize the main intrapersonal variations in the following: different face expressions and gestures, ageing, wearing complements (usually glasses, caps or scarves) or changes in hair-styles.

Regarding the *extrinsic* factors that most affect to faces, the following can be found:

- **Physical Geometry**: the three-dimensional nature of the head affects directly to the representation of the face. Depending on the angle of the camera that is recording the faces and the orientation of the face itself, the pose may change completely. Slight changes in the pose lead to high differences in the representation.

- **Scenario set-up**: the external conditions derived from the scenario set-up are an important source of variance. Basically, these conditions are provoked by two factors: variations in illumination and occlusions that may add noise to the extraction of the facial features.

  Regarding the illumination, strong variances can make the task of facial analysis really hard to perform, as great changes in shade and color are registered. Also, the position of the focus of light might affect on the casting of not desired shadows in the face. Typically, indoor illumination can be controlled better than outdoors.

  Regarding the occlusions, external objects placed in front of the faces may partially or fully interfere on the information that can be retrieved. The higher the occlusion, the more difficult to perform any facial analysis.

- **Camera set-up**: the parameters of the camera also affect to the performance of face detection and face recognition. The most significant factor related to the camera is the resolution, which has an influence on the scale of the faces in the image. The greater the size of a face, the higher amount of information that can be retrieved from it. Other parameters, such as the focus or the acquisition noise may also have an influence on the facial analysis process.

The main challenge in facial analysis is after all to come up with a solution that tackles the greatest number of intrinsic and extrinsic factors.

1.4 Supervised Algorithms

In machine learning, supervised algorithms are used to tackle with classification problems. Supervised algorithms are structured into three different phases: the supervised learning phase, which is also called *training phase*, the *validation phase*, in which the set-up parameters are defined and the classification phase, which is usually called *test phase*:

- During the training phase, a classification function is inferred from the information contained in a set of samples, called the *training dataset* (also, the *training database*).

  In supervised tasks, the training samples are labelled, so that their actual class is always known. Also, the training sample should be representative of the real context of the classification

- During the validation phase, a set of labelled samples is tested, so that the internal set-up parameters of the system can be adjusted to fit the current problem.

- During the test phase, the inferred classification function is applied to a test set to predict the label of each of the samples.

Figure 1.3 summarizes the main steps of the supervised learning process and the interactions among them.

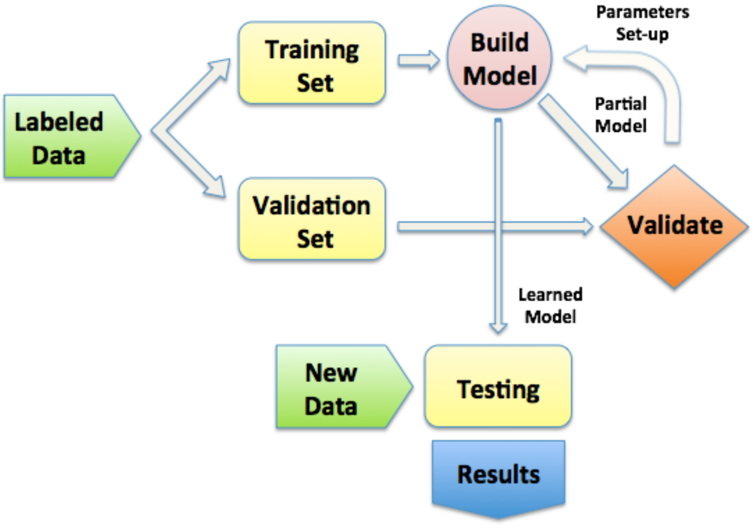


Figure 1.3: Diagram of a Supervised Learning process.

Both, face detection and face recognition processes, can be seen as classification problems. In face detection, the classification is typically a binary problem: *faces* against *non-faces*. In face recognition, the classification complexity is higher, as in this case each identity (different person) constitutes its own classification class.

Also, other sub-processes related to the facial analysis, such as the location of specific face elements, like the eyes, can be approached in terms of a two-class classification.

To achieve this goal, this thesis relies on the use of local texture descriptors, which are present along the different stages of the systems. The goal of extracting local features is to provide a highly discriminative context for the supervised classification processes. Thus, the local features have to be as invariant to the extrinsic factors as possible (so that they do not affect the results), while highly descriptive so that they can take advantage on the intrinsic variations.

To learn the classification function provided by the local features, this thesis has mainly used two widespread off-the-shelve supervised classifiers: the *Adaboost* classifier [113] and the *Support Vector Machines* [29].

More specifications on these approaches and the algorithms that have been employed are detailed through the different chapters of this work.

1.5 Summary of Contributions

The main contributions of this work are summarized in the following:

- We have developed a fully automatic system for recognizing people from images, based in a two-stage solution: face detection and face recognition. The system is aimed to overcome this challenge for stand-alone images of faces in frontal view, at any scale, with in-plane rotations and in partially uncontrolled scenario set-ups. Both stages are based on the use of supervised algorithms.

- We have developed a face detection with eye location stage, based on a hierarchical framework in which first two boosting classifiers detect the face regions and some eye candidates, and next a SVM classifier uses texture features based on the local gradients (HOG descriptors) to select the best eye candidates.

  The boosting and the SVM classifier have been used and train specifically for the purposes of this thesis. This stage has been extensively tested on a set of face databases and compared to other state of the art eye location algorithms.

- We have developed a novel face recognition algorithm in the category of Face Graph Algorithms (FGAs), which locates specific facial landmarks using an EBGM approach and extracts HOG features to locally describe each of them. The biometric signature of each face is the concatenation of the descriptor for each facial landmark that integrates the Face Graph.

  The experimentation of this stage has been carried out using different face datasets. To perform a fair comparative, other HOG-based Face Graph algorithms, such as the HOG-Grids and the HOG-AAM, have been developed. Also, an study of the inclusion of color cues in the face stage is done.

In all cases, a brief study on dimensionality reduction algorithms is carried out to determine the best techniques to reduce the descriptive information of faces.

- We have assembled a fully automatic face recognition system. This system has been tested in realistic and uncontrolled scenarios: it has been proved on video-sequences of people video-calling with a mobile phone. The system in this scenario has been compared to other current approaches in the framework of the MOBIO international competition. In this competition, we show that the fusion of our detection and recognition stages works and the results are quite competitive with other approaches.

1.6 Outline of the Dissertation

The remaining chapters of this work are organized as follows:

- Chapter 2 reviews the concepts of face representation and the necessity of normalizing the faces to establish a common context of analysis.
- Chapter 3 analyzes the techniques used to evaluate supervised classification algorithms, specifically applied on face detection and face recognition
- Chapter 4 studies and selects the local texture descriptors that can be used for the facial analysis and their applicability to facial analysis processes.
- Chapter 5 presents the design of face detection with eye location algorithms and the experimentation derived from such developments. .
- Chapter 6 presents the design and comparative of some feature-based face recognition approaches using local textures, and the experiments that are carried out to validate them.
- Chapter 7 presents a fusion of the face detection and face recognition algorithms developed in the previous chapters, in the context of MOBIO, an international research competition.
- Chapter 8 summarizes our approach and provides a discussion of the advantage, disadvantages and applications derived from it. It also suggests some directions for future research on this area.

Also with this work two appendices are given to complement the information regarding work-related topics that are not the goal of this work but help to understand the development and results that are obtained. Specifically, Appendix A deals with the topic of dimension reduction algorithms while Appendix B briefly describes the contents of the face databases used to test all the system designs.

Chapter 2

Facial Representation and Normalization

Prior to processing a face, the tasks performing facial analysis, like face detection and face recognition, need to work on a set of images with homogeneous representation. Inevitably, one question arises when considering this problem: *what is the optimal way of representing faces for such tasks?*

The goal of the current chapter is to provide an answer to this question. The methods here exposed will determine the normalization (*preprocessing*) stage that all the images will need to undergo before any processing in this thesis[1].

We can overcome the problem of facial representation from two distinct approaches:

1. Studying the properties of the spatial projection of a 3D object (the face) in a 2D world (the image).
2. Studying the representation of the faces once in the image world, which is related to the intensity and color information of the objects.

The rest of this chapter is structured as follows: first the problem of the spatial representation of the face as 3D objects is introduced, followed by a study of the intensity and color representation of the objects in the image. Regarding the latter topic different color spaces useful for the task of recognition are analyzed closely. The final section is dedicated to describing the face normalization stage, derived from the procedures outlined in the aforementioned sections.

2.1 Spatial Representation of Faces

Most of the time, the biological approach employed by the human brain to solve cognitive problems related to the visual system can provide practical clues to tackling the same issues in computer vision. In this vein, the previous question can be reformulated more generally: *how does the human visual system represents objects for later recognition?*

Since the decade the 1980s, many researchers have attempted to address this question [66, 108, 104]. In these works, a 3D object (a face), needed to

[1] In page 103, Figure 6.1 there is a detailed diagram of this process.

be recognized from its 2D representation. The decision criterion for recognition in [66] can be stated in the following simplified form:

$$\|\mathbf{PT}X^{3D} - X^{2D}\| < \theta, \tag{2.1}$$

where $\mathbf{T}$ is an aligning transformation, $\mathbf{P}$ is a projection operator from 3D to 2D; the norm $\|\cdot\|$ measures the dissimilarity between the projection of the transformed 3D model X^{3D} and the input image X^{2D}. A recognition decision is then made based on a comparison between the measured dissimilarity and the threshold θ.

The first conclusion to be drawn from the cited works is the necessity of aligning the projection of the 3D face and the actual image. A normalization procedure must be performed that considers the location, scale and in-plane rotation of the 2D image projection of the face.

We know from experience that the shape and features of a face vary considerably when the images compared correspond to different views. For example, it is very difficult even for the human brain to recognize a face when it has to match images in frontal view with images in profile (i.e. faces with a rotation of $\alpha = 90^o$).

This work tackles the solution to this issue based in two assumptions:

1. A 3D model of the face will not be employed, as this thesis limits itself to using only 2D specific view-related face representations. The use of 2D computer vision algorithms is mainly motivated by the fact that from a practical point of view, the major use of video devices corresponds to classical cameras, usually in fixed positions. The use of specific devices for extracting 3D information, like the extraction of depth-maps with stereo-vision or infrared grids, is not spread, restricting the scope of applications where the algorithms could run and considerably increasing the overall costs of the final systems. Moreover, working with 3D models of the face considerably increases the complexity of the representation; the use of 3D models is incompatible with the algorithms studied herein.

2. A single-view approach will be used, with frontal view as a reference model, in opposition to the multi-view approach [112], in which a face model is represented by several 2D views (such as frontal, mid-profile, profile, etc.). The single-view approach is the most common found in the literature; all the works mentioned in this thesis are single-view.

   The use of specific algorithms (such as the feature-based approaches proposed in Chapter 6 for face recognition) offers the opportunity to deal with frontal face images as well as a range of image views close to the frontal representation.

2.2 Importance of the Intensity-based Facial Representation

Once the issue of spatial representation of the faces has been addressed, a second point arises relating to illumination information. The human brain is very adept at recognizing faces based only on the light intensity received. Recent research on face recognition in humans [123] indicates that color information can be very helpful in certain contexts (e.g. when images have an extremely low resolution). It is worth asking whether including color when representing faces might provide a better context for facial analysis. The objective of this section is to review the fundamentals of intensity-based representation of faces in order to understand the discriminating properties that can be considered when using color cues.

With digital images, the intensity (or luminance) of objects in a scene is the most elemental information to describe them. This information is given as a set of values $I(x, y)$, unique for each pixel of the image. Usually, each pixel value is quantized with 8-bits to designate the intensity (also known as gray-scale values), so that $I(x, y) \in [0, 255]$. The intensity values can be considered as *raw data* used to describe the objects. As such, two types of descriptive algorithms can be derived:

1. Algorithms that base their descriptions directly on this *raw* data. In these algorithms, only the pixel information is used.
2. Algorithms that include some intelligence to extract more complex information after processing the intensities.

The *holistic* facial analysis methods, that consider the face as a whole, typically belong to the first group [13, 107, 12]. They work with features extracted from the intensity values of the face image, usually vectorized. As part of these methods the vector of intensities undergoes some post-processing such as dimensionality reduction methods, which is the case of the PCA analysis [107] or the LDA technique [12][2].

Nevertheless, intensity representation is also the basis for more complex means to describe objects. This is the case of the so called *feature-based* methods, which are aimed at extracting descriptive features to describe specific elements of the face, such as local textures. This will be examined in detail in further chapters.

The study of intensity methods to represent facial images provides a baseline useful to compare with more advanced feature-based extraction techniques.

2.3 Study of Color Spaces for Face Representation

Although the majority of works use exclusively the intensity information from images, color is still considered an important cue in object recognition. Some authors [123, 121, 44] have even shown that working on color can yield more distinctive features useful for classifying face images. Given that variations in illumination may be highly detrimental to recognition rates, the additional information provided by color features could mitigate these effects.

Unlike intensity, there are a variety of ways to represent the color information of an image, depending on the models that describe it. One of the most relevant aspects to be considered when using color information to extract biometric features is the selection of the color model that best suits the problem. These models represent the color information reflected in each of the pixels, commonly described by three channels: C_1, C_2 and C_3. Each color model represents the

[2] Further details on both techniques can be seen in Appendix A

information in a different format, depending on the channel. Sometimes, these channels are referred to as color coordinates; hence, the color models can be treated as a geometrical representation and discussed in terms of *color spaces*.

The geometrical distribution of the color samples in an object depends on the color space that is used to represent it. A specific distribution may favor or detract from a better description of the object in question. For this reason, this work devotes a detailed study on the issue of color spaces in order to discern which models provide higher discriminative power.

To date, a number of color models have been developed to represent the various data images. The present work reviews those models known to retain more discriminative information useful in subsequent separability by classes, using as a reference the research work done by Van de Sande *et al.* [109] and Yang *et al.* [120] respectively. However, the issue of discovering the optimal color subspace for face recognition is still open. To tackle this topic we have followed a strategy where we combine a study of three baseline conventional color spaces (RGB, HSV and Opponent Color), motivated by the study of Van de Sande *et al.* [109], with a particular solution proposed by Yang and Liu in [120], called *Discriminant Color Space* (DCS). The latter adapts the problem of the color space to the specific problem that is intended to solve; in this case, face recognition.

The rest of this section is dedicated to describing the characteristics of each of the aforementioned color models.

2.3.1 RGB Model

The RGB model is the most common color representation used in images as the acquisition systems are based on it. In this model, the three color channels of the image correspond to the red (R), green (G) and blue (B) components of light: $C_1 = R, C_2 = G, C_3 = B$. It is therefore an additive model based on the primary colors that follows the biological mechanism of the eye to perceive colors. The geometry of the color space generated by this model is commonly represented by a cube, as seen in Figure 2.1.

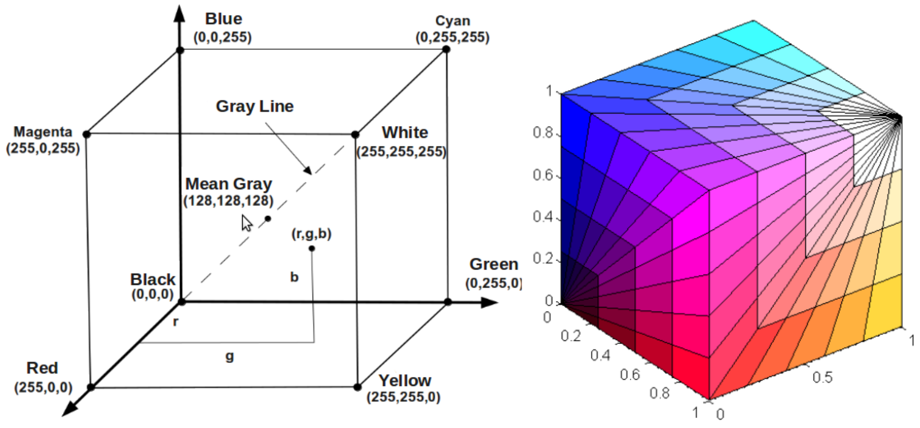


Figure 2.1: Representation of the color space generated with the RGB model.

The RGB model offers no remarkable discriminative properties for describing objects in recognition problems. Nevertheless, it is considered a fundamental model: it constitutes not only the reference against which other methods are compared, but also the base from which the vast majority of the other classic color models are derived.

Despite its simplicity, the quantity of information provided by the RGB model is much higher than that extracted directly from a gray-scale image. Actually, it is worth mentioning that the intensity value of a pixel can be directly inferred from the RGB components in that pixel. One of the most common methods to solve this conversion is given by the equation [95]:

$$I(x, y) \simeq 0.3 * R + 0.59 * G + 0.1 * B \tag{2.2}$$

2.3.2 HSV Model

This model represents colors based on three channels: hue (H), saturation (S) and value (V): $C_1 = H, C_2 = S, C_3 = V$. These three components are commonly represented into a cylindrical coordinate system, which is obtained through a non-lineal transformation of the RGB color space. If C_{MAX} and C_{MIN} are defined as the maximum and minimum values of the RGB components, then the transformation from the RGB space to the HSV space can be expressed as:

$$\begin{gathered}
H = \begin{cases} \left(0 + \frac{G-B}{C_{MAX}-C_{MIN}}\right) \times 60, & \text{if } R = C_{MAX} \\ \left(2 + \frac{B-R}{C_{MAX}-C_{MIN}}\right) \times 60, & \text{if } G = C_{MAX} \\ \left(4 + \frac{R-G}{C_{MAX}-C_{MIN}}\right) \times 60, & \text{if } B = C_{MAX} \end{cases} \\
S = \begin{cases} 0 & \text{if } G = C_{MAX} \\ 1 - \frac{C_{MAX}}{C_{MIN}}, & \text{rest of the cases} \end{cases} \\
V = C_{MAX},
\end{gathered} \tag{2.3}$$

where $C_{MAX} = max(R, G, B)$ and $C_{MIN} = min(R, G, B)$. Figure 2.2 displays the color space generated by the HSV model.

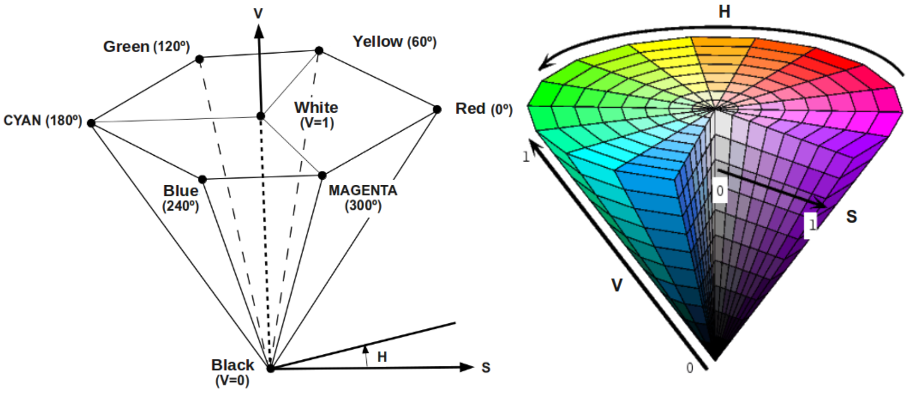


Figure 2.2: Color space generated by the HSV model.

One known problem of the HSV color space is that the hue component, H, becomes potentially unstable when the points approach the gray line[3]. This fact can have significant implications when extracting discriminative features. As a solution, Van de Weijer *et al.* [110] analyzes the error propagation for the transformation of the Hue component, H. Their analysis reveals that the certainty of the hue component is inversely proportional to the value of the saturation component, S. Thus, Van de Weijer *et al.* demonstrates that the model can achieve higher levels of robustness by multiplying each hue sample by the value of the saturation. Accordingly, this solution will be applied when using an HSV color space.

Some authors, such as Bosch *et al.* [19], have used the color space generated by the HSV model to extract discriminative features. Specifically, they used it in the context of scene classification, as a basis for extracting local features, including the SIFT descriptors [65].

In the HSV color model, the channel of the Value is the one that suffers greater variation when the intensity of the light changes. The channels of Hue and Saturation tend to remain stable with regard to the variations in light intensity.

2.3.3 Opponent Color Model

The Opponent Color Model is based on the theory proposed by Hering (1885) and concerns the natural ability of the eye to detect opponent colors. According to this theory, the visual experience is produced in the eye by three opponent processes: green-red, yellow-blue and black-white. This model proposes a color space generated by two chromatic axes (green-red and yellow-blue), and a luminance axis. Figure 2.3 renders the geometric representation of this model for a constant value of intensity. Hering used his opponent color theory to explain why it is not possible for our eye to perceive an object that is simultaneously red and green, while it is possible to see, for example, an orange object, that is red and yellow.

Based on Hering's theory, Hurvich and Jameson [48] have shown that in the opponent color model there is a cancellation of the Hue component produced in the HSV model. As described earlier, Hue is an unstable component of the HSV model; thus, the opponent color model becomes a more robust option and one suited for tasks involving feature extraction. Numerous studies have been carried out following these guidelines [109, 110].

This color space is derived from the linear combination of RGB-model components. Therefore, the color space can be defined as:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}, \tag{2.4}$$

where the channels C_1 and C_2 contain the chrominance information of the red-green and yellow-blue axes respectively, and the component C_3 is the intensity. The two chrominance components C_1 and C_2 remain invariant to displacements in both location and light intensity.

[3] As it can be seen from Figure 2.1.

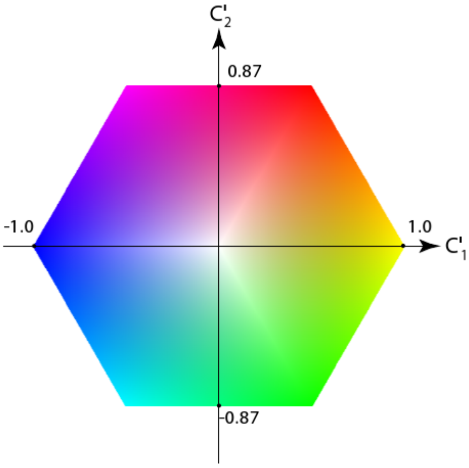


Figure 2.3: Color space generated by the Opponent Color Mode with a constant value of intensity

2.3.4 Discriminant Color Space - DCS

Discriminant Color Space is a color model proposed by Yang *et al.* in [120], with the aim of creating a space completely adapted to the needs of face recognition. The principal idea behind this model is that a specific color space can be trained to adapt a set of sample images, using an optimal combination of the RGB components from the color image:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} q_{11}R + q_{12}G + q_{13}B \\ q_{21}R + q_{22}G + q_{23}B \\ q_{31}R + q_{32}G + q_{33}B \end{pmatrix} = Q^T[R, G, B] \tag{2.5}$$

where $R \in \mathcal{R}^{N \times 1}, G \in \mathcal{R}^{N \times 1}, B \in \mathcal{R}^{N \times 1}$, are the vectorized color components of the original image $X \in \mathcal{R}^{m \times n}$, with dimension $N = m \times n$, and $Q \in R^{3 \times 3}$ is the matrix of coefficients to perform the transformation.

To calculate the optimal weights q_{ij}, this model stipulates minimizing the intra-class scatter matrix of the samples, while the inter-class scatter matrix is maximized. In other words, this method essentially consists of applying a LDA algorithm to the original RGB samples[4].

As in all LDA-based algorithms, this model adopts the Fisher Criterion which makes use of the traces of two scatter matrices. From the analysis performed in [120], the inter-class scatter matrix, S_b, and the intra-class scatter matrix, S_w are defined from a set of training samples. A total of n samples are divided into c distinct classes, such that:

$$S_b = \sum_{i=1}^{c} P_i(\bar{X}_i - \bar{X})^T(\bar{X}_i - \bar{X}) \tag{2.6}$$

$$S_w = \sum_{i=1}^{c} P_i \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (X_{ij} - \bar{X}_i)^T(X_{ij} - \bar{X}_i), \tag{2.7}$$

[4] For further information about basics of the LDA algorithm, see Appendix A.3.

where $X_{ij} \in \mathcal{R}^{m \times n}$ is the j-th sample corresponding to an image labeled as it belongs to class i, $\bar{X}$ is the mean image from all the training samples, $\bar{X}_i$ and N_i are the mean image and the number of samples in class i, respectively, and $P_i = \frac{N_i}{n}$ is the *a priori* probability of class i.

The solution to the maximization of the Fisher Criterion can be obtained from the solution of a generalized eigenproblem (Equation A.7). Therefore, the transformation matrix from the RGB original color space to the Discriminant Color Space can be obtained from the eigenvectors corresponding to the three eigenvalues that result from $S_b = \lambda S_w \Phi$.

Note that in this method only three eigenvectors are available, as they represent the three color components $C1$, $C2$ and $C3$ from the new color subspace.

2.4 Face Image Normalization

Before processing, all face images need to be homogenized as part of a preprocessing normalization stage. The goal of this homogenization is to preserve a common set of properties for all images. The basic block diagram of this normalization stage can be observed in Figure 2.4, and will be used for the face detection and recognition algorithms detailed in Chapter 5 and Chapter 6, respectively.

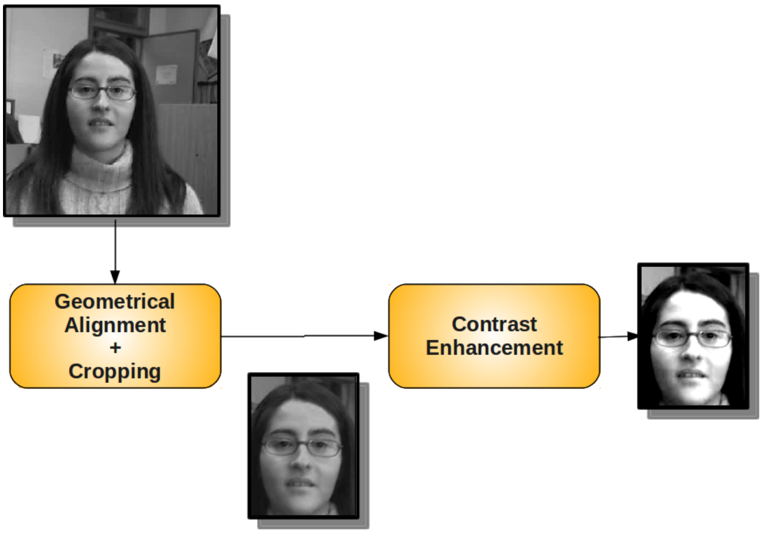


Figure 2.4: Steps in the face normalization stage.

The representation information given in the previous sections of this chapter leads to the definition a geometric normalization. This normalization is in alignment, scale and in-plane rotation of the face. An intensity-based normalization is likewise obtained by stretching the histogram, with an extension to color-based images. Finally, the normalization generated for the experiments carried out in the present work is presented.

2.4.1 Geometrical Normalization

To achieve a complete homogenization of facial images, we need to assume that they represent the same view. The faces also need to be aligned in terms of location, scale and in-plane rotation. With this geometrical normalization, the limits of the normalized face can then be defined by cropping the original image. This eliminates noisy background information that does not belongs to the face.

The problem of geometrical normalization can be summarized in the following question: given a set of faces, is it possible to align them? To answer this, first a set of criteria must be established for what an *aligned face* should look like:

- A face is considered aligned in *rotation* when the line that links the eye-centers is completely parallel to the horizontal axis.
- A face is considered aligned in *location* when all its facial features (e.g. the eyes, nose or mouth) are symmetrically places with regard to the central vertical axis of the image.
- A face is considered aligned in *scale* when at least two of the facial features are always placed in the same coordinates.

As part of this process, a two-dimensional representation of a face is treated as a matrix of pixels. To meet the normalization criteria established above, an affine transformation is applied. Given the intensity of a pixel $I(x, y)$ (with coordinates x and y) from a source image, a linear affine transformation is obtained by multiplying it with a transformation matrix, $\mathbf{T}$, such that:

$$I_{Norm} = \mathbf{T}I(x, y), \tag{2.8}$$

where I_{Norm} belongs to the normalized and cropped face. One of the most important properties of the linear transformation matrix is that it can be defined as the concatenation of any number of affine transformations, such as the translation (T_t), rotation (T_r) and scale (T_s). In our case, the transformation matrix can be expressed as:

$$\begin{aligned} \mathbf{T} = T_t T_r T_s &= \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} s_x cos(\theta) & s_y sin(\theta) & t_x \\ -s_x sin(\theta) & s_y cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned} \tag{2.9}$$

where $t = [t_x, t_y]$ is the translation in pixels to be applied, θ represents the angle of rotation and s_x, s_y is the scaling in both axes, respectively.

Thus, given any two fixed points in the transformed image, if their correspondence in the original image is known and both main axes are assumed to be equally scaled, $s = s_x = s_y$, then the matrix (2.9) can be completely solved.

In the case of faces, the two points selected to solve the transform matrix could correspond to any of the facial features. The two eye-centers were chosen in our case, as they are the most characteristic and potentially able to be detected with computer vision algorithms. The eyes provide direct information about the rotation angle θ (corresponding to the angle between the eyes and the horizontal), the scaling factor s (proportional to the inter-ocular distance), and the displacements t_x and t_y (when the eyes at the transformed image are placed in known coordinates).

Nevertheless, this normalization requires knowing the exact location of the eyes, which is not a trivial question. In fact, this issue constitutes one of the principal motivations for developing an accurate face detection algorithm, such as that proposed in Chapter 5.

As a final remark, the properties of transformation matrices can be utilized so that given any of the affine transformations proposed in the present work, the inverse transformation can always be calculated by directly inverting the transformation matrix, $\mathbf{T}$:

$$\mathbf{T}_{inverse} = \mathbf{T}^{-1} \tag{2.10}$$

This is useful for extracting the normalized facial image, since the process consists of running the inverse transformation pixel-by-pixel (via linear interpolation) on the original image. The entire transformation is delineated in Figure 2.5.

2.4.2 Intensity-based Normalization and application to Color Images

As mentioned previously, a geometric alignment of the faces is insufficient given that variances in illumination may lead to other types of misalignments. One of the most direct ways of providing robustness in the illumination is to homogenize the histograms of the various face samples. In this sense, the stretching of the histogram, which is essentially an increase in image contrast, becomes a useful tool. The fundamentals of this procedure are described here.

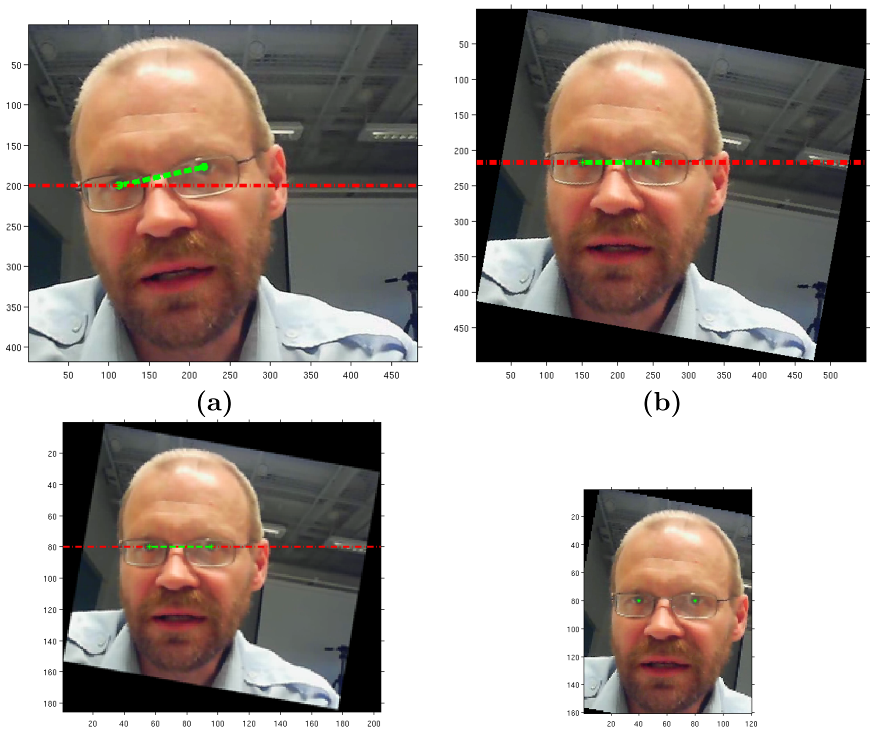
Given a gray-scale image, represented by an intensity matrix where its pixels are $I(x,y)$ (being x and y the pixel coordinates), an increase in contrast is defined as an stretching of the intensity histogram, such that every output pixel, $\hat{I}(x,y)$ is:

$$\hat{I}(x,y) = \big(I(x,y) - c\big)\left(\frac{b-a}{d-a}\right) + a \tag{2.11}$$

where a and b are the upper and lower limit values for the normalization, and c, d are the minimum and maximum intensity values in the image, respectively. In 8-bits gray-scale images, the values of a and b are typically $a = 0$ and $b = 255$, while the values of c and d are relative to the content of each image.

Figure 2.6 presents the increments of the contrast for a typical face recognition image, as well as their histograms. It is worth noticing how the image with stretched contrast actually stretches its histogram to the full extent of its range.

Given the importance and simplicity of the normalization techniques used here, the remainder of this thesis will assume hypothetically that for every intensity image used in any algorithms, a previous normalization step has already been applied. So as to avoid confusion, normalized images will be formulated as I, rather than $\hat{I}$, unless otherwise noted.



(a) **(b)**

(c) **(d)**

Figure 2.5: Phases of the geometrical normalization given two initial points and their correspondent location in the transformed image (i.e. the eyes). We have marked in red the horizontal reference and in green the inter-ocular segment. **a)** Original image, **b)** rotation of the image, **c)** image scaling and **d)** translation and cropping of the image.

On the other hand, when working with color images, normalizing in intensity does not make much sense. Instead, the individual stretching of the color channels could be of greater utility. To this end, the same steps used to stretch the intensity histogram are instead applied to the histograms from channels $C1$, $C2$ and $C3$. This normalization process remains independent of the color model used for representation. Henceforth, it will be remarked when the proposed color solution has been pre-processed in this fashion.

2.4.3 Normalization for Experiments

Having presented the geometric and intensity normalizations that comprise the *face normalization* stage, this final section details how faces for all experiments will be normalized.

Figure 2.4 presents the order of the various normalizations, beginning with spatial normalization and followed by the contrast enhancement. This order must be maintained, as contrast enhancement is only effective when applied to the cropped sub-image of the face, thereby avoiding noise in the intensity histogram due to background measures.

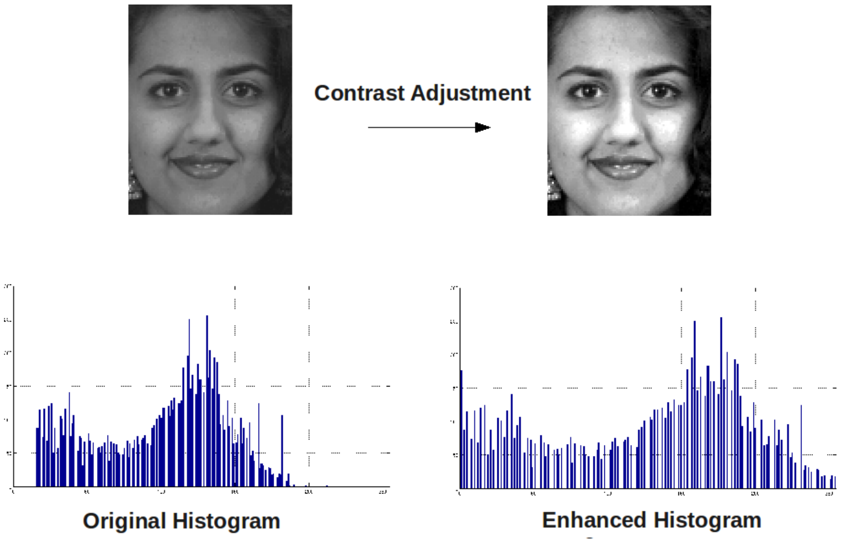


Figure 2.6: Example of the effects of normalization (contrast stretching) over images for face recognition.

As mentioned above, for all experiments where face normalization is required, the exact location of both eyes is assumed to be known *a priori*. This simplifies the solution for the transformation matrix since only then it deals with scale and rotation variations. Chapter 5 presents a novel technique to locate the eyes based on the extraction of local features. Notice that small variations on the localization of the eyes would produce deformations on normalized faces, similar to those shown in Figure 5.15 of the aforementioned chapter. Some experiments in face recognition performed in Chapter 6 show that minor displacements on the position of the eyes (with a subsequent bad normalization stage) produces great variance in recognition rates[5].

The normalized face selected for the experiments will thus correspond to a cropped image of size 120×160 pixels, in which the left and right eyes are located at the pixel coordinates $c_{left} = (40, 80)$ and $c_{right} = (80, 80)$ respectively.

The normalized faces showed in Figure 2.4 are a example of a face normalized using the aforesaid normalization process.

As it will be detailed in future chapters, this normalization stage not only resolves the problem of representation, but it can be of particular importance for extracting reliable local features.

[5] The experiment mentioned can be found in Section 6.8

Chapter 3

Evaluation Methodology in Facial Analysis

The foremost aim of this work is to study a novel set of theoretical methods that address the topics of of face detection and face recognition. Moreover, it is important to implement these methods and then assess their impact compared to other state-of-the-art algorithms within the same field of research.

To this end, this chapter provides an overview of the evaluation methodology that will be employed in the experiments carried out throughout the rest of the work. Furthermore, the tools required in the practical analysis of the proposed algorithms will also be defined.

The current chapter is structured in three blocks. It starts with a brief introduction concerning the generic problem of classifying objects, followed by the evaluation methods needed for the two tasks of this work: eye location for precise face detection and feature-based face recognition.

Finally, a detailed explanation of each of these topics is presented, including a technical description of the tools used in each case.

3.1 Introduction

The development of new algorithms is always the confluence of a good theoretical base and a valuable context for experimentation; together, they enable us to discover the principal properties of a new approach. From an experimental point of view, developing a technique necessarily entails evaluating its performance and establishing a framework wherein the technique can be compared to that of other approaches.

The selection of appropriate evaluation methods provides greater understanding of the developments under examination. In this work, two different fields of facial analysis are studied: the precise detection of faces performing eye location (discussed in Chapter 5) and the recognition of faces, for both identification and verification purposes (presented in Chapter 6). One might think that two different evaluation methodologies are required –one for each of the problems proposed– but this is not. In this work, solutions are offered that rely on supervised learning classification approaches[1], but focused differently. In the case of face detection, the task of classification is framed to address a binary problem, separating *faces* from *non-faces*. This problem can be adapted to detect any other facial feature, such as the eyes. In face recognition, the classification stage is more complex, as it has to solve a multiple-class problem (i.e. to differentiate all subjects in a database).

In the following sections, the general problem of object classification is reviewed from a perspective useful for both binary and multiple-class problems. An additional problem-specific methodology applicable to face detection and face recognition tasks will also be presented and evaluated.

- In face detection algorithms, the problem-specific methodology proposed is aimed at defining an accuracy measure of the locations of the eyes.
- In face recognition algorithms, we have to emphasize the different needs for the problems of face identification and face validation. Then we propose some tools that can be specifically used for each case.

3.2 Methodology for Classification Algorithms

All the facial analysis algorithms studied herein can be regarded as supervised classification methods and are treated as such. Therefore, the evaluation methodologies derived from the field of object recognition can also be applied. To evaluate a classifier, this works subscribes the evaluation methodology proposed for the database of FERET [93], where the validation and the test sample sets are each organized in two subsets:

- **target set**: A collection of labelled samples, where each label refers to the class L to which each sample belongs. In face recognition, the target set is composed of images of all *known individuals* against which testing images are subsequently matched. In this case, each subject contains at least one image in the target set. The literature frequently refers to a *gallery* set as a subset of the target set. In this work, both terms will be henceforth used interchangeably.
- **query set**: A collection of unlabelled test samples whose class is not known a priori. In supervised algorithms (with a training set), the test set does not include any of the samples used during the training. In face recognition, the query set is made up of images of unknown individuals to be recognized. A *probe* set is often referred to as a subset of the query set. In this work, both terms will be employed interchangeably throughout.

When a classifier is evaluated, the query set samples must match up with the samples belonging to the target set. To achieve a complete and methodical process for matching, the following important issues must be taken into consideration:

1. **Selection of similarity distances**: With appropriately selected distances, one can measure the degree of resemblance between any two matched samples (one each from the target and query sets). Note that this issue is related to a stage preceding classification.

[1] As detailed in Chapter 1

2. **Organization of the samples**: Samples are organized so as to extract subsets for the training, validation and test phases of the supervised algorithms[2]. Sample organization and the application of a similarity measure to the matches together yield the so-called *distance matrices*. These matrices align all samples from the target set (typically, the columns of the matrix) against all samples from the query set (typically, the rows of the matrix).

3. **Methodology to interpret the results**: With this, one can extract from the distance matrices the information needed for classification . To do this, all possible classification cases summarized in the *confusion matrix* must first be reviewed.

4. **Study of the performance evaluators**: These evaluators are directly derived from the classification cases presented in the confusion matrix. The measures extracted will serve as tools to understand and determine the performance of the classifiers presented in this work.

The remainder of this section focuses on addressing the four aforementioned topics in greater depth.

3.2.1 Similarity Measures

For every classification problem, the matching step is usually reduced to an evaluation of a distance matrix, M, based most of the times on the *nearest neighbor* approach, although other techniques are also possible. In the distance matrix, the element M_{ij} corresponds to the similarity (the distance) between a query sample i, represented by the i-th row of the matrix, and a target sample j, represented by the j-th column of the matrix.

In the specific case of facial analysis, matching is performed to assess the recognition results from identification and validation. The similarities are calculated in what is called the *feature space*. This space contains all of the vectorized target and query samples. They represent the face feature vectors obtained following post-processing (such as normalization or dimension reduction). Given a feature space, there is no unique similarity measure that optimizes all classification problems. Therefore, for every new approach, the most appropriate measure must be evaluated and selected with care from amongst various options. Given two final feature vectors belonging to the query and target sets, $x_i = [x_i^1, x_i^2, \ldots, x_i^k, \ldots, x_i^N]$ and $y_j = [y_j^1, y_j^2, \ldots, y_j^k, \ldots, y_j^N]$ respectively, where N is the dimension of the feature space, the following similarity measures become available:

- **Euclidean distance** - The shortest distance between two points, per Pythagorean geometry. Also, the most commonly used distance measure.

$$d_{Euc}(x_i, y_j) = \sqrt{\sum_{k=0}^{N} (x_i^k - y_j^k)^2} \tag{3.1}$$

The Euclidean distance measures the range $R_{Euc} = [0, \infty[$.

[2] See Chapter 1

- **Cosine similarity** - Measure based on the calculation of the angle formed by the two feature vectors, Θ_{ij}. Instead of a raw angle, this similarity uses the cosine as more general measure. The cosine distance is commonly used as a cohesion measure within clusters, which is especially useful in tasks such as classification or data mining. It is defined as:

$$d_{cos}(x_i, y_j) = cos(\Theta_{ij}) = \frac{x_i \cdot y_j}{\|x_i\|\|y_j\|} = \frac{\sum_{k=0}^{N} x_i^k y_j^k}{\sqrt{\sum_{k=0}^{N} (x_i^k)^2}\sqrt{\sum_{k=0}^{N} (y_j^k)^2}} \tag{3.2}$$

  The cosine distance measures in the range $R_{cos} = [-1, 1]$, where 1 and -1 indicates the same direction, while 0 indicates an orthogonal direction.

- **Manhattan distance** - Suitable for discrete data measurements, as it represents the rectilinear distance between two points measured along axes at right angles. The expression of the Manhattan distance is:

$$d_{Mn}(x_i, y_j) = \sum_{k=0}^{N} |x_i^k - y_j^k| \tag{3.3}$$

  The Manhattan distance measures in the range $R_{Mn} = [0, \infty[$.

- **Chebyshev distance** - The longest distance between the vector elements. This measure is particularly useful when computation time is absolutely imperative:

$$d_{Ch}(x_i, y_j) = max_k |x_i^k - y_j^k| \tag{3.4}$$

  The Chebyshev distance measures in the range $R_{Ch} = [0, \infty[$.

- **Minkowski distance (of order m)** - A generalized metric, commonly used with ratio scales (i.e. when an absolute zero is present). The general expression of this distance is:

$$d_{Mi}(x_i, y_j) = \sqrt[m]{\sum_{k=0}^{N} (x_i^k - y_j^k)^m} \tag{3.5}$$

  Regarding this expression, note that the Manhattan, the Euclidean and the Chebyshev distances are specific cases of the Minkowsky distance for values of m equal to $m = 1$, $m = 2$ and $m \to \infty$, respectively. This distance additionally measures in the range $R_{Mi} = [0, \infty[$, independent of the value of m.

- **Mahalanobis distance** - A measure of the divergence between groups in a feature space, in terms of the variation along each of the dimensions. In other words, the Mahalanobis distance can be expressed as the distance between two N dimensional points, scaled by the statistical variation in each point component. Thus, this similarity measure takes into account the covariance matrix Σ of the distribution of samples:

$$d_{Mh}(x_i, y_j) = \Delta_{ij} = \sqrt{\sum_{k=0}^{N} (x_i^k - y_j^k)^T \Sigma^{-1} (x_i^k - y_j^k)} \tag{3.6}$$

> It can be seen that since Σ is a non-singular covariance matrix, it is positive-definite and hence Δ_{ij} is a metric, in the range $R_{Mh} = [0, \infty[$. It is likewise worth remarking that if the variables in the distribution were completely uncorrelated and normalized, then Σ would be the identity matrix and the Mahalanobis distance would correspond to the Euclidean distance.
>
> The Mahalanobis distance has played a fundamental and important role in statistics and data analysis, specifically in the fields of classification, numerical taxonomy and statistical pattern recognition.

Depending on the specific problem, different similarity measures provide different results. After evaluating several of these measures as part of the present work, we decided to use *Euclidean distance*, *cosine distance* and *Mahalanobis distance.* This selection was motivated by the fact that these distances have become a *de facto* standard to compare different approaches for facial analysis, specifically for face recogntion.

3.2.2 Distribution of the sample datasets

In supervised approaches, sample classification is performed as a three-step process: the *training phase*, where the algorithm learns various classification rules, the *validation phase*, where these learned rules are verified and tuned, and the *test phase* where the rules are applied to a novel set to extract the performance results.

Each step necessitates different sets of samples, with specific features and requirements in each set:

- As part of the training phase, the set of training samples is organized so as to retain information useful for learning discrimination rules. Depending on its implementation, this set of samples can be gallery images or an independent set.
- During validation and test, sets are perused for one large and varied enough to extract useful performance information. As explained earlier, the validation and test sets are both divided into target and query subsets.

When evaluating and testing a classifier, it is sometimes more appropriate to employ numerous, smaller sets rather than one large set. However, given a sufficiently large set of samples, some techniques can be applied to obtain a methodical division into multiple subsets. Two classical examples of these techniques are the *Holdout Approach*, in which the sample set is randomly split [87], and the *Cross Validation* (CV) [56], which splits the sample set according to a predefined criterion.

Cross-Validation offers a means of obtaining a more reliable evaluation when faced with scarce data. As part of the *k-fold CV* approach, the database is randomly divided into k disjoint blocks of samples, usually of equal size. The classifier is trained using $k - 1$ blocks (training set), with the remaining block serving as the test set. In the training set, at least one representative sample from the test classes must be included. This process is iteratively repeated for each of the k blocks, and the final results obtained from each iteration are subsequently averaged. A useful variant of the *k-fold CV* is the *leave-one-out*

CV, which selects the number of blocks, k, equal to the number of samples in the original set. This approach leaves just one sample in the test set, thereby maximizing the number of iterations.

The classification process for a dataset using a *3-fold Cross Validation* approach is depicted in Figure 3.1.

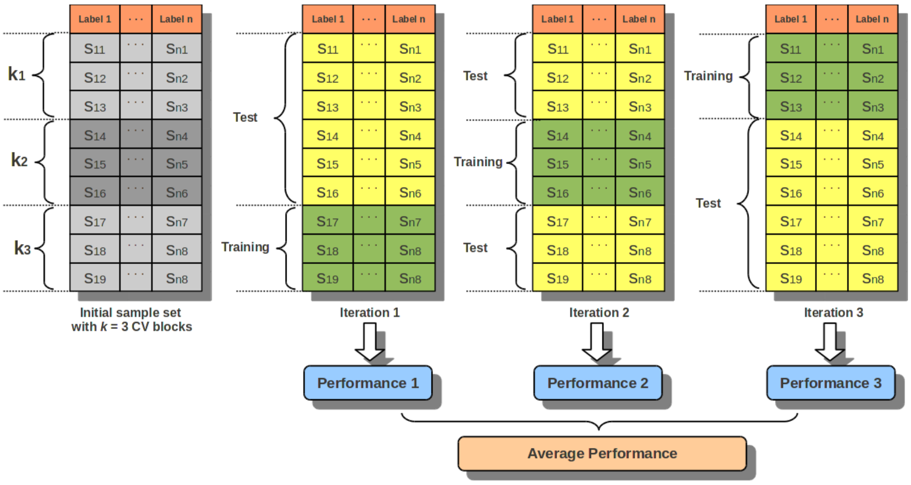


Figure 3.1: Evaluation of $k = 3$ blocks using 3-fold Cross Validation. Initial sample dataset organized in n classes.

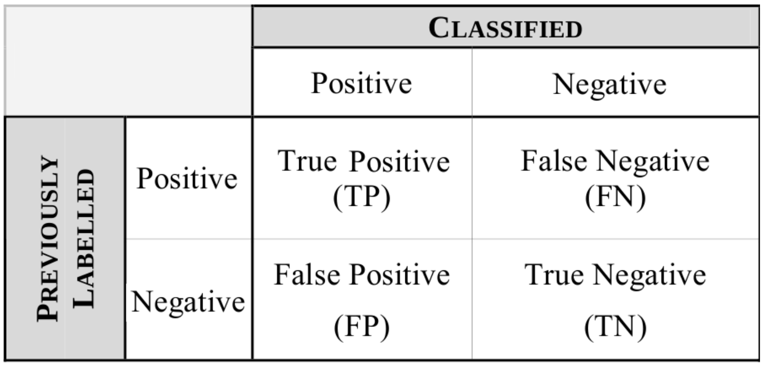
3.2.3 Binary Classification Cases

Prior to classifying test samples, a precise set of terminologies must first be defined for all potential classification cases. Specifically, the true class of a sample (given by the *actual label*) must match the class assigned to it after processing (i.e. *classification label*). To this end, a dichotomous classification problem will be employed where a sample can belong to only two different classes: class $C = 1$ if the sample is in a specific group we are looking for (e.g. the group *faces*), and class $C = 0$ if the sample belongs to any other group (e.g. the group *non-faces*). For simplicity, $C = 1$ is labelled as *Positive*, and $C = 0$ as *Negative*. This terminology can be easily extended to the multiple-class problem, where $C = 1$ if the sample belongs to our target group and $C = 0$ when it belongs to any of the remaining classes.

The cases associated with the binary classification problem can be summarized in the so-called *confusion matrix*, shown in Table 3.2.3. From the confusion matrix, four possible matching cases are derived:

- **True Positives (TP)**: All samples belonging to the target class and are correctly classified by the detector. In face detection, this corresponds to a real face labeled as such. In face recognition, a true positive is given when the classified identity of an image corresponds to its real identity.

- **True Negatives (TN)**: All samples which do not belong to the target class (non-objects) and are correctly classified as non-objects by the detector. In face detection, this corresponds to a *non-face* labelled as such.



		CLASSIFIED	
		Positive	Negative
PREVIOUSLY LABELLED	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 3.1: Terminology of the classification problem.

In face recognition, a true negative is given when a person is verified as not corresponding to a certain identity, corroborated by the real data.

- **False Positives (FP) or False Alarms (FA)**: All samples which do not belong to the desired class (non-objects) but are classified as belonging to the target class. In face detection, this corresponds to a *non-face* labelled as a *face*, producing a *false alarm.* In face recognition, a false alarm is given when a person is verified as belonging to a certain identity, when in fact, it does not. This specific error is called *false acceptance.* This case is quite common, especially in detection tasks, and pre-processing is generally aimed to avoid them.

- **False negatives (FN)**: All samples which do belong to the target class but are classified as non-objects. In face detection, this corresponds to a *face* labelled as a *non-face*, which is considered a miss. In face recognition, a false negative is given when a person is verified as not belonging to a certain identity, when it does. This specific error is also called *false rejection.* The aforementioned error types represent the least desirable cases as they diminish the performance of the algorithm. Therefore, the different techniques in face recognition attempt to minimize them.

This nomenclature also remains valid for classification in multiple-class problems, as these definitions are directly applicable. In the next point, various elemental measures that can be derived from these four matching cases are defined.

3.2.4 Basic Classification Measures

From the parameters presented in the confusion matrix, some fundamental performance evaluators can be defined, which help determine the correctness of a resultant classification:

- **True Positive Rate (TPR)** - Also known as *Recall*, *Sensitivity* or simply the *Hit Rate.* The expression of the TPR is:

$$TPR = R = sensitivity = \frac{TP}{TP + FN} \tag{3.7}$$

- **False Positive Rate (FPR)** - Also known as *False Acceptance Rate.* It is a measure usually combined with the *True Positive Rate*:

$$FPR = FAR = \frac{FP}{TN + FP} \tag{3.8}$$

- **True Negative Rate (TNR)** - Also known as *Specificity*, this measurement is combined with *Sensitivity*:

$$TNR = Specificity = 1 - FPR \tag{3.9}$$

- **Precision (P)** - Commonly combined with *Recall*:

$$P = \frac{TP}{TP + FP} \tag{3.10}$$

- **False Rejection Rate (FRR)** - The measurement of the probability a biometric system will fail to identify an individual who is properly enrolled. This measure is combined with the *False Acceptance Rate*:

$$FRR = \frac{FN}{TP + FN} \tag{3.11}$$

Defining these basic measures offers sufficient information to evaluate a classifier. However, sometimes the validation of face detection and face recognition processes needs more specific and complex information. Hence, the remainder of this chapter is devoted to presenting some problem-specific evaluation tools frequently found in face detection and face recognition literature.

3.3 Evaluation Methodology for Face Detection with Eye Location

The need for developing effective and reliable evaluation methodologies for face detection has led to increased interest in discovering suitable tools for measuring the accuracy of the location of the eyes. Those face detection methods based on the precision of the location of the eyes become suitable for a general approach to the majority of the algorithms developed. Jesorsky *et al.* [51] propose measuring the error generated when the located coordinates of the eyes are compared with the actual coordinates in the ground-truth data[3].

Jesorsky's definition is valid and quite simple, but before using it some clarifications must be made. The error measure is performed independently of the location algorithm used, and this measure is quantified in reference to the *inter-ocular distance*, *iod*, which is the Euclidean distance between the eye centers, in pixels.

The error of the location, N_{error}, is defined for each of the eyes independently, as delineated in the following expression:

[3] It should be noted that the studies pertaining to eye detection aim to fully automatize their location; consequently, to set the ground-truth data still requires marking all eye positions by hand.

$$N_{error} = \frac{|E_{det} - E_{gt}|}{iod} \times 100, \quad (3.12)$$

where E_{det} is the automatic position of the eyes generated by the locating algorithm under evaluation, and E_{gt} is the actual location of the sample eyes in the ground-truth. In the literature, there is common agreement [51, 68, 130] that an eye is considered to be correctly located if the error is $N_{error} < 25\%$. This error corresponds approximately to a distance smaller than the eye socket, as shown in Figure 3.2. However, this level of precision might be insufficient for some tasks (as is often the case in face recognition), so authors have worked with precisions of 15%, 10% or even 5% of the inter-ocular distance.

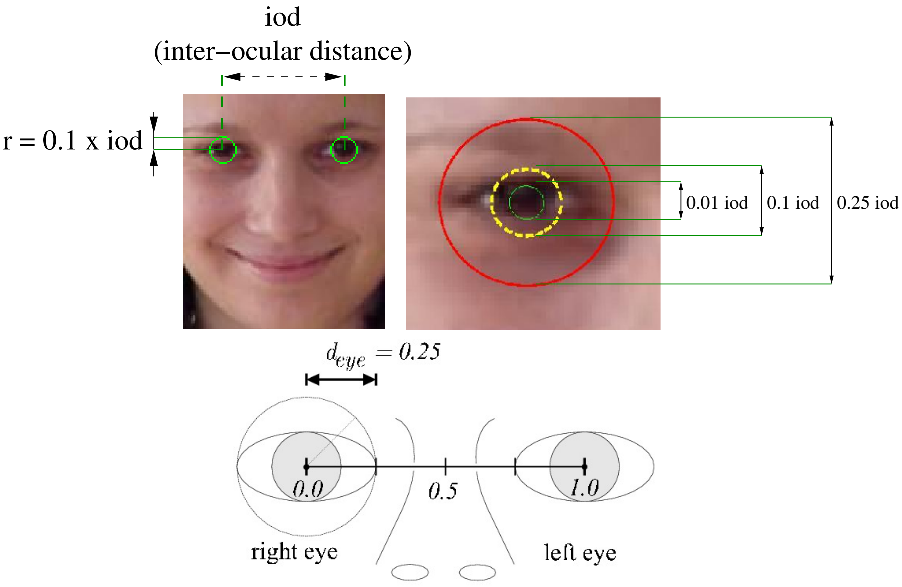


Figure 3.2: Error distance for the location of the eyes, based on the inter-ocular distance, *iod*.

Recent works have developed more accurate methodologies to measure the eye location errors. In [98], the authors propose breaking down the location error into four different error types: horizontal, vertical, scale and rotation. Based on the configuration shown in Figure 3.3, these errors are defined as:

$$\Delta_x = \frac{dx}{\|C_l - C_r\|}(horizontal), \Delta_y = \frac{dy}{\|C_l - C_r\|}(vertical), \quad (3.13)$$

and

$$\Delta_s = \frac{\|\tilde{C}_l - \tilde{C}_r\|}{\|C_l - C_r\|}(scale), \Delta_\alpha = \widehat{\overrightarrow{C_lC_r}\overrightarrow{\tilde{C}_l\tilde{C}_r}}(orientation), \quad (3.14)$$

where all the variables in the expressions, $dx, dy, C_l, C_r, \tilde{C}_l, \tilde{C}_r$, refer to their homonyms in Figure 3.3.

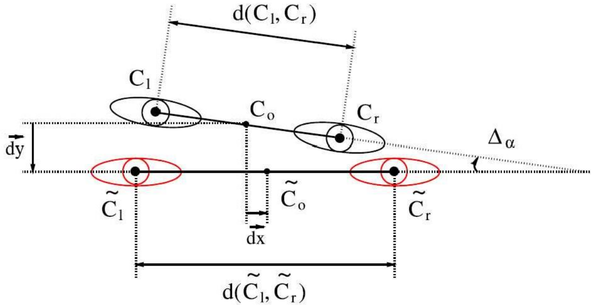


Figure 3.3: Measures for calculating errors in eye location.

These error measures can be of immense utility if, for example, it is reckoned that some location algorithms might perform reasonably well for some of the errors, but not as well for others. Despite their clear advantages, few studies have been found that make use of them and therefore they are not useful to compare different approaches.

3.4 Evaluation Tools for Face Recognition

Face recognition is a specific case in the more general object classification problem. This means that all the evaluation tools described at the beginning of this chapter apply to this context. However, a number of problem-specific tools can also be employed efficaciously, as will be addressed in this section.

To start, it should be taken into account that face recognition can actually be presented as two-facet problem: face identification and face verification. The tools that optimize the evaluation process of a face recognition system depend on which of the two problems we are attempting to solve.

This section is divided into a description of the differences between face identification and face verification, the definition of some classification error measures and the presentation of three relevant tools for performance evaluation: rank curves, ROC curves and DET curves. All of these operators are incorporated into the various experiments carried out throughout this work.

3.4.1 Distinction between Face Identification and Face Verification

It has already been mentioned that all face recognition systems can be evaluated as two different cases: face identification and face verification. Each of these topics is described in greater detail below:

- **Face Identification:** Given a test face image belonging to the query subset, the recognition system assigns it the most probable class, comparing it to each of the target images. To illustrate this scenario, let T represent a target set of individuals, with N classes, $\mathcal{T} = [\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N]$, where each class $\mathcal{I}_i$ represents a different individual. Then, given a query image

q_i, and a matching evaluator between two samples $dist(q_i, \mathcal{I}_j)$, the class of the test sample can be defined by:

$$class(q_i) = min_j(dist(q_i, \mathcal{I}_j)), 1 \leq j \leq N. \tag{3.15}$$

Satisfactory results in face identification are achieved when the distance between the predicted and the actual classes is less than that of the remainder of the classes.

- **Face Verification:** Given a claimed identity from the *target set* and face image belonging to the *probe set*, the recognition system has to determine whether the probe sample belongs to such identity or not. Verification experiments entail greater complexity than pure identification tests. This is due to the fact that identification tests determine the best match by evaluating each query face, but using the information from the whole gallery; whereas in verification experiments, each match is performed using only the information from the images belonging to the claimed identity.

  In verification, a good result is given when there is a true match of the query sample with the claimed identity, and also there is a true negative match for the resting labels in the target set. To illustrate this case, let T represent a target set of individuals, with N different labels, $\mathcal{T} = [\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N]$, where each label $\mathcal{I}_i$ belongs to a different individual. Then, given a query image q_i, which actually belongs to the class $\mathcal{I}_i$, one would expect the verification algorithm to exhibit the following behavior:

$$verification(q_i, \mathcal{I}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{3.16}$$

3.4.2 Error Measures for Verification Problems

With a verification system, two situations can occur with the person being verified: the identity claimed corresponds to the individual true identity, or, the identity claimed by the person is not the true identity. In the first case, the verified person is called a *client*, while in the second, the individual shall be known as an *impostor*.

Thus, the system may generate two types of errors:

- **False Acceptance**: When the system accepts an impostor. The *False Acceptance Rate* is defined as the number of false acceptances over the number of impostor accessed. This rate yields the percentage of impostors that are considered clients.

$$FAR = \frac{\text{false acceptances}}{\text{number of impostors}} \tag{3.17}$$

- False Rejection: When the system rejects a client. The *False Rejection Rate* is defined as the number of false rejections over the number of client accessed. This rate yields the percentage of clients that are considered as impostors.

$$FRR = \frac{\text{false rejections}}{\text{number of clients}} \tag{3.18}$$

Note that both errors depend on the threshold used to separate clients and impostors. To evaluate the performance of a face recognition system performing verification tasks, two operators derived from these errors are commonly used. One is the Half Total Error Rate (HTER), which is an evaluation methodology that combines the two types of error rates: the *False Rejection Rate* (FRR) and the *False Acceptance Rate* (FAR). The HTER operator is defined as:

$$HTER(\tau, \mathcal{X}) = \frac{FAR(\tau, \mathcal{X}) + FRR(\tau, \mathcal{X})}{2}[\%], \tag{3.19}$$

where $\mathcal{X}$ denotes the dataset of samples that is used, and τ is an arbitrary threshold. Since both the FAR and the FRR depend on the threshold τ, they are closely related to each other: increasing the FAR will reduce the FRR, and vice-versa. Due to the complexity of choosing the optimal threshold, a variant of the HTER operator is employed in this work, the Equal Error Rate (EER). This operator is the HTER defined at the threshold τ that generates the same value for both errors, FRR and FAR.

Both operators are related to DET curves (explained below). Specifically, the EER operator corresponds to the point of the DET curve which intersects with the line $FRR = FAR$. This can be seen in Figure 3.6[4].

3.4.3 Rank Curves

Rank Curves, also known in literature as *Cumulative Match Scores* [93], constitute a basic tool for classification problems. These curves are useful for determining not only the best match (i.e. the predicted class) of a test sample, but also the top n matches (i.e. the n classes closest to the sample). In this work, rank curves become of particular interest when validating face recognition problems related to identification.

Given a classification problem such as facial identification, every testing sample is compared with the class models, in accordance with Equation (3.15). From this equation, it is assumed that a smaller similarity score implies a closer match of the predicted label.

In this case, a true positive is given if the actual class of the test sample corresponds to the class j that has the closest distance $dist(q_i, \mathcal{I}_j)$. Rank curves can broaden this concept. If the distances in Equation (3.15) are sorted in increasing order, and then plotted in the form of a vector of distances,

$$\mathcal{D} = [dist_{min}(q, Gallery), dist_{min+1}(q, Gallery), dist_{min+2}(q, Gallery), \dots] \tag{3.20}$$

A true positive with rank $k = 1$ can then be defined if the *actual class* of the sample is contained in the unitary set,

$$\mathcal{R}_1 = \{dist_{min}(q, Gallery)\} \tag{3.21}$$

The same occurs for a true positive with rank $k = 2$ if it is contained in the binary set,

$$\mathcal{R}_2 = \{dist_{min}(q, Gallery), dist_{min+1}(q, Gallery)\} \tag{3.22}$$

[4] In page 38.

And more generally, a true positive with rank $k = n$ will occur if the *actual class* of the test sample is contained in the set,

$$\mathcal{R}_n = \{dist_{min}(q, Gallery), dist_{min+1}(q, Gallery), \ldots, dist_{min+(n-1)}(q, Gallery)\} \tag{3.23}$$

which means that the probe sample q is one of the nth smallest scores for the Gallery.

The rank curve plots the hit rate obtained from true positives with rank $k = 1, 2, \ldots, n$. In other words, it measures the cumulative percentage accuracy that can be attained by a classification system when the first k matches of the test image are considered relative to all gallery images (which correspond to the identities in face recognition problems). Due to its cumulative nature, a rank curve is, by definition, monotonically increasing. It starts in the null value for rank $k = 0$ (null distance set, $\mathcal{R}_0$), increases through the standard classification hit rate for $k = 1$ and tends towards an accuracy of $hitrate = 1$ as the rank $k = n$ approaches the total number of gallery images.

Rank is a reliability measure and it is held to be very important to video-surveillance applications in uncontrolled environments. In indexed video sequences, rank helps reduce the amount of information needed to be retrieved when performing specific queries. A clear example of this is a scenario wherein a video sequence contains labels for the identities of all the people prior to performing a search. If only one rank is considered, $k = 1$, the probability of finding that specific person (meaning the matched person is exactly the one searched for) is always lower than if a higher rank, e.g. rank $k = 5$, is selected (meaning the person does not necessarily need to be found in the first match, but rather within the top five matches).

Figure 3.4 shows an example of a Cummulative Matching Score. Note that the rates corresponding to some of the first ranks (i.e. $r = 1, r = 2, r = 5, r = 10$) have been also indicated.

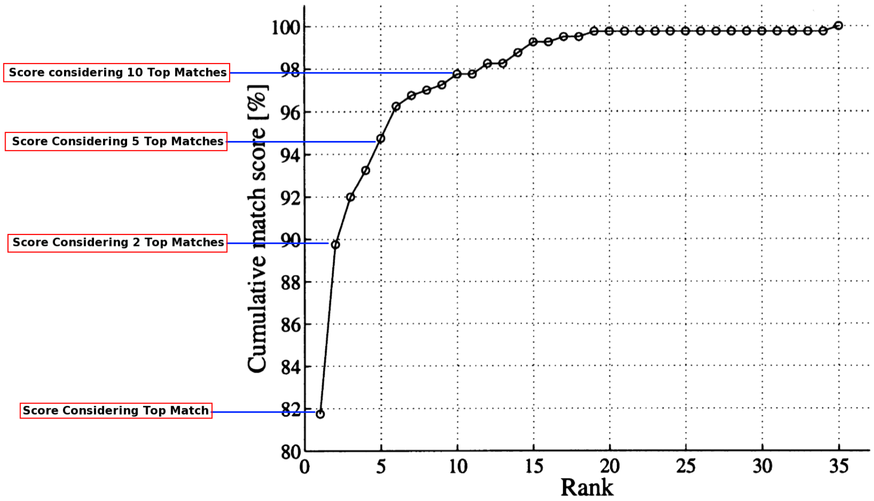


Figure 3.4: Example of a Rank Curve indicating the cummulative rate on some of its first ranks.

3.4.4 Receiver Operating Characteristic - ROC curves

The *Receiver Operating Characteristic* (ROC) analysis [116] has become one of the most powerful tools in the field of statistical decision theory in recent years. Originally applied to radar-image analysis around the 1940s, its versatility soon inspired its application to a number of disparate fields: medical test diagnosis, particularly in radiology and imaging; classifier assistance in machine-learning applications; and discriminate effects evaluation for different procedures.

The ROC of a classifier represents a trade-off between its *specificity* and its *sensitivity*, measured by the True Positive Rate (TPR) and the False Positive Rate (FPR). In other words, the ROC may be considered a measure of the hit and false alarm rates of the systems evaluated. In parametric classification systems, the TPR and the FPR vary as a function of a tolerance threshold that separates positive from negative samples (clients and impostors, in the case of face verification).

The performance of each system varying this threshold is represented by a (FPR, TPR) pair. The ROC curve is the graph generated when these pairs are concatenated. The study and subsequent analysis of the ROC curves provides the information necessary to determine the effectiveness of a system.

Therefore, a ROC curve essentially plots the FPR (i.e. $1 - specificity$) values on the X axis and the TPR (i.e. *sensitivity*) values on the Y axis, thus combining the most relevant information from the confusion matrix. In ROC curves, three (FPR,TPR) pairs become particularly relevant:

- **(FPR,TPR) = (0,0)**: Implies that all the objects of the database are classified as negatives, independent of their actual class.

  This means that the threshold was extremely stringent, and therefore no alarms are detected.

- **(FPR,TPR) = (1,1)**: Opposite to the $(0, 0)$, this point is achieved when all the samples are classified as positive, raising false alarms for all negative samples. This indicates that the threshold was extremely imprecise.

- **(FPR,TPR) = (0,1)**: The ideal classification performance. All the positives are successfully classified, while all the negatives are rejected.

All the points in an ROC curve are bounded by these special cases. The worst scenarios –a detector that randomly classifies the objects as positive or negatives– is displayed on the ROC curve as a straight line running from $(0, 0)$ to $(1, 1)$. Every system differing from that of a random classification has an ROC curve that consistently falls above this line. The curve of the optimal classifier would be that which has $TPR = 1$ for all FPR values, while the worst case would be the random classifier. A classifier performing worse than random guessing (i.e. with an ROC curve below the random classifier line), would simple swap positives for negatives, thereby reversing the ROC curve of such a classifier and turning it into a regular one.

In its left-hand image, Figure 3.5 reveals an example of three significant ROC curves: the best case, a standard case and the worst case.

When different classification systems are compared, ROC curves become quite useful. From an examination of an ROC curve, we know that the upper areas of the graph indicate greater discrimination power, and therefore greater

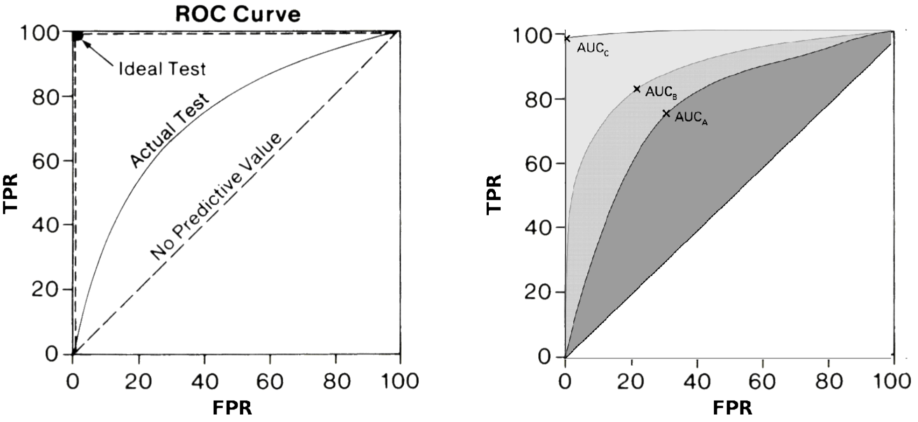


Figure 3.5: Examples of the most significant ROC Curves and the measure of the area under the curve (AUC) for three ROC implementations.

effectiveness. A system with higher values in its ROC curve indicates better performance. Figure 3.5 compares two detection systems and their respective performance relative to the worst and ideal cases.

In addition to the previously mentioned visual analysis of graphs, it is worth mentioning that researchers have established some analytical methods for measuring the goodness of a system based on its ROC curve.

Essentially then, two methods gain importance. The first consists of measuring the shortest distance between the points of a curve and the (FPR, TPR) point equal to $(0, 1)$. The shorter this distance, the better the performance of the system under evaluation.

The second method was proposed by [41] and consists of measuring the accuracy of the detector using the area under its ROC curve (AUC method). The area of the ROC curve of a standard classifier will achieve values between 0.5 (corresponding to the random classifier) and 1 (the area reached by ideal classifiers).

Those detectors whose area approaches that of the ideal value will logically perform better. In the right-hand image in Figure 3.5, an example of three ROC curves can be observed where the different areas are colored in grayscale. An 'X' also marks the optimal operation point for each curve (i.e., the point closest to the $(0, 1)$).

Regarding the goals of this work, ROC curves are utilized to evaluate face recognition problems, specifically those related to face verification. Also, researchers agree that measuring the performance of an algorithm with a false acceptance rate value of 10^{-3} usually constitutes a good working point for the algorithm.

3.4.5 Detection Error Trade-off – DET curves

The Detection Error Trade-off curves are a common tool used to represent the performance of any classification algorithm. This family of curves is a variant of ROC curves and its use has become more extensive in the field of biometric analysis due to its efficacy in identity validation. DET summarizes the verification performance of the biometric algorithm on the sample population for which it is calculated. Also, DET curves are related to such error measures as HTER and EER, explained in Section 3.4.2.

Given two sample subsets containing images of matching (client) and non-matching (impostor) identities, for every image pair a match score is obtained, t. This score is then used to estimate the match score distributions for the client, $c(t)$, and the impostor, $i(t)$. From these score distributions, the DET curve typically plots error rates on both axes, uniformly treating both types of error. The errors represented are the False Acceptance Rate (FAR) on the x-axis, against the False Rejection Rate (FRR) on the y-axis. In some face detection systems, one of the errors may have greater or lesser importance than the other.

In this sense, the DET curves help set the optimal working point. An example of a DET curve can be seen in Figure 3.6.

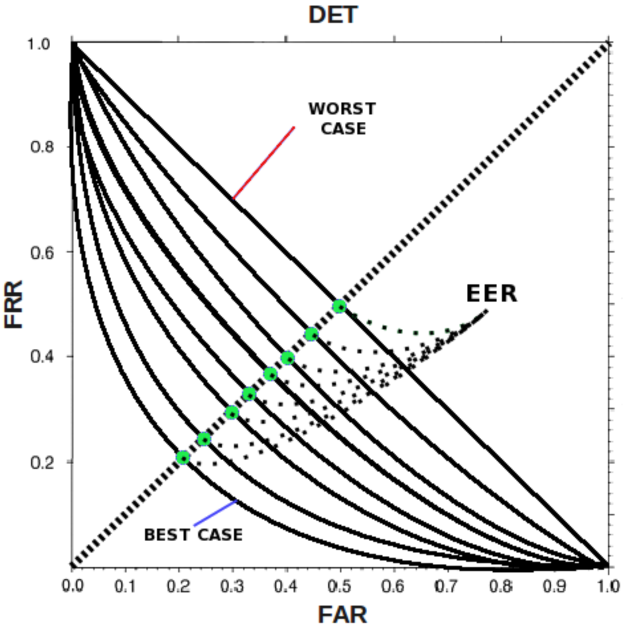


Figure 3.6: Example of various DET curves, with the Equal Error Rate Point marked.

The points on a DET curve are generated by varying a threshold, τ, and calculating the FAR and FRR rates from the match score distributions. As both axes of the DET represent errors, the optimal working point of the biometric system corresponds to that which minimizes them. That is, the closer the distance of an operating point to the (0,0), the better the system performance.

When comparing two DET curves that correspond to different systems, one can opt to compare only the difference in detection rates obtained for each of the systems, given a specific pair of errors. Alternatively, one can also evaluate them more globally by considering the area between curves. The greater the DET area between two curves, the greater the difference in system performance. The curve that delimits the bottom portion of this area is always the best performing.

Chapter 4

Texture Feature Analysis for Biometric Characterization

The process necessary to analyze, recognize or classify people with the information generated from their facial images is quite complex. Since individual faces are specific realizations of an object, they are subject to be described using specific features.

This chapter is focused on the study and analysis of the extraction of *local texture features* in order to perform face recognition tasks, specifically emphasizing our main contributions in this area. More in detail, the chapter exposes our motivation for the use of local texture features, specifically those based on local gradients, as is the case of the Histograms of Oriented Gradients (HOG) features. Then, the chapter makes an introduction to the texture features followed by a short state of the art of the current works on this topic. The following sections are aimed to describe some specific techniques, such as the Gabor Filters, and the HOG features, as a derivation of the the Scale Invariant Feature Transform (SIFT). The chapter ends with some experiments performed to set and validate the HOG features for facial analysis.

4.1 Motivation and Contributions

The study of local texture features has been an active topic for the last years and is still the target of several works. Depending on each specific classification challenge a different set of features may fit better. This thesis takes into consideration the works developed for other computer vision problems, to extract useful features that could be extrapolated to be used in facial analysis.

The main goal in this chapter is to study some relevant local texture features, some already extended and also new ones. Our goal is to adapt their use for the specific problem of the description of faces. Along the work developed in this thesis, local textures are used to describe facial elements, such as the eyes, the nose or the mouth.

This chapter has focused on the study of two local texture features, which in the text are also referred as *local descriptors* :

1. **Gabor Filters**: a classical feature descriptor, extended by Daugman in the late 80s [34]. This descriptor is based on the application of two-dimensional Wavelets and it is worth of study as it is related with the behavior of the human vision system. Due to its assessed good results in facial analysis, these features are a good baseline to compare to other approaches.

2. **Histograms of Oriented Gradients (HOG Features)**: a novel feature descriptor, based in Lowe's SIFT [65]. This approach makes use of histograms to extract and store the information relative to a region of an image. SIFT descriptors have been widespread for the last years, arising as a powerful tool that combines a structured approach with an more statistical point of view.

The novelty of the HOG features applied as local texture descriptors for facial analysis makes them an important topic of this thesis. The analysis of the HOG features for face description is a major goal in this work.

The motivation to study the local texture features can be summarized in the following points:

- Perform a theoretical study of the two widespread algorithms proposed: Gabor Filters and SIFT Transform (as the precursor of the HOG features).

- Perform a theoretical study of the HOG features as a novel descriptor. It is also important to analyze its relation and differences with the rest of the texture features.

- To validate the HOG descriptor and set the best configuration of its parameters in facial analysis. These experiments are described in the chapters dedicated to face detection and face recognition (Chapter 5 and Chapter 6, respectively).

This chapter is focused on the theoretical description of the Gabor Filters, the SIFT Transform and the HOG features.

4.2 Introduction and biological context of the texture features

In Computer Vision, some common elements such as faces have a complex nature, difficult to describe. To relieve this problem, it is good to examine the mechanisms and processes that enable us to extract descriptive information about such elements. A good approach is to understand how the biology deals with this task. Specifically, this work has addressed this issue studying with great interest the basic mechanisms of the human visual system.

In humans, the process to describe the objects using the visual system starts at the eye. The eyes sense the information provided by the light to build images using specialized cells in the brain. However, the light information alone is not enough as the brain is unable to directly infer the qualities of the objects or other contents that compose such images. Therefore, the impulses that contain the visual information are sent to specialized neurons, organized in what is called the *primary visual cortex* (area *V1*). The response of the neurons in V1 is activated when some low level characteristics of the visual stimuli are detected, such as lines, angles or edges. Making use of oriented patterns, spatial frequency or colors, among others, the rest of the layers in the brain dedicated to the visual sense function (also called visual areas *V2*, *V3*, *V4* and *V5*, respectively) are then able to interpret the contents of the image in higher level. Thus, after going through the visual area V1, the brain would be able to distinguish object, such as faces, in an specific image. Figure 4.1, shows the brain areas that are involved with the process of visual learning.

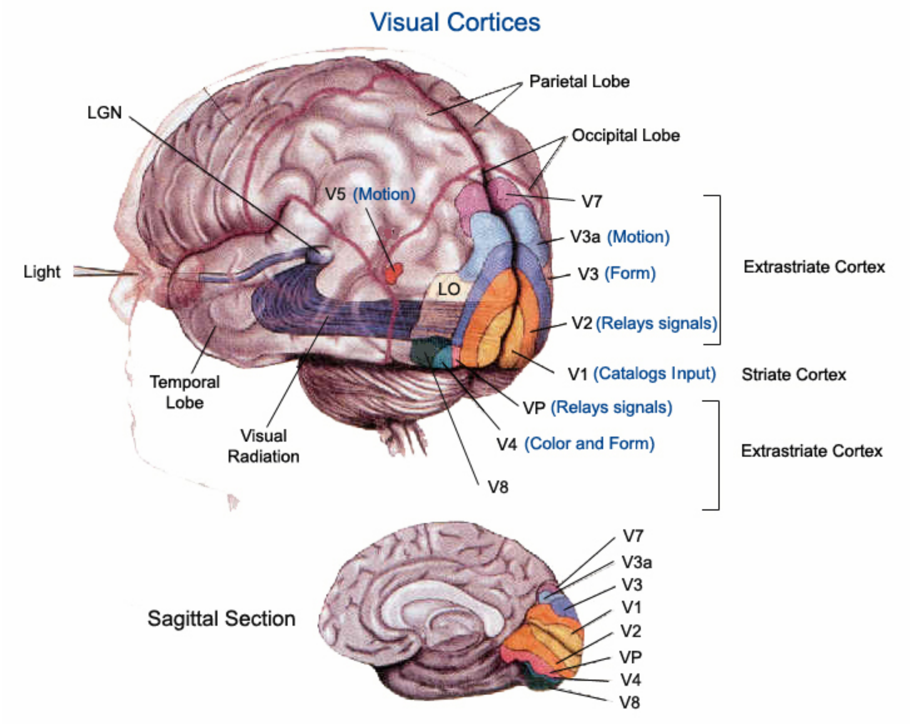


Figure 4.1: Global Scheme of the visual learning process and the brain areas that are involved with it. This image is taken from the book of R. Joseph [53].

In an analogue way to the process of visual understanding produced in the brain, when a computer describes an object from an image, the bare information contained at a pixel level is usually not enough; a processing of the intensities or even the color cues is a better source of data to describe complex objects. Following the ideas exposed in Chapter 2, human faces are 3D objects but the images captured are a 2D projection of them. The complexity of describing a face comes from the variability of the representation of such projections. Depending on the acquisition conditions and the context, a face can offer very different aspects.

The intensity levels directly represent a measure of the light that is reflected over the object without taking in account intrinsic characteristics such as its localization, scale or orientation. Despite the light is not a consistent descriptor, the problem of object description can be converted into one where the information is inferred from the intensity levels of the pixels. Such conversion is performed using algorithms that generate *image descriptors* (also called visual descriptors) and they become the link between pixel-based information and the intrinsic information of the object described.

As it can be seen from Figure 6.1[1], feature extraction is always an intermediate step between the image acquisition and the classification. Comparing the scheme the biological vision model, it can be observed a clear relationship between the functionality of the descriptors with the neurons present in the visual area of the cortex V1. A number of local descriptors can be defined, providing information about color, shape, movement or texture, among others.

In computer vision, the most extended features for facial analysis are those that provide information on the texture of the face. Local texture descriptors provide information of the objects at specific regions of the image. But, *what do we understand by texture?* Different definitions of texture based on the human perception are given: Haralick [42] analyzes it from an *structural approach*, defining a texture as an organised area which can be decomposed into primitives having specific spatial distributions; Cross *et al.* [33] studies it from a *stochastic approach*, defining a texture as a stochastic, possibly periodic, two-dimensional image field.

In this thesis, we understand that the texture information is defined by the repeating patterns of pixel intensities. Plain regions of an image (e.g. a plain wall) show few repetitive patterns and therefore low texture information (*inhomogeneous texture*). On the other hand, areas containing more shape repetitive patterns, produce more discriminative information (*homogeneous* and *weakly-homogeneous textures*). In Figure 4.2 an example of two images sorted regarding their texture information are shown.

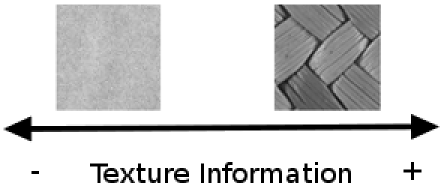


Figure 4.2: Example of three images with different degrees of texture information.

In this work, the analysis of descriptive features has focused on thoroughly studying local texture descriptors, used to describe faces in detection and recognition tasks.

4.3 State of the Art of Texture Descriptors

The description of image objects using texture features has been a vivid topic during the last years, and thus our research is directly related to some relevant works. In [106], Tuceryan *et al.* provide a classification of the texture feature

[1] See page 103.

extraction algorithms in four groups: *statistical*, *geometrical*, *model-based* and *signal processing*.

In facial analysis, the *statistical* and the *signal processing* methods can be unified. Both involve transforming original images using filters and calculating the energy of the transformed images. The *local texture features* can be considered as a part of these methods, as opposed to the *holistic features*. The main difference between the holistic and the local approaches is that the first ones consider the faces as a whole, while the second group uses information from local regions or keypoints within the face. Some typical landmarks are the eyes, the eyebrows, the nose, the mouth or the chin.

Initially, holistic features received more attention from the scientific community, mainly due to the fact that they are more easily computed. In the methods that extract global features, images are represented by a high-dimensionality vector containing all the values associated to the individual pixels (like the intensities or the color channels). Many of the holistic methods are focused on just simply reducing the dimensionality of this vector, trying to reject correlated or redundant information. Example of holistic methods are the Principal Component Analysis (PCA) [107], or the Linear Discriminant Analysis (LDA) [12]. As both methods are a specification of two general reductive methods, their implementation can be seen in Appendix A.

Texture analysis based in local descriptors has attracted more attention during the last years in the field of facial analysis. Local descriptors are more robust against localized distortions of the face, due to changes in expression, to occlusions or changes in illumination.

For the tasks involving the detection and recognition of faces, some of the most extended local texture features are, for example the Haar-like features [90, 113], which simply contrast regions of the face by adding and subtracting pixels, the Local Binary Patterns (LBP) [6] features, which compare a pixel with its neighborhood to create some relation histograms, Gabor filters [34], which describe the variations at different spatial frequencies and orientations, and the Scale Invariant Feature Transform (SIFT) [65] and more recently Histograms of Oriented Gradients (HOG) [14] features, that make use of local histograms of orientations from the gradients of the image.

Even though HOG features (deriving from the SIFT features) have proved to be of great interest in other computer vision tasks, this is the first work in which they have been used for the description of facial elements. The rest of this chapter is aimed to describe the theoretical bases of the main local texture descriptors used in this work.

4.4 Gabor Filters

During the decade of the 1980s Daugman modelled the specialized cells in the visual cortex of the human brain using Gabor functions [34]. Specifically, these functions emulate some of the functionalities of the neurons in the V1 layer of the human brain, offering information related to the orientation and the spatial frequency of the local textures. The Gabor Filters are related to the wavelets functions, which are a set of mathematical operators that apply on two-dimensional matrices; the Gabor filters are self-similar as all of them are generated from one mother wavelet by rotation and stretching.

It is because of their biologically-inspired origin and due to the high robustness this operator achieves that have put the Gabor Filters in the fore of many researches aimed to the development of techniques which include the extraction of local features in image objects.

The two-dimensional Gabor Filters are mathematically defined as a set of harmonic functions, modulated by a Gaussian enveloping function. These harmonic operators are actually kernel functions of sinusoidal flat waves that are directly convolved with the image $I(x, y)$ in a certain region centered at coordinates $\vec{x} = (x, y)$. More in detail we have that:

$$\mathcal{J}_i(x, y) = \int I(x', y')\psi_j(x - x', y - y')dx'dy' \tag{4.1}$$

In this case, ψ_j represents the a family of Gabor kernel functions defined as

$$\psi_j(x, y) = \frac{\vec{k}_j^2}{\sigma^2} e^{-\frac{\vec{k}_j^2 \vec{x}^2}{2\sigma^2}} [e^{i\vec{k}_j\vec{x}} - e^{-\frac{\sigma^2}{2}}] \tag{4.2}$$

where the flat waves $\vec{k}_j(\nu, \mu)$ are a function of the actual number of spatial frequencies, ν, and the actual number of orientations, μ. Let's remark that the total number of orientations is N_o, such that $0 \leq \mu \leq N_o - 1$. The definition of a flat wave $\vec{k}_j(\nu, \mu)$ is given by:

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu cos_{\varphi_\mu} \\ k_\nu sin_{\varphi_\mu} \end{pmatrix} \tag{4.3}$$

$$k_\nu = 2^{-\frac{\nu+2}{2}\pi}, \varphi_\mu = \mu\frac{\pi}{N_o} \tag{4.4}$$

In Figure 4.3 some realizations of a set of Gabor Filters are shown. All the realizations use the same spatial frequency in $N_o = 8$ different orientations (notice that the first and the last filters in the picture are the same one).

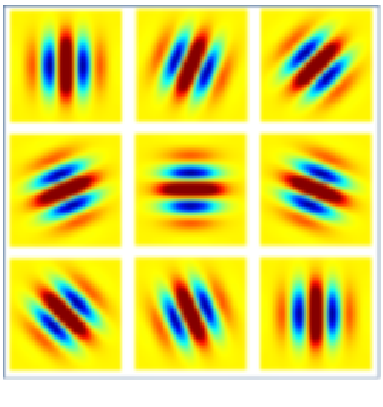


Figure 4.3: Example of a bunch of Gabor Filters with $N_o = 8$ different orientations.

The Gabor Filter results can be expressed in the imaginary and the real plane, as they define the two directions of the space that are mutually orthogonal. They achieve invariance to the light bright eliminating the continuous component (implicit in the algorithm) while the invariance regarding the contrast is achieved using a descriptor normalization. Other invariant properties are directly derived from the limitations in space and frequency. Given that the Gabor Filters offer information of the texture on a region centered in a point $\vec{x}$, one of the most important issues that has to be taken into account is not to select border areas where the texture of the background might be mixed with the actual information of the object.

In this thesis the implementation of the Gabor Filters used is that defined by Wiskot *et al.* in [117], maintaining the same configuration of the parameters. In the work of Wiskot *et al.*, the local features are defined by concatenating the results obtained when a specific region of the image, centered in a point $\vec{x} = (x, y)$, undergoes a total of 40 Gabor Filters which are obtained varying the parameters μ and ν. Specifically, they use $N_o = 8$ orientations equidistantly distributed, $\mu = 0 \ldots 7$, and 5 different spatial frequencies, $\nu = 0 \ldots 4$.

Figure 4.4 displays a bunch of Gabor Filters producing results when applied on a specific landmark of a face, the eye.

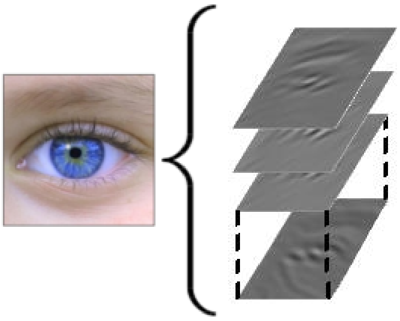


Figure 4.4: Real example of a bunch of Gabor Filters centered on an eye.

4.5 Histograms of Oriented Gradients

The Histograms of Oriented Gradients [14], also known as HOG features are a set of descriptors derived from the SIFT [113] local texture features. Both families of descriptors extract texture information from the orientation of the gradients at a level pixel. However, the main difference between them is in the location and preprocessing of the keypoints that are described.

This section summarizes the main properties of the SIFT and HOG features. The two different approaches used for the location of the keypoints are emphasized and the better suitability of using the HOG features instead of the SIFT for face recognition tasks is remarked.

4.5.1 SIFT Features

The Scale Invariant Feature Transform, also known as SIFT features, are described in the work of Lowe [65]. Initially, these features were designed to solve the problem of the spatial matching of an object. Given two views of the same object on different perspectives, the SIFT transform is able to match specific points between the images, regardless of the differences in scale, angle or 3D rotation. This makes the SIFT transform suitable for applications related with movement trackers, stereo cameras or the identification of 3D deformable objects.

Among the advantages of using the SIFT Transform with regard to other local texture descriptors, some should be remarked: their high robustness against in-plane rotations and scale variations of an object in different views and their ability to achieve partial invariance for three-dimensional rotations of the camera and changes in illumination. The algorithm used to locate the SIFT descriptors grants them high discrimination between textures, making them suitable for recognition tasks of objects and scenes.

The process developed by Lowe to extract the SIFT descriptors of an object (aimed to match an object in two views) can be summarized in the following stages:

- **Location of SIFT keypoints:** the system locates relevant points with invariant properties for different views of the object.
- **Extraction of the Histograms of Oriented Gradients:** aimed to extract local texture features at the SIFT keypoints.
- **Matching:** the SIFT features extracted for the object in the two images are matched.

In Figure 4.5 we can see an example of the matching of a deformable object (a hand) in two images using the SIFT Transform.

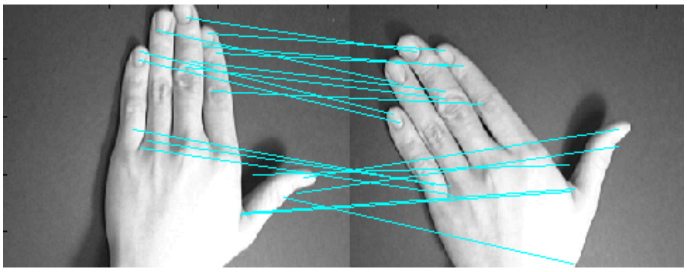


Figure 4.5: Example of the location and matching of the SIFT features corresponding to a pair of images of the same object.

Regarding the first stage of the algorithm, Lowe proposes a transformation of the image following a cascade filtering (the *scale-space transform*), such that the operations with higher cost are computed only in specific locations and not in the whole image. Therefore, the stage of the location of keypoints for the SIFT transform can be split in the following steps:

1. **Extraction of candidates**: a detection of the extrema of the *scale-space* transform of the image is performed to extract candidates of invariant keypoints. Using Difference of Gaussians (DoG), points potentially invariant to changes in scale and rotations are located.

2. **Location of the keypoints**: each *maxima* of the *scale-space* transform is matched with a model to determine its precise position and scale. The candidates with higher stability are selected to be keypoints where the descriptors will be extracted.

3. **Estimation of the orientation**: additionally with the location of the landmarks, the algorithm estimates the main orientation of the keypoints, based on local gradients information. This step is aimed to achieve higher invariance against rotations.

At the end of the SIFT location stage, the SIFT keypoints finally extracted are defined by an specific scale, location an orientation. However, the location of such keypoints is not controlled by the user as the location of the maxima on the local-space transform is particular for each object.

After the location, the local features based on Histograms of Oriented Gradients are extracted. In this thesis, we have studied and worked independently on the local descriptor used by the SIFT transform. Henceforth, the descriptors of the SIFT Transform will be referred as *HOG features.*

The last step of the SIFT Transform, the *matching step*, is skipped in this work as in our different implementations our own matching methodologies are applied.

4.5.2 HOG features

The location method of the SIFT keypoints above explained does not fit completely with the problem of face description due to its uncontrollable nature. However, the use of the Histograms of Oriented Gradients is still quite promising for the tasks of face description. This drives us to define a new set of descriptors based on the SIFT Transform but with an independent location method. This section explains the necessity of using the HOG features and details its main properties.

The necessity of a different set of features

To solve the problem of face description based on local features a robust location of facial keypoints hast to be performed. To extract a determined set of keypoints to describe a face it is advisable to follow some guidelines:

- The keypoints should be related to the local areas of the face with more information. The most remarkable features of a face, considering local textures, are obtained from the facial elements: the eyes, the nose or the mouth, among others. Therefore, the location stage of the SIFT Transform, based on the *space-scale* transform is not suitable.

- To compare features between different faces, it is necessary to find a common set of keypoints describing exactly the same elements. That is, given a new face, a fixed set of keypoints should be automatically located and extracted. This consistency on the location cannot be achieved using the SIFT Transform, in which for different samples of an object (e.g. a face), different keypoints are located.

The uncontrolled and partially random results on the location of the SIFT features has driven us to skip its first stage, maintaining the use of the HOG descriptors.

However, the need of skipping the SIFT location stage produces new needs. Without the SIFT location stage, the keypoints lose their properties regarding their invariance to scale, location and orientation. With the aim to cover this lack, it will be necessary to perform an image normalization process before extracting the HOG features. This normalization will focus on homogenize the representation of the face elements, as described in Chapter 2.

Properties of the HOG features

The HOG descriptors are derived from the second step of the SIFT Transform: extraction of the Histograms of Oriented Gradients. The study of the properties of the HOG descriptors presented in this work is based on the original definition of this stage detailed by Lowe [65].

Lowe proposes a mathematical operator to describe local textures, able to achieve robustness against variations not solved during the previous stages, such as the illumination variance and affine transformations.

For simplicity, henceforth in this work there is a distinction between the *SIFT descriptor* and the *HOG descriptor*; when the former is referred it will be assumed that the location, scale and orientation of the descriptors has been obtained using the SIFT location stage; when the latter is referred it will be assumed that the location, scale and orientation of the features has to be obtained by any other means.

The HOG features belong to the family of the local texture descriptors, and they represent specific keypoints of an image through the values of the gradients of the pixels in an area surrounding that point. An image gradient is defined as the first order derivative of it in x and in y directions. Therefore, it gives information on the directional change in the intensity of each pixel on the image. As it is a vectorial measure, it can be represented by its modulus (representing the magnitude of the local variations) and a phase (representing the direction of such variations).

The goal of the HOG descriptor is to define a series of histograms representing the patterns of the orientations in some areas (subregions) around that central point.

Basically, the extraction of the HOG features can be summarized in a few steps:

1. **Determination of the feature window**. This window represents the area described by the HOG feature. Given an image element to be describe, its HOG descriptor is generated defining a square window of size $p \times p$ pixels centered at its central point. The size of the window has to be adapted to the size of the described element.

2. **Division of the descriptor area in *cells***. Each of these cells, here referred as $N_p(i, j)$, will be further described by a histogram of orientations. More in detail, the descriptor is organized into a square grid embedded in the feature window. In some studies [14] the authors have proposed different shapes for the grid, such as circular instead of square. Our moti-

vation in this work was to follow Lowe's original work, which uses square uniform grids of size $N_p \times N_p$.

An example of square cells or subregions of a HOG descriptor can be seen in Figure 4.6.

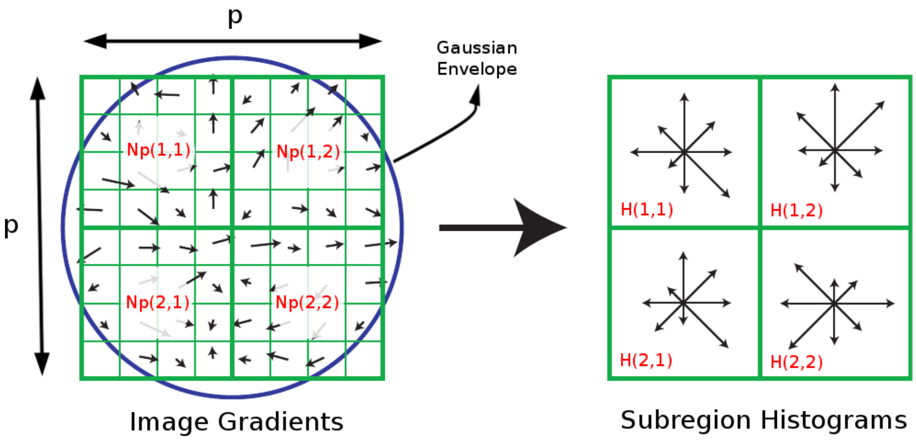


Figure 4.6: Example of HOG feature window, divided in a square grid of $N_p = 4$ bins along each side. Also there is the histograms of orientations corresponding to each of the cells.

3. **Calculus of the local orientations**. For each pixel in the feature window, its orientation is estimated based in its gradient $\nabla(x, y)$. Once the descriptor area is organized, there is a need to extract the module $\|\nabla(x, y)\|$ and the phase $\nabla(x, y)|_{phase}$ of the image gradient at each pixel of the feature window.

   However, the HOG describes an area around a specific keypoint. The information provided by the pixels near the borders of the window should be of less relevance than the information provided by the pixels closer to the central point. To achieve this fact, the HOG descriptor uses a two-dimensional Gaussian envelope function that weights the module of the gradients at each pixel of the feature window:

$$\bar{\nabla}(x, y) = G_{\sigma_{HOG}}(x, y)\nabla(x, y), \tag{4.5}$$

   where $G_{\sigma_{HOG}}(x, y)$ is a Gaussian function centered at the keypoint, and with standard deviation σ_{HOG}, for both the x and y axis. Also in Figure 4.6 we can observe the Gaussian envelope applied over the local gradients.

4. **Generation of the orientation histograms**. A histogram per cell is defined assigning the contribution of each pixel gradient to their corresponding histogram. This contribution is weighted by different factors which are combined in a trilinear interpolation (which corresponds to a bilineal spatial interpolation and a lineal orientation interpolation, as explained next in this section).

The underlying idea on the HOG descriptors is that the information on each of the cells of the $N_p \times N_p$ square grid, may be condensed in a single histogram of directions. All histograms are structured into N_o possible bins, which correspond to the same number of possible local orientations. The set of local orientations in a histogram is defined as $\Theta_{HOG} = [\Theta^1, \Theta^2, \ldots, \Theta^{N_o}]$.

The HOG histograms represent the pattern of orientations at a determined cell of the grid; the histogram belonging to the cell $N_p(i, j)$ is denoted as $H(i, j)$. Figure 4.6 displays an example of a HOG descriptor with four cell histograms.

Inside the HOG window, each pixel has a tri-linear contribution up to its four closest cell histograms. Let's consider a pixel $I(x, y)$ (with spatial coordinates x and y), with a local gradient such that its modulus is $\|\hat{\nabla}(x, y)\|$ and its orientation is $\alpha_{x,y} = \nabla(x, y)|_{phase}$. Then, the tri-linear contribution of the local gradient to the histograms can be broken down in the following:

- **Bilinear interpolation for the spatial location**: this interpolation grants the descriptor robustness against small displacements. Each pixel orientation contributes to the orientation histogram of the cell where it is located and also to three histograms of the closest cells.

  Let's consider a pixel $p(x, y)$, belonging to the grid cell $N_p^1 = N_p(i, j)$, and surrounded by the closest cells $N_p^2 = N_p(i+1, j)$, $N_p^3 = N_p(i, j+1)$ and $N_p^4 = N_p(i+1, j+1)$, with orientation histograms $H1 = H(i, j)$, $H2 = H(i+1, j)$, $H3 = H(i, j+1)$ and $H4 = H(i+1, j+1)$, respectively. An example of this configuration can be seen in Figure 4.7.

  Assuming and square grid, the distance between the central point of the grid cells in both axes can be normalized, such that $D_x = 1$ and $D_y = 1$. Then, the distance between $p(x, y)$ and the central point of N_p^1 is defined as $d = (d_x, d_y)$. Following the scheme in Figure 4.7, the bilinear contribution based on the pixel gradient is as follows:

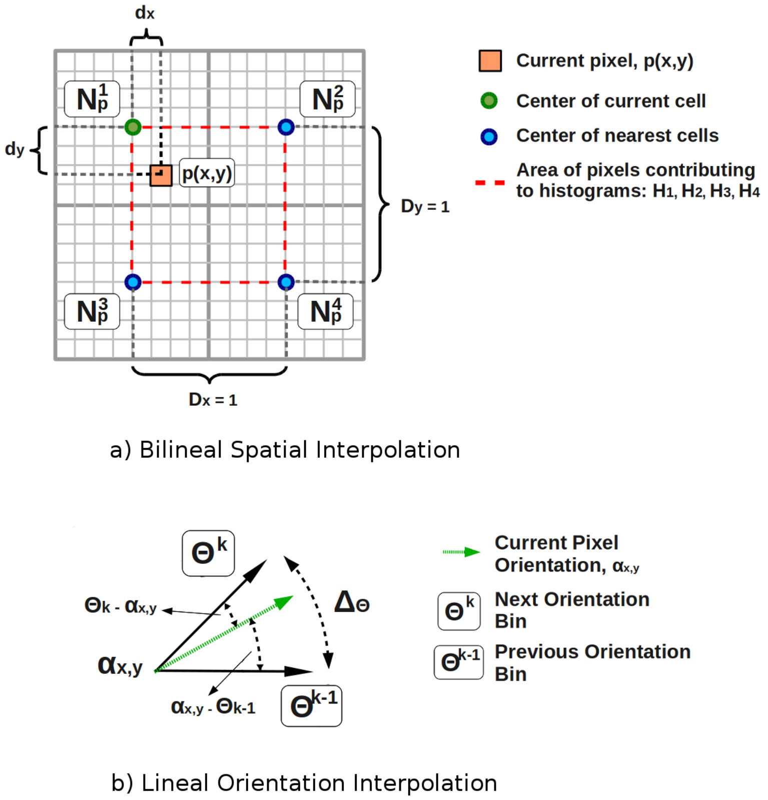
$$\begin{aligned}
&\|\hat{\nabla}_{H1}(x, y)\| = (1 - d_x)(1 - d_y)\|\nabla(x, y)\| \\
&\|\hat{\nabla}_{H2}(x, y)\| = \begin{cases} d_x(1 - d_y)\|\nabla(x, y)\| & \text{if } d_x > 0 \\ 0 & \text{otherwise} \end{cases} \\
&\|\hat{\nabla}_{H3}(x, y)\| = \begin{cases} (1 - d_x)d_y\|\nabla(x, y)\| & \text{if } d_y > 0 \\ 0 & \text{otherwise} \end{cases} \\
&\|\hat{\nabla}_{H4}(x, y)\| = \begin{cases} d_x d_y\|\nabla(x, y)\| & \text{if } d_x > 0 \text{ and } d_y > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4.6}$$

  where $\|\hat{\nabla}_{Hk}(x, y)\|$ represents the contribution of pixel gradient $I(x, y)$ to each of its surrounding orientation histograms, based on the relative position of the pixel (x and y) regarding the grid cells centers (as it can be seen from the right side of the expressions in Equation 4.6).

- **Linear interpolation for the orientation**: this interpolation grants the descriptor robustness against small rotations, dividing the contribution of each pixel orientation between two bins of the cell histogram.
  Let's assume a pixel $p(x, y)$ with a gradient orientation $\alpha_{x,y}$. If $\alpha_{x,y}$ corresponds to an orientation Θ^k in Θ_{HOG} then, the module $\|\hat{\Delta}(x, y)\|$ is added to the k-th bin of the orientation cell histogram. In a more general case, the phase of the pixel gradient is in the middle between two orientations. In that case, a single pixel contributes to each of the bins as follows:

$$\text{If } \Theta^{k-1} < \alpha_{x,y} < \Theta^k \text{, we add } \begin{cases} \frac{|\alpha_{x,y} - \Theta^{k-1}|}{\Delta_\Theta} \times \|\hat{\nabla}(x, y)\| & \text{to the (k-1)-th bin} \\ \frac{|\Theta^k - \alpha_{x,y}|}{\Delta_\Theta} \times \|\hat{\nabla}(x, y)\| & \text{to the k-th bin} \end{cases}$$

  where $\Delta_\Theta = \Theta^k - \Theta^{k-1}$.
  For a better understanding of this interpolation, see Figure 4.7.


dx
dy
N_p^1
N_p^2
N_p^3
N_p^4
p(x,y)
Dy = 1
Dx = 1
Current pixel, p(x,y)
Center of current cell
Center of nearest cells
Area of pixels contributing to histograms: H1, H2, H3, H4
Θ^k
$\Theta_k - \alpha_{x,y}$
Δ_Θ
$\alpha_{x,y}$
$\alpha_{x,y} - \Theta_{k-1}$
Θ^{k-1}
Current Pixel Orientation, $\alpha_{x,y}$
Θ^k Next Orientation Bin
Θ^{k-1} Previous Orientation Bin

a) Bilineal Spatial Interpolation

b) Lineal Orientation Interpolation

Figure 4.7: Scheme of the parameters involved in the a) spatial contribution and b) orientation contribution of a pixel to the cell histograms .

5. **Generation of the HOG feature vector**. The extraction of the HOG feature is done concatenating into a single vector the $N_p \times N_p$ orientation cell histograms extracted from the descriptor window. That is,

$$HOG(x,y) = [H(0,0)^T, \ldots, H(i,j)^T, \ldots, H(N_p, N_p)^T]^T,\ 0 \leq i,j \leq N_p, \quad (4.7)$$

which is a column vector of dimensionality $dim_{HOG} = N_p \times N_p \times N_o$. Henceforth, the HOG descriptor will be characterized always as vector.

6. **Normalization of the feature vector**. This step was introduced by Lowe to remove light dependences. More in detail, the last step of the extraction of the HOG descriptor consists on reducing the noise introduced by the changes in illumination over the histograms of orientations. In the original work of Lowe, the final descriptor undergoes a two-step normalization step: first the feature vector $HOG(x,y)$ is normalized in magnitude and then the resulting vector is trunk so that the maximum value of each of the components cannot be greater that 0.2. With the firs step of this normalization stage we remove the effects produced by affine changes in illumination, while with the second step we reduce the influence of sharp edges.

The following section describes the experiments performed to determine the optimal parameters of the HOG descriptors when used for face analysis. More specifically, our interest lies on exploiting its potential to describe some of the facial elements such as the eyes, the nose or the mouth.

4.6 Study of the HOG features to describe Facial Elements

The use of HOG descriptors to describe specific facial elements is completely novel. In consequence, before performing any facial analysis it is important to set some of their intrinsic parameters. These parameters can be summarized in the following:

- The shape of the feature window and the number of cells it contains, N_p, that determines the number of histograms, H_{Np}.
- The number of bins of the orientation histograms, N_o.
- The parameters of the Gaussian envelope function, specifically its standard deviation, σ_{HOG}
- The total size of the HOG window (which determines the number of pixels contributing to each of the histograms), p (also called P_{HOG}).

For almost all the parameters, a generic solution is selected. These solutions have already proved to be valid in other works. However, for some specific cases additional experiments should be performed to determine their optimal values. Next, the generic solution adopted for the parameters N_p, N_o and σ_{HOG} is described. The experiments performed to determine the size of the HOG window, P_{HOG}, are included in Section 6.6.1[2] which addresses the set-up of the face recognition subsystem.

[2] P. 119

4.6.1 Selection of generic values

The HOG descriptors have been tested in a number of scenarios in other works. In that sense, Lowe [65] carries out a thorough analysis of the internal parameters of the descriptor. Its conclusions are useful for the research community.

In order to study the HOG descriptors for facial analysis, in this thesis some of the original values from Lowe for the SIFT Transform have been directly adopted:

- **Shape of the feature window and number of cells**: We decided to use square regions organized as coarse grids. This configuration is the most simple of the ones presented by the authors and also the most effective due to its generic nature. This is an advantage for our work as we intend to describe different facial elements, each of them having its own properties (e.g. shape, texture and spatial configuration). A generic shape for the descriptor is the best option to adapt the particularities of each of them.

  Regarding the grid, a spatial layout of $N_p = 4$ cells in both axis, horizontal and vertical, was selected. This configures a total of $H_{Np} = N_p \times N_p = 16$ histograms corresponding to the 16 cells of the grid. Figure 6.11 displays a real example of the shape of the feature window and its grid for a HOG describing an eye.

- **Number of orientations**: It is important to determine the number of orientations (bins) of the histograms that are extracted at each subregion. A low number of bins produces very simple patterns, reducing the discriminative power of the descriptor, while a high number of bins increases the complexity of the feature and may lead to the *overfitting* problem, in which the local descriptor contains too much information to generalize common features for a family of elements (e.g., features generic for all the eyes).

  Lowe also determined after his experimentations that for the description of objects the optimum number of orientations is $N_o = 8$. This corresponds to the set of gradient angles $\Theta_{HOG} = \{0, 45, 90, 135, 180, 225, 270, 325\}$ degrees. This work selects the same configuration. In Figure 4.6 an example of four cell histograms with $N_o = 8$ orientations can be seen.

- **Parameters of the Gaussian envelope**: To localize the information produced by the HOG descriptor, the information closer to the central keypoint needs to be enhanced. This is performed using a Gaussian envelope function, centered in such point, weighting the module of the gradients at each pixel. These weights are related to the standard deviation of the Gaussian, σ_{HOG}. Lowe considered it appropriate to make the standard deviation proportional to the size of the area that is described. In this work the value of the standard deviation of the Gaussian has been set to be half the size of the described area. That is, $\sigma_{HOG} = \frac{P_{HOG}}{2}$, where P_{HOG} is the size in pixels of the region that is described.

Using all these generic parameters in our work the dimension of the final feature vector, $HOG(x, y)$, is $D_{HOG} = N_p^2 \times N_o = 128$. Once again, this is the original length of the descriptor proposed by Lowe.

Table 4.1 summarizes the most important values of the parameters of the HOG descriptor used in our work.

HOG Feature	**Symbol**	**Value**
Window Shape	$P_{HOG} \times P_{HOG}$	Square
Window Size (pixels per direction)	P_{HOG}	20
Grid Shape	$N_p \times N_p$	Square
Nr. Cells (per direction)	N_p	4
Cell Pixels	p	5
Nr. of Histograms	N_H	16
Histogram Orientations	N_o	8
Standard Deviation of Envelope	$\sigma_{HOG} = \frac{P_{HOG}}{2}$	10
Feature Vector Dimension	D_{HOG}	128

Table 4.1: Summary of the HOG descriptor parameter values in this work.

Chapter 5

Precise Face Detection using Eye Location

5.1 Introduction

In biometric face analysis, most of the times it is necessary to include a face location stage. The faces on an image are first detected and then some spatial cues are given to facilitate their normalization. However, the location stage has been avoided in many works; this has made the development of face detection algorithms as an interesting topic for this research.

This chapter is aimed to study fully automatic face detection algorithms to perform eye location. In this thesis, the precision of the location algorithms is achieved using a intermediate stage of eye pairs location. The detection algorithms are important to be used as a previous stage for face recognition, as it is shown in Chapter 7. In the text, the term *detection* is referred the process of determining if an object of an specific class is in an image. However, when the term *location* is used, in most cases it implies that the output of the process is a set of coordinates referred to the image.

5.1.1 The necessity of automatic eye location

First of all, it is necessary to explain the necessity of introducing an eye location step to provide more precision to the face detection algorithms. The location of the facial elements provides information for different face analysis tasks, such as the classification of facial gestures, eye tracking or, most commonly, for face recognition. In this work, an eye location step is used to provide the desired accuracy to the face detection algorithm.

In the literature, the majority of the developments avoid this location step, performing a manual marking of the facial landmarks of all the faces in the training and validation images. This information is considered the groundtruth data of the datasets.

The manual marking of facial landmarks is a non-scalable process. It cannot be tackled when the number of sample images increases. This has pushed the researchers to develop automatic techniques, to the detriment of reducing the degree of precision in the location of these landmarks.

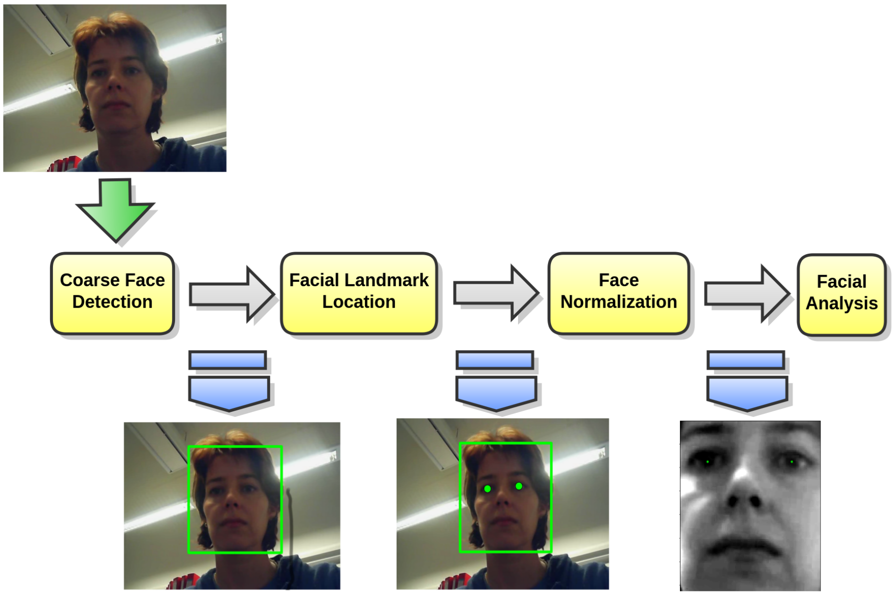


Figure 5.1: Diagram of the most common stages between coarse face detection and any facial analysis processing.

On the other hand, it is important to achieve as much precision as possible in face recognition; small variations in the facial representation information, such as spatial displacements, rotations or scale variations can have an adverse effect.

Many authors [97, 100, 25, 58] have studied the impact that the degrading of the location of a detected face has on a face recognition system. It has been proved that for the majority of face recognition algorithms, the worse the location of the detected faces, the lower the recognition rates achieved. Even small displacements may lead to a significant decrease in the face recognition rate. In [97], the authors prove that perturbations of a 5% deviation in the eye location may induce an increase in the recognition errors of up to 20% for baseline algorithms such as PCA [107] or EBGM [117].

Figure 5.1 shows some of the most common steps that help link coarse detection with the final analysis. In the last few years, a number of algorithms to locate facial landmarks have been studied. The most common landmarks to locate are the eyes, the eye brows, the mouth or the nose-tip, as they provide useful information for a precise location of the face.

Precise location is essential to ensure the effectiveness of the normalization step. A necessary requirement for normalizing faces is to locate at least two points from the original image and their correspondence in the normalized image. For further details, see Section 2.4.

The eyes fulfil these requirements, as demonstrated in works such as [25]. The use of the eyes as keypoints for normalization is motivated by some of their properties:

- The eyes mark the horizontal line of the face and are also symmetric to the central vertical axis. Using the eye information, the center of the face can be set in the middle point between them. Also, the angle of the head can be directly inferred. If we set a constant inter-ocular distance, the scale uncertainty can also be solved during normalization.

- The symmetric resemblance between the two eyes makes it possible to develop a unique algorithm to detect both the left and right. This is much more efficient than locating two completely different landmarks, as this would entail developing independent algorithms, probably with different performances.
- The location of the eyes is a good starting point for many feature-based algorithms that extract the position of other relevant landmarks from them [61]. In Chapter 6, all the feature-based algorithms analyzed use the eyes as starting points.
- The eyes can be easily located using special cameras working on wavelengths of near-infra-red light (NIR) [84]. The detection with these cameras can be combined with the detection of images working on the visible part of the spectrum, making the results much more consistent as many false alarms may be avoided during the process.

The location of eyes entails some additional benefits for recognition tasks. For example, it is a powerful mechanism to validate the *Coarse Face Detection* step (see Figure 5.1). In face detection, the positive samples used to train the algorithms are face images that usually have very low resolution. For example, sometimes the samples have resolutions of 24×24 pixels or smaller, as is the case for the samples used in the OpenCV library classifier[1], widely used in computer vision.

Figure 5.2 offers an example from the FERET database[2] of a face image extracted at low resolution. Logically, these sample images provide a very limited quantity of information, which sometimes leads to a greater number of false detections. In a system without facial landmarks detection, the false alarms are also reduced by decreasing the number of hits (i.e. increasing the permissibility). However, the eyes can be used as an additional mechanism to validate the faces that are detected: if the eyes are not detected in a face, it can be considered a false alarm. Thus, the location of the eyes helps reduce the rates of false detections, keeping the number of hits at reasonable levels.

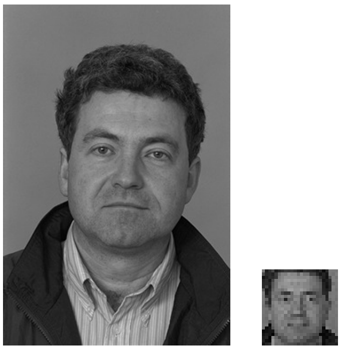
5.2 Motivation and Contributions

The present focus on the topic of face detection with eye location was mainly motivated by the lack of sufficient literature addressing the problem of unifying face detection and face recognition. Many works have done research on isolated approaches for both issues, but it is still challenging to integrate them in a single system.

In this chapter, a novel solution for face detection with eye location is developed. One of the novelties of the proposed approach is the use of a multi-resolution eye location system trained with a set of local descriptors never previously used for this task. In our case, the study is based on an approach focused

[1] OpenCV is a C++ image processing public library developed by Intel: http://opencv.willowgarage.com/wiki

[2] See Appendix B.



(a) (b)

Figure 5.2: Extraction of a low resolution face image (24 × 24 pixels) from a FERET face image. **a)** Original Image, **b)** Low Resolution Image.

on the extraction of HOG features[3] to describe the eyes. This choice is motivated by the positive response of the HOG features versus some illumination problems and small spatial variations.

The hypothesis behind our approach can be summarized in the following:

- **Hypothesis 1**: *The location of the eyes performed with a multi-resolution design can provide the system with more precision than a monolithic approach.*

  This hypothesis is supported by some of the results in the literature, as explained in further sections. The eye location algorithm proposed in this thesis works at three levels of resolution. First, a coarse face detection algorithm is performed, giving the position of faces as a bounding box. Second, some eye candidates are extracted from the face area with a fast and efficient classification approach. Finally, high-level descriptors are used to determine the optimal pair of eye candidates. This last step is quite exhaustive and thus needs to be performed on a limited number of samples.

- **Hypothesis 2**: *The use of HOG descriptors as local features for eye location can lead to a precise set of coordinates.*

  The use of local descriptors with high descriptive power, such as the HOG features, is expected to lead to an increase in eye location accuracy. The HOG descriptors have been proved to be robust when the illumination conditions change, or when there are small variations in the position of the landmark, in rotation or even in scaling. This flexibility reduces the complexity of the requirements in the extraction of the eye candidates.

From these hypothesis, three goals arise:

1. *Analyze automatized and precise face detection systems.* This analysis is focused on obtaining information for the face representation stage, prior to the recognition stage.

[3] See Section 4.5.2 for further details

2. *Design a fully automatic solution to the accurate face detection problem.* Our design has to be able to generate input data for any of the feature-based algorithms studied in Chapter 6.

3. *Establish a comparative study between the eye location approach proposed here and other state-of-the-art-works.* A set of experiments has been designed specifically to achieve this goal, using a workbench of datasets.

5.3 State of the Art

The necessity of precise eye location has encouraged the design of several different strategies. This section reviews some of the most salient works in the literature.

The section starts with a brief explanation of the most significant research in eye location developed recently. Then, a selected group of these algorithms are described in detail; these algorithms will be used in this work as referents for the comparative analysis.

5.3.1 Algorithms for Face Detection with Eye Location

Jesorsky *et al.* [51] were pioneers in the development of a metric to quantify the precision of eye location systems. They studied an approach based on contour models that starts with a coarse tuning, followed by a fine tuning of the position of the eyes. As explained below, the technique of doing the eye location in two steps, from rough location to a fine solution, has become widespread. The algorithm designed in the current thesis follows the same strategy: locating a bounding box around face areas, then preselecting eye candidates and then locating the best eye pair.

With time, new eye location methods have tended to use more and more complex features and classifiers. The use of classifiers discriminates between *eye* and *non-eye* areas. Inspired by the results in face detection, authors such as [69] and [67] include some boosting classification stages, based on the AdaBoost classification performed by Viola and Jones [113]. One of the main features of the boosting is its high efficiency with low computational loads. In [69], three boosting classifiers are used: one for the Coarse Face Detection stage and two more to detect each of the eyes, with a vertical division of the face into two different regions. In this work, the eyes are then paired up using probabilistic methods. In [67], the authors also make use of a boosting classifier as a preliminary stage for coarse eye location.

In the framework of this thesis, the boosting classifiers are used to preselect eye candidates from which some local features are extracted. Further details are given in Section 5.6.

The evolution of the algorithms for eye location has proved that the systems based just on boosting classifiers do not achieve the precision that is required for the majority of the facial analysis techniques. Therefore, many prominent approaches perform the extraction with other kinds of supervised classifiers, the most common being those based on linear discrimination. In such approaches, the data extracted to locate the eyes is vectorized; then, these eye feature vectors are projected into spaces of a lower dimensionality than that of the original space, making it easy to separate positive samples (*eyes*) from negative samples (*non-eyes*). There is also a branch of approaches based on kernels that achieve non-linear discrimination by projecting the samples to spaces of higher dimensions.

One of the most successful and widespread classifiers is the Support Vector Machine (SVM) [29], a kernel-based algorithm. A clear example of one approach using this classifier is found in [103]. In this work, a mix of classifiers is performed: in the first stage, a boosting classifier extracts some candidates, and then a SVM is applied to achieve the final location. In this case, the SVM works on geometrical features of the eyes. In [52], the authors use SVM integrating features from the eye-pair, building a hybrid classifier. The features extracted from the eyes are selected using a filtering and clustering method based on a simplification of the maximum likelihood theory. After the extraction of the features, they are verified using Template Matching techniques to reject the false detections. The SVM classifier has also been proposed in other referenced works, such as in [24].

In our work, SVM classifiers are used to classify eye candidates. The features used to train such classifiers are HOG descriptors. In Sections 5.7.3 and 5.7.5, a detailed explanation of the SVM for its training and validation phases is provided.

Another key topic relevant to eye location has been the study of the local features used to discriminate the eyes from the rest of facial elements. Some works try to make use of our *a priori* knowledge of the biological disposition of the eyes and their physiognomy. One exponent of this tendency can be found in [39], where the author exploits the fact that the eyes have distinctive horizontal-like borders and also the fact that the area of the pupil is much darker than its surroundings. Other works use other mathematical backgrounds to extract complex descriptors,. For example, in [40], the authors perform some Gabor-wavelets filtering to look for different facial landmarks (not only eyes), and then end up using a SVM classifier to determine valid triplets of these landmarks. In [114], for example, the authors design an AdaBoost-like classifier, substituting the Haar-like wavelets with more advanced features, extracted after applying an Recursive Non-parametric Discriminant Analysis (NRDA).

One of our main interests in the development of this thesis has been to go through a set of mathematical features that could deal with the regular conditions after a coarse eye location, such as small variations in the position, in the rotation or the scale. The HOG features before mentioned are a good candidate to be studied, as they perform well for these variations. More details on this issue are given in Section 5.7.3.

Finally, it is also remarkable to note the number of works that have solved the topic of eye location from completely different perspectives. Some works perform a holistic approach, using templates, both predefined [38] and adaptive [3]. More recently, the work performed by Behnke [11] is worth mentioning. The author proposes the use of Neural Networks (NN), trying to emulate the work of a human brain. The location of the eyes is achieved using hierarchical and multi-resolution Neural Networks with local recurrent connectivity. Another noteworthy work is that presented by Cristinace y Cootes [32], in which some face modeling algorithms are adapted to the eye location task. Specifically, they use a variant of the AAM method [30], called the Constrained Local Models (CLM), in which $N_p = 17$ facial landmarks are located (including eyes, eyebrows, or the nose tip). These approaches are quite ambitious and also significantly increase the complexity of the systems, but also usually offer higher accuracy in the location of the landmarks. However, one of the targets of this work is the search for simplicity.

Some of the works mentioned before are used in this work as benchmarks for establishing a comparative analysis with our solutions in the experimental analysis performed in Section 5.8.2. Thus, an extensive analysis of these works is given below.

5.3.2 Selection of relevant Eye Location algorithms

In order to evaluate the eye location algorithm developed here, some reference works are needed. Thus, three outstanding works [52, 24, 114] have been selected. This selection has been based on three aspects:

1. The three algorithms are highly innovative, having a big impact on the field of eye location research. Currently, these algorithms have become a reference not only for this work but for many other authors.

2. These works provide a full set of information about the implementation and the results obtained. Specifically, the results are given in a common evaluating framework, making them accessible and quite easy to compare.

3. One feature relevant to this thesis is how to measure the time efficiency of our solution. Thus, some algorithms that provide additional information about real computational costs (i.e., execution times) have been selected.

Selecting algorithms with such features ensures that any new approach (as is the case of ours) can be directly validated under a common evaluation reference. To complete the comparative study, the results are analyzed against a commercial eye location system. The system we selected for this purpose was VeriLook [73], developed by Neurotechnologija.

Next, a brief description of the reference algorithms selected is given, noting the main similarities and differences with regard to our proposal. However, the goal of this section is only to provide an informative insight into the algorithms; for a complete understanding of them, the reader is referred to the original works in [52, 24, 114].

A hybrid classifier for precise and robust eye detection (Jin *et al.* [52])

In the work developed by Jin *et al.*, the eye location algorithm designed is based on the use of a hybrid classifier that combines the extraction of features directly from the two eyes and also from the eye-pair. This hybrid classifier effectively combines the use of a SVM classifier with one of maximum likelihood, both with supervised training stages. The goal of these classifiers is to detect areas with eye-like geometries. By concatenating the two classifiers in a cascade, some eye candidates are selected and then they are validated using template matching techniques for eye-pairs. The steps of the eye location (candidates extraction and validation) are similar to our algorithms. Figure 5.3 shows a diagram of the methodology used in [52].

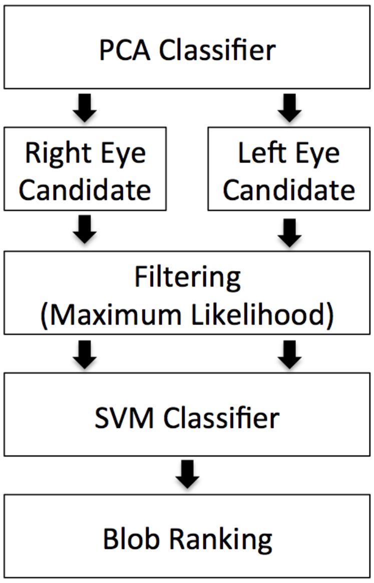


Figure 5.3: Block diagram of the algorithm proposed by Jin *et al.*. in [52].

The evaluation of the performance of Jin *et al.* has been done using the FERET database[4].

Precise eye localization through a general-to-specific model definition (Campadelli *et al.* [24])

Campadelli *et al.* explore the precise location of the eyes using a two-step approach: first a coarse eye detection, followed by a fine location. Starting from the results of a coarse face detection stage, the first step of this approach extracts some low-level sample features which also help to validate the output from the previous stage. The second step does a fine eye location using a SVM classifier trained with the samples extracted before. Figure 5.4 shows a diagram of the methodology used in [24].

The multi-resolution approach of this work is similar to that proposed in this thesis. It starts with a model with some low-level features that extracts general information about the eyes, and then uses specific classifiers for the fine location.

The results of Campadelli *et al.* have been validated using images from the FERET and the FRGC databases[5].

Automatic Eye Detection and its validation (Wang *et al.* [114])

The algorithm proposed by Wang *et al.* uses a boosting classifier, although its main contribution is that it is trained with a set of features different from the Haar-like ones used in [113]. The authors propose extracting Recursive Non-parametric Discriminant Analysis (RNDA) features, simplifying the boosting

[4] See Appendix B.
[5] See Appendix B.

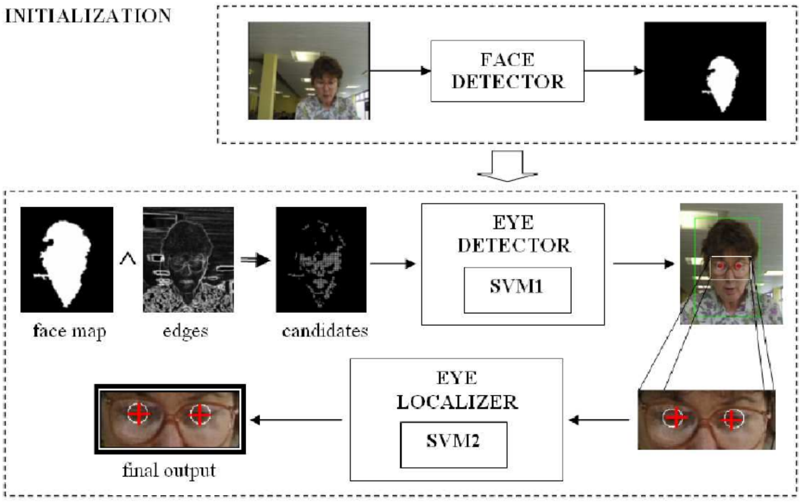


Figure 5.4: Block diagram of the algorithm proposed by Campadelli *et al.*. Courtesy picture, extracted from [24].

classifier into two stages: a cascade of two steps, followed by a fine location using less than a hundred features (compared to the thousands usually used with AdaBoost). This simplification reduces computational costs, allowing the system to perform faster than other algorithms, but with almost the same high efficiency. Figure 5.5 shows a diagram of the methodology used in [114].

The evaluation of this solution has been done using the FRGC dataset and, similar to the system developed by us, it studies the use of a complex set of features for the description of the eyes, which contributes to higher achieved accuracy in the location.

5.4 Overview of our Eye Location approach

To fulfil the main goals cited earlier, a multi-resolution face detection algorithm with eye location has been developed. It is structured in a low number of stages that combine low-resolution and high-resolution feature extraction, combined with supervised classifiers. This structure looks for an effective way of mixing stages that are efficient in terms of computing time with stages that are efficient in terms of feature description power.

The general process of the eye location stages can be found in Figure 5.6. As we can see from the figure above, the multi-resolution philosophy is embedded in our design, as we start extracting some generic information that, stage by stage, becomes more refined. All the steps are concatenated in such a way that the outputs from every stage are the inputs for the next one. To fully understand our design, the evolution of three key parameters through the stages is analyzed; these are the *quantity* of data processed, the *complexity* of this data and their velocity in terms of *execution time*:

- *Quantity* of data analyzed: The system starts working with large amounts of data (e.g., scanning a whole image in the beginning), but then it de-

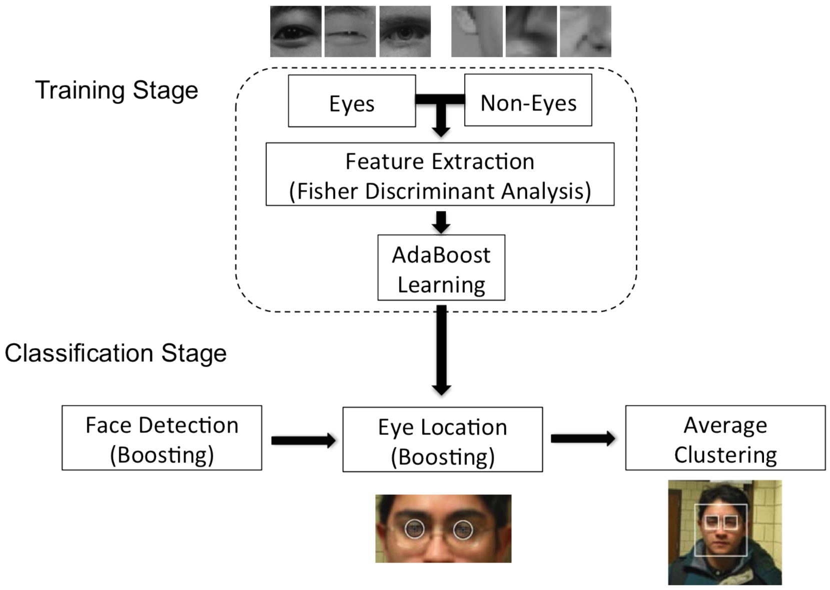


Figure 5.5: Block diagram of the algorithm proposed by Wang *et al.*. in [114].

System Feature	Evolution
Quantity of Data	*Decremental*
Complexity of the algorithms	*Incremental*
Computational Cost	*Balanced*

Table 5.1: Evolution of three parameters in our eye location solution.

creases step by step, so in the end, the algorithm is working just on the data extracted from certain eye candidates.

- *Complexity* of the features: As the quantity of data to process decreases with the steps performed, we can make use of more complex techniques.
- *Execution time*: The main intention is to keep it balanced along the stages. The first stages process more data but in a simpler fashion, while in the last stages the complexity of each step is greater, but with a lower quantity of data.

Table 5.1 summarizes the evolution tendency of these three parameters throughout our system.

The system here proposed is structured into four main stages: first, it starts with a Coarse Face Detection, based on the AdaBoost algorithm developed in OpenCV. To simplify, we assume that the input image is in grayscale, has an arbitrary size and contains one only face to be detected. In the case of multiple faces, the steps of the system would be repeated for each face detection. In the second step, a two-folded boosting stage is performed on the face regions delimited by the bounding boxes previously detected, in order to extract some

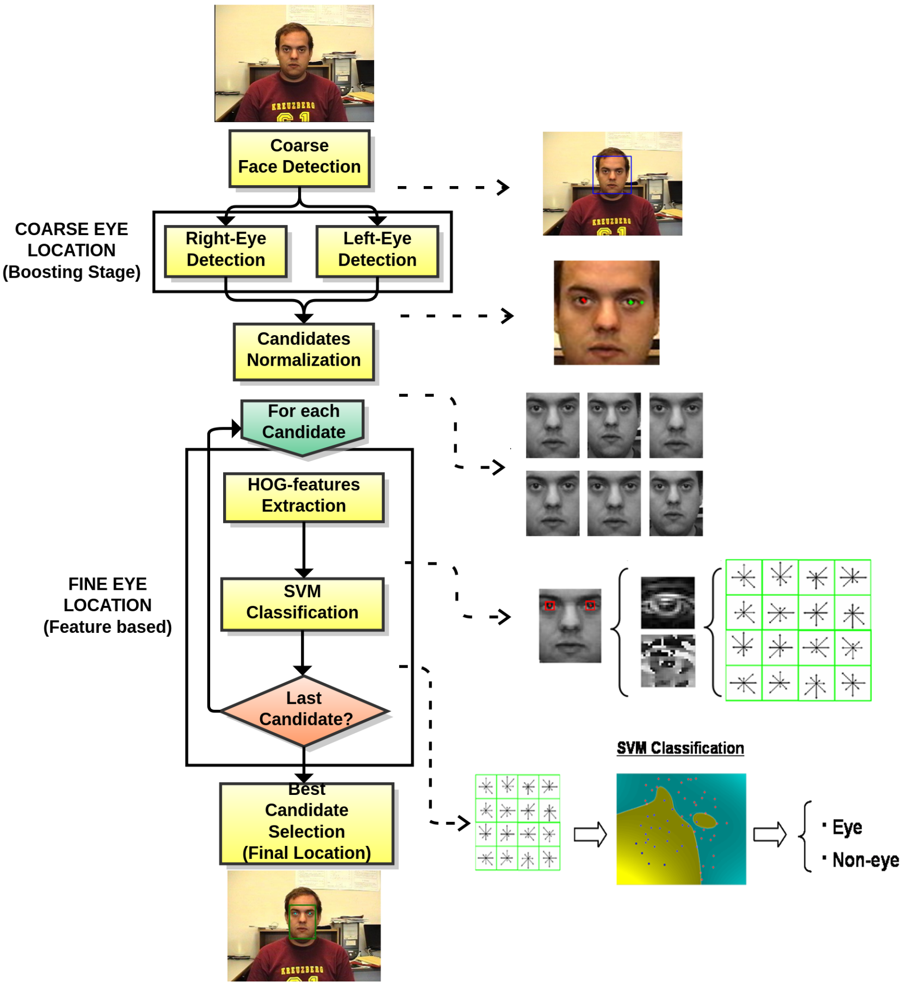


Figure 5.6: Block diagram of the stages of our eye location solution.

eye candidates. Then, all the eye candidates are combined to configure a few face candidates, which are then normalized. After the normalization, HOG features are extracted from the eye-positions of the face candidates, and then a SVM classifier determines which is the optimal candidate. The final pair of selected eyes determines their precise position.

The two classifiers used in this thesis are the **boosting** for the first stages and **SVM** for the final step. These classifiers are known for their high performance when classifying biometric samples. Due to the efficient training of such classifiers, high accuracy in the location results was expected. Another factor that was expected to contribute to the overall good performance was the use of the HOG features, as will be proved later.

To provide greater clarity for our developments, the rest of this chapter is devoted to an exhaustive theoretical analysis of each of the individual steps of the system (following the structure shown in Figure 5.6), and a practical evaluation of their individual performance. Then, the experiments performed to evaluate the overall system are described, with a comparative analysis of our solution against other state-of-the-art algorithms.

5.5 Coarse Face Detection - AdaBoost

This section describes the *Coarse Face Detection* stage of our system, in which the AdaBoost classifier provided by OpenCV was selected for use. Our motivation for using it is outlined first and some results obtained in this step are also analyzed.

5.5.1 Motivation to use the AdaBoost classifier

The first stage in our location system is a Coarse Face Detection, aimed at establishing an initial location of the face determining a bounding box around it. The following stages of the system are focused on refining the location given by this step. Locating a face though its bounding box is not a goal of this work; however, the final performance of our eye location system is linked to the performance of this step.

The Coarse Face Detection stage should essentially cover the following aspects:

- Considering these directives, it is important to select a face detection algorithm able to achieve high face detection rates, regardless of potential inaccuracies in the location (i.e., no facial elements are located, just the face as a whole). The output bounding box provides information regarding the regions of the image where a face is detected. The lack of a need for accurate location results may lead to the use of algorithms with low computational cost. These algorithms combine the use of fast techniques with an optimal image-scanning organization. The result is that even large images can be processed in a very short period of time.

- The selected algorithm for the Coarse Face Detection should minimize the number of *false alarms* per image. In this case, a false alarm is produced when there is a detection in a region where there is no true face. This point is less critical than the previous one, as a high number of false alarms can be reduced in subsequent stages, while a non-detected face in this first step would be inevitably lost for the remainder of the system.

This thesis uses the AdaBoost classification method proposed by Viola and Jones [113] to perform the Coarse Face Detection stage (specifically, the OpenCV implementation for frontal faces). The AdaBoost supervised method first performs a step of simple features extraction, in this case the Haar-like features, followed by a classification step. This structure fits well with the requirements of this stage: the simplicity of the Haar-like features in [113] produce efficient results for the detection of *face regions*.

In the training phase provided by OpenCV, the classifier is performed with a dataset of male and female individuals, with a resolution of 24 × 24 pixels. During the training stage, the AdaBoost does not learn rules for a single complex classifier, but rather for a cascade of multiple simple classifiers. This is known as the *boosting cascade*. When a sample is evaluated, it will be rejected by the classifier in the first stages of the cascades when it is efficiently detected as a negative, and only in the case of true positives and false alarms do the samples need to be classified by all the classifiers in the cascades. This fact significantly speeds up the classification process.

However, as mentioned before, the AdaBoost achieves high hit rates with low accuracy. This inaccuracy in the location of the face regions is mainly a result of the concatenation of three sources of error:

- *Resolution of the Training Faces*: Low image resolution implies lower accuracy of the detections performed at this stage. An example of a low resolution training image can be seen in Figure 5.2.
- *Image Scan Technique*: The AdaBoost needs to fully scan the input image to extract the regions which could contain faces. To avoid overloading the system with a high number of operations, the classification is sped up thanks to two factors: the intrinsic efficiency achieved when integral images are used to extract Haar-like features, and the fact that the image scan is non-intensive. The latter means that the image is not processed pixel by pixel, but with certain scanning steps. This inevitably leads to a loss in the precision of the face location.
- *Results Clustering*: After the scan of the image, the boosting classifier needs to cluster the clouds of detections around each potential face region into single and representative points. Usually, these clouds of points are clustered following a *neighborhood* criteria which ultimately adds more variability to the location of the central point of the detected face.

The inaccuracy of the location of the faces and the lack of any facial element point (e.g., the eyes), motivates the remaining stages of the system. Next, some experiments are presented to assess the use of the AdaBoost algorithm for the Coarse Face Detection stage of our system.

5.5.2 Validation of the Coarse Face Detection

To understand the global results of the proposed eye location approach, it is important to know the performance of the first stage. For the evaluation of the boosting classifier at the Coarse Face Detection stage some tests were performed. This tests were designed to work on four different face databases: the FERET database (due to its great variety of individuals); Experiment 4 images of the FRGCv2 dataset (mainly due to the great quantity of samples uncontrolled scenarios); and the Yale and AR datasets, as in these sets the images for each individual contain a controlled variety of expressions, face complements (glasses or scarves) and different grades of illumination[6]. In the case of FERET and FRGCv2, the datasets used in the experiments were directly extracted from the evaluation protocols of the databases.

In these experiments, a true positive was given when the system detected a face whose bounding box contained the face center given by the data groundtruth. Table 5.2 summarizes the performance of the experiment in terms of hit rates (i.e., actual faces labeled as *face*), and the number of false alarms (i.e., *non-face* regions labeled as a *face*).

[6] All these databases are detailed in Appendix B

FERET Dataset

Dataset (Nr. images)	**Hit Rate** (%)	**False Alarms**/image
gallery (1196)	99.75%	0.001
fb (1195)	99.67%	0.003
fc (194)	100%	0
dup1 (722)	99.86%	0.001
dup2 (234)	99.57%	0

FRGCv2 Dataset

Dataset (Nr. images)	**Hit Rate** (%)	**False Alarms**/image
Training (12776)	98.05%	0.054
Target (16028)	99.93%	0.023
Query (8014)	99.06%	0.0120

Yale and AR Datasets

Dataset (Nr. images)	**Hit Rate** (%)	**False Alarms** (per image)
Yale (165)	100%	0
AR (279)	96.77%	0.004

Table 5.2: Hit Rate and False Alarms for the AdaBoost classifier in Coarse Face Detection.

The analysis of the results shown in the table leads us to confirm that the two main requirements for the Coarse Face Detection are fulfilled:

1. For all the experiments, high hit rates were achieved– in some cases even 100%, meaning that all the faces in that set were correctly detected.

2. For all the experiments, an acceptably low number of false alarms was produced. Note that in FERET, the number of false alarms is around one in every thousand images, while in FRGCv2 it is one in every ten images. The explanation for this difference is found in the fact that most images in FERET show people with a plain background, while the great majority of the images in FRGCv2 were recorded in a realistic scenario with complex backgrounds. The probability of finding a false face in a plain background is much lower than finding it in a non-homogeneous scenario.

   For the case of the Yale and AR databases, the results for the number of false alarms are not significant; they tend towards zero, mainly due to the highly-controlled recording scenarios.

In conclusion, the AdaBoost classifier is confirmed to be a good selection for the first stage of our eye location system. The use of an open and widespread algorithm in this thesis has permitted to focus our attention in the following stages.

5.6 Extraction of Eye Candidates - Boosting

The second stage of the eye location system we propose is aimed to refine the information derived from the Coarse Face Detection by extracting eye candidates within the bounding boxes (from now on *face-like regions*) previously detected. As can be seen in Figure 5.6, this search for candidates is approached as a *Boosting Eye Location* stage.

This section is structured in the following parts: it starts by defining the architecture of the Boosting Eye Location stage, describing all its steps one by one. After that, the necessity of creating our own eye database for the boosting location is explained, and finally, the algorithms proposed are trained and evaluated.

5.6.1 Architecture of the Candidates Extraction

Unlike face location, the process of eye location needs to be highly accurate. Therefore, introducing a complex and advanced set of features to precisely describe the eyes might be necessary. A comparison between the original size of the input image and the size of the *face regions* reveals that in regular scenes, the latter are usually around 15% to 40% smaller than the former.

Despite the smaller size of the detected *face regions* compared to the size of the whole image, applying in such regions an intensive extraction of complex descriptors to locate the eyes would still require a large number of operations. To avoid increasing the computational cost, the inclusion of a stage prior to the complex features extraction is studied. In this thesis this stage is known as the Extraction of Eye Candidates, or simply the Boosting Eye Location.

The aim of this stage is to extract candidates for the left and the right eyes from detected face regions. It makes use of simple features to locate a controlled number of candidates, called *eye-candidates*. As in the previous stage, developing this task using a boosting classifier will allow an exhaustive location mechanism without triggering an increase in the computational cost.

Summarizing, the structure of the current stage can be organized into three different steps, as shown in Figure 5.7:

1. **Boosting Location**: This is a phase of exhaustive feature extraction in which a configuration of boosting classifiers is trained with Haar-like features. The outcome of this step is in the form of clouds of points corresponding to a set of right- and left-eye candidates. However, notice that at this point we do not distinguish between *left* and *right* candidates.

2. **Clustering of Candidate Clouds**: This step is aimed at clustering the previous clouds of detections into single eye candidates using a clustering approach. In this step, the quantity of eye candidates should be reduced to a low and controllable number.

3. **Eye Candidates Classification**: Finally, the nature of the clustered candidates has to be determined to classify them into *right eye* candidates and *left eye* candidates. The criteria used in this step is based on simple geometrical cues.

In the following sections, the previous steps are analyzed and evaluated.

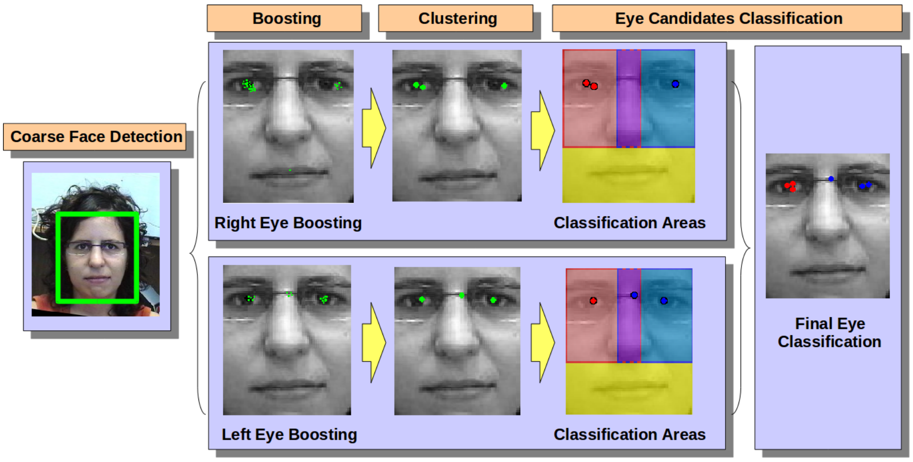


Figure 5.7: Diagram of the steps in the Coarse Eye Location stage.

5.6.2 Boosting Location

Similar to the Coarse Face Detection stage, the requirements to extract eye candidates are summarized in the following: achieving high hit rates (i.e., the actual eyes of a face should be present in the set of eye-candidates), along with a limited number of false alarms (i.e., a low number of false candidates). The intention of this step is to reduce the number of potential candidates within a *face region*.

The previous determining factors led us to use a boosting classifier using Haar-like features. The main motivation to use such classifiers is their proven effectiveness and low computational cost, as explained in the previous section.

In order to perform the training phase of the boosting classifier for eye-candidates extraction, we generated our own dataset, built up with positive and negative samples of eyes, as detailed in Section 5.6.4. Compared to the classifiers in the Coarse Face Detection stage, the eye boosting classifiers work with much higher resolution images: while the face samples had very low resolution, $r_{face} = 24 \times 24$ pixels, in the eye-candidate boosting classifier the resolution of the eye samples was actually of $r_{eye} = 15 \times 15$ pixels.

Three issues from the eye-candidate boosting classifier are discussed more in detail: the preprocessing prior to the classification, the configuration of the classifier itself and its geometrical restrictions.

Face Region Rescaling

In principle, the eye-candidate boosting cannot detect candidates with dimensions smaller than r_{eye} and neither smaller than r_{face}. This could be a problem with small-sized Faces, as from the Coarse Face Detection the smallest face-region that can be detected has a size of r_{face}; such face-regions contain eyes with sizes around 5 to 10 pixels, which smaller than r_{eye}.

To avoid these situations, a rescaling of the *face region* is performed prior to eye-candidates boosting. To determine a good solution for the rescaled size of the face region, a set-up experiment was performed. This test searched for a high number of hits (positive eye-candidates) with a low number of false positives (negative eye-candidates). The size of the rescaled face region was varied in the range $24 \leq r_{scaled} \leq 240$, which corresponds to the range $r_{scaled} \in [r_{face}, 10r_{face}]$.

The result of an empirical study in this thesis on the size of the rescaled face region was the following:

- The optimum size for the face region in order to perform the eye-candidates boosting is $r_{rescaled} = 115 \times 115$ pixels.

After this rescaling, the typical size of the eye regions in the face is around 15 to 40 pixels, which is always within the detection range.

Classifier Configuration

Regarding the eye-candidate boosting classifier, two feasible configurations were proposed to increase its robustness: *single classifier* and *double classifier*. Next the main features of the two configurations are summarized:

- *Single Classifier*: In this configuration, there is a unique boosting classifier. A single cascade of weak classifiers is trained using the whole dataset of samples, including images of right and left eyes.

  This configuration is based on the hypothesis that the similarity between the right and left eyes is high enough, so that the two eyes of a face can be detected with the same classifier. This speeds up the classification process, but may also reduce the performance as the assumption of the left and right eye being equal is not completely true.

- *Double Classifier*: In this configuration, two specialized boosting classifiers are trained to detect each the right eye and the left eye candidates, respectively.

  This configuration tries to exploit the differences between the right and the left eye to gain more accuracy. However, the use of two classifiers instead of only one doubles the number of operations during the boosting step.

The composition of the training sets for each of the two configurations of the classifier is detailed in Section 5.6.4.

Geometrical Restrictions

To limit the number of false alarms (i.e., negative candidates) produced by the boosting classifier in its two configurations –and also to increase the velocity of the stage– some simple geometrical restrictions on the area analyzed by the classifier are set. These restrictions are motivated by the fact that given a detected frontal face, the eyes are always located in the upper half of the face region, and thus the lower half can be excluded from the search.

Given a *face region* of size $h \times w$ pixels, where h is the height and w the width of the region, the search area, R, to extract eye candidates is:

$$R(x, y) = \{x | 0 \leq x \leq w, y | 0 \leq y \leq 0.6h\} \tag{5.1}$$

Depending on the classifier configuration, we find that:

- In the case of using the *single classifier* configuration, R is the scan area of the boosting classifier.

- In the case of using the *double classifier* configuration, the two classifiers work on the same area, R. The overlapping of the two boosting classifiers is intended to provide the algorithm with further consistency: the left eye boosting classifier may help to find right eyes, and vice versa. Also, when a face is not completely frontal (e.g., a face in half-profile), the eyes can be located displaces from its corresponding half of the face-region.

In Figure 5.8, the geometrical restrictions can be seen, restricting the search areas of the boosting classifier for both configurations. In the picture, the search area R corresponds to the union of the regions painted in blue, red and their intersection.

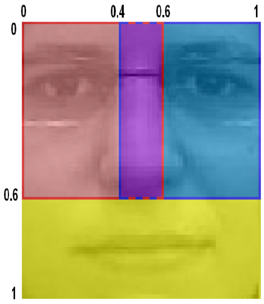


Figure 5.8: Diagram of the geometrical restrictions set to restrict the searching areas of the boosting classifier.

With these restrictions, 60% of the area that has to be scanned is skipped.

5.6.3 Clustering and Classification techniques

During the location of eye candidates, the boosting classifier exhaustively scans the *face region* delimited by the face bounding-box. Due to its nature, the boosting classifier is robust against small variations in translation and scale. This means that around each eye and each false alarm multiple detections will be found.

Therefore, every classification is presented as a *cloud* of detections around the candidates. These *clouds* make the number of detections, n_d, sometimes extremely large, even when the classifier is working properly. Figure 5.7 shows two examples of clouds of points around the eyes, specifically in the images corresponding to the *Boosting* column.

It is a goal of the Eye Candidates Location stage to reduce the information contained in a *face region* into a few candidates, potentially centered on the eyes of the individual. To fulfil this goal, it is critical that a technique to produce single eye candidates from the clouds of detections be discovered. This is equivalent to reducing as much as possible the number of final candidates per eye, $n_c < n_d$.

The extraction of single candidates is achieved through a two-step process. First, all the detections given by the boosting classifier, $d_k(x, y), 0 \leq k \leq n_d$, are clustered into candidate sets, $[A_1, A_2, \ldots, A_i]$, and then a unique representative candidate is given for each of these sets, $A_i \rightarrow c^i(x, y), 0 \leq i \leq n_c$.

Given a candidate set A_i, its representative candidate, c_i, is calculated as:

$$c^i(x, y) = \frac{1}{n_{di}} \sum_{n=1}^{n_{di}} d_{ki}(x, y), \forall d_{ki}(x, y) \in A_i, \tag{5.2}$$

where n_{di} is the number of detections, $d_{ki}(x, y)$, that generate the set A_i after the clustering step.

Regarding the reduction of the number of candidates, this thesis analyzes different clustering techniques, which can be found in the literature [50].

Generally speaking, the clustering techniques can be classified into two groups: the *Partitioning Methods* and the *Hierarchical Methods*. The *partitioning methods* comprise all those techniques in which there is an *a priori* knowledge about the precise number of sets that will be obtained after the clustering process; while in the *hierarchical* approaches, there is a clustering rule that conditions the way the elements are joined until this rule cannot be satisfied any more. In this case, the number of final sets is not known until the clustering process ends.

In our system, it is completely impossible to know *a priori* the final number of eye candidates, n_c, as it depends on the results obtained after the boosting classification. Therefore, as the *partitioning methods* for clustering do not fit with our problem, the focus of this thesis is on the *hierarchical* techniques.

The *hierarchical methods* are usually based on a bottom-up clustering approach. They start by assuming that each individual sample constitute their own set in the first iteration; then, recursively, each of these sets is clustered with those that fulfil distance rules, generating the so-called *tree-structure* or *dendrogram*, as Manning and Schütze detail in [70].

In Figure 5.9, samples and the tree-structure generated after applying *hierarchical* clustering can be seen. In this example, a minimum clustering distance, d, between clusters is established as a rule; the iterative process to generate such a tree-structure is shown for minimum distance values between d and $4d$.

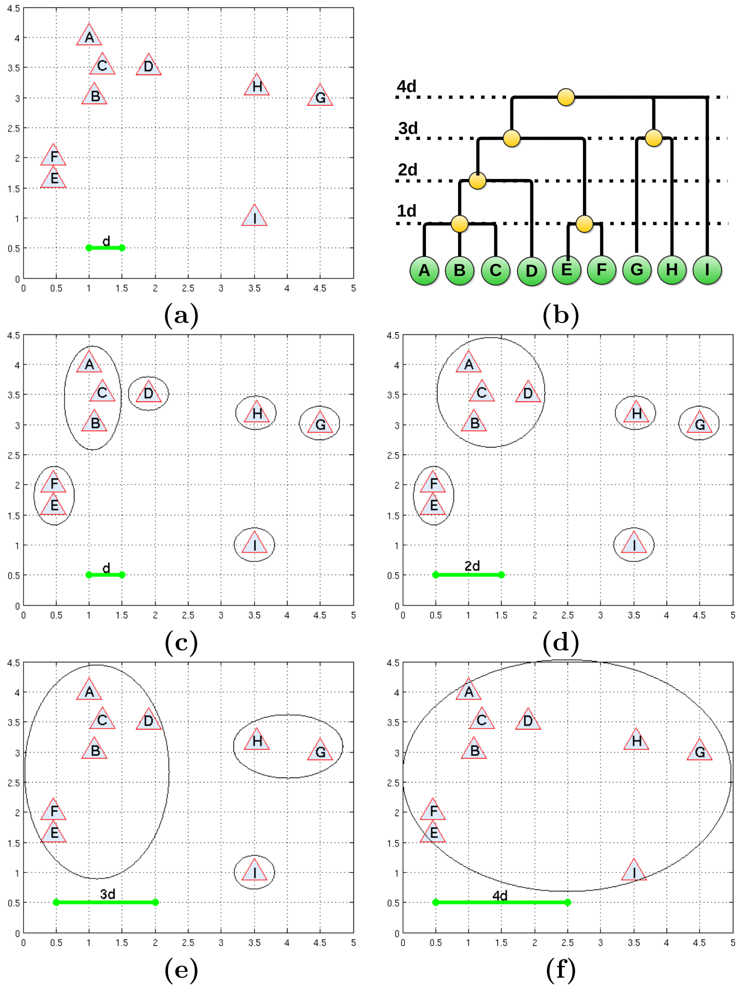
Single-linkage clustering

The most widely-used hierarchical clustering technique is the *single-linkage clustering*. This method is characterized by its simplicity: in each iteration, two sets are clustered if the closest pair of samples within them (each of the samples belonging to each of the sets) are at a distance lower than a maximum distance, d_{max}, defined as the clustering criterion.

Given two sets A and B, of sizes k_a and k_b and generated by the samples $a = (x_A, y_A)$ and $b = (x_B, y_B)$, respectively, a maximum distance between samples of the same set, d_{max}, is set such that the condition to cluster A and B is given by the expression $D_s(A, B) < d_{max}$. This distance between sets is given by:

$$D_s(A, B) = min_{ij}(dist(a_i, b_j)), 0 \leq i \leq k_a, 0 \leq j \leq k_b, \tag{5.3}$$

where $dist(a_i, b_j)$ is the Euclidean distance of the two samples. Even though other kinds of distances can be defined (such as the Manhattan, Mahalanobbis



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 5.9: Hierarchical clustering, with a distance criterion *d*. **a)** Initial sample data, **b)** Hierarchical tree for different distance criteria, **c)** to **f)** actual clustering with different distance criteria.

or the maximum norm), the Euclidean distance has empirically been proven to be sufficient for this problem.

During the design of the Eye Candidates Extraction stage in our system, the first experiments were performed with the single-linkage clustering. However, after some initial tests, it was verified that in some cases this method could not fulfil all the requirements of the stage. Specifically, there were cases where the results after the clustering were not accurate enough.

These results were a consequence of the nature of the detections given by the boosting stage. As mentioned earlier, the boosting classifier returns some clouds of detected points around the eyes, inhomogeneously scattered. Around a real eye, the clouds of points are frequently clustered in two regions: the center of the eye (more often around the pupil) and the inner corner of the eye.

In Figure 5.10, the left image shows a real example of the clouds of detections after the eye-candidates boosting stage, gathered around the eyes. In the figure, the two aforementioned regions can be clearly observed. When the eyes are clustered as shown in the figure, the performance of the stage decays considerably.

Regarding the distance criterion, d_{max}, two different problems can be observed:

- When *low* values of d_{max} are set, the clouds of points are clustered into several sets and thus generate large number of candidates, placed between the eye center and the corner of the eye. The main problem is that in most of the cases none of the candidates are finally located in the eye-center, which is the goal of this stage.
- When *large* values of d_{max} are set, all the detections around the eye center and all the detections around the corner of the eye are clustered into a single candidate, usually placed in the gap between the two areas, thereby decreasing the accuracy of the location. This example can be seen in Figure 5.10.

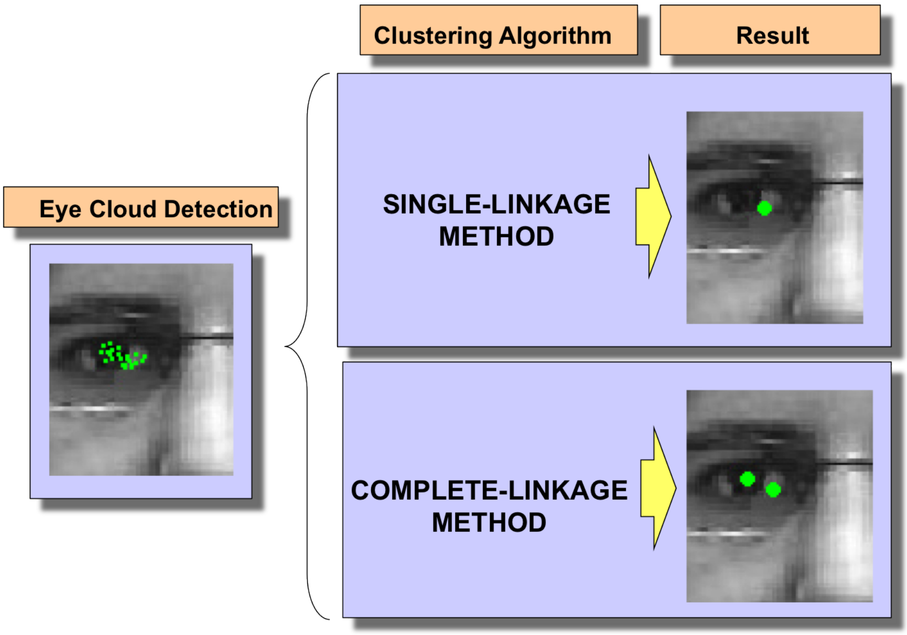


Figure 5.10: Example of a cloud of detections around the eye region and the clustering using two different clustering algorithms.

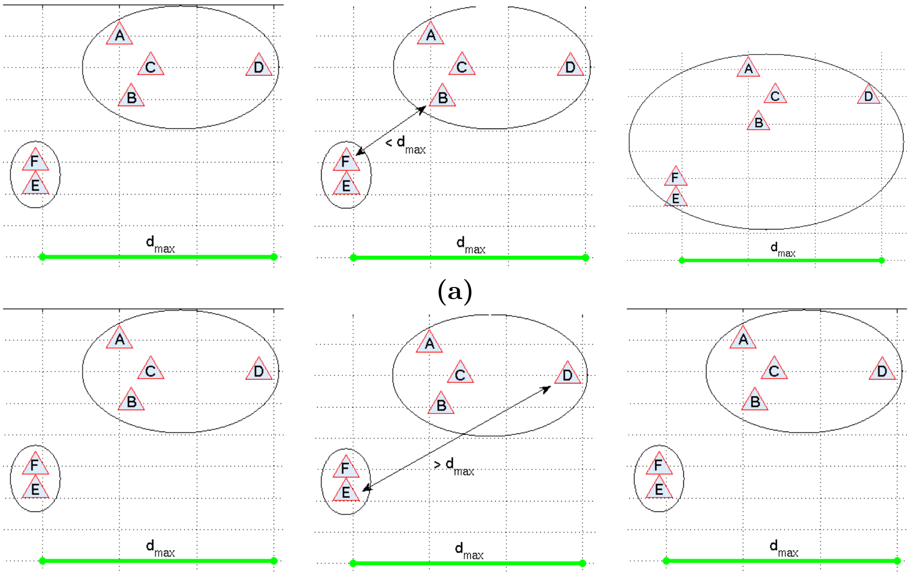
Complete-linkage clustering

To avoid the problems of the *single-linkage clustering*, a second branch of clustering methods to extract single eye candidates has been analyzed: the *complete linkage clustering* methods [50], also known as the *furthest neighbor* clustering. These techniques are iterative; on each iteration, two sets are clustered into one if the distance between the furthest pair of elements within them is lower than a maximum distance. This distance criterion is the same as in the *single-linkage clustering*, $D(A,B) < d_{max}$, but the distance itself is defined differently:

$$D(A,B) = max_{ij}(dist(a_i,b_j)), 0 \leq i \leq k_a, 0 \leq j \leq k_b, \tag{5.4}$$

where $dist(a_i,b_j)$ refers to a Euclidean distance.

Setting up the value of the distance d_{max}, the clustering produced by the complete linkage method normally leads to a greater number of sets of smaller size (i.e., they are more compact). Figure 5.11 shows a theoretical example of clustering performed with the *single-linkage clustering* and the *complete-linkage clustering*. Note the different results obtained, depending on the technique performed.



(a)

(b)

Figure 5.11: Hierarchical clustering for **a)** Single Linkage Method and **b)** Complete Linkage Method.

The disposition of the clusters obtained with the *complete-linkage clustering* method are in tune with the goals of this stage. With an appropriate d_{max}, the clouds of false positives that are placed around the corner of the eyes will not interfere in the location of the candidates corresponding to the clouds actually centered in the eyes, as they will be clustered as different sets.

Figure 5.10 depicts a real example of the extraction of eye-candidates on a face region after applying single-linkage and complete-linkage clustering techniques. As mentioned in Section 5.6.2, the detected *face regions* are rescaled to a size of $r_{rescaled} = 115 \times 115$ pixels, allowing a constant value for the maximum distance to be fixed, d_{max}. For our eye location system, we empirically arrived at the conclusion that the optimum distance is $d_{max} = 16$ pixels, which corresponds to distances usually around 20% of the inter-ocular distance. This distance maximized the number of cluster detected around a circumference of radius $d \leq 0.5iod$, using the databases of YALE and AR[7].

It should be noted that the distance value obtained with the experimentation is less than half the size of an eye.

Candidates classification

After the clustering, the last step of the eye-candidates extraction consists of a simple classification. This is done by labeling each candidate with its position in the *face region*: *right eye-candidate* or *left eye-candidate*, respectively. If the boosting classifier is in *double configuration*, all the candidates are labeled without taking into account if the detections came from the right-eye boosting classifier or the left-eye boosting classifier. To ensure more stability in the system, a certain degree of redundancy has been favored: to exploit the similarities of the right and left eye, the right-eye boosting classifier is allowed to detect left eye-candidates and vice versa. The labeling process is completely independent from the result of the classifier, as seen in Figure 5.7. For example, a candidate detected by the left-eye classifier should not be necessarily labeled as a left-eye. The labeling process will be performed in a subsequent stage depending on geometrical restrictions.

5.6.4 Training of the Boosting Classifier

The stage of Eye Candidate Extraction works on the detections given by a supervised boosting classifier, as mentioned earlier.

During the training phase of the classifier, the discriminative features are selected, and the classification rules are learned from them. The performance of the classifier directly depends on the nature of the features (Haar-like, subtraction of pixels, etc.), but also on how representative the training samples are, both positive and negative.

In this work, a large and consistent dataset has been exclusively developed to fulfil the requirements of the eye-candidates boosting classifier. This dataset consists of a large number of *eye* images and also *non-eye* images, both directly generated from the face images of the BioID public database [8]. The eye samples extracted from these images are not only cropped, but also include some rotation, scaling and post-processing has also been applied. From the same face images, the two training subsets have been extracted: positive samples (i.e., *eye* samples), and negative samples (i.e., *non-eye* samples).

Before the extraction of the samples from the images, the faces are geometrically normalized using the ground-truth position of the eyes. The result of this normalization is a face image of size 125×160, with the eyes located in the coordinates $(25, 30)$ and $(100, 30)$, respectively.

Next, the positive and the negative samples of the eye dataset are described:

- **Positive samples**: They are the models of the elements to be detected (the eyes). To build this subset, we generated positive samples by extracting two windows of size $r_{eye} = 15 \times 15$ pixels from the normalized face images, centered in the middle of the eye. Originally, up to 3400 eye

[7] See Appendix B

[8] See Appendix B

images were selected, corresponding to the face samples of 25 different individuals. The eyes in this set were selected in different gestures to cover a wider range of situations (opened, partially closed, frontal gaze, lateral gaze, etc.).

In real images, usually when a face is detected the eyes are not horizontally aligned with the borders of the face bounding-box. The result of this is that an eye may present different aspects regarding the specific degree of misalignment. Also, in real images is easy to find noise coming from different sources.

As Li *et al.* probe in their work [62], artificially incrementing the set of samples to be classified by AdaBoost can lead to a better performance when the new samples add significant information. In our case, a good way of incrementing the number of positive candidates in the training set was to generate new images directly derived from the initial set of eyes. Summarizing, the number of images was artificially augmented by two means:

– All the samples were rotated, taking the center of the eye as the fixed point, a total of $0°$, $\pm 2°$ y $\pm 4°$.

– Random Gaussian white noise was added to all the samples, including the samples generated by rotation, to emulate the noise that can be found in real scenarios.

If I_{eye} is defined as the initial set of samples (i.e., the initial 3400 eye images), the final set of samples, directly derived from the first, $\tilde{I}_{eye}$, is defined as:

$$\tilde{I}_{eye}(i) = \Delta_\theta I_{eye}(i) + N(i),\ \theta = 0°, \pm 2°, \pm 4°, \tag{5.5}$$

where Δ_θ is the transformation matrix that rotates the original eye image a total of θ degrees, i is the current sample and $N(i)$ is a random component of white Gaussian noise. Considering a grayscale image with pixel intensities in the range $0 \leq I(i) \leq 1$, the Gaussian noise $N(i)$ is generated with zero mean, $\mu = 0$, and a standard deviation of 10%. The modelation of noise in many cameras can be modelled with a Gaussian with the parameters before mentioned, as it is proved in the work of Irie *et al.* [49].

After the combination of both processing steps, the final set derived from the original set of positive samples consists of more than 60.000 images for each type of eye (right and left). In the *eye candidate boosting classifier*, the main difference between the *single* and *double* configurations is the composition of the subsets of positive samples. In the *single classifier* configuration, the boosting cascade of classifiers is trained using the 120.000 samples corresponding to both subsets, the right and the left eyes. In the case of the *double classifier*, each of the boosting cascades is trained using only one of the subsets, the right or the left, respectively.

- **Negative samples**: They are an example of what an eye *does not* look like. Regarding this set, it must be borne in mind that the eye-candidate extraction is performed directly on previously detected *face regions*. Therefore, the negative samples should be extracted from all the areas in a face except for the eye regions.

Starting from the BioID normalized face images, the negative samples were generated by applying a low pass filter in two square patches of size 50×50 and each centered on the middle point of the eyes.

Figure 5.12 shows some examples of positive and negative samples used to train the eye-candidates boosting classifier. In the case of the positive samples, both the original images directly extracted from the BioID database and the processed images with rotation and noise addition can be seen.

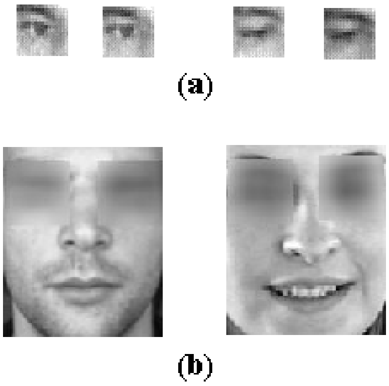


Figure 5.12: **a)** Positive samples and **b)** images from where negative samples are extracted and used to train the boosting classifier for the coarse eye location.

In the following section, a validation of this stage is given as a stand-alone subsystem, without taking into account the remaining parts of the eye location system.

5.6.5 Validation of the Eye-Candidates Extraction Stage

In this section, some evaluation results are obtained after applying the boosting stage for the eye-candidates extraction. These experiments try to prove that this stage generates a limited number of eye-candidates with high reliability; there should be at least one true candidate for the right eye and one for the left eye in the list of candidates. Also, the number of false alarms among the candidates should stay at some controlled levels.

Before running the performance experiments and selecting an appropriate dataset, some internal parameters have to be tuned during the training phase (independent of the configuration of the classifier selected).

Next, the tuning and the performance experiments are explained, along with the results obtained.

Tuning of the boosting classification cascades

One of the most important aspects of parameters during the training of the boosting technique is the number of weak classifiers that constitute the training cascade (known as the number of stages of the cascade). Generally, a high number of weak classifiers trained will produce accurate results (with a low number of false detections), but the classifier may also lose some true positives (in our case, some eyes might not be detected). On the contrary, reducing the number of stages in the boosting cascade would increase the recall of the detector, producing a higher number of detections (usually we can reach a 100% of eyes detected), but at the expense of incrementing the number of false alarms. In this case, the classifier could become useless, as it would not be able to discern between true and false positives.

To avoid both extremes, a trade-off between having high detection rates and an affordable number of false alarms had to be reached. For the experiments, the three classifying cascades (one for the *single classifier* configuration, and two for the *double classifier* configuration) were trained with the same number of weak classifiers, $NC_{initial} = 15$. The value of $NC_{initial}$ was automatically fixed for a given classification rate of 99.99% of positive classifications on the training data. Then, the number of stages in each of the cascades was reduced until the performance on the training set was close to 99% (decreasing the probability of false alarms as much as possible).

- After the cascade tuning experiment, the number of stages in the three boosting classification cascades was set to $NC_{optimal} = 12$.

Further on, all the experiments were performed considering this number of stages of the classifying cascade.

Validation of the boosting classifier

A series of experiments were designed to evaluate the performance of the Eye-Candidates Extraction stage. In these experiments, the extraction algorithm is applied to a set of evaluation images, and the location of the eye-candidates extracted is compared to the ground-truth positions of the actual eyes of the faces.

To evaluate the method, the performance measures introduced in Section 3.3 were used. If the error in the location of the eye was $N_{error} \leq 5\%$, relative to the inter-ocular distance, the eye was considered a positive. This criterion is quite demanding compared to the most commonly-used criterion in the literature, where the threshold is set to $N_{error} \leq 25\%$.

The experiments were performed using the FRGCv2 Experiment 4[9] dataset, motivated by its high number of images, and also to be consistent with the validation tests performed in the previous stage.

Table 5.3 summarizes the results of the experiments. These results are the hit rate and the number of false alarms obtained after applying the boosting stage for eye-candidates location. Specifically, the hit rate in this experiment is the percentage of face images in which at least a valid candidate is extracted for each of the eyes. The number of false alarms is the number of false candidates extracted around each of the eyes.

[9] See Appendix B

Given that the boosting classifier in this stage has been designed to perform with two different configurations, the *single* and the *double* (see Section 5.6.2), the table shows the results for each of them. To make the evaluation for both configurations consistent, the hit rate always makes reference to the percentage of faces in which a true candidate for both eyes was extracted simultaneously.

Simple Classifier Configuration		
Dataset (Nr. images)	**Hit Rate** (%)	**False Alarms**/eye
Training (12776)	92.87%	0.396
Target (16028)	97.86%	0.463
Query (8014)	90.69%	0.259

Double Classifier Configuration		
Dataset (Nr. images)	**Hit Rate** (%)	**False Alarms**/eye
Training (12776)	95.35%	0.630
Target (16028)	99.33%	0.660
Query (8014)	93.15%	0.554

Table 5.3: Hit Rates and False Alarms with the eye-candidates boosting classifier using two configurations on FRGCv2 Experiment 4.

A first analysis of the results shows the following results:

- The boosting classifiers extract eye-candidates with high rate values, making them suitable for our system. In all of the tests performed, the hit rate was over 93%.

  It should be noted that the results in this stage accumulate the errors from previous stages, as a non-detected face in the first stage is counted as a failure. Also, the number of false alarms detected around each eye is lower than one per image, which increases the reliability of the stage.

- Comparing the performance obtained with the two boosting configurations, it can be concluded that for all cases the *double classifier* achieves a higher performance. In other words, when two different boosting cascades are used, one for each of the eyes (left and right), the results are more accurate than using a single boosting cascade for both.

In subsequent phases of our system, the eye-candidates boosting were always configured using the *double classifier*.

5.7 Feature-based Eye-Candidate Selection

The two first stages of the eye location system proposed in this thesis extract information from detected faces, producing a preselection of eye-candidates, labeled *right-eye candidates* and *left-eye candidates*.

In this section, the selection of the optimal eye-candidates is described to finally decide on the precise location of the eyes. The synthesis from multiple candidates to the final eye-pair is done using a feature-based mechanism based on the extraction of HOG descriptors and a later classification step using SVM machines.

This section starts describing the global architecture of the stage, followed by the description of each of the individual steps. Finally, some validation experiments on the SVM classifier are performed.

5.7.1 Architecture of the Eye-Candidates Selection

After the previous boosting stages, the eye location system has to deal with an undetermined number of eye-candidates. To determine the final location of the eyes, all possible combinations of *eye-pair candidates* are generated, always considering pairs constituted by right and left eye-candidates, and then determining which eye-pair defines the location of the face with greater accuracy. In this text, the eye-pair candidates are also called *face-candidates*, as an eye-pair combination generates a potential face-like image.

The current stage is aimed at moving from eye-candidates to face-candidates and extracting discriminative information from them to perform a classification. Thus, for every detected face a total of $C_k, 0 < k \leq n_{candidates}$ face-candidates is generated, where $n_{candidates}$ is the final number of eye-pair candidates.

The output of the boosting classifiers is conditioned by the trade-off between detection rates and false alarms. From the results obtained during the validation of the previous stage, it can be seen that the number of false alarms is too high for the goals and requirements of the final system.

In the current stage, when the eye-candidates are paired to generate face-candidates, the previous false alarms multiply the errors in the current stage. One single negative eye candidate generates several negative eye-pair combinations. To avoid these false alarms deteriorating system performance, it becomes necessary to find some mechanism to discriminate between positive and false eye-pair candidates. In other words, the goal of this stage is to perform an advanced mechanism to select the best face candidate, without incrementing the computational cost.

The architecture of the eye-candidates selection stage can be summarized in the following steps:

1. **Generation and Normalization of face-candidates**: All eye-candidates are combined to generate normalized face-candidate images. These face-candidates place each pair of eye candidates in fixed positions. The goal is to determine the best face-candidate, that is, the one having the most accurate eye-candidates combination.

   The normalization of the face-candidates helps avoid problems with the scaling or the rotation of the eyes. This can be useful for the extraction of local texture features.

2. **HOG Features Extraction**: After the normalization of the face-candidates, some local descriptors extract biometric information from the *a priori*-known locations of the eyes. In this thesis, the local texture features

used for this goal are the Histograms of Oriented Gradients (HOG) descriptors[10]. The decision to use these features was made based on the high descriptive power of the HOG features, reinforced by the previous normalization step, which reduces the variability in the eyes.

In this step, two different feature vectors are extracted for every face-candidate, covering both of the *a priori* known eye locations. However, the information of these vectors is somehow combined later, producing a unique signature for each combination of eye-candidates.

3. **SVM classification**: The selection of the optimum eye-pair after the normalization of the face candidates is based on the information extracted from the eye locations using HOG descriptors. The feature vectors of each face-candidate are used to train a Support Vector Machines (SVM) classifier.

   The SVM [29] is a supervised classifier that sets a non-linear separation among positive and negative samples. In our system, the SVM determines whether the potential eye-pair of the face-candidate is a true positive or a false alarm (i.e., with at least one of the candidate not being an eye). Along with the classification results, the SVM provides a soft-output, which is an indicator on how reliable that classification is. Finally, the best eye location corresponds to the eye-pair classified as a true positive and having the highest reliability.

Next, each of the steps integrating the Eye-Candidates Selection stage is described more in detail.

5.7.2 Generation and Normalization of face-candidates

To select the best eye-pair (in terms of accuracy) from all the eye-candidates, all possible combinations of right and left eye-candidates are explored. Given a *face region* following the Eye-Candidates Extraction stage, the boosting classifier has n_{CR} samples labeled right eye-candidates, $c_R^i(x,y), 1 \leq i \leq n_{CR}$ and n_{CL} samples labeled left eye-candidates, $c_L^j(x,y), 1 \leq j \leq n_{CL}$. A total of $n_{comb} = n_{CR} \times n_{CL}$ eye-pair combinations are then performed. From these combinations a equivalent (or lower) number of normalized face candidates, $n_{candidates} \leq n_{comb}$, is extracted.

The possibility of obtaining less face-candidates than eye-pair combinations is given by the fact that some geometrical restrictions can be applied, so that some combinations that logically cannot come from positive candidates can be rapidly discarded. The geometrical restrictions can be summarized in two points:

- A minimum separation distance between the right and the left eye-candidates, iod_{min}, is required. The reason for this restriction is purely physiological: it is known that the distance between eyes is proportional to the width of a face and thus, eye-pairs with very close candidates contain at least one negative candidate with high probability.

[10] See Chapter 4

- The eye-pairs in which the candidates are at an angle with the horizontal of the face greater than a maximum angle, α_{max} can also be disregarded. This limitation is imposed by the specifications of the Coarse Face Detection stage, as the AdaBoost used is limited to detecting *faces* with an inclination lower than α_{max}; any eye-pair with higher angles inevitably has at least one negative eye-candidate.

Some problems are derived from the case in which given a detected *face region*, the eye-candidate boosting fails to detect at least one of the two eyes (i.e., $n_{CR} = 0$ or $n_{CL} = 0$), added to the case where there is no valid combination of eye-pairs after applying the geometrical restrictions. In both cases, the number of final eye-pair candidates is *null*, $n_{candidates} = 0$ and thus the *face region* is considered a false alarm from the Coarse Face Detection stage. This way, the extraction of eyes becomes a mechanism to reduce the number of false faces detected by the AdaBoost classification in the first stage; nevertheless, poor performance in the later stages could also reduce the number of true detections.

Figure 5.13 displays a real example of the face-candidates generated from all combinations of the eye-candidates detected in a face-region. In this example, two right and three left eye-candidates were extracted in the boosting stage, and from them a total of six eye-pair combinations were generated. Note that some of the face candidates corresponding to the eye-pair combinations with at least one negative candidate do present a very different aspect from the rest of the faces.

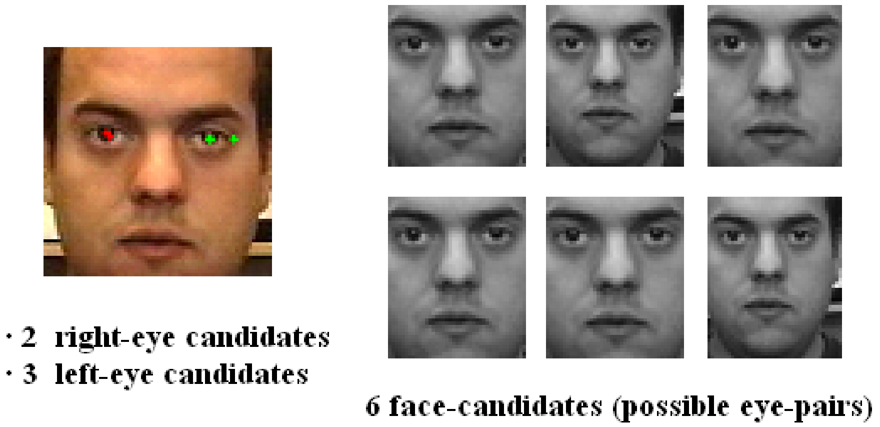


Figure 5.13: Face candidates extracted from all the eye-candidate combinations from a single *face region*.

Following the specifications of the AdaBoost classifier and the cascade for frontal faces provided by OpenCV, in this work the maximum angle is set to $\alpha_{max} = \pm 20°$. With regard to the criterion of minimum distance between eyes, considering that all the detected *face regions* are rescaled to a fixed size of $r_{rescaled} = 115 \times 115$ pixels, the distance iod_{min} is set empirically to 20 pixels, which is between the 15% and the 20% of the width of the face.

From every eye-pair combination, a normalized face candidate of size 125×145 is generated. In this normalized face, the left and right eye-candidates are located in fixed positions, $r(x, y) = (25, 35)$ and $l(x, y) = (100, 35)$, respectively, establishing an interocular distance of $iod = 75$ pixels.

5.7.3 HOG Features for Eye Description

After the normalization of the face-candidates, a single feature descriptor, D_k, is generated for each of them. This feature descriptor consists of the concatenation of the two feature vectors obtained after the extraction of local texture features from the two eye locations of the face-candidate. That is:

$$D_k = descr(\overline{c^i_{Rk}}) \cup descr(\overline{c^j_{Lk}}), 0 < k \leq n_{combined}, \tag{5.6}$$

where the function $descr(x)$ makes reference to the extraction of a local descriptor on the coordinates of a specific candidate, x, and $\overline{c^i_{Rk}}$ and $\overline{c^j_{Lk}}$ are the left and right eye candidates c^i_R and c^j_L after the normalization of the face C_k.

It is important to remark that each eye-candidate, for both the right and the left (c^i_R or c^j_L), generates not only one face-candidate, but a whole *bunch* of them ($\overline{c^i_{Rk}}$ and $\overline{c^j_{Lk}}$, respectively). Each of the face candidates is completely independent, and therefore contains unique information.

Intuitively, it is easy to understand that the information given by each eye-candidate on a face-candidate is relative to the information provided by the complementary eye-candidate. For example, a true right eye-candidate will produce a positive face-candidate when combined with a true left eye-candidate, but it will produce a negative (distorted) face-candidate when combined with a false left eye-candidate. The distortion of the negative face candidates is seen as a misalignment and a poor scaling of the normalized image. In Figure 5.13, some examples of positive and negative normalized face candidates and their different aspects are shown.

During the classification of eye-pairs, the most sensitive case is given when an *eye* candidate is combined with a *non-eye* candidate. A real example of this combination is displayed in Figure 5.15.c. In this case, although the left eye-candidate is correctly located, the eye-pair should be classified as negative. Thus, the necessity of using a precise selection mechanism. The design of such a mechanism is influenced by the following issues:

- The local descriptors are extracted only on preselected points and their location in the normalized faces is fixed. This helps avoid intensive scans and allows the use of complex local descriptors without increasing the computational load.

- After their normalization, the positive face-candidates present a similar aspect, while the aspect of the negative candidates tends to be distorted in scale and rotation. Thus, the descriptor selected can be less restrictive, as the invariance to these factors for the positive samples is given beforehand.

A great variety of local descriptors can be selected to fulfil these requirements (as shown in Chapter 4). The results obtained in recognition tasks led us to examine the use of the Histograms of Oriented Gradients (HOG), which are local statistics of the orientation of each pixel in an area around a central point and the landmark we want to describe. The use of HOG descriptors is novel in the matter of fine eye location.

If the HOG features are considered, the equation 5.6 that represents the most significant information for each eye-pair becomes:

$$D_k(\overline{c^i_{Rk}}, \overline{c^j_{Lk}}) = HOG_{C_k}(25, 35) \cup HOG_{C_k}(100, 35), 0 < k \leq n_{nface} \qquad (5.7)$$

where $HOG_{C_k}(x, y)$ represents the extraction of a HOG local descriptor in centered in the pixel with coordinates x and y of the candidate image C_k.

Regarding the internal parameters of the HOG descriptors (detailed in Chapter 4), in this stage we used the standards defined by Lowe [65]: a square layout with $N_p = 4$ spatial cells and $N_o = 8$ orientation bins per histogram, producing a HOG feature vector of $N_p^2 \times N_o = 128$ elements. The selection of these values was motivated by the results obtained in Lowe's work.

Also, from the results presented in the experiments in Section 6.6.1, the size (in pixels) of each spatial cell is set to 6×6 pixels, producing a square HOG region of 24×24 pixels.

Finally, regarding the Gaussian envelope that weights the module of the gradients for each pixel in the area described, empirical results show that its optimal workpoint is achieved when the function is centered on the landmark and has a standard deviation equivalent to half the size of the HOG region (which in our case is $\sigma = 12$ pixels).

Next, the classification of the HOG features to discriminate between positive and negative samples of the eye-pairs is described.

5.7.4 SVM classification of HOG features

After selecting the HOG descriptors to describe the eye locations in the normalized face-candidates, it is necessary to design a methodology to determine the most representative eye-pair. The best option is to use a classifier able discriminate between the positive eye-pairs (where both eye-candidates are positive) from negative eye-pairs (in which, at least one is a negative eye-candidate).

The selection of the classifier was determined by a series of context factors, summarized in the following:

- Running the previous stages of the system on training images generates as many positive and negative normalized eye-pair as necessary. This led us to select a supervised learning classifier. The more representative a set of samples, the better performance the classifier will achieve.

- The main goal in this step was to select the optimum eye-pair, and therefore it is critical to obtain a confidence value from each of the samples classified. A higher confidence implies in this case a better location of the positive samples.

- As this stage provides the final location with higher precision, the boosting solution used in previous stages did not fit here. In the previous stage, the eye-candidates were extracted through an exhaustive scan performed at a pixel level. On the contrary, the classifier in the current step does not work directly on the image information, but on local HOG descriptors extracted from specific locations.

- Due to the complexity of the problem, the high dimensionality of the data and the lack of information about its distribution in the feature

> space, it was difficult to achieve good discrimination using linear classifiers: a *non-linear criterion* appeared to be more practical. The decision of using SVM was taken considering its high accuracy, ability to deal with high-dimensional data such as local texture descriptors, and flexibility in modeling diverse sources of data. By the use of kernel functions, SVM gains flexibility in the choice of the form of the decision surfaces that separate eyes from non-eyes, which needs not be linear.

Considering these factors, the classifier selected was the Support Vector Machines (SVM) [111].

SVM classifiers usually treat the problem of classifying samples as a binary problem. Given an initial feature space where all the feature vectors from positive and negative samples are placed, the goal of the SVM is to define a discriminant criterion to classify them. However, as the samples in the feature space usually follow a complex distribution, the SVM cannot solve this problem at first as a linear classification. To convert a non-linear problem into a linear one, the SVM creates a hyperspace, where the samples have greater dimensionality than in the feature space. This hyperspace is generated using some *kernel functions* combined via dot products. The sample distribution in the new space intends to be simpler than in the original one. If this is the case, the samples are linearly separable by means of a hyperplane that constitutes the non-linear separation criterion in the feature space. The selection of an appropriate kernel function determines the distribution of the samples in the hyperspace, and thus the discriminant criterion that is learned, in the feature space. Figure 5.14 shows an example of two-dimensional data classified with linear, polynomial and Gaussian kernels.

The SVM classifiers also provide *soft* output, which means that the classified samples are not just labeled with a class, but also obtain a value that measures the confidence in that classification. With the SVM classifier, the samples are labeled positive and negative, and a real value indicates how close such samples are to the separating criterion (coming from the hyperplane in the projected space); the samples with higher confidence are farther from said hyperplane, which means they are easily discriminated from potential outliers, closer to the hyperplane.

In a first set of preliminary experiments, two ways of combining the descriptors from both candidates in each eye-pair were studied: first, a *previous combination* approach was tried, which consisted of concatenating the feature vectors from the candidates before performing the SVM classification, and a second option was to perform the SVM classification independently on each of the eyes of the eye-pair, and then do a *subsequent combination* of the classification results. Using the *subsequent combination* configuration of the SVM, the final confidence value from an eye-pair is the sum of the individual confidence values obtained separately for each of the candidates.

The preliminary results obtained for the two configurations of the SVM classifier showed results similar to those obtained after experimenting with the two configurations of the boosting classifier (Section 5.6). The independent classification of samples in the *subsequent combination* performs better than the combined classification using the *previous combination.* Therefore, the *subsequent combination* configuration was selected for further experiments.

For simplicity, the positive samples of the eye-pairs are henceforth called

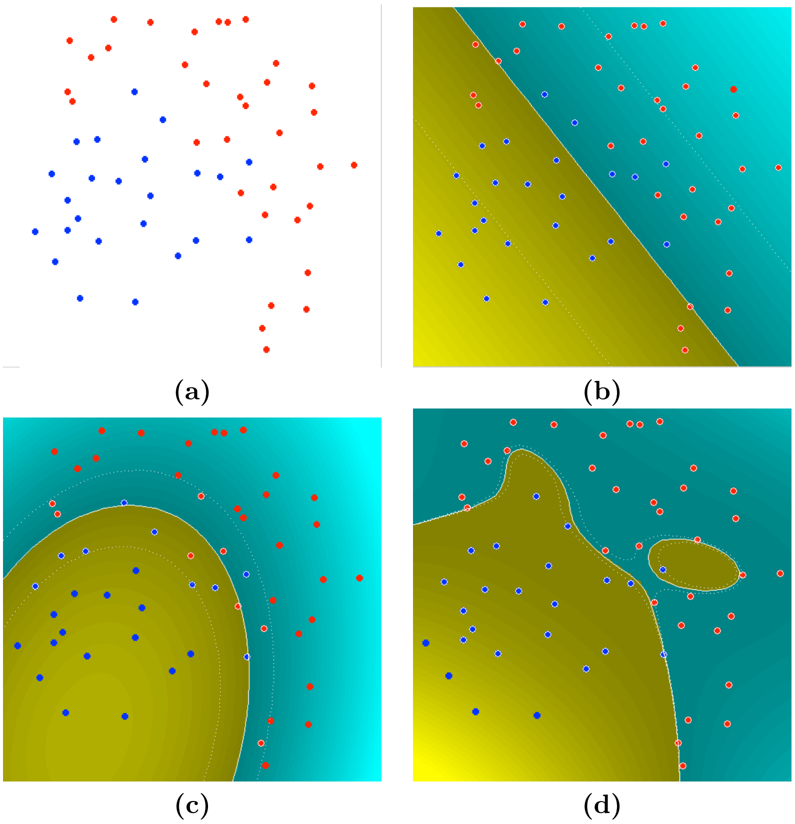


Figure 5.14: Example of discrimination of a set of samples using SVM with different kernels: a) Original sample set, b) Linear, c) Polynomial and d) Gaussian discriminations.

eyes in this text, while the negative samples, which are the false alarms from the previous stage, are called *non-eyes*. In this work, negative eye-pairs are those containing one or two candidates labeled as *non-eyes*. Also, all the images where all the eye-pair combinations are classified as negative are considered false alarms.

The following part of this section is aimed at studying the training and setting of the SVM classifier to adapt its performance to the specific problem of fine eye location.

5.7.5 Training Methodology for the SVM classifier

To set up the SVM supervised classifier, two requirements should be considered:

1. The training sample set is organized into two groups, positive samples (*eyes*) and negative samples (*non-eyes*). These groups need to contain a wide variety of significant samples, as the classification rules to be learned must cover a wide range of cases. Also, all training samples have to be produced under the same conditions as those produced during the validation and test of the system.

2. A kernel function has to be selected to determine the distribution of the positive and negative samples in the feature space. This distribution is not known beforehand, and therefore the experiments should lead to choosing the best option to separate the samples more precisely.

Next, the composition of the training datasets generated to train the SVM classifier is detailed. Then, the last section of this chapter is devoted to explaining the experiments performed to determine the best kernel function fitting our problem.

Generation of the training subsets for the SVM classifier

The samples classified by the SVM are the HOG feature vectors that describe the region centered on the eye locations of the normalized face-candidates. To generate the training samples for the SVM classifier, it is fundamental to emulate all the previous processing –eye-candidates extraction, eye-pair combination and face-candidate normalization– step by step. From each training image, we expected to extract up to two positive feature vectors from the eyes, and an undefined number of negative feature vectors from the false alarms. The diagram in Figure 5.15 shows a few samples used to train the SVM classifier.

To generate the training samples, a subset of images of BioID and the CVL datasets[11], consisting of 223 images that belong to 118 individuals was required. The need for ground-truth information about the location of the eyes motivated the selection of these two datasets.

As a result of the boosting stages on the training samples, near 2000 unlabeled eye-candidates were obtained, containing both positive (real eyes) and negative (false alarms) samples. Due to the high quantity of false alarms (around three per eye), it became necessary to automatically classify them. The criterion to perform this labeling was based on the ground-truth information about the actual eye locations, compared to the location obtained for each candidate. A sample was considered positive when its actual distance error was $N_{error} < 5\%$ of the *iod*. This error is approximately equivalent to the size of the pupil. The rest of the detections were considered negative candidates. With this criterion, approximately one third of the 2000 samples were labeled positives and two thirds, negatives (false alarms).

Depending on the eye-candidates, the normalized face-candidates that are generated can be classified into three different categories:

- **Positive Face-Candidates**: Normalized faces generated by the combination of two positive eye-candidates. These faces present a homogeneous aspect regarding the orientation, relative position and scaling of the face, and produce positive feature vectors for both eyes. An example of this category is displayed in Figure 5.15.a.

- **Negative Face-Candidates**: Normalized faces generated by the combination of two negative eye candidates (i.e., two false alarms). These faces are highly distorted in relative position, rotation and scale. The feature vectors extracted from the fixed eye locations produce two negative samples. An example of this category is displayed in Figure 5.15.b.

[11] See Appendix B

- **Hybrid Face-Candidates**: Normalized faces generated by the combination of one positive and one negative eye-candidate. In this set, the faces are distorted only to a certain degree. In some cases, the distortion can be so slight that it would be difficult to be detected by the naked eye. The feature vector extracted from the negative sample can be clearly used as a negative sample. However, it is difficult to define the utility of the positive candidate, which is centered on an actual eye. Although it represents a true sample in location, the complementary eye produces a distortion in scale and rotation of the normalized candidate. Therefore, these samples will not be considered for the training phase. An example of a hybrid normalized face is is displayed in Figure 5.15.c.

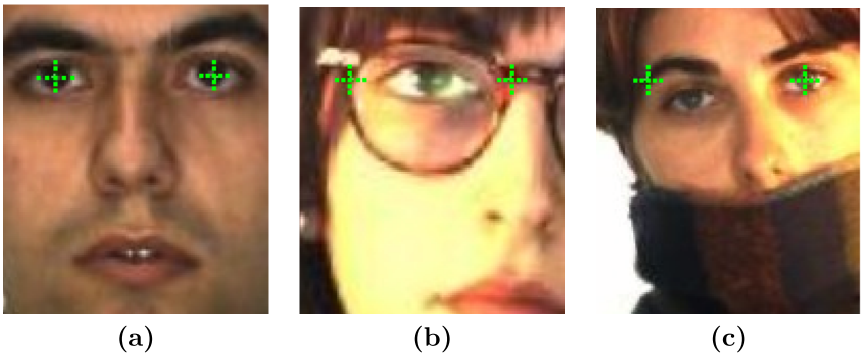


Figure 5.15: Example of the three categories of normalized face candidates: a) Positive Face Candidate, b) Negative Face Candidate and c) Hybrid Face Candidate.

After extracting both HOG feature vectors from the eye-pairs generated with the 2000 eye-candidate samples, up to 11078 positive samples (two from each positive face-candidate) and 22160 negative samples (from the negative and hybrid face-candidates) were obtained. With the size of these training sample sets, a representative variety of samples was obtained.

Study of the optimum kernel function in the SVM classifier

Once the subsets of positive and negative samples are generated, the next step consists of analyzing their distribution in the initial feature space to perform the optimal classification. There are no previous works in the literature that use SVM classifiers on HOG descriptors for eye classification. Therefore, there is no criteria by which kernel functions may address the eye location problem. This forced us to study the selection of an appropriate kernel function.

In this thesis, two kernel functions were used for the SVM classifier: Polynomial and Radial Basis Function (RBF). Both kernels are parametric, so we needed to vary their parameters to find the best configuration. Specifically, the equations that describe the projection of the original samples to the hyperspace using these two kernels are as follows (see [29] for further details):

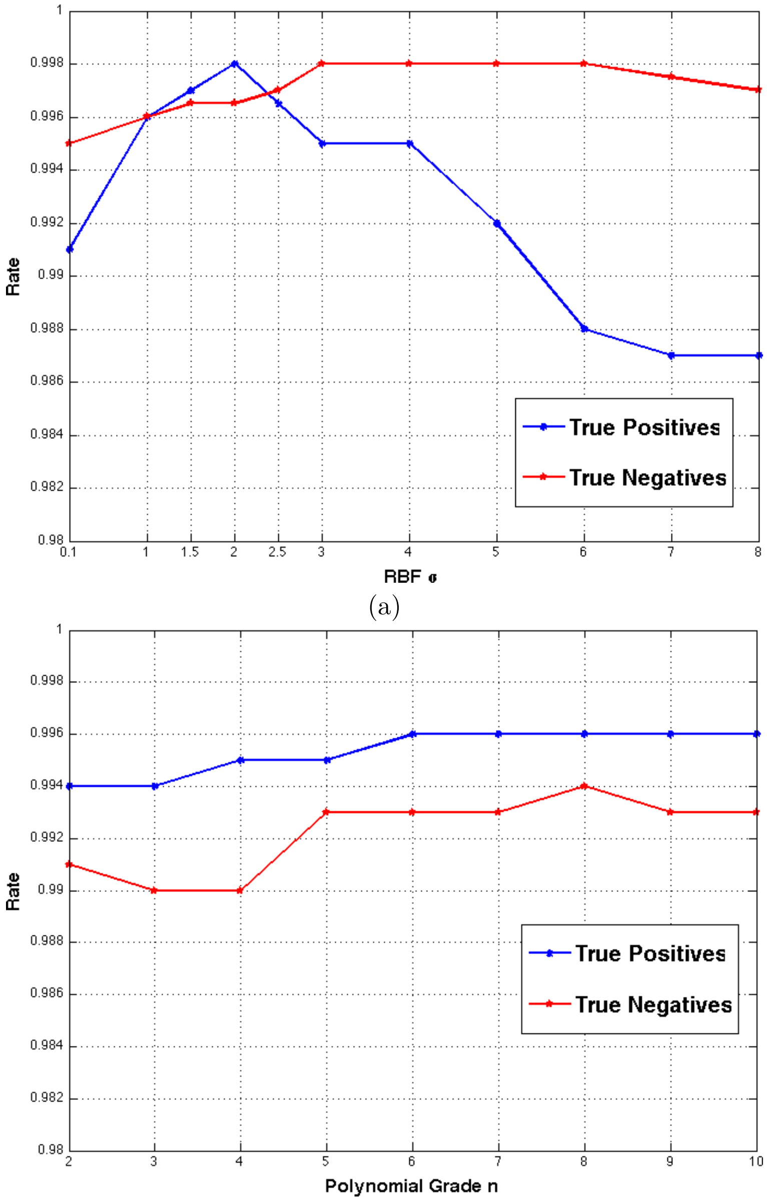
$$k(x_i, x_j) = (x_i \cdot x_j)^n, \tag{5.8}$$

in the case of polynomial kernels, and

$$k(x_i, x_j) = exp(-\sigma\|x_i \cdot x_j\|)^2, \text{ for } \sigma > 0, \tag{5.9}$$

in the case of the Radial Basis Function kernel.

The parameters in both functions are the grade of the polynomial, n, in the first case, and the standard deviation of the Gaussian function, σ, in the latter.



(a)

(b)

Figure 5.16: True Positives and True Negatives of **single eye** detections obtained with SVM using two kernels functions: **a)** RBF and **b)** Polynomial.

A set of experiments was designed to determine the optimal values of the parametric kernel functions. In these experiments, some classification rates of the SVM were obtained when applied to the training samples by varying the value of the parameters n and σ for both kernel functions:

- *Polynomial Kernel*: The grade of the polynomial is in the range $n \in [2, 10]$. The best value is achieved when the number of true positives is maximized for the lowest grade n possible. Also, a preliminary study showed that the linear discrimination, $n = 1$, had quite a poor performance and therefore it has not been included here.

- *Radial Basis Function*: In this Gaussian kernel, the variance σ is tested in two range orders, $\sigma \in [0.1, 1] \cup [1, 10]$. Again, the best value is achieved when the number of true positives is maximized for the lowest σ possible.

In the experiments, a *leave-one-out Cross-Validation* methodology was used. In each iteration, one element of the training dataset is extracted and becomes the *query set*, while the rest of samples constitute the *target set*. At the end of the iteration, the probe sample is labeled and the result compared to its actual class (positive or negative). After all the iterations, a mean classification rate is obtained[12].

The results of the kernel experiments are presented in three sets of curves:

- **Individual Eye Classification**: These curves give an idea of the discrimination of the SVM classifier in classifying each single candidate as *eye* and *non-eye*, without considering the eye-pair.

  Figure 5.16 shows the results obtained for the True Positives (Hit Rate) and the True Negatives. As the total number of actual positive and negative samples is known *a priori*, the complementary classification parameters (i.e. False Postives and False Negatives) can be directly inferred. In the figure, the two sets of plots correspond to the two kernel functions studied, RBF and Polynomial, respectively.

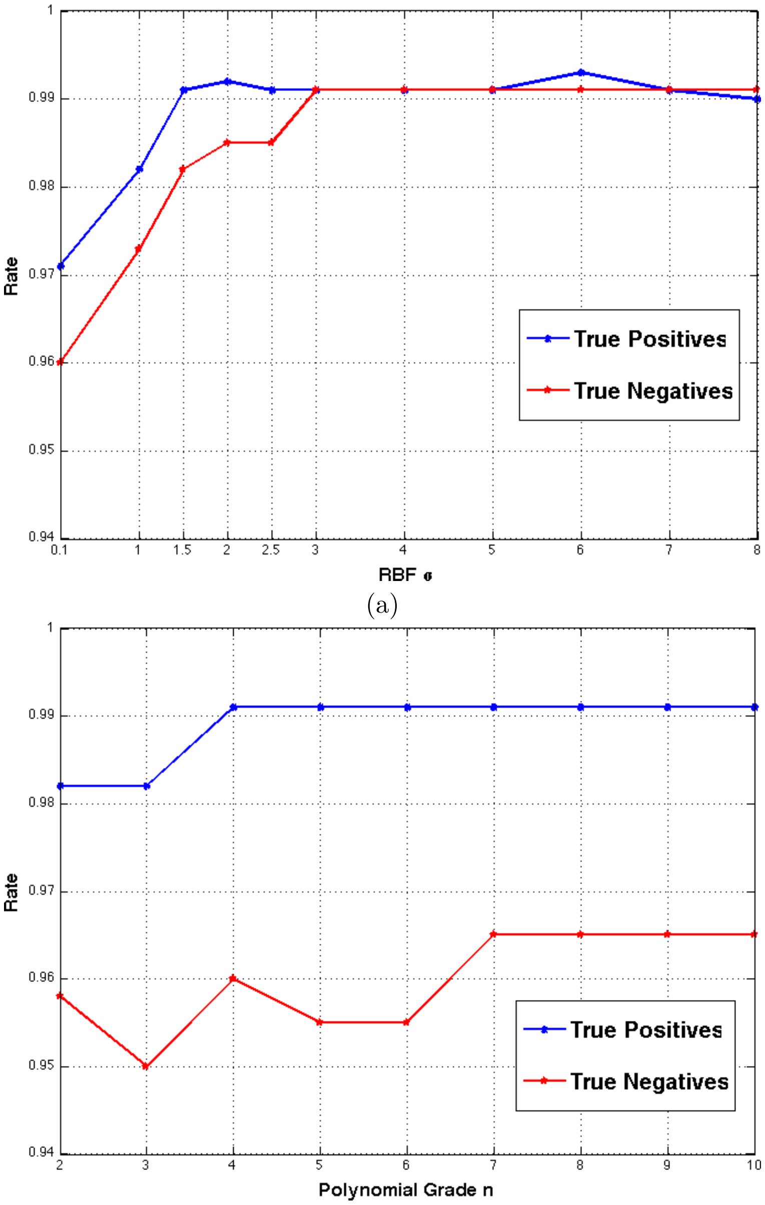
- **Eye-Pair Classification**: These curves add information about the performance of the system to discriminate the best eye-pair from the remaining face-candidates.

  Figure 5.17 shows the results obtained for the True Positives and the True Negatives of the eye-pairs. Similar to the previous case, the *a priori* knowledge about the number of positive and negative samples directly leads to the extraction of the False Positives and False Negatives. In the figure, the two sets of plots correspond to the two kernel functions studied, RBF and Polynomial, respectively.

- **Joint analysis of the Kernels**: This curve provides a comparison between the performance of the classifiers with the two kernels studied. It allows the kernel with best performance and the optimal value of the parameters to be selected.

  Figure 5.18 displays the results in the form of a Precision-Recall curve (see Chapter 3 for further details).

[12] Further explanations on this methodology are given in Chapter 3



(a)

(b)

Figure 5.17: True Positives and True Negatives of **eye-pair** detections obtained with SVM, using two kernels functions: **a)** RBF and **b)** Polynomial.

From the analysis of the experiments shown in Figure 5.17 and Figure 5.16, it can be observed that when a RBF kernel function is used, an increase in the σ pushes all the rates to the ideal case, except for the True Positives in single eyes, which becomes lower for values of $\sigma \geq 3$. For the polynomial kernel function, all the curves monotonically increase with the value of the grade, n, for $n \leq 7$. For $n \geq 8$, the rates do not suffer great variations, as the curves remain more or less consistent.

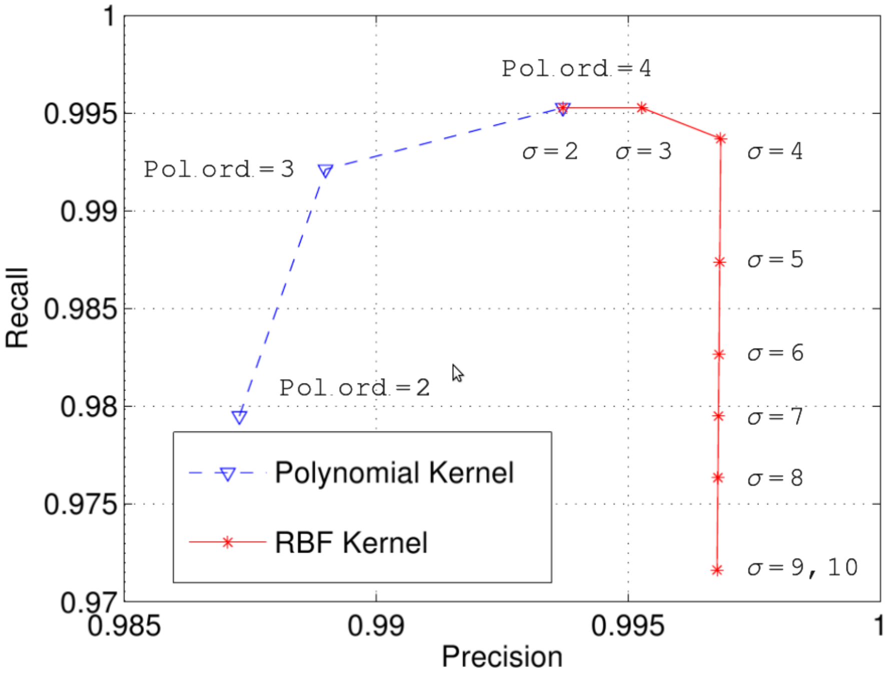


Figure 5.18: Precision-Recall curves of the setting of the polynomial and RBF kernel functions of the SVM classifier.

Figure 5.18 plots the two Precision-Recall curves obtained after applying the SVM classifier, using both kernels and varying their parameters. Given that the optimum working point of the PR curves are those closer to the upper-right corner of the graph, it can be concluded that the optimal kernel configuration is achieved with a Radial Basis Function approach with variance $\sigma = 3$.

- For the forthcoming experiments in this thesis, the SVM classifier for the eye location is trained using a RBF kernel with $\sigma = 3$.

5.8 Evaluation of the Eye Location System Developed

To validate all the initial hypotheses presented in Section 5.2 about the implementation of a novel Eye Location system, some tests were developed. In all, three different sets of experiments were carried out. First, an evaluation of the eye location system as a control mechanism of the number of face false-alarms, and then two studies comparing our system against some state-of-the art reference algorithms: one in terms of performance results and another to evaluate the computational load and time response of the algorithms.

Dataset	WITHOUT EYE DETECTION		WITH EYE DETECTION	
(Nr. of images)	Nr. of FA	FA Rate	Nr. of FA	FA Rate
AR (279)	18	6.45%	0	0%
FERET (3542)	23	0.65%	2	0.05%

Table 5.4: Control of the False Alarms produced by the Coarse Face Detection stage using eye locations in AR and FERET.

5.8.1 Control of the face alarms of a face detection system

Providing a face detection system with an eye location stage does not only improve the accuracy on the location of the faces, but it also works as a filter to reject some of the false alarms initially generated. This section describes a set of experiments that try to probe that our system can be used as a mechanism to reduce the number of potential false alarms given by some previous face detection system.

When using robust classifiers, the probability of detecting positive eye-pairs in regions where there is no real face is very low. Thus, by discarding these detections the number of false alarms in the Coarse Face Detection stage can be reduced, thereby increasing the reliability of the system.

To evaluate the control of the false alarms produced in the first stage, we designed an experiment that evaluated two databases with a known number of false alarms after the Coarse Face Detection: the FERET and the AR datasets[13]. In this experiment, the number of false alarms generated in face detection without using eye location is compared to the number of false alarms generated after using eye location.

The results from this experiment can be seen in Table 5.4

The analysis of these results confirms that the eye location stages can be used as an additional mechanism to control the information provided by the Coarse Face Detection stage. In both test datasets, there is a significantly high false-alarm rejection rate, showing that the system can detect nearly 100% of the errors in the detected faces.

5.8.2 Comparative Study of the Eye Detection system

The experiments proposed in this section compared our eye location system and the approaches proposed by the works detailed at the beginning of this chapter[14]: [52, 24, 114]. For the experiments, commercial software developed by Neurotechnologija [73] was also used: *VeriLook*.

In this comparative study, it is important to use exactly the same datasets and performance evaluation methods for all approaches. The two datasets selected were: FERET and the initial version of the Face Recognition Grand Challenge (FRGCv1)[15].

For these experiments, our eye location system was tested using all the steps displayed in Figure 5.6, which is basically the Coarse Face Detection stage,

[13] See Appendix B

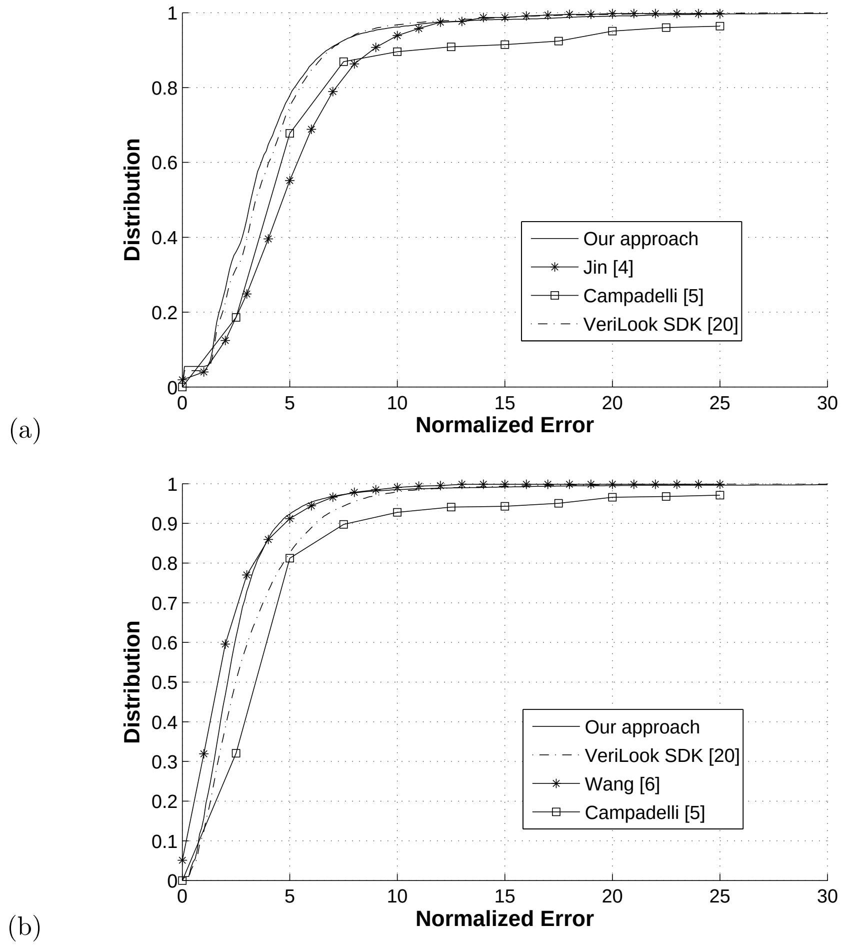
[14] Section 5.3.2, p. 61

[15] See Appendix B

followed by an Eye Candidates Extraction step (both using boosting classifiers), and finally, by performing a fine eye location stage with the extraction of HOG local features that were validated using the SVM classifier.

A summary of the results of these experiments can be seen in Table 5.5 and Table 5.6. The tables show not only the performance results, but also the values within a confidence interval of 99%. For a detailed description of these evaluation method, the work in [118] is recommended. It should be remarked that the confidence intervals may vary with the number of the samples and the estimated probability.

In FERET, the probability of finding a true positive eye-pair was $hit = 96.2\%$, while in FRGCv1, the percentage was even higher, $hit = 98.5$, considering a relative location error of $N = 10\%iod$. Figure 5.19 shows the curves corresponding to the true eye-pairs detected (as a function of the eye location error), and defined by the expression (3.12). The results are plotted for the two datasets evaluated.



(a)

(b)

Figure 5.19: Curves with the detection rates of the eye-pairs versus their relative errors in **a)** FERET y **b)** FRGCv1.

Maximum N_{error}	5% ***iod***	10% ***iod***	25% ***iod***
Our Solution	78.0%,[76.73, 79.22]	96.2% ,[95.58, 96.73]	99.6%,[99.36, 99.75]
L. Jin [52]	55.1%,[53.02, 57.17]	93.0%,[91.86, 93.99]	99.8%,[99.51, 99.92]
P. Campadelli [24]	67.7%,[61.21, 73.57]	89.5%,[84.70, 92.92]	96.4%,[93.00, 98.18]
VeriLook [73]	74.6%,[73.29, 75.86]	96.8%,[96.24, 97.28]	99.9%,[99.75, 99.96]

Table 5.5: Eye detection percentages and confidence intervals in FERET achieved with different criteria for maximum error distance for the detected eyes.

Maximum N_{error}	5% ***iod***	10% ***iod***	25% ***iod***
Our Solution	92.3%,[91.63, 92.92]	98.5%,[98.17, 98.77]	99.6%,[99.41, 99.73]
P. Wang [114]	91.2%,[90.47, 91.88]	99.0%,[98.72, 99.22]	99.7%,[99.53, 99.81]
P. Campadelli [24]	81.2%,[73.53, 84.39]	92.8%,[90.19, 94.76]	97.1%,[95.23, 98.25]
VeriLook [73]	82.6%,[81.66, 83.50]	97.8%,[97.42, 98.13]	99.9%,[99.79, 99.95]

Table 5.6: Eye detection percentages and confidence intervals in FRGC 1.0 achieved with different criteria for maximum error distance for the detected eyes

From the figures, it can be seen that our eye location solution achieves the best results in both databases, compared to the rest of works studied. Regarding the commercial software in the comparison, VeriLook obtains better results in the FERET dataset than in FRGCv1. This can be explained by the fact that in the specifications of this software the images used for training the algorithm were a selected subset extracted from the FERET and the XM2VTSDB databases. Therefore, the results obtained with the VeriLook application are less conclusive.

From the study of the results obtained for FRGCv1, our system achieves slightly better results than the ones obtained by the algorithm proposed by Wan *et al.* [114] when an error $N_{error} < 5\%$ is considered. However, while selecting less restrictive rates for N_{error}, the difference between the two approaches are statistically irrelevant.

It is important to remark that, although the same face databases were used by all the approaches, not all of them use exactly the same samples, which could distort the results of the study. To test our approach, we used 99.28% of the images in the database, while the rest of the authors used a smaller subset of images. In the case of Wang *et al.* [114], the evaluation of the location results used only the 94.5% of images in FRGCv1 –which corresponds to the quantity of images in which they detected a face and an eye-pair. Thereby, our results and the conclusions achieved are validated.

5.8.3 Comparative Study of the computational cost for the Eye Location system

Depending on the application using the face detection with eye location algorithms, the computational cost (measured as execution time) of the whole system can be in many cases as critical as the detection hit rate. However, it should be remarked that the execution time is a relative measure, as it tends to decrease with the use of more powerful machines.

Table 5.7 summarizes the time the system required when performing all stages of our eye location system, as well as the execution time for the rest of the authors selected for the comparative study [24, 52] (see Section 5.3.2) for further details). Also, notice that for all the algorithms displayed in this table, the execution time was evaluated without taking into account the coarse face location stage. When analyzing the table, it can be seen that our eye location algorithm has a behavior similar to the remaining state-of-the-art algorithms studied, in the sense that non of them can work at real-time (considering real-time the algorithms operating at 25 frames per second), but still several images per second could be processed. This allows all the algorithms to work on time-demanding scenarios; also they can be run on the images of a video frame by frame. The only exception to this statement is the work of Campadelli *et al.* [24], which takes several seconds to analyze a single image.

	Execution Time	**Technical Specifications**
Our Solution	$195ms \pm 12ms$	1.85GHz Dual-Core PC
L. Jin [52]	$105 \pm 19ms$	2.93GHz PC
P. Campadelli [24]	$> 4s$	3.2GHz PC
P. Wang [114]	$> 100ms$	2.7GHz PC

Table 5.7: Execution Time per image for some eye location algorithms (without coarse face location).

Achieving such results in our system is a good indication that the distribution of the computational load among the different stages of our system is, in general, quite efficient, and also that this efficiency is extended to the concatenation of all the steps. However, after performing a more exhaustive analysis, we can see that this is not completely true, as 70% of the execution time in our system was devoted to the boosting stage for the extraction of eye-candidates. Despite the simplicity of the classifier in this step, its intensive scanning of the face region for each possible location and scale still makes it the most demanding stage.

5.9 Conclusions

In this chapter, a fully automatic eye location algorithm was developed, targeting gray-scale frontal faces in semi-controlled scenarios where the two eyes were visible and preferably open. The novelty of our approach was to mix a first set of stages of fast and robust boosting classifiers (to detect face-regions and some eye-candidates) with a second set of steps, where HOG local features were extracted from the candidates and a SVM classifier was applied to select the optimal eye-pair.

The main conclusions drawn from the experiments for eye location can be summarized in the following:

- **Boosting Stages**: After experimenting on different datasets, it can be determined that the boosting classifiers can extract true eye candidates with high confidence. In none of the tests performed was the hit rate under 90%. Moreover, the results for the negative candidates show that on average the classifiers produce only a reduced set of false alarms per eye, which confirms the reliability of the stage.

  Also, the performance obtained when using two different classifiers, each trained to detect the left and right eyes, respectively, is greater than when a single classifier is used.

- **Local Descriptors**: The use of HOG local descriptors combined with a binary SVM classifier leads to a robust selection of the best eye-pair from a set of candidates. For a Radial Basis Function kernel in the SVM, setting the variance parameter to $\sigma = 3$, the detection rates achieved values of around $TP = 99\%$, with a false alarm rate of around $FP = 1\%$ during the training of the classifier. These results outperform the ones achieved using a polynomial kernel approach.

- **Multi-resolution Approach**: The location of the eyes performed with our multi-resolution design provides more precision than other state-of-the-art approaches. Compared to some key works and a commercial software, our algorithm achieves good results with two extensive datasets, FERET and FRGC. For an error of $N_{error} < 5\%$ in the inter-ocular distance, our approach surpassed the results of the other solutions compared.

Chapter 6

Face Recognition using Face Graph Algorithms

6.1 Introduction

Face recognition is known as the process through which the identity of a subject is provided using only facial biometrics extracted from an image. The identity is given in the form of a label. Given a database of known individuals, During a recognition process the label of the new face image is matched to one of those in the database.

The implementation of a face recognition system is not trivial as it has to tackle the issue of the differences between any pair of face images:

- A first group of differences are derived from the *extrinsic* variations of the facial analysis, as reviewed in Chapter 1. As an overview, the extrinsic variations are those that do not depend directly on each specific face, but on the context where the images are acquired. Most of the extrinsic variations are thus inherent to image conditions, such as illumination variations, different image resolutions or uncontrolled occlusions in the scene.

- A second group of differences are derived from the *intrinsic* variations for each person. Each face is unique, and thus it defines a person identity. However, the face of a person is still subject to multiple variations, some related to its physiology and others related to the appearance of a face at a specific moment. The identity is a mixture of characteristic features such as the shape of the face, the skin color, the age or the gender, while the appearance variations are related to the changes in the gestures, hair-style, beard or the use of glasses.

When a face undergoes a face recognition process, a biometric representation is needed. This representation is compared to the models in the database. Due to the intrinsic and extrinsic variations, a person can show several appearances and ideally the models in the database should account for them. Face recognition methods are designed most often as supervised algorithms. They learn the biometric information from a database of models and, depending on the information they provide, the algorithms can be classified into two main categories: *holistic* and *component-based* approaches.

Holistic approaches consider the face as a whole, while component-based approaches define faces from the information from their individual elements. Component-based approaches usually locate the elements on a face, extract a set of features and then define relations between them. This results in component-based methods having more accurate information than the holistic, though they increase the complexity of the system and are also quite sensitive to the feature-extraction step.

The use of local descriptors– such as the texture features (Chapter 4)– can help to relieve the effects derived from the *extrinsic variations*. On the other hand, to tackle the *intrinsic variations* a flexible solution should be employed. This solution should locate and extract the local descriptors adaptively for each person. The feature-based *Face Graph Algorithms* (FGA) address this specific problem. Given a face, a Face Graph Algorithm consists of locating a number of facial landmarks interconnected in a graph. The graph represents the face that has to be recognized and fixes the location of the local descriptors to be extracted. Figure 6.1 provides a flow chart of a generic FGA, consisting of the following steps:

1. **Preprocessing**: In this phase, the face is normalized for a common representation, running a normalization process such as those presented in Chapter 2. The initial information for this stage is usually provided by a face detection algorithm.

2. **Feature Location and Extraction**: This is the main phase of the FGA. The landmarks in the face are located and the local features are extracted from these landmarks to constitute the graph that identifies each person. Different FGA approaches are characterized by their respective implementation of this stage. The graph resulting from this stage is represented by the *face feature vector*, which contains biometric information of the person.

3. **Postprocessing**: The postprocessing stage is basically a dimensionality reduction phase that addresses this issue. Many reduction techniques can be applied to the face feature vector. In Appendix A, a summary of the most common techniques used in this work is provided.

4. **Matching**: As in any face recognition algorithm, the FGA finishes by matching the reduced *face feature vector* with the models that are stored in the database. The methods used in this stage are described in Chapter 3.

The following sections start explaining the motivation for using feature-based algorithms, followed by an introduction of the research yielded in this field. Then two widespread Face Graph Algorithms employed herein are explained, the Non-deformable Grids and the Active Appearance Models, and our two FGA proposals are introduced: the HOG-EBGM and the Colored HOG-EBGM. The last sections of this chapter detail the experiments to evaluate the performance.

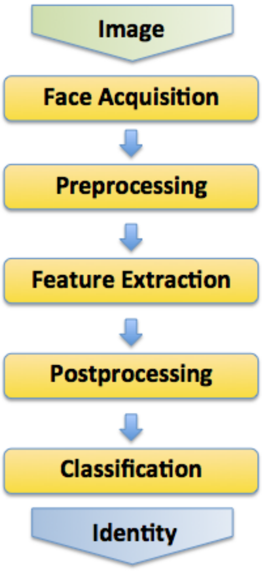


Figure 6.1: Global flow chart of the steps in a FGA for face recognition, based on the extraction of local features.

6.2 Motivation and Contributions

Although several works have already been developed, face recognition is still an open topic that attracts many researchers. Due to the great quantity of practical applications that can make use of it, finding new solutions able to fit real-scenario constraints in face recognition has become one of our main motivations for studying it. In order to develop our own system , we focused on developing a Face Graph Algorithm that took advantage of the properties of the local texture descriptors studied in Chapter 4, finding a practical solution for the problem of recognizing frontal faces in semi-controlled scenarios.

Some techniques in literature, like Wiskott's Elastic Bunch Graph Matching (EBGM) [117] have succeed in combining a good graph location stage with the extraction of local features. Specifically, this algorithm provides facial information using Gabor jets, but it opened the door to explore approaches beyond the original set-up.

- In this chapter we present the *HOG-EBGM*, a novel Face Graph Algorithm based on Wiskott's approach.

The HOG-EBGM combines the essentials from the classical Elastic Bunch Graph algorithm (EBGM) for landmark location with the properties of HOG features to describe the face using local gradients. Taking the EBGM approach as a baseline, the idea is to replace the Gabor jets with a more robust set of local descriptors, the HOG features.

In contexts other than related to face recognition, HOG features have proved to be useful in retrieving biometric information, mainly due to their robustness. This raises some hypotheses about the inclusion of HOG descriptors in the EBGM algorithm, which will have to be proven later:

- **Hypothesis 1**: *The inclusion of HOG descriptors in the original EBGM graph extraction can lead to a more precise set of coordinates for the keypoints located in the Face Graph.*

  This hypothesis is supported by the fact that Wiskott's EBGM uses both geometrical and texture information. When swapping from the Gabor jets to a set of local descriptors with more descriptive power (i.e. the HOG features), the accuracy of the location of the keypoints is expected to increase.

- **Hypothesis 2**: *The use of HOG descriptors to generate the face feature vector face can lead to an increase in the effectiveness of the original algorithm for face recognition purposes.*

  This hypothesis predicts an improvement in the descriptive, biometric power of a face, not only through better locating of the keypoints, but also due to the type of information extracted at those specific landmarks.

Considering these two hypotheses the HOG-EBGM should witness an improvement over other FGA used in the literature. The set of experiments designed as part of the present research was aimed at corroborating this fact.

To validate the two hypotheses, in this chapter we should be able to:

- Determine the improvements when we include HOG features into the original EBGM, specifically addressing accuracy and descriptive issues.
- Analyze the inclusion of color cues into the HOG-EBGM using a range of color spaces.
- Compare the usability of the HOG-EBGM algorithm with other widespread non-EBGM-based face graph algorithms.

To achieve the last of these goals, it is of great importance that fair comparisons be performed. This was done by adapting some of the most widespread non-EBGM face graph location algorithms to the use of HOG features. Therefore, this chapter includes another significant contribution: the adaptation of two common FGAs, the Non-Rigid Grids and the Adaptive Active Models, to use HOG features.

6.3 State of the Art

Face recognition has been studied in depth over the last few decades. The most common recognition methods are designed to detect individuals who usually appear in a specific view, most often the frontal view, and in controlled or semi-controlled scenarios. The more advanced the algorithms, the more difficult the scenarios they are able to overcome, providing perks such as a greater flexibility in tackling multiple views, a more robust handling of occlusions, etc.

Researchers have been working diligently on this issue, applying several approaches to the recognition problem. Some surveys on this topic may be consulted for further descriptions of the most common algorithms [128, 125]. Next, an overview of the main tendencies for face recognition is given, remarking the importance of those algorithms based on the generation of a face graph. This overview is followed by the analytical description of two off-the-shelf Face Graph Algorithms: Non-deformable Square Grids and Active Appearance Models.

6.3.1 Works on Face Recognition

Taking the aforementioned surveys [128, 125] as a reference, the different approaches for face recognition can be broadly classified into the following categories:

- **Statistical Holistic Methods**: These are the methods that consider the face as a whole. All biometrics are extracted from the face use statistical approaches which are able to create a specific face representation. The raw data to create such representations is generally the grayscale information from the face images.

  The most basic approaches of the statistical methods are the correlation-based techniques (also known as *Template Matching*) [22, 85]. These techniques model each person using a template (sometimes adaptive), so that each novel face is directly compared to each of the templates models in the database. The problem with these approaches is that the templates are extremely dependent on the extrinsic variations of faces.

  Another family of popular solutions try to optimally represent the faces by reducing the dimensions of the biometrics (face feature vectors), in such a way that the new face representation contain the maximum information. A good representative of the dimension reduction algorithms is the Principal Component Analysis (PCA) [107]. Regarding the classification methods used to classify face vectors in a reduced space, these solutions can be divided into linear and non-linear.

  Zhang *et al.* [127]. gives offer an analytical summary of the main approaches for the linear discrimination. One of the most widespread techniques for linear discrimination is the Linear Discriminant Analysis (LDA) [12]. This approach discriminates maximizing the separation between classes, while minimizes the separation of the members of the same class. That is, individuals can be discriminated by maximizing the distance between the models of different people in the final subspace. A number of variations from the LDA have also been derived, such as the Null Spaces-PCA [46], Orthogonal-LDA and Uncorrelated-LDA [122] or the Regularized-LDA [59], being the latter implementation directly derived from the *Lanczos algorithm*[1]. Another popular linear discriminant method is the Independent Component Analysis (ICA) [63], in which the data at the subspace is statistically independent.

  Regarding the non-linear solutions, the Kernel methods are quite common. These methods use non-linear functions to project the face vector to spaces of higher dimensionality, where a linear discrimination is possible. Some good representatives of these approaches are the Kernel Fisher Analysis [64] and the Kernel Independent Component Analysis [9].

  Finally, among the statistical approaches, another family of algorithms should be mentioned: the Hidden Markov Models (HMM) [86]. These algorithms define a number of states associated with a set of probability density functions.

[1] These methods are explained in Appendix A as reduction techniques for face recognition

- **3D Face Recognition**: Human faces are represented as a projection of 3D surfaces, as shown in Chapter 2. Many authors have tried to take advantage of the 3D information to handle some of the extrinsic variations (i.e. pose, illumination, etc.), as well as some of the intrinsic variations (i.e. facial expressions). In the beginning, this option involved very high computing costs, but greater advances in technology have allowed researchers to explore this field.

  In this category are found 3D Morphable Models [79, 15], which extract texture and shape information from single face images. Given the difficulty of this task, this remains a hot issue under study today. Kemelmacher and Basri [55] propose a novel approach that exploits the similarity of faces to obtain 3D models from single images in uncontrolled conditions (e.g. from Internet images).

  Methods based on 3D facial surface reconstruction [21, 20] are likewise found. The idea is that a face surface can be acquired, insensitive to variations such as head orientations and facial expressions, from which a model can be extracted. A Surface Interpretation Measure (SIM) has been recently defined as a mechanism for describing surface areas and extracting scores for subsequent face matching [101].

- **Feature-based Methods**: These methods extract biometrics based on local information of the faces, that we call *keypoints* (unlike the holistic methods). These methods are mostly component-based, and tend to locate the most discriminative keypoints building graphs overlaid on the faces.

  From the beginning, many face recognition systems focused on the detection of individual face elements, such as the eyes, nose, mouth and head contours. Some of these early studies can be found in [16, 57]. These approaches proved, however, insufficient to handle problems such as the variation produced by intrinsic and extrinsic factors.

  To solve this, other algorithms were developed, as Elastic Graph Matching (EGM) [60]. This has been one of the most advanced feature-based methods developed. It makes use of neural information, included in the Dynamic Link Architecture (DLA). The DLA is employed to solve some of the problems in conventional artificial neural networks, such as the expression of syntactical relationships in neural networks. In EGM, a rectangular grid is placed over the test face, and a set of feature vectors is extracted from each of the nodes of the grid (for instance, the Gabor wavelet responses in [60]), and then compared to those extracted from the face database. The amount of features extracted constitutes the main of drawback of this approach, since the computational time increases considerably. The EGM approach represents a natural base for two of the FGAs studied in this work: the Rigid Grids and the Elastic Bunch Graph Matching [117].

  The Active Appearance Model (AAM) [27] is a Face Graph Match algorithm which is partly *feature-based* and partly *statistical.* It is an integrated statistical model containing information about the shape and appearance of the face. Then, the model is fitted to a new face in a way that the differences become minimized. In the current chapter, the AAM approach is explained in greater detail.

Due to the relation of the Feature-based face recognition methods and the approach presented in this thesis, the rest of this section is aimed to present two of the aforementioned Face Graph algorithms: Non-deformable Square Grids and Active Appearance Models.

6.3.2 Non-Deformable Rigid Square Grids

The simplest method to locate landmarks for the extraction of local features is the use of Non-deformable Rigid Grids. In this case, a grid of points whose distribution is known *a priori*; the landmarks extracted from an object are the result of overlapping this grid directly onto the image.

The grids are considered non-deformable and rigid in the sense that they are not able to fit the object they are describing. Non-deformable rigid grids can be used as a Face Graph Algorithm to locate facial points where local information may be extracted. The keypoints are located at the positions obtained after overlaying the grid over a face normalized in scale and rotation (Chapter 2). The relative location of the landmarks is always the same, although the distribution depends greatly on the specific shape and configuration of the grid. Figure 6.2 shows three grids with different shapes (i.e. square, round and star) and distribution configurations (homogeneous and non-homogeneous).

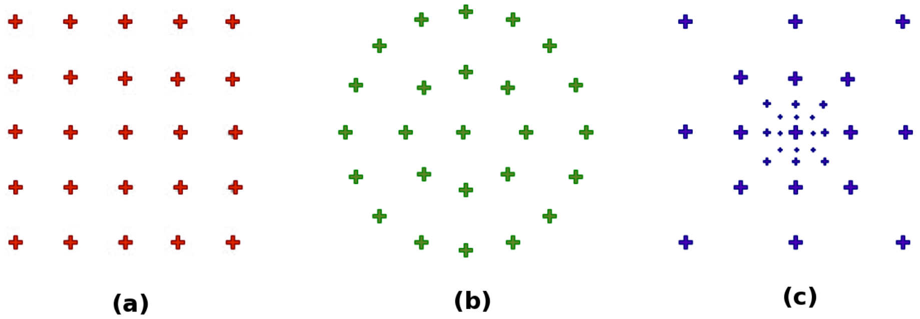


Figure 6.2: Three non-deformable rigid grids with different configurations: **a)** Homogeneous Square Grid, **b)** Homogeneous Circular Grid and **c)** Non-homogeneous Star Grid.

To ensure the usability of non-deformable rigid grids as a FGA, the location of each grid point is assumed for different individuals to correspond to a common relative position. That is, the distribution of the landmarks should be equivalent for all face images. To achieve this assumption some requirements have to be fulfilled:

- All face images have to be obtain under the same camera view. That is, if the grid is extracted from a frontal face, then it will only be comparable to other grids extracted from frontal faces.

- All face images have to undergo exactly the same geometrical normalization process. This normalization determines not only the rotation but also the scale of the images. Grids extracted from different faces will be comparable only if the relative scale and orientation of the images are the same. In our case, this scale is related to the inter-ocular distance, *iod*, as shown in Chapter 2.

- All grids need some initial points to be an anchored on the face. In FGAs, a pair of facial landmarks common to all the images are used to set the relative position when overlapping the grid. The position of the eyes completely addresses this problem, so in this work the eye-centers will coincide with two of the grid points for all images.

It is the interest of this thesis to generate simple facial graphs; therefore, the two-dimensional square grids were considered. In these grids, the keypoints are located according to a square pattern, and homogeneously distributed. For simplicity, the directions of the axes of the grid are defined as parallel and orthogonal to the inter-ocular line, respectively. The number of grid vertices, N_n, corresponds to the the number of landmarks and is an indicator of the quantity of information that can be retrieved from the face (regardless of the quality of the information). However, it should be noted that a higher number of keypoints also implies higher computational and storage costs.

In this work, two kind of configurations for the square grids are studied:

1. **Sparse Square Grids:** These grids have a low density of vertex, which means that the keypoints located are sparsely distributed over the face. In this configuration, the local features generally have a window size such that the overlap between them is null or small (less than 25% of the window size).

2. **Dense Square Grids:** These are grids with a high density of grid vertices, which means that the keypoints located are condensed over the face. In this configuration, the local features generally have a window size greater than half the separation between landmarks, which implies that the local descriptors have a remarkable degree of overlap.

In Figure 6.16, an example of overlapped dense and sparse square grids can be observed.

6.3.3 Active Appearance Models - AAM face graphs

In 1998, Cootes *et al.* [28] developed one of the most popular methods for matching a model to an image: the Active Appearance Models (AAM). AAM models contain information about shape and appearance, learned through supervised training. Also, these models use statistical information to be matched to new images. The AAM can be used as a Face Graph Algorithm: a model is generated from a series of facial landmarks, matched to the input faces and then a local descriptor is extracted from each graph point.

One of the most employed configurations of the AAM models is that designed by Milborrow *et al.* [78]. This thesis is based on Milborrow's design, and therefore this chapter is aimed at studying its main features. However, for a detailed explanation of this implementation of the AAM model matching algorithm, the reader is referred to the original work.

In general, AAM models need to learn how to adapt to new faces. The training stage is aimed to learn the variance in shape and appearance of different faces. Prior to this training, a model face graph for all training images is located, marking the points by hand. This constitutes the ground-truth of the AAM models; the information about the shape and the appearance will be extracted from this set of graph models.

Regarding shape information, a statistical model is generated using the Principal Component Analysis (PCA) as a representation technique, described in Appendix A. This is achieved during the training, using the ground-truth. The PCA technique is applied to each of the keypoints, which have been previously aligned, such that a face graph shape, x, can be expressed as:

$$x = \tilde{x} + P_s b_s, \tag{6.1}$$

where $\tilde{x}$ is the mean face graph shape, P_s is the set of eigenvectors obtained after applying PCA and b_s are the coefficients for face graph x.

To generate the appearance model, all training images undergo a warping process. The goal of this process is to align the model face graphs using a triangulation algorithm. To build the appearance model, we apply a PCA representation technique that extracts appearance information from the warped images. Each normalized model, g is in grayscale and is defined as follows:

$$g = \tilde{g} + P_g b_g, \tag{6.2}$$

where $\tilde{g}$ is the normalized mean of the grayscale vector, P_g is the matrix of eigenvectors obtained after the PCA analysis and b_g are the coefficients that describe the sample g.

The last stage of AAM training is aimed at exploiting the correlation between shape and appearance. A combination vector is generated, $b = [b_s^w, b_g]^T$ for each training image, where b_s^w is the weighted version of b_s. If the PCA reduction method is applied to this vector, a projection matrix, Q, is obtained. After projecting b with Q, the combined appearance parameters, c, are obtained, such that $b = Qc$.

Once the AAM models are generated, the process of creating a face graph in a new image is iterative. An initial configuration of the face graph is assumed, and then it iteratively adapts to locate in the new face the points equivalent to those in the models. Figure 6.3 shows the iterative process of the AAM algorithm from the initial configuration to the final matching.

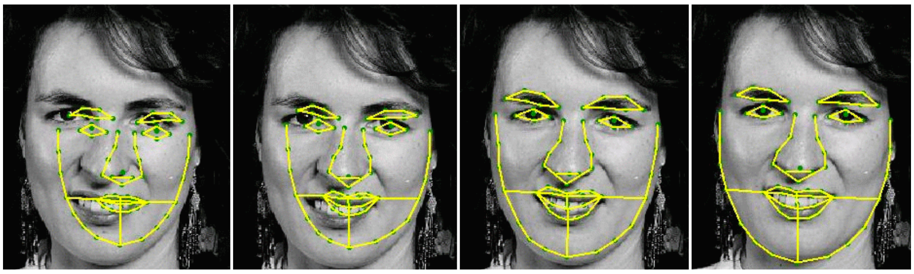


Figure 6.3: Iteration process to adapt a graph model to a face using Active Appearance Models.

6.4 HOG-EBGM: HOG-based Elastic Bunch Graph Matching

6.4.1 Introduction

HOG-based Elastic Bunch Graph Matching (henceforth *HOG-EBGM*). is an algorithm specifically designed and developed in this work for the extraction of biometric information in face images.

Although HOG-EBGM is a novel algorithm, it is based on a common and widespread algorithm, the Elastic Bunch Graph Matching (EBGM) [117]. EBGM is an algorithm that describes faces from the information that can be retrieved at significant landmarks while these landmarks are organized in the form of graphs. Thus, the main idea of the algorithm is that given a novel image of a face, a Face Graph (FG) can be placed using the local information that is extracted from some specific keypoints in a set of training images serving as models. Each Face Graph consists of spatial and texture information extracted from those facial landmarks.

Traditionally, EBGM algorithms have been performed using Gabor filters (also known as Gabor jets) as local features to retrieve the facial information. As will be seen, local features play a double role in the creation of the Face Graph: they are partially responsible for the final location of the facial landmarks, and they constitute the final descriptor of the face. Our main interest lies in finding an alternative to EBGM that substitutes the original Gabor filters with the Histograms of Oriented Gradients.

For the remainder of this thesis, the original EBGM algorithm will be referred to as *Gabor-EBGM*, or just *EBGM*.

6.4.2 Theoretical Development

To describe the operation of the HOG-EBGM algorithm, the steps performed by EBGM are taken as a baseline. However, in this section the differences between both algorithms will be highlighted.

The process to generate a new EBGM Face Graph can be broken down into three main steps: *image normalization*, *Face Graph location* and extraction of the local descriptors associated with the facial landmarks previously located. A diagram outlining these steps can be seen in Figure 6.4. Also, the inputs and outputs corresponding to each of the stages can be observed.

The normalization is solved as described in Chapter 2, and it has to be done essentially before the extraction of the local features. The variance of the noise in the descriptors is much higher when non-normalized images are used. In the case of using HOG features, the pixel gradients are more accurate in images with higher contrast, as the transitions are strongly marked. Also, this normalization step is associated with a gray-scale conversion of the image, as the HOG-EBGM is designed to work only on intensity images. For the inclusion of color information in our algorithm, see Section 6.5.

Definition of the Face Graph

Regarding the EBGM methods used in this work, during the extraction of the graphs, the facial landmarks are located in a Face Graph, following the structure

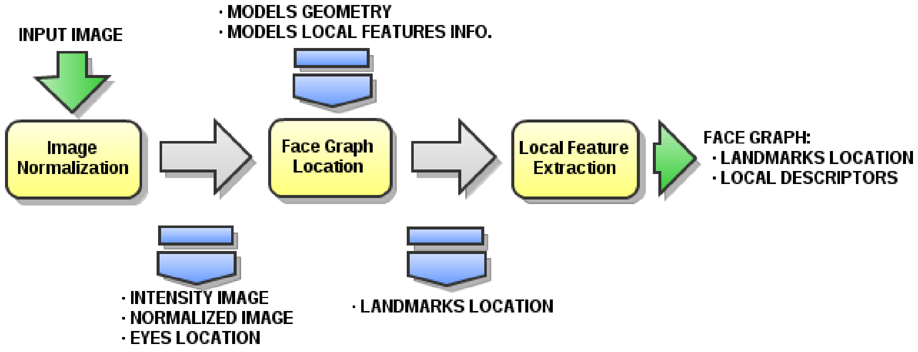


Figure 6.4: General steps performed by the HOG-EBGM algorithm.

proposed by the CSU project [17]. In this structure, the total number of nodes that constitutes each FG is $N_p = 25$, and consists of one location, $X_i = (x_i, y_i)$, and its corresponding local descriptor, $J_i(X_i)$.

Thus, a Face Graph can be defined as:

$$FG = \{X_i, J_i(X_i), 1 \leq i \leq 25\} \tag{6.3}$$

Figure 6.5 displays an example of a Face Graph generated with the HOG-EBGM algorithm after the location of the facial landmarks, where $N_p = 25$.

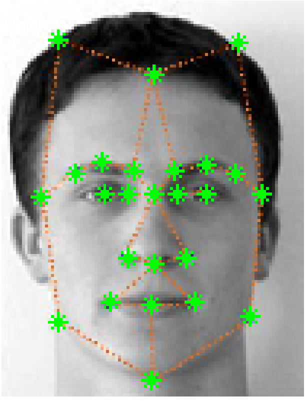


Figure 6.5: Example of the generation of a FG using the HOG-EBGM algorithm with $N_p = 25$ facial landmarks.

The graphs generated by Gabor-EBGM and HOG-EBGM locate the same landmarks, X_i (with different accuracies), but mainly differ in the descriptor J_i. The local features used are Gabor jets in the case of Gabor-EBGM, $J_i = Gabor(X_i)$, and HOG descriptors in the case of HOG-EBGM, $J_i = HOG(X_i)$.

Creation of EBGM training models

To automatically locate each of the facial landmarks, $fg_i = [X_i, J_i(X_i)]$, on a new FG, a set of models is needed. These models are generated during an off-line training phase, using a set of training images which should account for differences produced by the intrinsic and extrinsic facial variances.

The process to build the set of EBGM models is not trivial. For a better understanding of its difficulties, the steps outlined in Figure 6.6 will be followed. The list of the variables involved is summarized in the following chart:

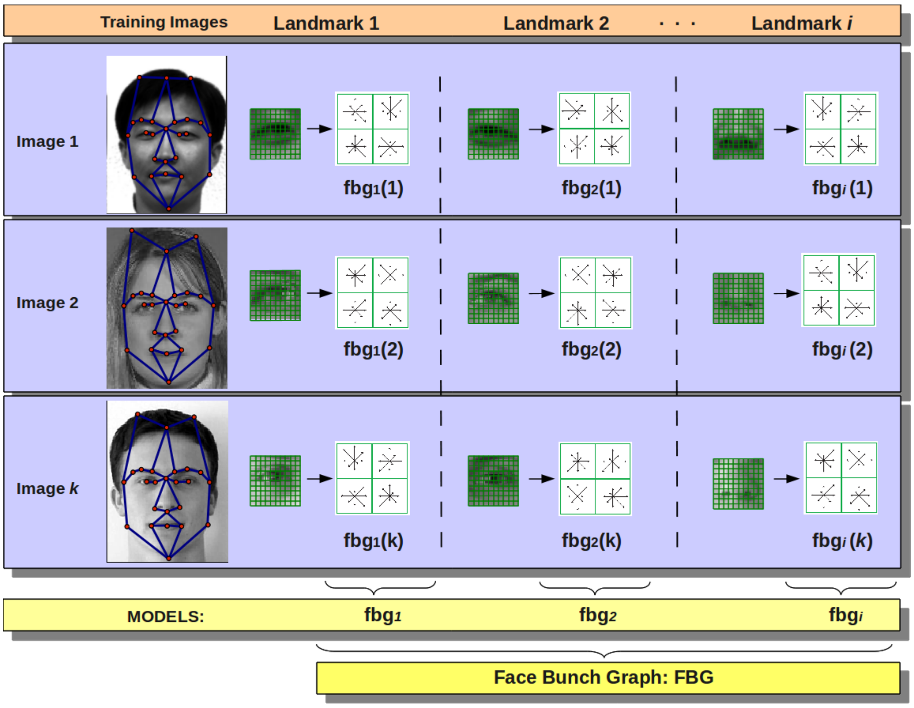


Figure 6.6: Outline of the composition of the models and the FBG used to generate a new FG in an input image.

From Figure 6.6 it can be seen that there is one model fbg_i for each facial landmark, accounting for a total of N_p models.

Each model contains the information for the automatic location of a specific landmark, fg_i, in new images. Each is built by aggregating the local descriptors $fbg_i(k)$ (HOG features for HOG-EBGM, instead of Gabor jets used in the standard EBGM), extracted for that landmark over the N_t training images. In Figure 6.6, each fbg_i is represented as a column. The association of all the models fbg_i is called the Face Bunch Graph (FBG):

$$fbg_i = \bigcup fbg_i(k) \forall k, 1 \leq k \leq N_t \tag{6.4}$$

$$FBG = \bigcup fbg_i \forall i, 1 \leq i \leq N_p \tag{6.5}$$

Note that the location of each trained landmark, $fbg_i(k)$, aggregated to the FBG is done manually on the training images, while the facial landmarks, fg_i, in the FG for new images are automatically located.

Global Parameters	**Symbol**	**Description**
Number of Landmarks	N_p	Landmarks in the graph
Number of Training Images	N_t	Images to extract models
Training Parameters	**Symbol**	**Description**
Models	fbg_i	Models the i-th Landmark
Face Bunch Graph	FBG	Formed by Models
Testing Image Parameters	**Symbol**	**Description**
Landmark	fg_i	Describes the i-th landmark
Face Graph	FG	Formed by Landmarks

Table 6.1: Summary of the variables involved in the HOG-EBGM.

Creation of the HOG-EBGM Face Graph in incoming face images

In contrast to the process of creating the FBG models, the location of landmarks in a new FG is achieved through an automatic iterative process. In this iterative process, the position of each new landmark is predicted using information from the landmarks previously detected, thereby reducing the search area.

The *a priori* information of this process is the position of the first two facial landmarks, which corresponds to the center of both eyes. As can be seen in Figure 6.4, this information is provided directly by the normalization step. For the rest of the landmarks, the process to detect the i-th facial landmark fg_i ($i > 2$) is graphically described in the flow chart shown in Figure 6.7.

The creation process for the HOG-EBGM Face Graph is defined as follows:

1. Coarse Location Step: This step is aimed at producing an initial estimation of the facial landmark location, X_i^s. This estimate is predicted using the mean of displacements between the i-th and the j-th ($j < i$) keypoints. More in detail:

   (a) Let $d^m(i,j)$ be the mean value of the displacements between key points i and j, estimated using the data in the FBG. This distance is calculated off-line with the information from the training images. In Figure 6.8.a, an example of the mean displacements calculated with the FBG information is displayed. In this example, only the calculation of the distances involved in the location of the landmark $i = 4$ is shown. Also in this figure, the FBG is assumed to consist of only three training images.

   (b) Let X_j be the coordinates of the j-th keypoint, which has been already located.

   (c) For each X_j, $X_i(j) = X_j + d^m(i,j)$ is defined as the initial estimation of the i-th keypoint, based on the j-th keypoint.

   (d) The initial estimate is $X_i^s = \frac{1}{i-1}\sum_{jj<i}(X_i(jj))$, i.e. the mean of the estimates of previous key points. Figure 6.8.b offers an example of the effect of the last three steps after being applied to a new image.

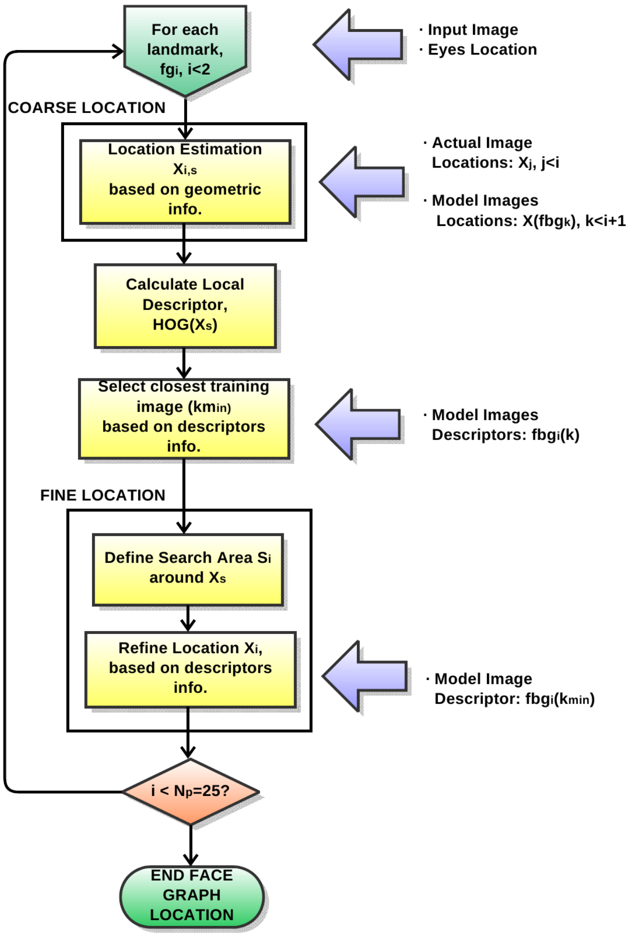

For each landmark, fgi, i<2
· Input Image
· Eyes Location
COARSE LOCATION
Location Estimation Xi,s based on geometric info.
· Actual Image Locations: Xj, j<i
· Model Images Locations: X(fbgk), k<i+1
Calculate Local Descriptor, HOG(Xs)
Select closest training image (kmin) based on descriptors info.
· Model Images Descriptors: fbgi(k)
FINE LOCATION
Define Search Area Si around Xs
Refine Location Xi, based on descriptors info.
· Model Image Descriptor: fbgi(kmin)
i < Np=25?
END FACE GRAPH LOCATION

Figure 6.7: Scheme of the iterative landmark location algorithm used to build the FG in HOG-EBGM.

2. Calculate the HOG feature on the previous location, $\text{HOG}(X_i^s)$.

3. Compare $\text{HOG}(X_i^s)$ with the $\text{fbg}_i(k)$ of the FBG and let:

$$k_{min} = \min_k \|(\text{HOG}(X_i^s) - \text{fbg}_i(k)\| \qquad (6.6)$$

   The training image corresponding to k_{min} will be the referent regarding the landmark i, as k_{min} and the actual new image have the most similar information for that landmark. Thus, for the rest of the iteration, all the calculations will refer not to the whole FBG, but to the information in $fbg_i(k_{min})$.

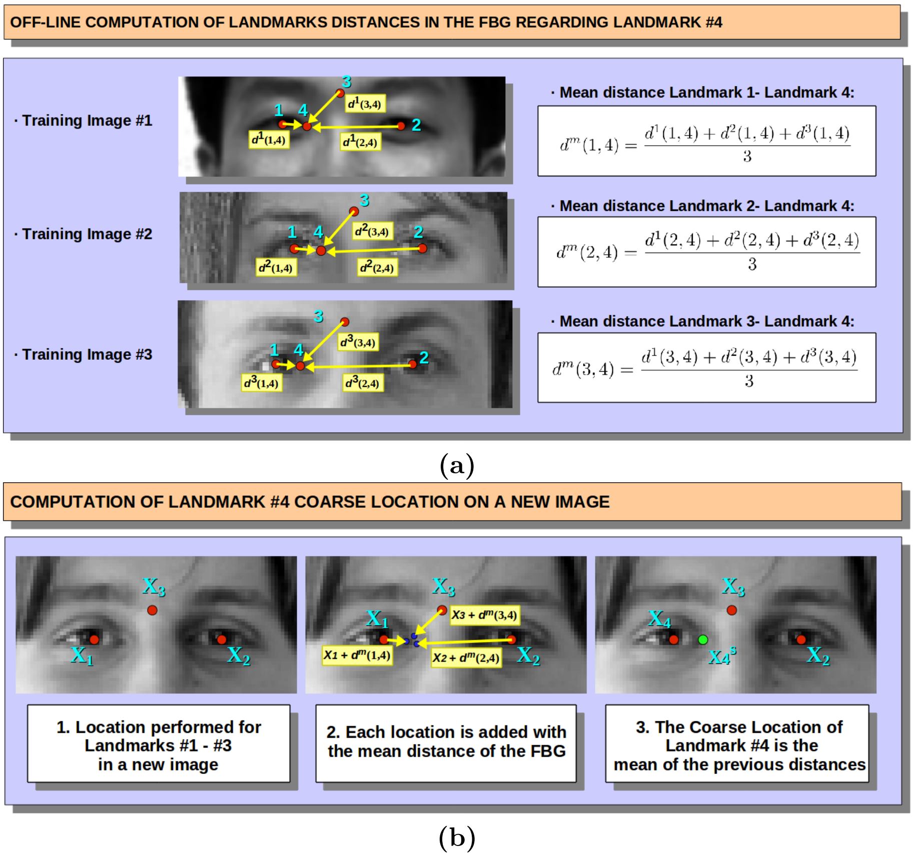
4. Fine Location Step: This step is aimed at refining the initial estimation of the location of the landmark in a closed and controlled area. The output of this step is the final location, X_i, of the landmark in the Face Graph. This step is divided into two stages:

(a) Define a search area, S_i, around X_i^s. The extent of the search area depends on the particular key point, as shown in Figure 6.9.b. The search areas are empirically set by considering the dispersion of the facial landmark locations in the FBG.

(b) Refine the initial estimate of the i-th key point using the descriptor $fbg_i(k_{min})$:

$$X_i = \min_{X \in S_i} \|\mathrm{HOG}(X) - \mathrm{fbg}_i(k_{min})\| \tag{6.7}$$

The idea of this step is that for every location inside the selected area and around the initial estimate of the landmark, X_i^s, an HOG descriptor centered in that point will be extracted and compared to the model descriptor, $fbg_i(k_{min})$. The final location, X_i, corresponds to the location with the local texture most closely corresponding to that of the model.

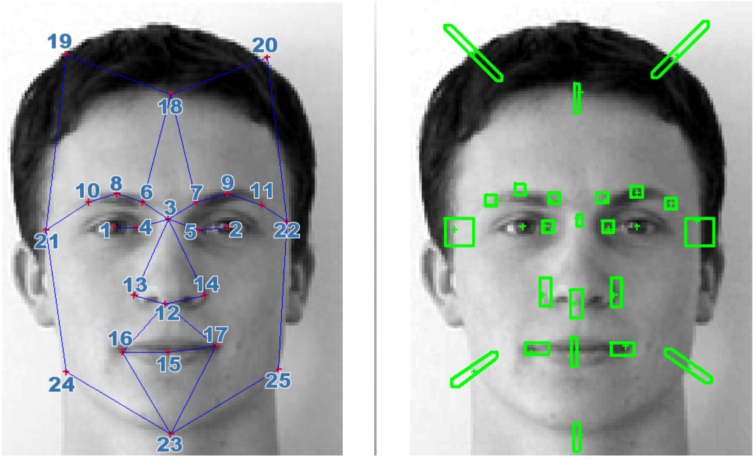

OFF-LINE COMPUTATION OF LANDMARKS DISTANCES IN THE FBG REGARDING LANDMARK #4
· Training Image #1
· Training Image #2
· Training Image #3
1
2
3
4
$d^1(1,4)$
$d^1(2,4)$
$d^1(3,4)$
$d^2(1,4)$
$d^2(2,4)$
$d^2(3,4)$
$d^3(1,4)$
$d^3(2,4)$
$d^3(3,4)$
· Mean distance Landmark 1- Landmark 4:
$d^m(1,4) = \frac{d^1(1,4)+d^2(1,4)+d^3(1,4)}{3}$
· Mean distance Landmark 2- Landmark 4:
$d^m(2,4) = \frac{d^1(2,4)+d^2(2,4)+d^3(2,4)}{3}$
· Mean distance Landmark 3- Landmark 4:
$d^m(3,4) = \frac{d^1(3,4)+d^2(3,4)+d^3(3,4)}{3}$
COMPUTATION OF LANDMARK #4 COARSE LOCATION ON A NEW IMAGE
X_1
X_2
X_3
X_4
X_4^s
$X_1 + d^m(1,4)$
$X_2 + d^m(2,4)$
$X_3 + d^m(3,4)$
1. Location performed for Landmarks #1 - #3 in a new image
2. Each location is added with the mean distance of the FBG
3. The Coarse Location of Landmark #4 is the mean of the previous distances

(a)

(b)

Figure 6.8: Example of the Coarse Location step of the 4-th landmark during the HOG-EBGM iterative location algorithm. In this example, the following can be observed: **(a)** the off-line distances needed to locate the landmark $i = 4$, assuming that the FBG consists of three training images, and **(b)** the calculations needed for a new image to estimate the location of the landmark.

In Figure 6.9, the order of the key points located by the iterative method can be observed, as well as the geometry and size of the search areas, S_i, used in the precise location of the landmarks.



(a) **(b)**

Figure 6.9: **(a)** Location through iterations of the 25 facial landmarks in the EBGM face graph algorithm. **(b)** Refinement Search Areas for each facial landmark.

Once all the points are located, each facial landmark, fg_i, is finally defined by its coordinates, X_i, and the HOG descriptor extracted at that location. In other words, given a gray-scale image, I, and a set of landmarks, X_i, the corresponding descriptor associated with them, J_i, is calculated as the vector:

$$J_i = \text{HOG}(I, X_i) \in R^{N_{hog}} \tag{6.8}$$

where N_{hog} is the dimension of the HOG descriptor. The standard definition of the HOG feature is $N_{hog} = 128$. For a number of external tasks– such as recognition– the faces, F, need to be represented lastly by a unique feature vector. In the case of HOG-EBGM, this vector is the result of concatenating the descriptors associated with the N_p landmarks of the FG:

$$F = [J_1, J_2, \ldots, J_{25}] \in R^{N_f}, N_f = N_{hog} \times N_p \tag{6.9}$$

where N_f is the dimension of the feature vector associated with the face, F.

Assuming $N_{hog} = 128$ and $N_p = 25$, the standard dimension of the vector would be $N_f = 3200$.

6.5 Colored HOG-EBGM: CHOG-EBGM

6.5.1 Basics of CHOG-EBGM

As shown in the previous section, the HOG-EBGM does not make use of the color information from the images. To avoid this question, images are converted to grayscale during the first step of the algorithm, as seen in the first block of Figure 6.4.

Generally, a color image, A, with dimensions of $m \times n$, consists of three components (channels): C_1, C_2, C_3, most commonly R, G, B. Treating these channels as column vectors, the image A can be expressed as the matrix:

$$A = [C_1, C_2, C_3] \in R^{N \times 3}, N = m \times n \tag{6.10}$$

When performing a grayscale conversion of the color image A into an intensity image I, these three components are combined linearly, such that:

$$I = [w_1 C_1 + w_2 C_2 + w_3 C_3] \in R^N, N = m \times n, \tag{6.11}$$

reducing the dimensionality from $N \times 3$ to N, and subsequently reducing the amount of information.

The method proposed here, Colored HOG-EBGM (henceforth CHOG-EBGM), is an adaptation of HOG-EBGM, using the information given in C_1, C_2 and C_3. CHOG-EBGM addresses the problem of the loss of information during the conversion of the color images to grayscale by including the color information. To this end, two aspects are taken advantage of:

1. As proven in Section 6.7.1, the facial landmark localization step in HOG-EBGM has better accuracy than other algorithms, such as Wiskott's original EBGM [117].

   This allows the grayscale HOG-EBGM location algorithm to be used by the CHOG-EBGM. Including color information in the location stage would considerably increase the computational burden.

2. Some authors [4, 19, 109] have shown that HOG descriptors are empowered for recognition tasks when they embed color information.

   With this knowledge, the CHOG-EBGM is expected to perform better than the intensity-based HOG-EBGM, as the color descriptors contain a greater quantity of information useful in recognition tasks. In the ideal case of uncorrelated information between C_1, C_2 and C_3, the texture information would be increased up to three times more than the information in the intensity image.

It remains an open question which color spaces contain more discriminative information suitable for recognition tasks. Next, the implementation and the details of the CHOG-EBGM are defined.

6.5.2 Theoretical Development

As stated before, CHOG-EBGM is a HOG-EBGM-based algorithm, thus it consists of the same three steps revealed in Figure 6.4: *image normalization*, *creation of the graphs* and *feature extraction*. Figure 6.10 displays an example of color face graph location and the extraction of the local color features from an input image.

Through these steps, similar to those in the grayscale algorithm, each face in CHOG-EBGM is described by a Color Face Graph (FG_C), composed of automatically located facial landmarks, X_i, and their associated color-HOG (CHOG) descriptors, J_i^c.

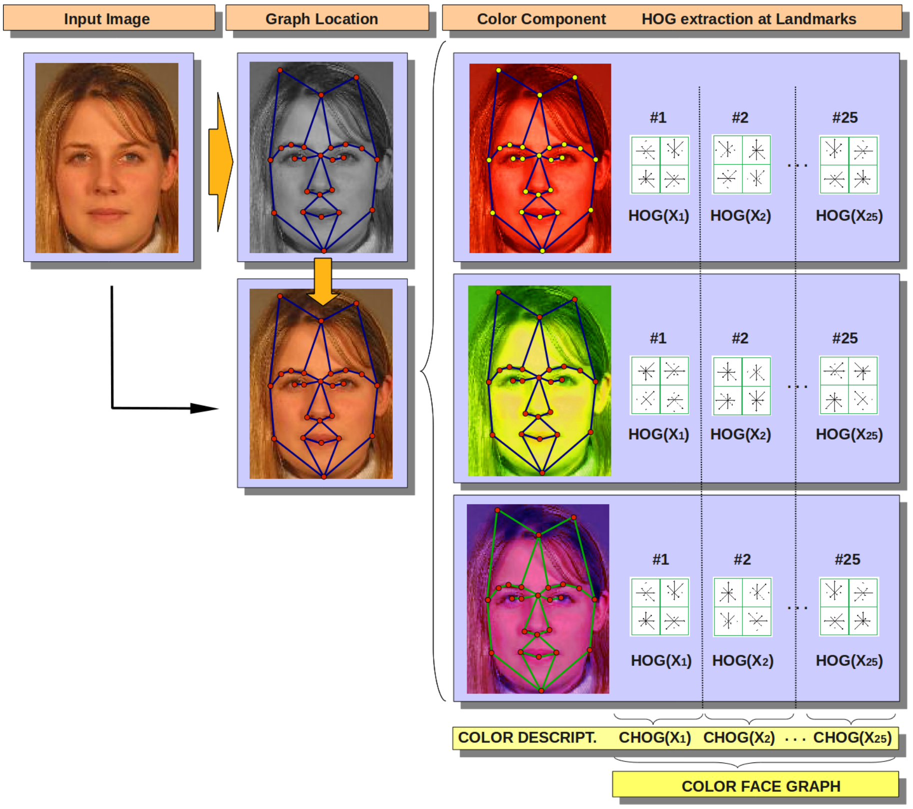


Figure 6.10: Example of the location of facial landmarks in CHOG-EBGM and the extraction of the local features.

Considering a Color Face Graph based on the one proposed for HOG-EBGM, it can built from $N_p = 25$ facial landmarks, fg_i^c, such that:

$$fg_i^c = [X_i, J_i^c(X_i)], \tag{6.12}$$

and

$$FG_C = \bigcup fg_i^c, \forall i, 0 \leq i \leq N_p \tag{6.13}$$

For the first and second steps of CHOG-EBGM, the same process described in Section 6.4 is reproduced exactly using the gray-scale information for the automatic location of the landmarks. This can be seen in the second column of Figure 6.10. Given an input color image, it is first converted to grayscale, the location of the facial landmarks are computed and then the resultant graph is translated back to the original color image.

The image normalization step is also carried out as in Section 6.4. But note that in this case, the grayscale conversion is only used for the location of the landmarks, and not for the final extraction of the descriptors.

The main difference between CHOG-EBGM and HOG-EBGM is in the feature extraction step. The color descriptor proposed, J_i^c, is extracted from every facial landmark, X_i, by independently applying an HOG descriptor for each of the color components, and then concatenating them:

$$J_i^c = [HOG(C_1, X_i), HOG(C_2, X_i), HOG(C_3, X_i)] \in R^{N_{chog}}, \tag{6.14}$$

where N_{chog} is the dimension of the color HOG descriptor, which is actually $N_{chog} = N_{hog} \times 3$. Assuming a dimension for the standard HOG descriptor of $N_{hog} = 128$, as seen in the previous section, the final dimension of the color version would be $N_{chog} = 384$.

For recognition tasks, a color face F_C is represented by the vector derived from the concatenation of the $N_p = 25$ color descriptors on the FG_C:

$$F_C = [J_1^c, J_2^c, \dots, J_{25}^c] \in R^{N_c} \tag{6.15}$$

where N_c is the dimension of the feature vector associated with a color face, F_C, and is defined as $N_c = N_{chog} \times N_p$. For standard values, the final dimension of the color feature vector is $N_c = 9600$.

6.6 Set-up Experiments for HOG-based Face Recognition Algorithms

The current section is devoted to set-up all the face recognition techniques developed. These techniques have in common the use of HOG local descriptors, and therefore, first a set-up on the features itself has to be performed[2]. The set-up experiments can be divided in the following;

- **Set-up of the size of the local HOG features for face recognition**: the experiments will be tested using hand-marked Face Graphs, although the results will be extrapolated to be used with the graph location techniques.
- **Adaptation and set-up of two Face Graph Algorithms to be used with HOG features**: two algorithms are adapted, Active Appearance Models and Rigid Square Grids. The resulting adaptation are called *HOG-AAM* and *HOG Rigid Square Grids*, respectively.

  Unlike the HOG-EBGM, the two FGAs proposed are not novel but only slightly adapted to the needs in this thesis. In these algorithms, the landmark location stage is not modified, and only during the feature extraction are the HOG features included.

6.6.1 Optimum size of the HOG descriptor for Face Recognition

The majority of HOG internal parameters are determined by the solution proposed by Lowe [65] for the description of objects. Yet, the size of the local feature window has not be defined to adapt to face recognition. The scale factor of the original SIFT features is determined during the location stage (using the *scale-space* transform). In the case of the HOG features, the scale factor of the image is fixed according to the normalization of the face (see Chapter 2) and therefore the feature wide size has to be arranged relative to it.

[2] This set-up is complementary with the one presented in Section 4.6

It is necessary to determine the optimal number of pixels P_{HOG} that best characterizes facial elements for a given normalization. The parameter P_{HOG} determines the *quantity* of information that can be retrieved from the element that is described, and also relates to the *specificity* of such information:

- **Quantity of information**: low values of P_{HOG} lead to a too small feature window, compared to the element described. This removes potentially useful information for discrimination.
- **Specificity of the descriptor**: high values for P_{HOG} may lead to descriptor windows greater than the element that is described. In this case the descriptor includes information of elements different from the goal. The result is a noisy and less discriminative descriptor. Therefore, the selection of a value for the feature window size is a trade-off between quantity and specificity of the discriminative information that can be extracted from an element.

The experimental set-up addresses the trade-off of this two parameters. The results are valid for the extraction of local features in face recognition, as well as for eye location (see Chapter 5).

In these experiments, three assumptions have been considered:

1. Regardless the facial element that is described, a square HOG descriptor window of size $P_{HOG} \times P_{HOG}$ is used. Thus, the value of P_{HOG} has to be valid for all the facial elements.

   The HOG descriptor is organized as a 4 × 4 square grid (as shown in Figure 6.11), limiting the range of values for P_{HOG}.

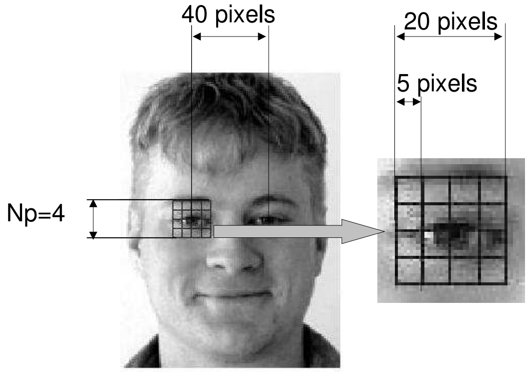


Figure 6.11: Representation of the HOG descriptor extracted the right eye, with an square window shape and a organization as a 4 × 4 square grid.

2. We assume that the size of the face is normalized such that inter-ocular distance, *iod*, is known *a priori*. Assuming this, the descriptors are extracted on facial elements with fixed relative sizes. In this experiment„ the inter-ocular distance is set to $iod = 40$ pixels.
3. The size of the most significant facial elements, specifically the eyes, the mouth and the nose, are approximately in the same order of magnitude

compared to the size of the face (and thus, regarding the inter-ocular distance). With this assumption, we can state that the value of P_{HOG} determined will be valid for all the facial elements without losing generality of the results.

The experiment to determine the optimum size P_{HOG} consists on describing faces extracting HOG features on a set of manually marked facial landmarks (corresponding to the $N_p = 25$ landmarks in the EBGM algorithm). The experiments are repeated for different sizes of the descriptor window and the hit rates are computed.

Given a point $p(x, y)$, the HOG descriptor centered at it and with window size $P_{HOG} = s$, is $HOG(x, y, s)$. For each image, the final face feature vector, D^i, is extracted as the concatenation of the 25 keypoints manually marked:

$$D^i(s) = [HOG(x^i_1, y^i_1, s), HOG(x^i_2, y^i_2, s), \ldots, HOG(x^i_{25}, y^i_{25}, s)], \tag{6.16}$$

where x^i_k and y^i_k, correspond to the location of the k-th keypoint (facial element) on the i-th image of a dataset.

The matching of the face feature vectors was done using a *Nearest Neighbor* classification with *Euclidean distance.* In the experiments the extraction of the face feature vectors was performed using different values of the HOG window: $P_{HOG} = 0.2iod, P_{HOG} = 0.3iod, P_{HOG} = 0.4iod, P_{HOG} = 0.5iod, P_{HOG} = 0.6iod$ and $P_{HOG} = 0.7iod$. Given the initial assumption of inter-ocular distance $iod = 40$ pixels, this size values configure HOG feature windows of size $S_1 = 8 \times 8$, $S_2 = 12 \times 12$, $S_3 = 16 \times 16$, $S_4 = 20 \times 20$, $S_5 = 24 \times 24$ and $S_6 = 28 \times 28$ pixels, respectively. Let's remark that all the P_{HOG} values tested are multiples of 4. This is due to the configuration on the HOG window, which is a square grid with $N_p = 4$ cells on each direction.

The results using the databases of Yale and CVL[3] are displayed in Figure 6.12.

From the identification rates observed in the figure, it can be deduced that the behavior for the two databases under analysis is different. In both cases, the curves present a maximum for middle values of P_{HOG}, in concordance to the previous reasoning. However, the optimum is reached with different window sizes for each database. In the case of the curve obtained using the images on CVL, the maximum is produced with a window size $P_{HOG} = 0.5iod$, while in the case of the results obtained when using the Yale database, the identification rate is approximately flat in the range of sizes $0.3iod \leq P_{HOG} \leq 0.5iod$, with a tendency of decreasing while P_{HOG} increases.

A reasonable explanation can be found on the divergence of results in the two databases: when the size of the HOG window is reduced, the local features cover a smaller area of the facial element and therefore are less sensible to noise introduced by extrinsic factors, like changes in illumination. This also explains the decreasing tendency of the rates for both databases for larger window sizes. Yale contains higher contrasts to illumination, and therefore for smaller sizes the performance is less affected.

As a conclusion, the optimum size of the HOG feature window describing facial elements is reached when using $P_{HOG} = 0.5iod$. Given the face normalization explained in Chapter 2, this size corresponds to a square HOG window of 20×20 pixels centered in the keypoint that is going to be described.

[3] See Appendix B

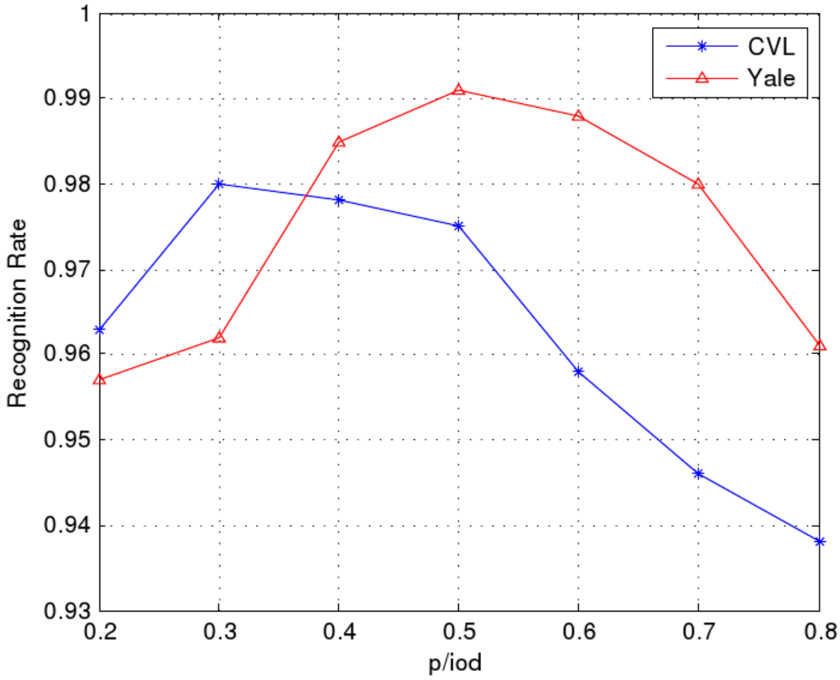


Figure 6.12: Study of the influence of the HOG window size, P_{HOG} in the performance for identification tasks. The curves plotted correspond to the results obtained for the Yale and CVL databases. The size of the feature window, $P_{HOG} \times P_{HOG}$ is normalized with the inter-ocular distance, *iod*.

Henceforth, in this work we will use this window size for the HOG features. Also for the sake of simplicity, we will refer to the descriptor $HOG(x, y, s = 20)$, just as $HOG(x, y)$.

6.6.2 Face Recognition using HOG-AAM

Several works have proven the efficiency of the location stage provided by the Active Appearance Model technique. When the AAM is used as a Face Graph Algorithm, the extraction of landmarks for biometric analysis can adapt to a number of facial variations, making it worthy of study for the goals of this thesis.

In this work, it seemed worthwhile to develop an adaptation of the AAM Face Graph Algorithm where the HOG features could be extracted from the landmarks after the graph was matched with a face. The algorithm resulting from this adaptation is called HOG-AAM, and for its implementation, the AAM location stage proposed by Milborrow *et al.* [78] was followed.

Regarding the HOG-AAM a preliminary set of experiments has been performed to decide the number of landmarks of the final face graph generated.

Experiments to determine the number of points in HOG-AAM

During the training stage of the Active Appearance Models, the number of keypoints that constitute the facial graph can be configured. Milborrow et al. work with graphs of $N_n = 58$ points. In Figure 6.13, an example of an AAM face graph generated using Milborrow's approach can be observed.

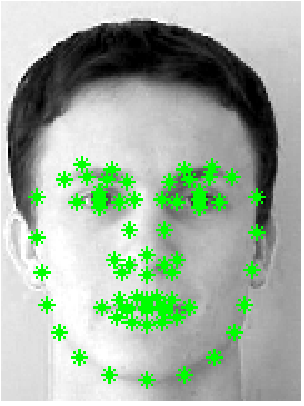


Figure 6.13: Real example of a face graph generated using the AAM model matching technique, with graph models of $N_n = 58$ landmarks.

In accordance with the ideas exposed for the EBGM-based techniques, a balance needs to be struck between the number of keypoints in a graph (i.e. the quantity of information that can be retrieved) and the computational and storage cost of having large N_n values. Compared to the rest of the Face Graph Algorithms proposed in this chapter, the Active Appearance Models have a larger quantity of keypoints, only surpassed by the Dense Square Grids, as it will be seen next. As the HOG-EBGM has $N_n = 25$, one goal was to reduce the number of points in AAM to make them comparable.

Three configurations were considered for the landmark locations using the AAM face graph algorithm:

- **Original AAM (58-AAM)**: In this configuration, the original AAM designed by Milborrow *et al.* [78] is used explicitly, with $N_p = 58$ facial landmarks. This model constitutes the foundation for the rest of the model configurations.

  The dimension of the face-feature vector in this configuration, using HOG features, is $dim_{\text{o}} = 7424$.

- **Coincident AAM (22-AAM)**: In this configuration, the AAM graph model consists of $N_p = 22$ keypoints. Starting from Milborrow's facial model, only those points which completely coincided in location with the facial landmarks extracted with EBGM (see Section 6.4) were selected. With this modus operandi, up to 22 coincidences were found. Specifically, the facial landmarks selected corresponded to the whole set of landmarks in the EBGM Face Graph, in addition to the upper-head and fore-front points. Figure 6.14 shows the equivalence of landmarks between the HOG-EBGM and the HOG-AAM graphs.

The dimension of the feature vector corresponding to the Coincident HOG-AAM is $dim_{\mathrm{c}} = 2816$.

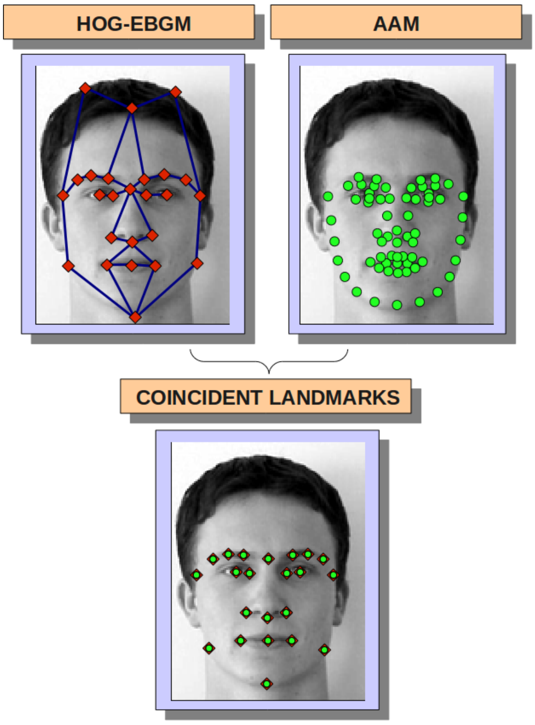


Figure 6.14: Example of the 22 coincident landmarks between the face graph location algorithms of HOG-EBGM and the standard AAM.

- **Extended AAM (25-AAM)**: In this configuration, the AAM graph model consists of $N_p = 25$ key points. This configurations uses the 22 landmarks located by the *Coincident AAM* and the three additional landmarks automatically generated from the rest of the points using geometrical cues. This configuration was developed to be fully comparable in number of landmarks between the Sparse Square Grid and the HOG-EBGM, and also comparable in location with EBGM.

  The dimension of the feature vector corresponding to the Extended HOG-AAM is $dim_{\mathrm{e}} = 3200$.

Table 6.2 summarizes the key parameters of the three AAM model configurations. The adaptation of the three AAM configurations was done by extracting a HOG local descriptor at each of the key points of the final graph. That is, for each graph keypoint, i, $HOG^i(x_i, y_i, s)$ was extracted, where $(x_i, y_i), 1 \leq i \leq N_p$ are the coordinates of the landmark and s is the size of the square HOG descriptor window. In our case, pixels were set at $s = 20$, as described in Section 4.6.

For the validation of the three HOG-AAM configurations, some experiments were carried out to compare the derived models ($N_{p1} = 22$ and $N_{p2} = 25$, respectively) with the model designed by Milborrow *et al.* ($N_{po} = 58$). The experiments were carried out as follows:

	AAM Model Configuration		
	Original	**Coincident**	**Extended**
Facial Landmarks	58	22	$22 + 3$
Feature Vector Dimension	7424	2816	3200
Coincident Landmarks (EBGM)	22	22	25

Table 6.2: Summary of the main parameters of the AAM models for HOG-AAM.

1. Two different datasets of face images were selected to perform the matches. In this experiment the images from the FERET face database were used. the *fa* subset for the target images and the *fb*, *fc*, *dup1* and *dup2* subsets for the query datasets [4].

   Given the groundtruth on the location of the eyes, all the images in these datasets were normalized following the guidelines in Chapter 2.

2. For all the configurations, the landmark location algorithm was run on all the datasets. In this step, the original, coincident and extended versions of the AAM models were used. The face feature vector was extracted by concatenating all the HOG descriptors resulting from the matched graph.

3. Due to the high dimensionality of the face feature vectors obtained (see Table 6.2), a reduction method needed to be applied. In this case, a simple LDA technique was selected, reducing the dimension for all the face feature vectors to $dim_{red} = 200$.

4. Finally, the identification performance results were obtained by matching the face graphs extracted from the query test images to the face graph extracted from the target images. In this case, the matching was obtained using a cosine distance similarity measure.

Figure 6.15 shows the curves obtained after this experiment, using the three HOG-AAM model configurations. The curves display the performance in identification versus the number of landmarks in the graph model.

From the analysis of the results, we observe that the curves tend to increase with the number of keypoints in the face graph. This fact corresponds to what was expected, as a higher value of N_p implies more biometric information retrieved. As we will see in further experiments, this is consistent with the results obtained with Dense Grids. The different performance between the 22-AAM and the 25-AAM is much higher than difference in performance between the 25-AAM and the 58-AAM.

This result is significant considering that the original configuration doubles the quantity of landmarks of the others. This is an indicator that many of the keypoints in the graph model proposed by Milborrow contain redundant information. The omission of such keypoints does not reduce significantly the discriminability of the graph. Observing the location of the 58 keypoints of the Original AAM, as shown in Figure 6.13, it can be noticed that the spatial

[4] See Appendix B for further details

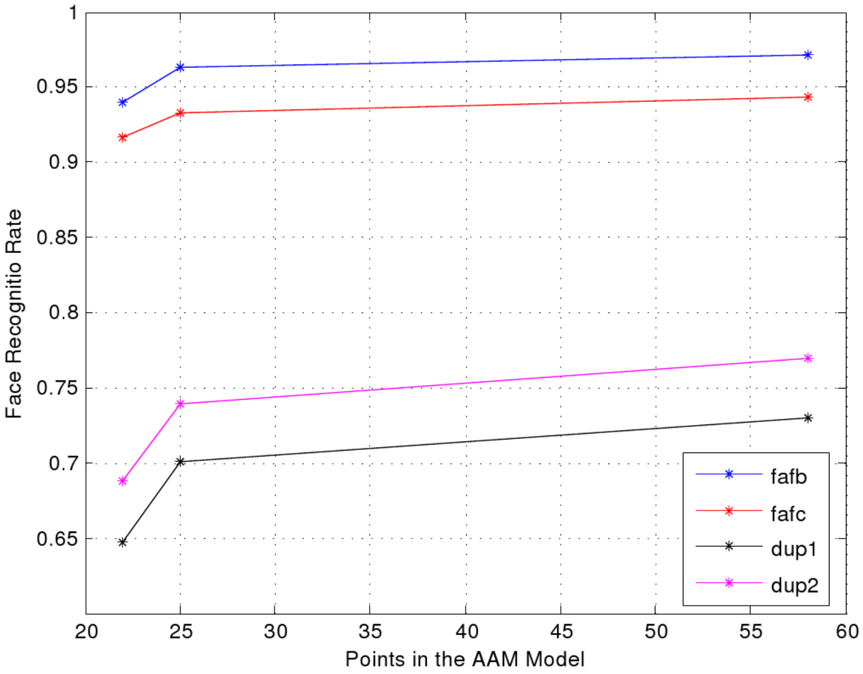


Figure 6.15: Identification hit rate in the FERET database versus the number of keypoints in the graph model, using the original ($N_p = 58$), coincident ($N_p = 22$) and extended ($N_p = 25$) configurations of the HOG-AAM models.

distance between landmarks is small, so when the HOG-features are extracted, high rates of overlapping take place.

From these validation experiments we can conclude that the HOG-AAM achieves higher results when using the Extended AAM configuration, using a face model with $N_p = 25$ keypoints. The number of points in this configuration is the same than in the graphs generated with the EBGM location algorithms, and also the quantity of information is not reduced much with regard to the Original AAM model with $N_p = 58$ keypoints.

6.6.3 Face Recognition using HOG-Grids

HOG Rigid Square Grids is an FGA algorithm produced by adapting the Non-deformable rigid square grids showed in Section 6.3.2, and therefore uses the mapping of the location of features described there. Once the keypoints are placed on a face, a HOG local texture descriptor is extracted from each. The facial graph obtained is the concatenation of the HOG feature vectors extracted from the grid nodes.

To validate the HOG Rigid Square Grids algorithm, some experiments were designed. This experimental set-up studied the behavior of HOG-Grids two possible configurations: the *Sparse Square Grids* and the *Dense Square Grids*.

For all the experiments, the face normalization stage proposed in Chapter 2 was used, which implies a face width of $w_f = 120$ pixels and a inter-ocular distance of $iod = 40$ pixels. Also used was the HOG feature window size proposed in Section 4.6, which is $P_{HOG} = 0.5iod = 20$ pixels.

In all, three different square grids were studied:

1. **Sparse Square Grid (25 HOG-Grid)**: This configuration came from a grid with 5×5 nodes, which corresponded to a total of $N_n = 25$ landmarks. In this configuration, the separation between neighbor landmarks was 30 pixels, which introduced an overlap between the HOG feature windows of $o_{25} = 25\%$. The face feature vector extracted, after concatenating the HOG features extracted at each landmark, had dimension $dim_{25} = 3200$.

   This square grid size of $N_n = 25$ located landmarks was chosen to offer maximum comparability between the rest of the FGAs described in this chapter, specifically the HOG-AAM and the HOG-EBGM. All the algorithms extracted an equivalent number of facial landmarks, so a fair comparative study could be performed.

2. **Dense Square Grid (81 HOG-Grid)**: This configuration came from a grid with 9×9 nodes, which corresponded to a total of $N_n = 81$ landmarks. In this configuration, the separation between neighbor landmarks was 15 pixels, which introduced an overlap of the windows of the local features of $o_{81} = 63\%$. The face feature vector extracted, after concatenating the HOG features extracted at each landmarks, had dimension $dim_{81} = 10368$.

3. **Highly Dense Square Grid (81 HOG-Grid)**: This configuration came from a grid with 13×13 nodes, which corresponded to a total of $N_n = 169$ landmarks. In this configuration, the separation between neighbor landmarks was 10 pixels, which introduced an overlap of the windows of the local features of $o_{169} = 75\%$. The face feature vector extracted, after concatenating the HOG features extracted at each landmarks, had dimension $dim_{169} = 21632$.

Figure 6.16 displays an example of the location of the facial landmarks using the three square grids described earlier. Additionally, Table 6.3 summarizes the parameters of these grids.

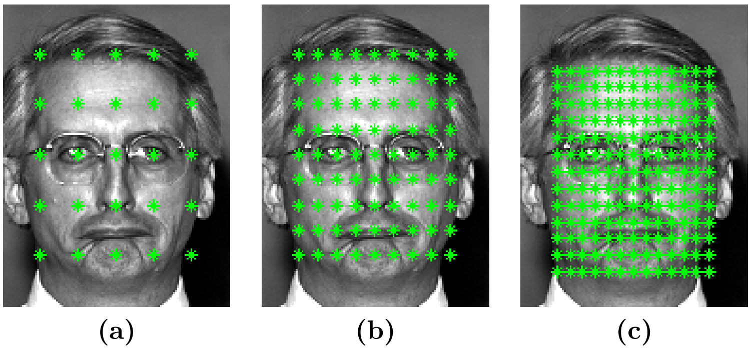


Figure 6.16: Example of facial landmark location using Rigid Square Grids: **a)** Sparse Grids ($N_n = 25$). **b)** Dense Grids ($N_n = 81$). **c)** Highly Dense Grids ($N_n = 169$)

To evaluate the behavior of the face graphs generated by these grids:

	Sparse Grid	**Dense Grid**	**Highly Dense Grid**
Grid Size	5×5	9×9	13×13
Facial Landmarks	25	81	169
Feature Vector Dimension	3200	10368	21632
Feature Overlapping	$^1/_4$	$^2/_3$	$^3/_4$

Table 6.3: Summary of the main parameters of the different HOG-GRID configurations.

1. Two different datasets were selected to build the target and query datasets. In this experiment, the images from the FERET face database were used[5]. Specifically, the *fa* subset for the target images and the *fb*, *fc*, *dup1* and *dup2* subsets for the query datasets were used.

   Given the ground-truth on the location of the eyes, all images in these datasets were normalized following the guidelines of Section 2.4.

2. For all the configurations, the keypoints in every image of all datasets were distributed on top of the image. In this step the sparse, dense and highly dense square grids were used.

3. Due to the high dimensionality of the face feature vectors obtained (see Table 6.3), a reduction method needed to be applied. In this case, an LDA technique[6] was selected, reducing the dimension for all the face feature vectors to $dim_r = 200$.

4. Finally, the identification performance results were obtained by matching the face graphs extracted from the query images to the face graphs extracted from the target images. In this case, the matching was obtained using a *cosine distance* for the similarity measures.

The results obtained in these experiments can be seen in Figure 6.17, in which the curves represent the identification rates for a given number of vertex in the grid (i.e. the number of keypoints located).

As expected regarding the behavior of the grid configurations, it was observed that the curves had a clear tendency to achieve better identification results as the number of points, N_p, were increased. This result is coherent with the analysis made before, as a higher number of keypoints leads to a higher quantity of information.

However, the difference in the results obtained with $N_p = 81$ and $N_p = 169$ is of low relevance, which indicates a tendency to the *overfitting*. Thus, it could be said that the best grid size would correspond to $N_p = 81$. As in the case of the Highly Dense Grid (169 HOG-Grid), the increase in the identification rates is not relevant, while the computational cost is higher.

In further experiments, the performance of the Square Grids needed to be compared to other FGAs, with $N_p = 25$ landmarks in each. Thus, it was decided to use not only the Dense Grid configuration (25 HOG-Grid), but also the Sparse Square grid (81 HOG-Grid) with $N_p = 25$ keypoints.

[5] See Appendix B

[6] See Appendix A for further details.

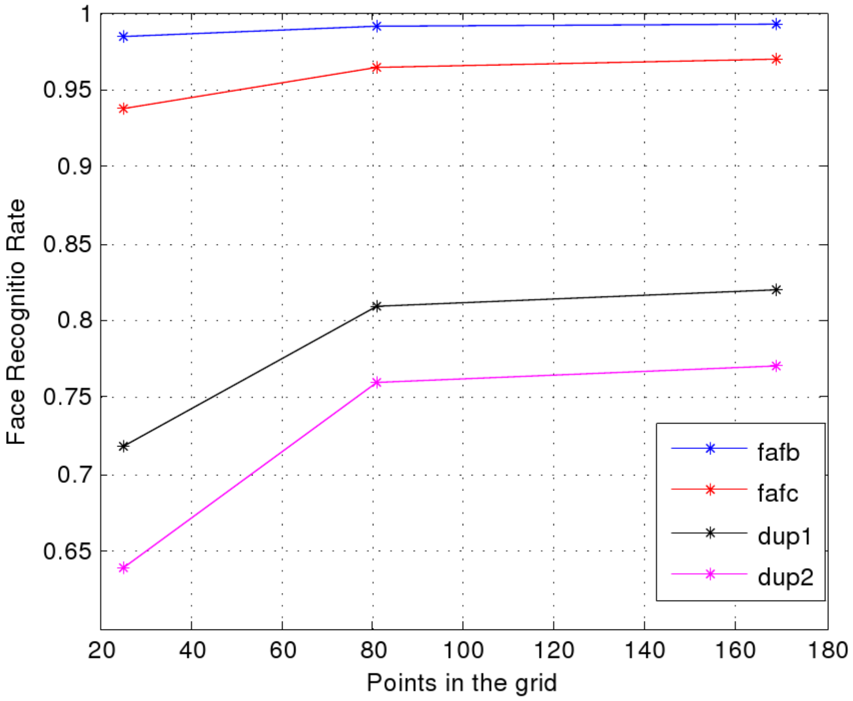


Figure 6.17: Identification rates for different square grid configurations using HOG features as local descriptors, for images from the FERET database. The square grids analyzed have the following sizes: 5×5, 9×9 and 13×13

.

6.7 Evaluation of the HOG-EBGM algorithm

This section is focused on proving the hypothesis concerning HOG-EBGM in order to generate Facial Graphs for recognition tasks. In proving these hypotheses, this algorithm is evaluated as an alternative to other face recognition algorithms. With this goal in mind, four sets of experiments were carried out in this work:

1. A comparative study between HOG-EBGM and Gabor-EBGM. The experiments focus on two main aspects, the accuracy on the location of the landmarks in the Face Graph and the descriptive power of the local features used.

2. Experiments to validate the accuracy on the location of the landmarks, compared to HOG-AAM

3. A comparative study between HOG-EBGM and some conventional, holistic face recognition algorithms. These techniques are used as a baseline useful to validate the results.

4. Analysis of different feature-based approaches using HOG descriptors to discriminate local textures. The experiments sought to validate the HOG-

EBGM against two adapted Face Graph Algorithms: HOG Rigid Square Grids and HOG-AAM.

The rest of this section provides the results of the four sets of experiments.

6.7.1 HOG-EBGM versus Gabor-EBGM.

This section is aimed at validating the HOG-EBGM algorithm comparing it to the original EBGM.

Some sets of experiments were conducted to analyze the effect of changing the local features (Gabor filters with HOG descriptors) on the descriptive potential of the EBGM Face Graph. Another goal was to compare the stability and the precise location stage achieved by both algorithms.

The three sets of experiments carried out for the comparative are:

1. Analysis of the performance for identification problems, given a prior location (ideal location) of the Face Graphs in the test images. In this case, the groundtruth is the same for both algorithms.

2. Influence of the number of images in the FBG training models on the recognition rates.

3. Accuracy obtained in the automatic landmark localization stage.

Notice that in all the experiments to compare HOG-EBGM and Gabor-EBGM, the classification of the Face Graphs is done using a *Nearest Neighbor* algorithm based on the distances obtained landmark by landmark.

Next, the implementation and results obtained in each set of experiments is described.

Face Identification using ideal location of the landmarks.

This first set of experiments was designed to compare the descriptive capacity of the feature vector obtained from each of the methods with independence from the landmark location. To achieve this independence, the experiments assumed ideal facial landmark locations; under this assumption, it was not necessary to use the training model information to build the Face Graphs. The location of the keypoints was manual, given as an input ground-truth in the data. As both algorithms shared the location of the FG, the comparative was only between the descriptors that were extracted.

As a consequence of the ideal location of the FGs, the identification results obtained can be considered the upper bound for the algorithms, as they establish their limit in performance.

In this experiment, the testing sets were the Yale and CVL databases[7]. They were selected since they contain a ground-truth with additional information about the location of the $N_p = 25$ facial landmarks that constituted both EBGM algorithms. The evaluation methodology used for the test was *leave-one-out*[8].

The performance results for face identification in each of the test sets are shown in Table 6.4.

[7] See Appendix B

[8] Explained in Chapter 3

Database	Gabor-EBGM	HOG-EBGM
Yale	85.5%	97.0%
CVL	96.5%	99.1%

Table 6.4: Identification Rates with Gabor-EBGM and HOG-EBGM, using ideal (hand-marked) location of the landmarks.

HOG-EBGM outperforms the rates of Gabor-EBGM in both test datasets. As the location of the keypoints was the same for both algorithms, the first conclusion that can be drawn is that the descriptive power of the HOG features for the same facial landmarks is greater than that achieved with Gabor coefficients.

Analyzing more in detail, it was found that the difference of rates obtained by both algorithms was greater using the Yale dataset. This can be explained by the fact that the images in Yale contain greater variance in illumination, and the HOG descriptors are less vulnerable to illumination changes than the Gabor filters.

Summarizing, the results of this experiment show that for specific facial landmarks, the HOG descriptors outperform the Gabor filters in extracting a more descriptive feature vector.

Influence of the number of model images in the FBG in face identification rates.

The goal of this second set of experiments was to determine the influence of the number of training models, N_f, on face recognition rates. The stability of the facial landmarks located with HOG-EBGM and Gabor-EBGM was measured when the number of models in the FBG varied. Notice that in EBGM, the landmark location relied on the information from the FBG models. The stability in the location provides useful information about two measures:

- Repeatability of the location algorithm: A high variance in the location of the keypoints when the number of models are changed would imply bad repeatability. This would mean a high dependence on the results obtained regarding the specific FBG selected.
- Stability of the recognition results: With this measure, one can determine if the recognition results obtained with the EBGM algorithms depend on the models in the FBG.

The experiments designed in this section consisted of generating Facial Graphs with the HOG-EBGM and Gabor-EBGM location algorithms, while varying the number of models in the FBG. Each FBG modification produced an execution useful to determine the performance of face identification.

The FBG modifications follow a systematic process: given a set of models, the number of images, N_f, to constitute the new FBG; then, model images were randomly included until the subset was complete. The resulting FBG subset was called FBG^i_{Nf}, where i refers to the specific realization of random models.

To avoid possible dependencies of the results on a specific FBG subset, each experiment was repeated a total of 20 times, randomly changing the contents of the subsets.

The landmarks in the model images needed to be manually annotated. For the experiment, two datasets with information about the location of the landmarks were used: Yale and CVL[9]. Independence between the training set (to generate the FBG subsets) and the testing set needed to be achieved. Thus, a combination of both databases was used, such that when one was used for the training stage, the other was used as test set for identification. Therefore, while the Yale images were selected to constitute the FBG, the face recognition system was applied to CVL images, and vice versa. This produced two different datasets configurations.

The results of the experiments performed for the two configurations are shown in Figure 6.18, for $N_f \in [10, 15, 20, 25, 30, 35, 40]$. The curves in this figure represent the mean value of the identification rates for the execution repetitions. There is also a representation of the standard deviation of the performance that gives information about the confidence of the results.

The results obtained show that the performance of both algorithms is highly stable as the number of model faces in the FBG varies. For both configurations, the higher rates were achieved by HOG-EBGM. This confirms the results obtained by the experiments in the previous section.

In Figure 6.19, one can see the effect of the dispersion of the located landmarks on a specific image after the 20 repetitions of the FBG subsets with $N_f = 10$ and $N_f = 40$. Notice that this figure does not give information about the accuracy of the location of the landmarks, only their dispersion.

The stability of the location of the keypoints is higher (lower dispersion) when the number of models in the FBG increases. However, the differences are not significant according the standard deviation that is seen in Figure 6.18. Also, the variance achieved by HOG-EBGM is lower, which can partially explain the higher rates of face identification shown in Figure 6.18.

Finally, notice that the rates obtained with automatic landmark location are similar to the results with ideal location showed in Table 6.4. This leads one to think that the automatic location of the Face Graph is not only stable, but also tends towards the ideal values.

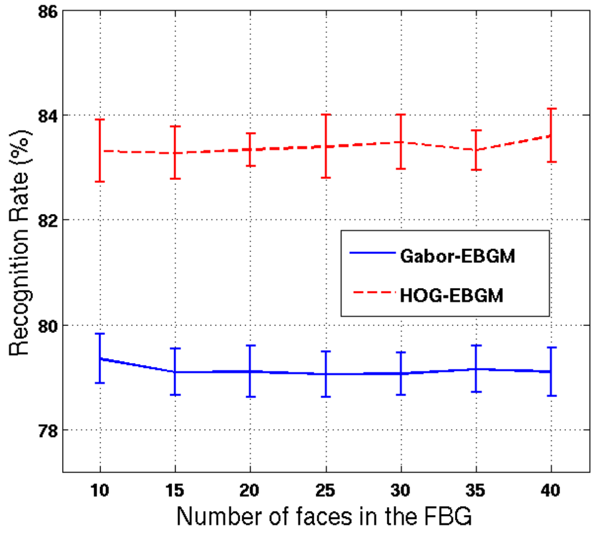
Comparison of the precision of the location of landmarks.

The EBGM iterative method has two goals: to precisely locate facial landmarks, and biometrically identify the face generating a facial feature vector. The previous experiments aimed to prove the descriptive capacity of the HOG feature vectors, but neither addressed the location.

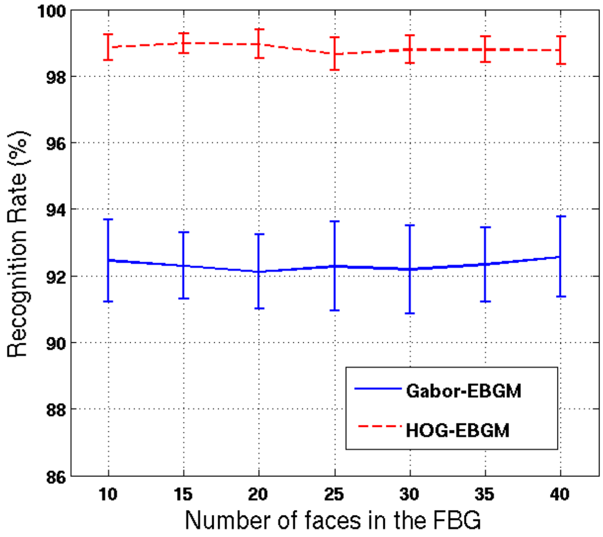
The HOG-EBGM location step is based on the original EBGM, but the use of different descriptors leads to different landmark locations. Therefore, the higher performance of HOG-EBGM versus Gabor-EBGM in identification might also be influenced by the precision of the location of the Face Graph. The stability of the location algorithm has been measured, but the accuracy of the final face graph has not been checked.

The experiments in this section attempt to compare the location of the Facial Graphs generated with both EBGM algorithms to their ideal counterpart, using a quantitative set of test images. Once again, the Yale and CVL databases were

[9] See Appendix B



(a) Yale database



(b) CVL database

Figure 6.18: Recognition rates for identification when changing the number of model images constituting the FBG in the HOG-EBGM and the standard EBGM algorithms.

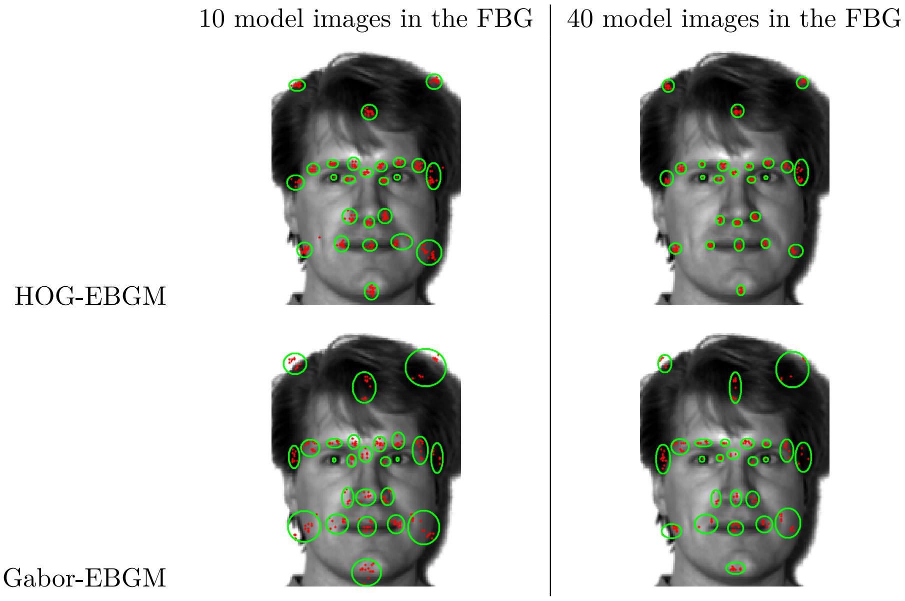


Figure 6.19: Dispersion of the facial landmarks automatically detected when using FBG model sets with 10 and 40 images and 20 repetitions.

chosen due to their extensive ground-truth information, thereby producing two configurations.

The methodology of the experiments consisted of three stages:

1. A Facial Graph was generated for each of the N_t images in the test dataset. In this case, the two test datasets used consisted of the whole set of valid images from Yale and CVL, respectively.

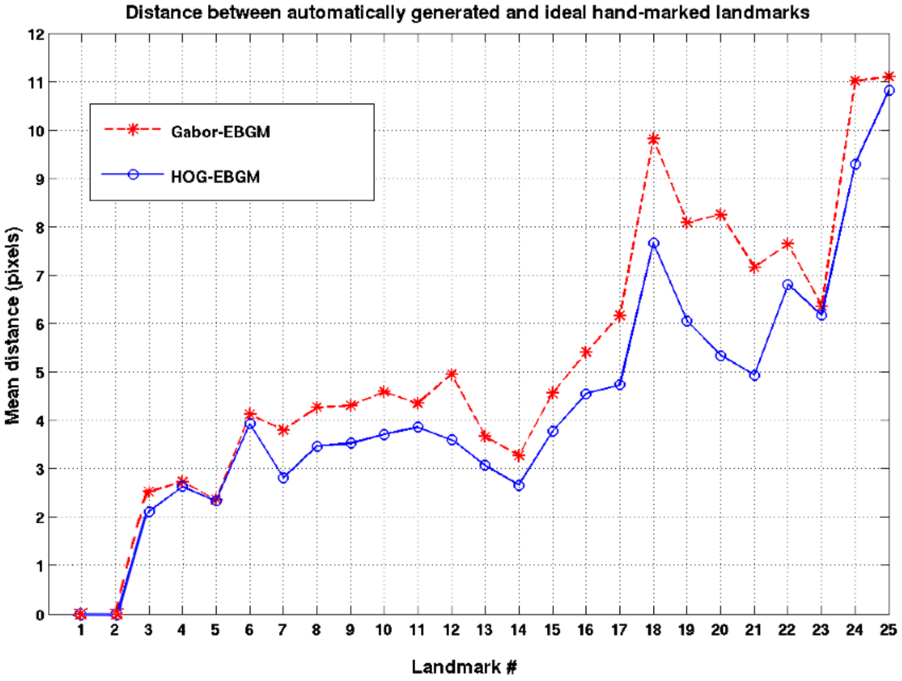
2. For each of the landmarks, the distance was calculated between the location given by the iterative method, $X_i(k)$, and their ground-truth information $X_{fbg_i}(k)$, where $1 \leq i \leq 25$ was the number of the facial landmark and $1 \leq k \leq N_t$ was the number of the image in the dataset.

3. For each set of distances belonging to the same landmark, i, the mean distance was extracted:

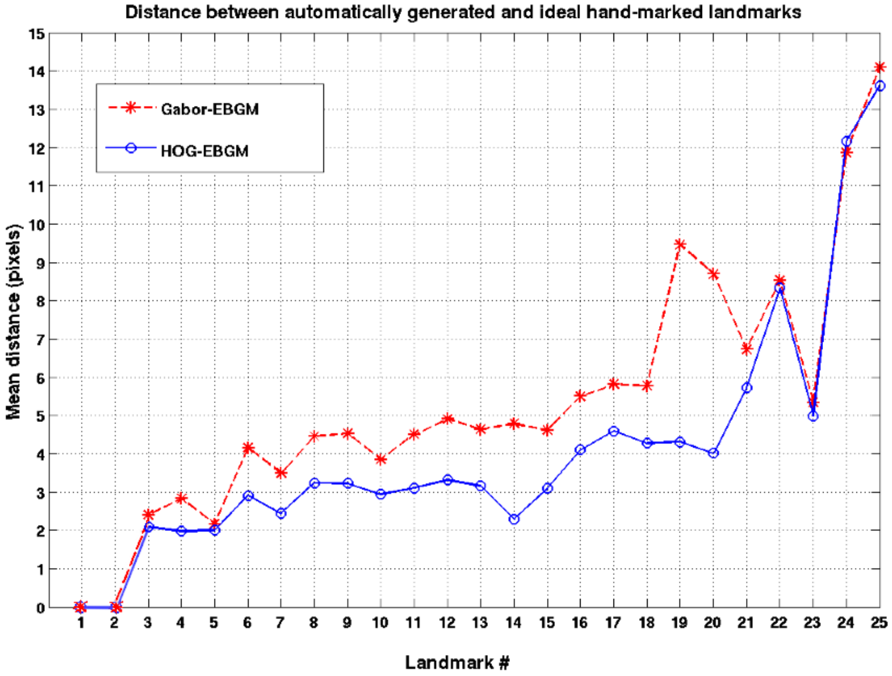
$$dist_i = \frac{1}{N_t} \sum_{k=0}^{N_t} \|X_i(k) - X_{fbg_i}(k)\| \tag{6.17}$$

The representation of the mean distances for each landmark (sorted by the order shown in Figure 6.9) can be seen in Figure 6.20. The two set of curves correspond to the two dataset configurations.

From the results, a clear tendency can be perceived: the location of the graph with HOG-EBGM is closer to the ideal for all the landmarks than with Gabor-EBGM. This means that HOG-EBGM achieves higher precision in the location step, which leads to an improvement in identification tasks. For most of the landmarks, the location error achieved using HOG features was between 5% and 15% lower than with Gabor Filters.



(a)



(b)

Figure 6.20: Curves of the distance between ideal landmark localizations (from the groundtruth) and the positions automatically obtained with HOG-EBGM and Gabor-EBGM for: **(a)** Yale database, **(b)** CVL database.

Figure 6.21 shows a graphic representation of the difference in the location between two Face Graphs, generated with HOG-EBGM and Gabor-EBGM, and the ideal location of those keypoints extracted from the ground-truth of the image.

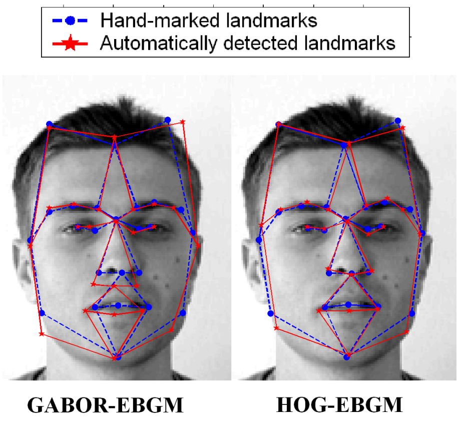


Figure 6.21: Comparison of Facial Graphs located using the iterative algorithms of Gabor-EBGM (left) and HOG-EBGM (right) with regard to the ideal location of the same keypoints.

The increasing tendency of the curves in Figure 6.20 should also be explained. During the EBGM Face Graph creation algorithm showed in Section 6.4, each keypoint was located using spatial information from the landmarks previously located. Assuming null initial error the eye centers, the spatial error of the following landmarks was always cumulative. That is, a landmark located with high inaccuracy increases the error in the location of the resting landmarks.

6.7.2 Landmark Location: HOG-EBGM versus AAM

In the previous experiments, it has been proved that the accuracy on the location of the Face Graph generated with HOG-EBGM is higher than the one obtained with the original EBGM. In this experiment, HOG-EBGM is compared to a completely independent Face Graph algorithm: Active Appearance Models.

To perform this comparison, the configuration of 22-AAM was selected. This experiment compared the precision on the location of the graphs generated with HOG-EBGM and AAM to a ground-truth of hand-marked keypoints. The Yale and CVL databases were used, as they contain such ground-truth information. From them, two different testing sets were generated, corresponding to whole sets of images in Yale and CVL, respectively.

The experimental set-up for the comparative can be summarized in the following steps:

1. Extraction of the facial graphs. For each face image in the test set, a face graph using HOG-EBGM and AAM was located. For each landmark, the location with the two algorithms can be expressed as $X_i^{ebgm}(k)$ and $X_i^{aam}(k)$, respectively, where i is the landmark number, $1 \leq i \leq 25$ and k represents the order of the image in the testing set, $1 \leq k \leq N_t$.

2. Estimation of the distance between the k-th landmark location given by each of the iterative methods, $X_i^{ebgm}(k)$ and $X_i^{aam}(k)$, to the ground-truth information, $X_{fbg_i}(k)$.

3. Extraction of a mean distance value for each method. The mean distance for HOG-EBGM and AAM (respectively), considering i-th landmark, can be expressed as:

$$dist_i^{ebgm} = \frac{1}{N_t} \sum_{k=0}^{N_t} \| X_i^{ebgm}(k) - X_i^{GT} \| \tag{6.18}$$

and

$$dist_i^{aam} = \frac{1}{N_t} \sum_{k=0}^{N_t} \| X_i^{aam}(k) - X_i^{GT} \| \tag{6.19}$$

The representation of the mean distances is displayed in Figure 6.22. The two sets of curves represent the distances obtained for the Yale database and the CVL database.

In both algorithms, the order of the landmarks is the one proposed by the CSU Evaluation System [17] for the standard EBGM. However, there is no data in the interval $18 \leq i \leq 20$ (the curves are displayed as dotted lines). This is explained by the fact that there were only 22 coincident landmarks between HOG-EBGM and the Coincident AAM. In Figure 6.14, an example of the coincidence of landmarks between HOG-EBGM and AAM is displayed.

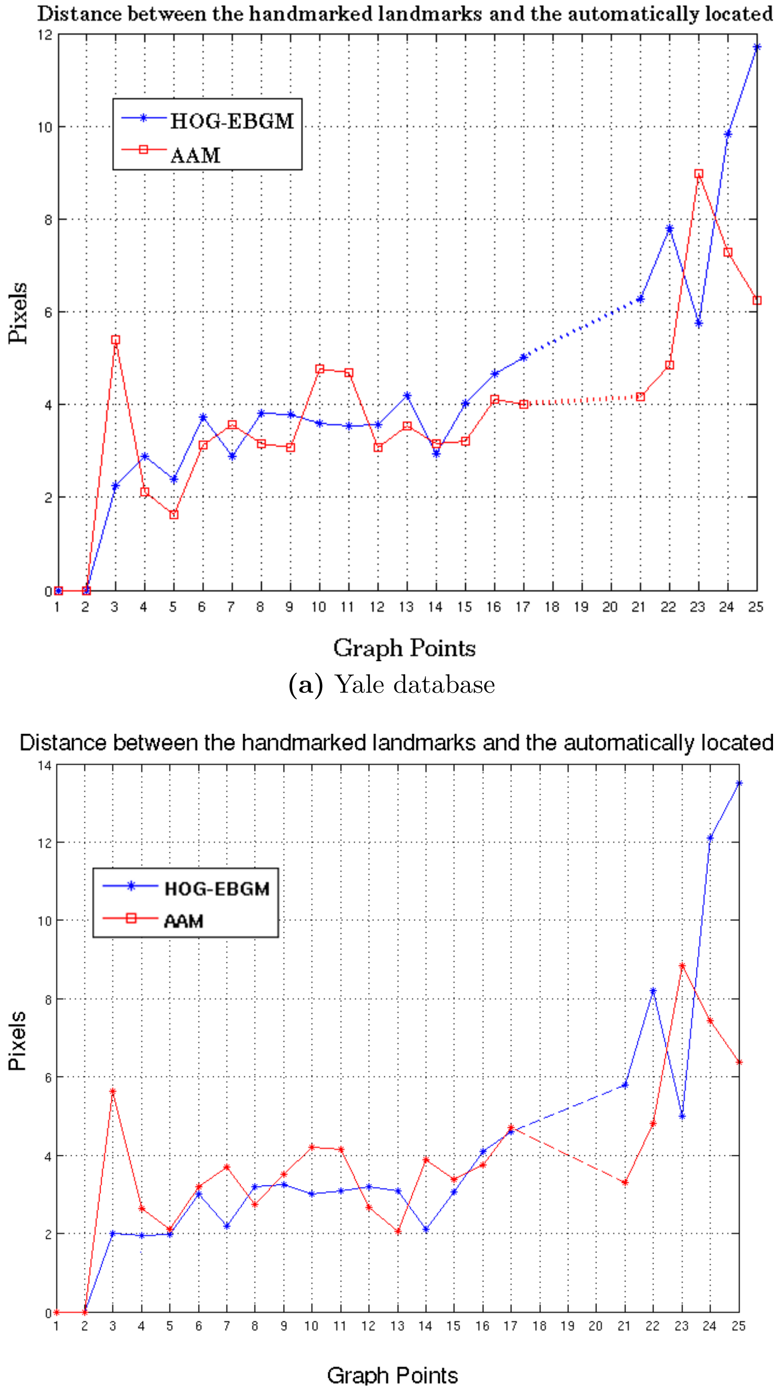
From a preliminary analysis of the curves, it is difficult to conclude which algorithm achieves better accuracy. AAM performs better than HOG-EBGM for 14 out of 22 facial landmarks when using the Yale database, and HOG-EBGM performs better than the standard AAM for 12 out of 22 points when using CVL. This means that, depending on the testing set, either AAM or EBGM locate their graphs with more precision. However, what is more significant is that HOG-EBGM tends to achieve higher accuracy for the landmarks that are associated with the inner and non-deformable keypoints of the face (i.e. from the landmark $i = 1$ until the landmark $= 14$), while AAM achieves better results for the borders of the face or deformable elements such as the mouth (i.e. from the landmark $i = 15$ until the landmark $i = 25$).

Additionally, it can be observed from the curves that the variance on the location is lower with AAM, as the distances for each keypoint when testing the two sets (Yale and CVL) are more stable than the distances obtained with HOG-EBGM. This implies that the latter is more dependent on the image conditions.

6.7.3 Evaluation of HOG-EBGM compared to other holistic face recognition algorithms

To determine the usability of HOG-EBGM as a face recognition algorithm, a set of experiments involving some holistic approaches needed to be carried out.

To achieve this goal, the CSU Face Identification Evaluation System was used in this experiment, as described in [17]. This Evaluation System provides a standard set of well-known face recognition algorithms and established experimental protocols, which are specifically tuned to be run on the FERET



(a) Yale database

(b) CVL database

Figure 6.22: Distance for each of the facial landmarks located by the graph-generating algorithms HOG-EBGM and AAM, with regard to the same landmarks manually marked in the groundtruth of the: **(a)** Yale database and **(b)** CVL database.

face database. This evaluation protocol intends to set a solid basis for drawing conclusions about the relative performance of the HOG-EBGM algorithm.

In the CSU Face Identification Evaluation System, up to three baseline algorithms using holistic approaches are compared, plus the classical, feature-based Gabor-EBGM algorithm:

1. **Standard PCA algorithm**, as described in Appendix A. In this case, during the matching step in the evaluation, two distances were considered: Euclidean distance and Mahalanobis Cosine. For further details, see Chapter 3.

2. **Fisherface algorithm** (which combines PCA and LDA) based upon the University of Maryland algorithm in the FERET tests. This combination is likewise described in more detail in Appendix A.

3. **Bayesian Intrapersonal/Extrapersonal Image Difference Classifier** based upon the MIT algorithm in the FERET tests. In this case, two variants of the classifier have been considered: a Bayesian classifier with Maximum A Posteriori (Bayesian MAP), and a Bayesian classifier with Maximum Likelihood (Bayesian ML).

4. **Elastic Bunch Graph Matching Algorithm** that uses localized landmark features represented by Gabor jets, on 25 locations (as described in Section 6.4), based upon the USC algorithm in the FERET tests.

In Figure 6.23 (extracted from [17]), an scheme of the CSU Evaluation System can be seen. In the experiments of this section, only the Rank Curve Performing was performed.

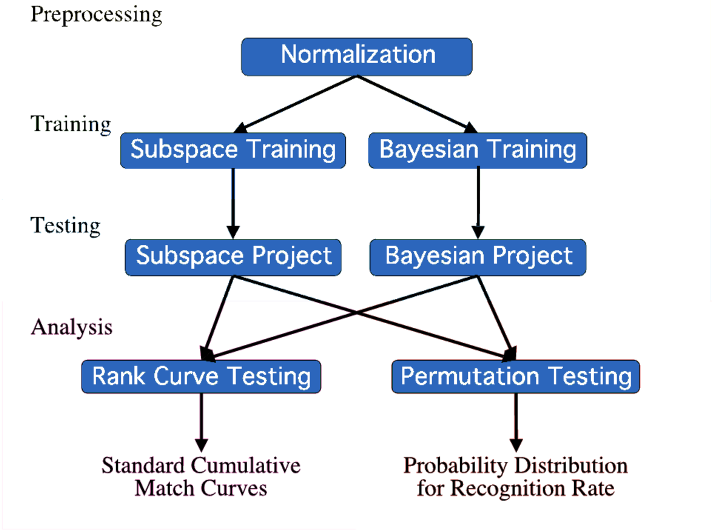


Figure 6.23: Overview of the CSU process for evaluating algorithms.

All curves were evaluated using the FERET *fa* subset as the gallery, and the *fb*, *fc*, *dup1* and *dup2* as probe sets. In Figure 6.24, the performance of the algorithms analyzed using the rank curves can be observed.

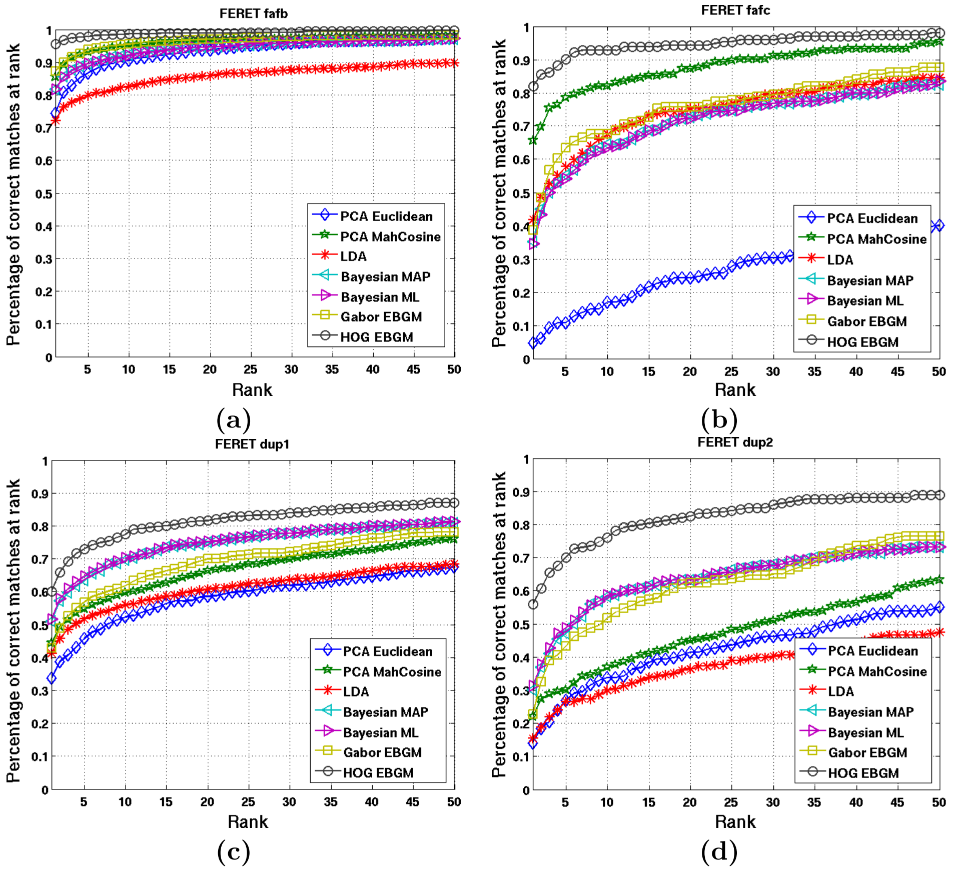


Figure 6.24: Performance evaluation of face identification for the CSU algorithms and the HOG-EBGM algorithm using the rank curves on the feret subsets: **a** fafb, **(b)** fafc, **(c)** fa-dup1 and **(d)** fa-dup2.

From these curves, it can be concluded that the HOG-EBGM algorithm performs better than the rest of the algorithms for all the tests. HOG-EBGM achieves higher results for rank $r = 1$, considered the recognition rate, than the rest of the algorithms. Sometimes it even achieves identification rates doubling those from the rest of the algorithms.

One may think that the performance differences obtained in these experiments for HOG-EBGM could be motivated by comparing holistic and feature-based approaches. However, in this experiment, a second feature-based approach, Gabor-EBGM, was tested. In the figure, it can be seen that Gabor-EBGM performs worse in some scenarios than other holistic approaches, such as the Bayesian classifiers.

6.7.4 HOG-EBGM compared to other HOG-based FGAs

In the previous experiments, the HOG-EBGM approach has been proven to outperform Gabor-EBGM algorithm. Also, at this point, the HOG-EBGM technique has been validated against other holistic face recognition algorithms, as well as feature-based algorithms using local features different from the HOG descriptors.

The goal of this section is to provide an experimental basis to validate the HOG-EBGM algorithm against other feature-based algorithms based on HOG descriptors.

In these experiments, the performance in recognition of the HOG-EBGM is evaluated against two Face Graph Algorithms: HOG Rigid Square Grids and the HOG-AAM. For a complete analysis on these algorithms, see Section 6.6.

Two different sets of experiments were designed to achieve the goals mentioned above:

- Performance evaluation of HOG-EBGM, Gabor-EBGM, HOG-AAM and HOG-Grid, for *face identification.*
- Performance evaluation of HOG-EBGM, Gabor-EBGM, HOG-AAM and HOG-Grid, for *face verification.*

Next, the results of the experiments are given.

Comparison of HOG-EBGM, HOG-AAM and HOG-Grid for face identification

The biometric information contained in a facial graph strongly depends on the algorithm that generated it. The experiments here evaluated the influence of the feature-based algorithms on the facial feature vector that is generated. The objective was to measure their descriptive power in identification tasks.

A comparative study comprising several variables was set up and the results were given in the form of tables containing the hit rates. The variables involved in these experiments can be summarized as follows:

- **Face Graph Algorithms:** The comparative study is between five algorithms: *HOG-EBGM*, *22 HOG-AAM*, *25 HOG-AAM*, *25 HOG-Grids* and *81 HOG-Grids.*

  The performance of each of the algorithm versions is individually displayed in a different table of results (Table 6.6 to Table 6.9).

- **Dimensionality Reduction Methods:** Due to the high dimensionality of the face graphs generated (see Table 6.5), an additional dimensionality reduction stage was included. The number of dimensions was reduced one order of magnitude, from $O(10^3)$ to $O(10^2)$.

  Five reductive algorithms were used: *PCA*, *LDA*, a *Regularized LDA* (*RLDA*), *Orthogonal LDA* (*OLDA*), *Null-Space PCA* (*PCANULL*) and *Kernel Fisher Analysis* (*KFA*)[10].

  Two remarks need to be made:

  1. The KFA is a non-linear method based on kernel functions. In this case, a polynomial function with degree $n = 2$ was selected.
  2. In Regularized-LDA, the value of the regularization constant α was tested for the range: $\alpha = [0.01, 0.05, 0.1, 0.5, 1, 5, 10]$. In this case, the performance results are those obtained with the best value of the constant α for each case.

[10] All these methods are explained in greater detail in Appendix A.

		Nr. Landmarks	**Feature Vector Dimension**
HOG-EBGM		25	3200
HOG-Grid	Sparse	25	3200
	Dense	81	10368
HOG-AAM	Coincident	22	2816
	Extended	25	3200

Table 6.5: Dimensionality of the feature vector for the different face graph algorithms tested.

In the tables of results, each dimensionality reduction method is represented in a different column.

- **Similarity Measures:** The experiment was evaluated with two distances: *cosine* and *Euclidean*.

  Both were used in all experiments, except for KFA, in which case only the *cosine* distance was performed. In [64], the author shows that the *cosine* works better for KFA.

  The results containing two values (i.e. val_1, val_2) correspond to measures with the *cosine* and *Euclidean* distances, respectively. Cells containing only one value (val_1) correspond to *cosine* distance.

- **Test Datasets:** The experimental sets of images used were FERET and FRGCv2[11]:

  - In FERET *fa* form the *target* set, while the *fb*, *fc*, *dup1*, *dup2* subsets acted each time as a different *query* set.
  - In FRGCv2, the images corresponding to *Experiment 4* were used with the predefined training, target and query sets.

Table 6.6, Table 6.7 and Table 6.8, contain the performance results from the experiment. For a better overview of the results, the best results (corresponding to each of the algorithms on each of the test sets) are highlighted in each table.

Table 6.9 summarizes the performance of the feature-based algorithms, collecting the best results obtained from each case. The table offers a comparison of the best results achieved by each algorithm. Also, the highest performances obtained on each of the test sets have been highlighted.

Due to the complexity of the tables, the results require an analysis of the different variables involved:

1. **Analysis of dimensionality reduction**: Using FRGC, the best results were always obtained with the Orthogonal LDA algorithm across all the tests. Using FERET, depending of the specific dataset used, the best performances were achieved with the LDA, Regularized LDA and Kernel Fisher Analysis reduction algorithms. The main problem with RLDA is

[11] See Appendix B

	HOG-EBGM (25 Landmarks)					
	PCA	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
FERET *fafb*	92.64, 92.05	**97.74**, **97.74**	97.66, 97.66	92.97, 92.72	97.66, 97.66	97.40
FERET *fafc*	69.59, 73.71	89.69, 89.18	89.69, 89.69	68.04, 76.84	**89.69**, 89.69	**100**
FERET *dup1*	57.76, 54.85	**69.80**, 69.40	69.52, 69.52	54.85, 54.71	69.52, 69.52	63.15
FERET *dup2*	52.14, 50.43	**65.81**, 65.38	64.96, 64.96	54.02, 50.43	64.96, 64.96	58.54
FRGC	48.67, 42.72	52.38, 34.25	65.03, 61.29	74.02, **74.34**	32.56, 36.06	69.01

Table 6.6: Face Identification rates with HOG-EBGM using FERET and FRGC.

	Sparse Square Grid 5×5 **(25 Landmarks)**					
	PCA	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
FERET *fafb*	93.40, 91.97	**98.41**, **98.41**	**98.41**, **98.41**	93.12, 92.23	93.40, 91.67	95.48
FERET *fafc*	82.47, 88.66	93.81, 93.30	93.81, 93.81	82.47, 86.08	82.47, 88.66	**95.36**
FERET *dup1*	56.51, 57.20	71.60, 71.47	**71.75**, 71.60	52.49, 55.82	56.51, 57.20	54.57
FERET *dup2*	40.17, 45.30	64.1, **64.53**	63.68, 63.68	31.20, 40.17	40.17, 45.30	32.47
FRGC	62.72, 62.55	74.17, 75.1	76.07, 76.07	**87.92**, **87.92**	62.72, 62.55	73.38

	Dense Square Grid 9×9 **(81 Landmarks)**					
	PCA	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
FERET *fafb*	94.73, 94.56	**99.16**, **99.16**	99.08, 99.08	95.15, 94.48	94.73, 94.56	97.23
FERET *fafc*	85.57, 89.69	96.39, 96.39	96.39, 96.39	85.58, 88.66	85.57, 89.69	**98.45**
FERET *dup1*	61.63, 62.74	80.61, 80.47	80.61, **80.75**	57.06, 59.69	61.63, 62.74	59.00
FERET *dup2*	45.30, 51.28	74.36, 74.36	**74.79**, **74.79**	35.04, 44.02	45.30, 51.28	38.88
FRGC	65.23, 66.15	75.76, 75.76	77.11, 71.11	60.46, 65.23	65.15, 72.87	**78.53**

Table 6.7: Face Identification rates with HOG-Grid with the *Sparse* and *Dense* configurations using FERET and FRGC.

	Coincident HOG-AAM (22 Landmarks)					
	PCA	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
FERET *fafb*	86.99, 85.64	**96.47**, **96.47**	96.39, 96.39	87.32, 86.31	86.99, 85.64	91.35
FERET *fafc*	76.30, 78.35	**94.33**, **94.33**	**94.33**, **94.33**	69.59, 74.23	76.30, 78.35	89.27
FERET *dup1*	50.97, 48.89	**70.08**, 68.70	69.85, 68.84	48.06, 48.62	50.97, 48.89	48.08
FERET *dup2*	49.57, 47.43	73.93, 73.08	**74.36**, 73.50	44.01, 44.87	49.57, 47.43	40.77
FRGC	38.17, 37.94	68.10, 67.73	72.55, 72.55	**81.53**, **81.53**	38.17, 37.94	62.33

	Extended HOG-AAM (25 Landmarks)					
	PCA	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
FERET *fafb*	96.57, 96.40	**96.65**, 95.05	96.14, 96.14	87.99, 86.57	88.12, 86.40	94.12
FERET *fafc*	75.75, 78.87	93.30, 93.30	92.27, 92.78	64.95, 72.68	75.75, 78.87	**100**
FERET *dup1*	50.83, 50.42	**72.99**, 72.02	72.71, 72.43	48.34, 49.44	50.83, 50.42	52.49
FERET *dup2*	49.15, 49.57	**76.92**, 76.50	76.07, 76.07	43.59, 45.30	49.15, 49.57	56.84
FRGC	37.30, 37.41	69.10, 69.10	75.25, 75.12	**85.61**, **85.61**	37.30, 37.41	63.99

Table 6.8: Face Identification rates with HOG-AAM with the *Coincident* and *Extended* configurations using FERET and FRGC.

	HOG-EBGM	**GRID** 5 × 5	**GRID** 9 × 9	**HOG-AAM**	**HOG-AAM**
	25 Landmarks	**25 Landmarks**	**81 Landmarks**	**22 Landmarks**	**25 Landmarks**
FERET *fafb*	97.74	98.41	**99.16**	96.47	96.65
FERET *fafc*	**100**	95.36	98.45	94.33	**100**
FERET *dup1*	69.80	71.60	**80.75**	70.08	72.99
FERET *dup2*	65.81	64.53	74.79	74.36	**76.92**
FRGC	74.34	**87.92**	78.53	81.53	85.61

Table 6.9: Summary of the best performances obtained from each of the face graph algorithms for identification tasks.

the uncertainty in determining the optimal value for the regularization constant, α. This prevents this algorithm from being used in a regular scenario.

2. **Analysis of similarity measures**: in the majority of tests, the cosine distance outperformed the Euclidean distance. Henceforth, the *cosine distance* was selected to be the best distance for face recognition.

3. **Analysis of the algorithms**: Due to the complexity of the data in the tables 6.6 to 6.8, this analysis is done using the summarized information in Table 6.9.

   - **HOG-Grid Variants**: in FERET, the performance was improved when the number of points in the grid was higher. The *Dense* (9×9) grid systematically outperforms the *Sparse* (5×5) grid.

     In the FRGCv2 dataset the *Sparse* grid outperforms the *Dense* grid. As the images in FRGC are more challenging, the worse performance of the 9×9 HOG-Grid could be caused by the *overfitting* problem.

     Although the HOG-Grid with higher number of points achieves higher rates in controlled scenarios (FERET), its instability due to the random location of the keypoints decreases its robustness for scenarios with uncontrolled conditions (FRGC).

   - **HOG-AAM Variants**: the *25 HOG-AAM* variant outperforms the *22 HOG-AAM* in all tests.

     The increase in the number of keypoints in AAM is followed by an increase in the performance. Unlike the grids, here the extra keypoints added to the face graph correspond to specific (not random) face landmarks, and therefore their information contribution is significant.

   - **HOG-EBGM versus HOG-Grids and HOG-AAM**: From the tables it can be observed that the three algorithms have a very close behavior.

     However, two issues should be remarked: on one side, it can be observed that the tendency of the graphs generated by HOG-AAM is to slightly outperform those of the other algorithms. On the other side, assuming frontal faces in which the angle variance is low, using grids is the simplest and fastest method to extract features.

Comparison of HOG-EBGM, HOG-AAM and HOG-Grid for face verification

This experiment addressed the comparison of the three HOG feature-based algorithms, analyzing their behavior in verification tasks (evaluated using ROC curves[12]).

The information obtained for the experiments of identification in the previous section were helpful to delimit the variables involved:

[12] See Section 3.4.4

- **Face Graph Algorithms:** The study compares the three FGAs used in the identification experiments: HOG-EBGM, HOG-AAM and HOG Rigid Square Grids.

  In relation to HOG-AAM and HOG-Grid, it was decided to use only the variants with the same number of landmarks as the HOG-EBGM, to be fully comparable: *25 HOG-AAM* and *25 HOG-Grid* (Sparse Grid).

- **Dimensionality Reduction Methods:** Similar to the identification experiments, some dimensionality reduction techniques had to be applied to reduce the face feature vector (see Table 6.5).

  Based on the results of the previous experiments, three reductive algorithms were tested for verification: *PCA* (as a baseline), the standard *LDA* and the *Orthogonal LDA* (*OLDA*).

  Each of the reduction methods were shown as different set of curves in the final results.

- **Test Datasets:** The experimental sets of images of FRGCv2 Experiment 4 were used. This database was selected because it defines a standard protocol focused on the performance analysis for face verification, using the extended ROC curves.

  The evaluation was performed using three subsets for the ROC curves: *ROC I*, *ROC II* and *ROC III.* For further explanations, refer to Section B.6. In the results, each variant of the ROC curves was displayed with a different curve-line.

- **Similarity Measure:** Only one distance measurement was performed in verification, the *cosine distance.* This choice was attributed to the results obtained in the previous experiment, in which the *cosine* distance outperformed the *Euclidean* distance in the majority of cases for the PCA, LDA and OLDA algorithms.

Figure 6.25, Figure 6.26 and Figure 6.27 display the ROC curves corresponding to all tests performed in this experiment.

The following conclusions can be drawn:

1. The best performance is achieved with the *Orthogonal-LDA* for all the approaches and image subsets.

   The performance of OLDA is nearly double that of PCA if a false alarm probability of $p_{fa} = 10^{-3}$ is considered. When OLDA is compared to the standard LDA algorithm, the difference is less remarkable.

   These results match those obtained in identification experiments using FRGCv2. With regard to reduction methods, it can therefore be concluded that LDA techniques outperform PCA in face recognition, with OLDA as the best case.

2. Regarding the dimensionality reduction technique used, when PCA was used, the curves had greater dispersion than with any of the LDA algorithms. That means that PCA is less stable than LDA in verification tasks.

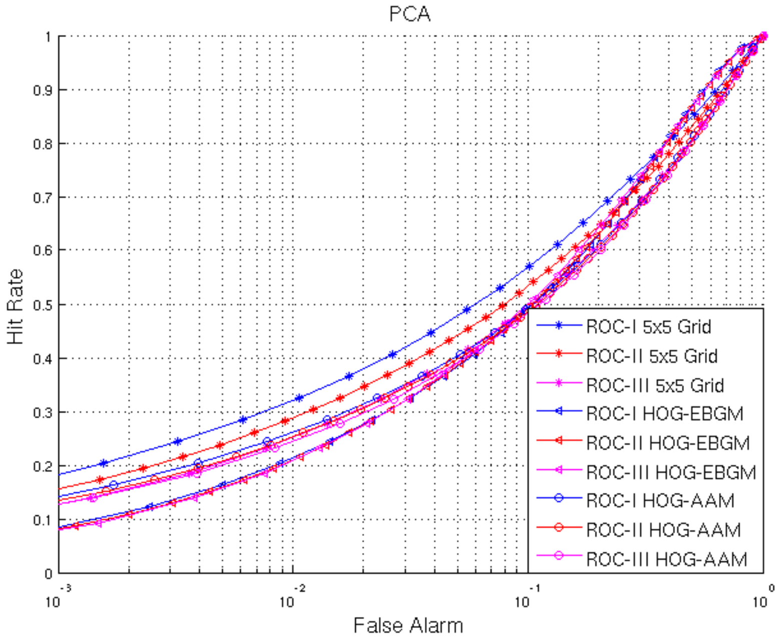


Figure 6.25: ROC curves for verification performance using the PCA reduction method in FRGCv2.

3. For all the cases, the HOG-Grid achieves the best results, especially when combined with the OLDA technique. For a probability of false alarm of $p_{fa} = 10^{-3}$, the hit rate is around 60%, while in the case of HOG-EBGM and HOG-AAM, the performance is around 55%.

4. Focusing on HOG-EBGM and HOG-AAM, we observe that the performance is very similar. In PCA and LDA, HOG-AAM outperforms HOG-EBGM, whereas in OLDA (the best case for all the algorithms), HOG-EBGM outperforms HOG-AAM.

   This similarity is due to the fact that the only difference between HOG-EBGM and 25 HOG-AAM is in the location of the keypoints, and in both cases it is quite accurate.

5. If the experiments with PCA are considered, for all the methods, the *ROC I* subset performs better than *ROC II*, and this in turn outperforms the results for *ROC III*.

   On the other hand, for all the tests performed with the two LDA-based methods, the results obtained with *ROC III* outperform those obtained with *ROC II* and *ROC I*.

Summarizing, it can be stated that in verification, *OLDA* is the optimum dimensionality reduction algorithm. This is especially true when it is applied to graphs obtained from the Sparse HOG-Grid, followed by the HOG-EBGM and the AAM-EBGM, which all perform similarly.

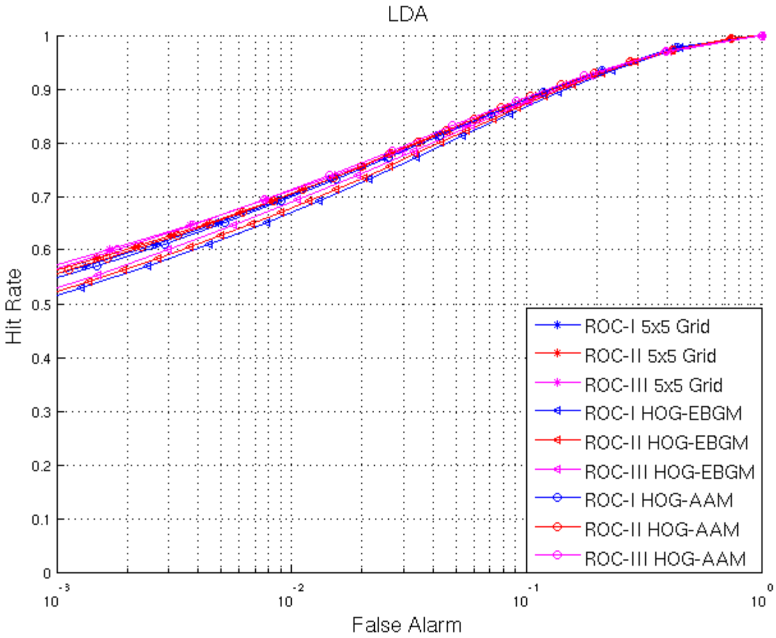


Figure 6.26: ROC curves for verification performance using the LDA reduction method in FRGCv2.

6.8 Effect of automatic eye location on face recognition

As introduced in Chapter 5, to determine the robustness of a face recognition system regarding its initial conditions, it is necessary to test it with regard to initial location of the eyes.

Here, an evaluation experiment is designed to analyse the effects of the variance of the location of the eyes on face recognition. To that end, we have selected HOG-EBGM (without using dimensionality reduction on the face feature vector). The results are also compared to a baseline *PCA* algorithm based on eigenfaces [107]. The selection of the PCA algorithm has been motivated by its use as a baseline and also because of its sensibility to small displacements on the location of the face.

In this experiment, three scenarios have been created for the test images:

1. **Hand-marked Location**: the location of the eyes is provided beforehand, as part of the groundtruth of the datasets.

2. **Automatic Location**: the location of the eyes is provided by the eye location stage proposed in Chapter 5.

3. **Randomly Displaced Location**: the location of the eyes comes from the groundtruth, but a component of white random Gaussian Noise, $N[\mu, sigma]$ generated with zero mean, $\mu = 0$ and maximum standard deviation 5% of the inter-ocular distance is added.

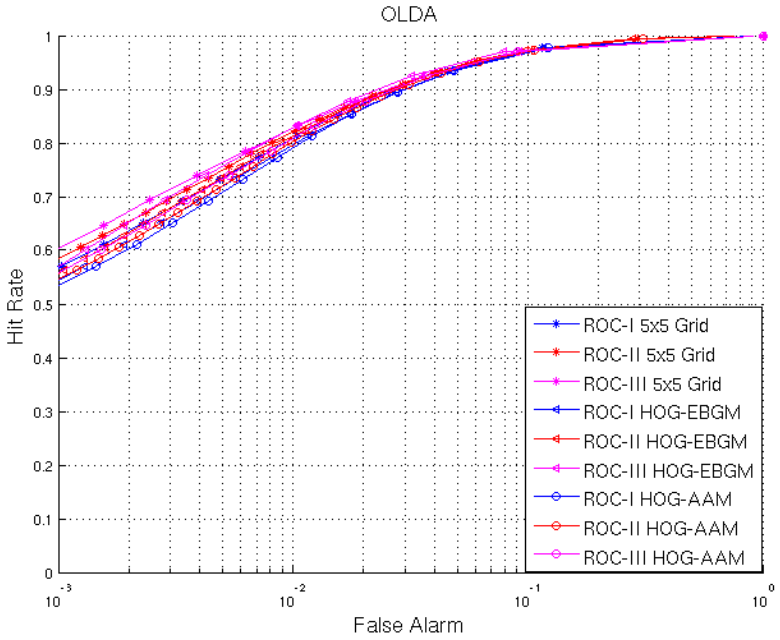


Figure 6.27: ROC curves for verification performance using the Orthogonal-LDA reduction method.

For all the experiments a subset of FERET images has been used. This subset corresponds to 200 individuals with three different images per person. The methodology used in this experiment was a *3-fold Cross-Validation*, where two random images for each individual generated the *gallery* subset, and the resting images the *probe* set. In this experiment it was important to select a controlled set of face images, so that the inaccuracies due tu external factors (such as bad resolution or changing light conditions) would not have a significant influence on the final results.

The results of the study of the performance of both algorithms using the three scenarios are displayed in Figure 6.28.

From the results, it can be concluded HOG-EBGM is quite robust to the variation in the initial location of the eyes: in none of the three scenarios remarkable differences are appreciated, unlike in the case of the PCA algorithm.

Also let's also notice that our eye location system performs for both recognition algorithms in a mid-point between the hand-marked results and the hand-marked results with additional noise. As the addition of noise was designed with a standard deviation of 5% of the inter-ocular distance this means that, the error of our location was less than that level.

These results create a favourable background to mix our eye location stage with the HOG-EBGM face recognition approach.

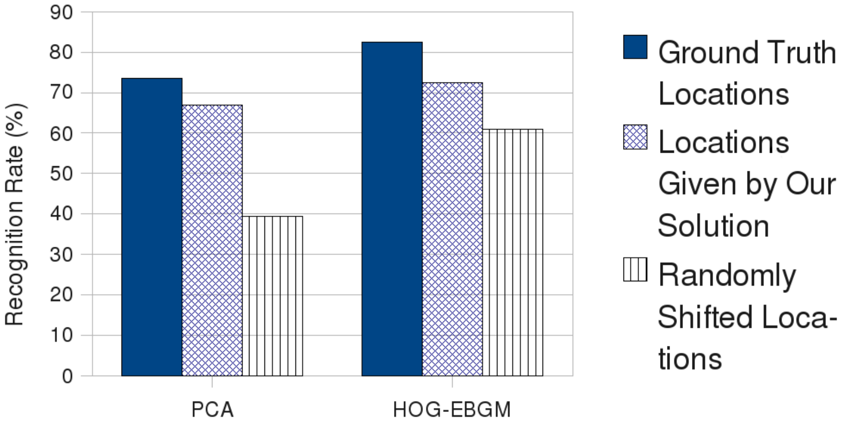


Figure 6.28: Analysis of the influence of the location of the eyes on the performance of some Face Recognition algorithms.

6.9 Evaluation of the Color HOG-EBGM algorithm

In previous experiments, the grayscale HOG-EBGM has been evaluated for face recognition problems. This evaluation could be significantly enhanced by performing the same tests with the color information, using the CHOG-EBGM algorithm described in Section 6.5.

The main target of this section is to experiment how the inclusion of color cues affects to the discriminability of the face feature vector that is extracted. To this end, both HOG-EBGM algorithms, the color-based version and the standard grayscale are compared.

Some additional issues were considered, such as the selection of the optimal color space (in terms of discrimination power) to be used with CHOG-EBGM. A second goal of this section is to analyze the recognition results as a function of the color space: the standard *RGB*, *HSV*, the *Opponent Color Space* (*OCS*) and the *Discriminant Color Space* (*DCS*). All these color spaces are explained in detail in Section 2.3.

These color spaces can be directly derived from the RGB, except for DCS, which needs to be trained off-line. In DCS, an optimal discriminant space is calculated for a specific set of training images. In our case, the FRGCv2 database was used in all experiments, specifically the images corresponding to *Experiment 4* [13]. The transformation matrix from RGB to the optimum color space for FRGCv2 is given in Liu [120]:

$$X_{FRGC} = \begin{pmatrix} 0.28 & 0.06 & 0.74 \\ 0.06 & -0.16 & 0.40 \\ 0.29 & -0.21 & -0.12 \end{pmatrix}$$

The rest of this section is focused on the description of a series of experiments

[13] See Appendix B

using the two versions of the HOG-EBGM face graph algorithm to validate the hypothesis:

- Comparison of CHOG-EBGM and grayscale HOG-EBGM face graph algorithms for face identification.
- Comparison of the two face graph algorithms for face verification.

The evaluation was done using FRGCv2, as it is one of the most extensive color databases. Due to the large quantity of images in FRGCv2, other authors tend to perform their experiments using reduced subsets. In our case, the original sets were approximately halved through random selection of the images. This allowed the computational burden to be reduced and eased the demands on the allocation of memory. Next, the experiments performed are performed, along with an analysis of the results.

6.9.1 Analysis of CHOG-EBGM for face identification

This experiment evaluates the influence of color information on descriptive power when a Face Graph is generated using a HOG-EBGM algorithm.

Following the steps proposed for the grayscale HOG-EBGM in the previous experiments, a comparative study was presented. The results are given in the form of tables that contain the hit rates and also the rank curves of some specific cases to obtain more detailed information.

Next, a summary is offered of the organization of the results and the parameters involved:

- **Algorithms and Color Spaces:** the comparison is between the CHOG-EBGM and the regular grayscale HOG-EBGM. For the CHOG-EBGM algorithm, four variants were analyzed which corresponded to four different color spaces: *CHOG-EBGM on RGB*, *CHOG-EBGM on HSV*, *CHOG-EBGM on OCS* (Opponent Color Space) and *CHOG-EBGM on DCS* (Discriminant Color Space).

  The performance of each of the algorithm variant is displayed as a row in the table of results, and as a different curve-line in the rank curves.

- **Dimensionality Reduction Methods:** The feature vector generated with the CHOG-EBGM had dimension $d_c = 3600$, three times larger than the dimension of the grayscale variant. Therefore, a reduction method was needed to decrease the number of dimensions.

  The reduction techniques tested were: *PCA*, *LDA*, the *Regularized LDA* (*RLDA*), *Orthogonal LDA* (*OLDA*), *Null-Space PCA* (*PCANULL*) and the *Kernel Fisher Analysis* (*KFA*), explained in detail in AppendixA.

  Two additional remarks are needed:

  1. For the KFA, a polynomial kernel function was used, with the degree set to $n = 2$.
  2. In the LDA-Iterative Power Method, tuning was required for the value of the regularization constant, α. The values were empirically set: $\alpha = [0.01, 0.05, 0.1, 0.5, 1, 5, 10]$.

     The result shown in the result table for the RLDA are those obtained with the best value of the constant α for each case.

In the table of results, each of the dimensionality reduction methods are represented in different columns, while in the rank curves, only two reduction methods are shown as different sets of curves: the Orthogonal LDA and the KFA. The reason for showing just these two sets of curves is explained below.

- **Similarity Measures:** Two distances were employed in these tests, *cosine* and *Euclidean*, except for KFA, as in [64] Li shows that the optimal similarity measure is the *cosine* distance.

  In the table of results, all the cells that contain two performance values (i.e. val_1, val_2) correspond to *cosine* and *Euclidean* distances, respectively. The cells with only one value correspond to *cosine*.

  Regarding the rank curves (Figure 6.29), *cosine* distance was used for the Orthogonal LDA. For greater clarity, the curve-lines obtained with the *Euclidean* distance are not shown in the results, as for all the methods, *cosine* outperforms *Euclidean*.

Table 6.10 contains the results obtained for the CHOG-EBGM and HOG-EBGM algorithms in face identification. The best results for each variant are highlighted.

Some conclusions can be drawn:

1. For all cases, CHOG-EBGM outperformed the grayscale HOG-EBGM.
2. Regarding the measurement distances, the results obtained using *cosine* outperformed the ones obtained with *Euclidean*. This result is in accordance with the results obtained with the grayscale HOG-EBGM.
3. Regarding the Color Space, the highest rates were obtained for all the cases with the Opponent Color Space.
4. Regarding the dimension reduction method, the best results were obtained with the Orthogonal LDA and the KFA for all the color spaces.

Since the performances of the OLDA and KFA reduction techniques are quite similar. For a deeper comparison of the methods, we also plotted the rank curves corresponding to all the CHOG-EBGM variants (along with the grayscale HOG-EBGM) were also extracted using both the OLDA and KFA techniques. The results are shown in Figure 6.29.

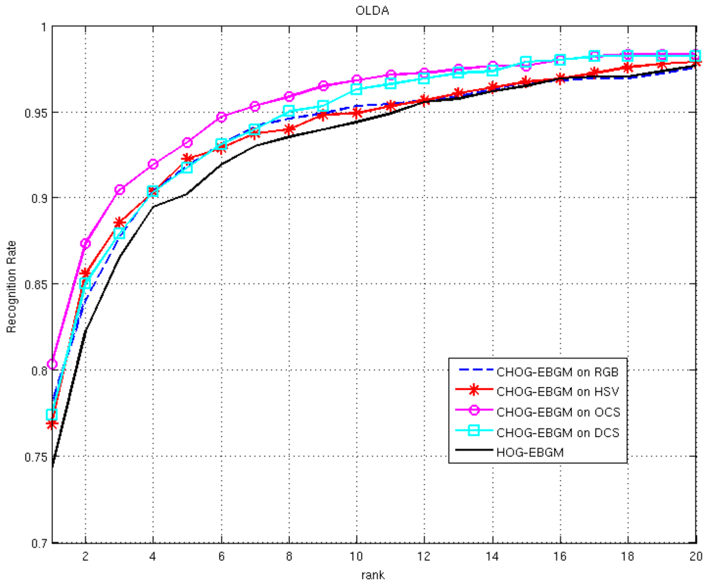
From the rank curves, it can be seen that the behavior using both reduction techniques is still quite similar. The results obtained with KFA for all the variants are better and more consistent than the results with OLDA. On the other hand, in the case of Orthogonal-LDA, the CHOG-EBGM algorithm performs slightly better on the Opponent Color Space. For example, an accuracy of $hit - rate = 95\%$ was achieved for rank $r = 6$ for the CHOG-EBGM on OCS with the OLDA technique. The same was achieved for rank $r = 8$ when using the KFA technique.

6.9.2 Study of CHOG-EBGM for face verification

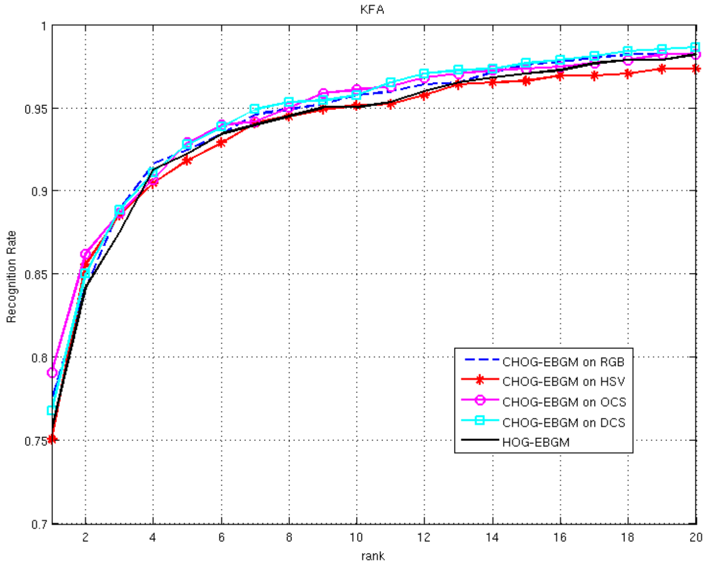
The last experiment designed to evaluate the CHOG-EBGM algorithm was aimed at analyzing its behavior in face verification. Following the experimental

	DIMENSIONALITY REDUCTION METHOD					
COLOR SPACE	**PCA**	**LDA**	**RLDA**	**OLDA**	**PCANULL**	**KFA**
HSV	50.37, 41.78	53.13, 36.48	64.58, 62.02	76.35, **78.37**	25.98, 20.07	75.08
OCS	51.01, 41.57	58.32, 41.57	71.16, 65.32	79.11, 79.32	37.96, 33.30	**80.59**
DCS	51.01, 43.48	58.43, 39.45	68.61, 64.58	**76.78**, 76.14	41.04, 42.42	76.78
Grayscale	48.67, 42.72	52.38, 34.25	62.46, 61.29	74.02, **74.34**	32.56, 36.06	69.01

Table 6.10: Face Identification rates with CHOG-EBGM for different Color Spaces using the FRGC database.



(a)



(b)

Figure 6.29: Rank curves corresponding to CHOG-EBGM performed on different Color Spaces in FRGCv2, using two dimension reduction methods: **a)** Orthogonal-LDA and **b)** Kernel Fisher Analysis.

set-up provided in Section 6.7.4, the performance of the two algorithms was evaluated using the ROC curves.

To optimize the experimental set-up, some of the conclusions from the previous CHOG-EBGM identification experiments were used. This information helped delimit the number of variables involved in the current set-up.

In this experiment, the FRGCv2 dataset was also used. Due to the amount of information in the ROC curves, only the ones corresponding to the *ROC I* set proposed in the CSU methodology are displayed, as the results for the *ROC II* and *ROC III* do not make a significant contribution.

The variables involved with the verification experiments are:

- **Algorithms and Color Spaces:** The comparison is between the CHOG-EBGM and the regular grayscale HOG-EBGM. For the CHOG-EBGM algorithm, four variants were analyzed, each corresponding to four different color spaces: *CHOG-EBGM on RGB*, *CHOG-EBGM on HSV*, *CHOG-EBGM on OCS* (Opponent Color Space) and *CHOG-EBGM on DCS* (Discriminant Color Space).

  The performance of each of the algorithms is displayed as a different curve-line in the figures displaying the ROC curves.

- **Dimensionality Reduction Methods:** From the experiments for identification, it can be concluded that the two most outstanding methods for reducing the dimension of the CHOG-EBGM feature vector are the Orthogonal LDA and the KFA techniques. In this experiment, the algorithm variants with both techniques were evaluated, and the PCA and LDA were added as baselines.

  Each of the tests run with the different reduction techniques are shown as different sets of curves in the final results.

- **Similarity Measures:** the final scores were evaluated with the *cosine* distance, as a consequence of the results in the experiments for identification.

In Figure 6.30, Figure 6.31, Figure 6.32 and Figure 6.33, one can observe the ROC curves corresponding to all the tests performed for this experiment.

Next, the results obtained from the ROC curves were analyzed:

1. **Analysis of dimensionality reduction algorithms:** For all the color spaces tested, the best performance is achieved with the Orthogonal-LDA and the Kernel Fisher Analysis.

   Considering false alarm probability $p_{fa} = 10^{-3}$, the results obtained for OLDA and KFA are quite similar. However, if the number of false alarms is increased, OLDA performs slightly better. The performance with the PCA technique is quite low compared to the other cases.

2. **Analysis of the performance of color spaces:** When the performance of the CHOG-EBGM on the different color spaces is analyzed, two conclusions can be drawn:

   - The curves obtained for the CHOG-EBGM indicate a higher performance than the curves for the grayscale HOG-EBGM in almost

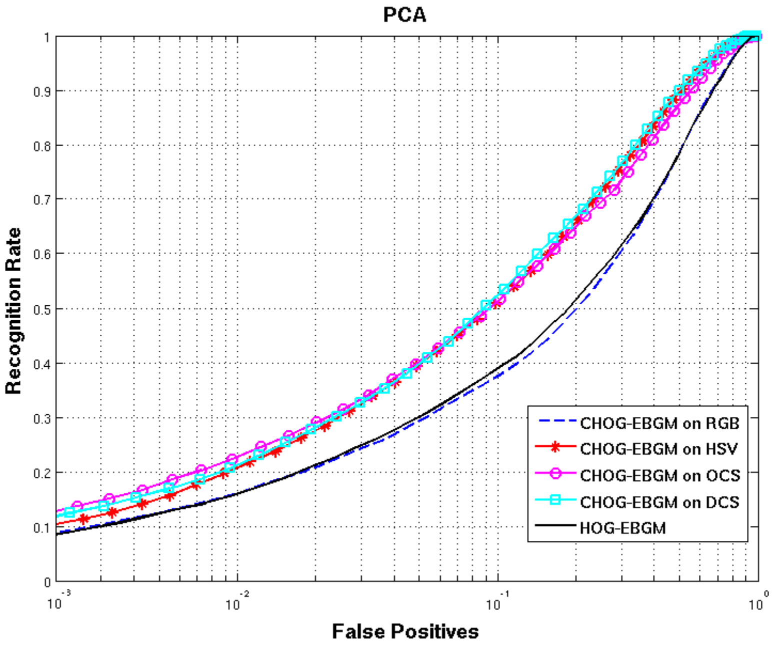


Figure 6.30: Verification performance of CHOG-EBGM with different Color Spaces using PCA on FRGCv2.

all the cases. However, the response of the CHOG-EBGM depends on the color space used, and in some cases, the HOG-EBGM can outperform the CHOG-EBGM for certain color spaces. This can be seen in Figure 6.30 and Figure 6.33, corresponding to PCA and KFA, respectively.

- For all the color spaces and all cases studied, the one with the best performance was the Opponent Color Space (OCS). The best curve was obtained using this color space and the KFA technique for dimension reduction, as seen in Figure 6.33.

6.10 Conclusions

This section is aimed to summarize the main conclusions extracted from the use of different Face Graph algorithms, specifically the one developed in this thesis HOG-EBGM, and its color adaptation CHOG-EBGM.

The section is structured in blocks corresponding to the conclusions extracted from each set of experimental analysis.

6.10.1 HOG-EBGM compared to Gabor-EBGM.

From the results comparing the two EBGM-based methods, it can be generally said that the use of HOG-EBGM to extract facial graphs is more convenient

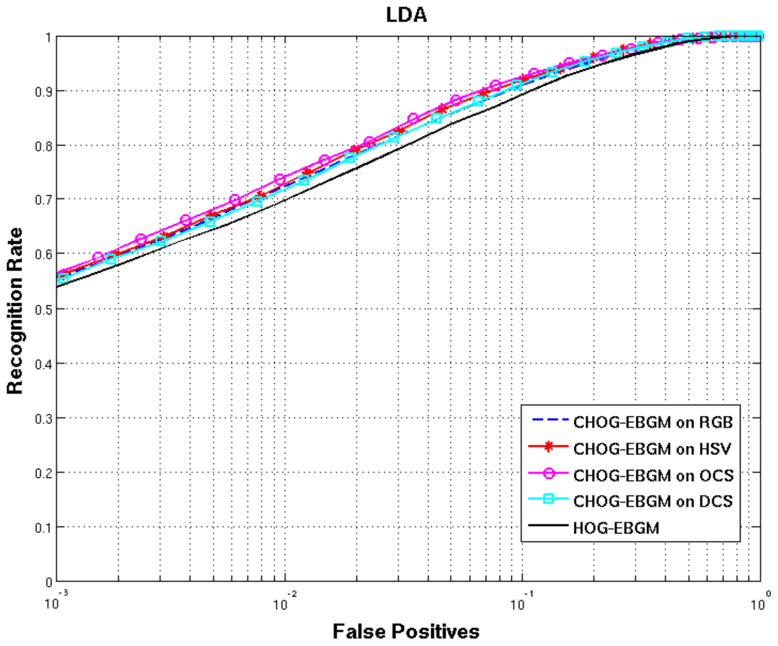


Figure 6.31: Verification performance of CHOG-EBGM with different Color Spaces using LDA on FRGCv2.

than the standard Gabor-EBGM. Analyzing this statement more in detail, it can be seen that:

1. The descriptive power achieved with HOG-EBGM to represent a given set of facial landmarks is greater than that obtained with Gabor-EBGM. This is a good indicator that it is preferable to use HOG descriptors over Gabor jets, even for configurations other than EBGM algorithms.

2. The automatic process of facial landmarks localization is more stable and accurate when the iterative EBGM algorithm uses information from the HOG descriptors previously stored in the models set.

3. The use of HOG-EBGM in identification tasks outperforms a number of the most common and widespread holistic algorithms.

All these conclusions support the use of the HOG-EBGM algorithm as a Face Graph Algorithm.

6.10.2 HOG-EBGM compared to HOG-AAM and HOG-Grid

One of the goals of this chapter was to evaluate different feature-based face recognition approaches, based in HOG descriptors. The experiments provided assess the use of the HOG-EBGM, comparing it with other HOG-based face graphs algorithms: HOG-AAM and HOG-Grid. Specifically, this validation

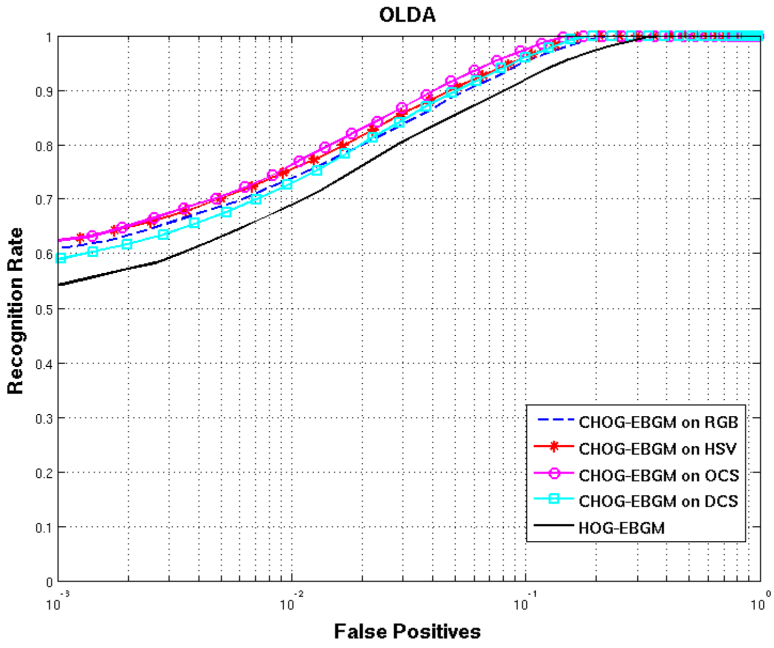


Figure 6.32: Verification performance of CHOG-EBGM with different Color Spaces using Orthogonal-LDA on FRGCv2.

was aimed at evaluating the accuracy of the three algorithms on the location of the keypoints of the face graph and their performance in recognition tasks.

From the results of the three sets of experiments performed, some valid conclusions can be drawn:

- Regarding the location of the face graph, the HOG-EBGM algorithm is compared to the standard AAM. Two main conclusions can be extracted from the experiments:

  1. The accuracy of the keypoints achieved by both algorithms is comparable. The HOG-EBGM achieves higher accuracy for the inner landmarks of the face and also those less flexible parts (e.g., the corners of the eye or the nose), while AAM achieves better results for the borders of the face or deformable elements (e.g., the mouth).
  2. The landmarks extracted with AAM are more stable (i.e. have lower dispersion), than those obtained with HOG-EBGM, meaning that HOG-EBGM is more dependent on the image conditions.

- Regarding the performance in recognition of HOG-EBGM, HOG-AAM and HOG-Grids using dimensionality reduction, the conclusions reached are:

  1. The HOG-Grid tends to outperform HOG-EBGM and HOG-AAM; however, this result needs to be contextualized. The location of the points in the a non-deformable grid is static, not obeying the characteristics of the image. Therefore, for normalized and frontal images,

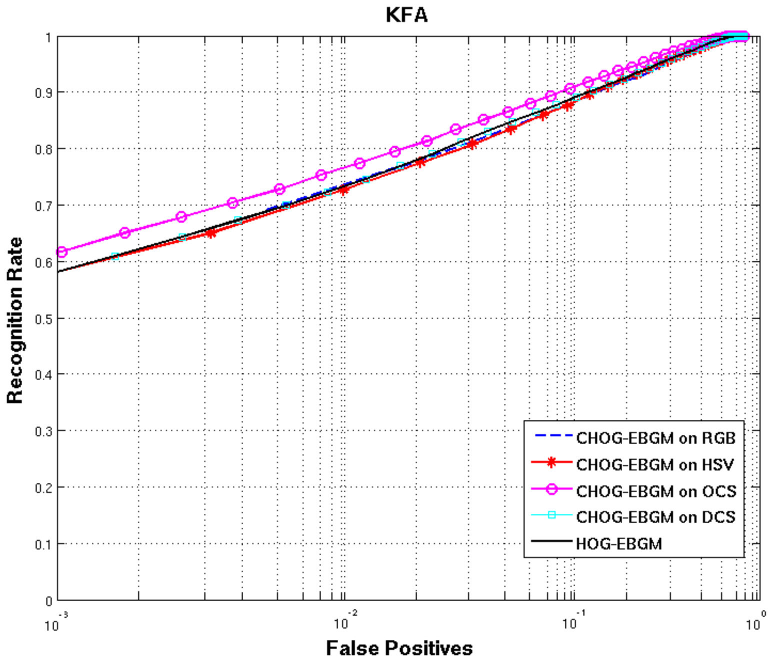


Figure 6.33: Verification performance of CHOG-EBGM with different Color Spaces using KFA on FRGCv2.

the HOG-Grid tends to naturally place the keypoints in the same relative position on the face. Meanwhile, for HOG-EBGM and HOG-AAM, badly conditioned images can lead to non-accurate locations. Nevertheless, the adaptability of the HOG-Grid should be tested when the faces are not completely frontal.

2. HOG-EBGM and HOG-AAM have similar performances in recognition and verification, being both suitable for such tasks, as the only difference is in the method to locate the keypoints.
3. In both experiments, a final step of dimension reduction was needed. For all the algorithms tested, the two achieving the most relevant performances were the Orthogonal-LDA and the Fisher Kernel Analysis. However, depending on the set of images, results may change, making it difficult to determine the priority of either one of these methods.
4. With regard to the matching algorithm, different distances were tested, leading to the conclusion that the cosine distance generally outperforms the Euclidean distance in recognition tasks.

6.10.3 CHOG-EBGM compared to the standard HOG-EBGM

In this section, a testing context was provided to validate the use of the color-based HOG-EBGM, comparing it to the grayscale HOG-EBGM face graph algorithms. The experiments were aimed at measuring the performance of the two algorithms in recognition with the use of different color spaces.

Next, some of the conclusions are summarized after performing the two sets of experiments:

- Regarding the descriptive power of the color features, CHOG, compared to the grayscale features, HOG, the first showed better results for all the experiments performed, indicating a higher descriptive power in those graphs that incorporate color cues.

  In terms of performance, the CHOG-EBGM is preferable to the standard HOG-EBGM, although its cost is higher in terms of the memory used and the number of operations.

- Regarding the five color spaces analyzed , some have proved to fit better with our problem than others. The classical RGB color space showed one of the poorest performances, while others, such as the Opponent Color Space, were more discriminative.

- Regarding the reduction of dimensions, two techniques proved to be more appropriate in dealing with the color feature vectors: the Orthogonal-LDA and the Fisher Kernel Analysis.

As a final conclusion, it can be asserted that the CHOG-EBGM achieves its best performance when it is used on the Opponent Color Space, with the OLDA or the KFA techniques for dimension reduction.

Chapter 7

Results of the MOBIO contest for a HOG feature-based solution

7.1 Introduction to the MOBIO contest

In the field of biometrics, great efforts have been made to develop non-intrusive techniques such as face or speaker recognition. These two topics have been studied for a long time: since the mid 1960s, in the case of face recognition [16, 31], while for the topic of automatic speaker recognition, research started in the 1970s [8, 72]. The research on these topics has often been done in parallel, without any feedback between them. Nowadays, images are usually associated with sound streams, and thus it is reasonable to think that both kinds of biometric analyses could be coordinated to achieve better results. The problem is that the historical isolation of these topics has produced few joint databases of face images and speaker voices.

During the last two decades, some academic and public institutions, such as the National Institute of Standards and Technology (NIST) [1], have organized a series of contests for face and speaker recognition. The aim of these contests was to spur development of novel algorithmic approaches and challenge the technology.

In the case of face recognition, some of the most remarkable and widespread competitions in recent years have been the 2004 ICPR Face Verification Competition [77] and the Face Recognition Grand Challenge [94], both organized by NIST.

Regarding the challenge offered by these databases, it must be remarked that some sets of images in the Face Recognition Grand Challenge[1], contain many blurry images with uncontrolled light and background conditions. Despite the challenging level of the images, there was still a general lack of samples showing *realistic* scenarios. The majority of these contests offered samples in semi-controlled conditions, with images acquired using known poses and angles for the face and camera. Such contexts move the experiments away from the

[1] See Appendix B for further description

conditions of real scenarios.

To overcome some of the limitations of previous competitions, a group of universities designed a new database in 2009 in the form of a competition: the Mobile Biometry Face and Speaker Verification Evaluation (MOBIO). The aim of MOBIO was to provide a unique opportunity to jointly analyze two mature biometrics, face and voice, in a realistic environment. The database was designed as a series of videos of people recorded from a common mobile phone. This mobile environment offered challenging conditions for the acquired images, including adverse illumination, noisy background and noisy audio data.

Note that the MOBIO database offered a joint dataset of video and audio, whereas the evaluation was focused on examining uni-modal face and speaker verification techniques. The participants in this contest were expected to confront the evaluation from only one of two perspectives: face recognition or speaker recognition.

The universities involved in the recording of the samples of the database were: the Brno University of Technology (BUT), the University of Manchester (UMAN), the Idiap Research Institute (IDIAP), the University of Avignon (LIA), the University of Surrey (UNIS) and the University of Oulu (UOULU).

Finally, the database was designed to be recorded in two phases: *Phase I* and *Phase II*. *Phase I* was used for the competition in the 2010 International Conference on Pattern Recognition (ICPR). This phase was open to the public and it is the only one object of study in the current chapter. *Phase II* was intended to be developed independently, and thus its study is beyond the aim of this work.

7.2 Motivation to participate in the MOBIO contest

The MOBIO contest targeted participants willing to match their algorithms for facial analysis and voice recognition with the solutions given by the other participants. However, the major challenge that this competition offered was going beyond the state-of-the-art of recognition methods to overcome a number of problems derived from the mobile context in which the samples were acquired.

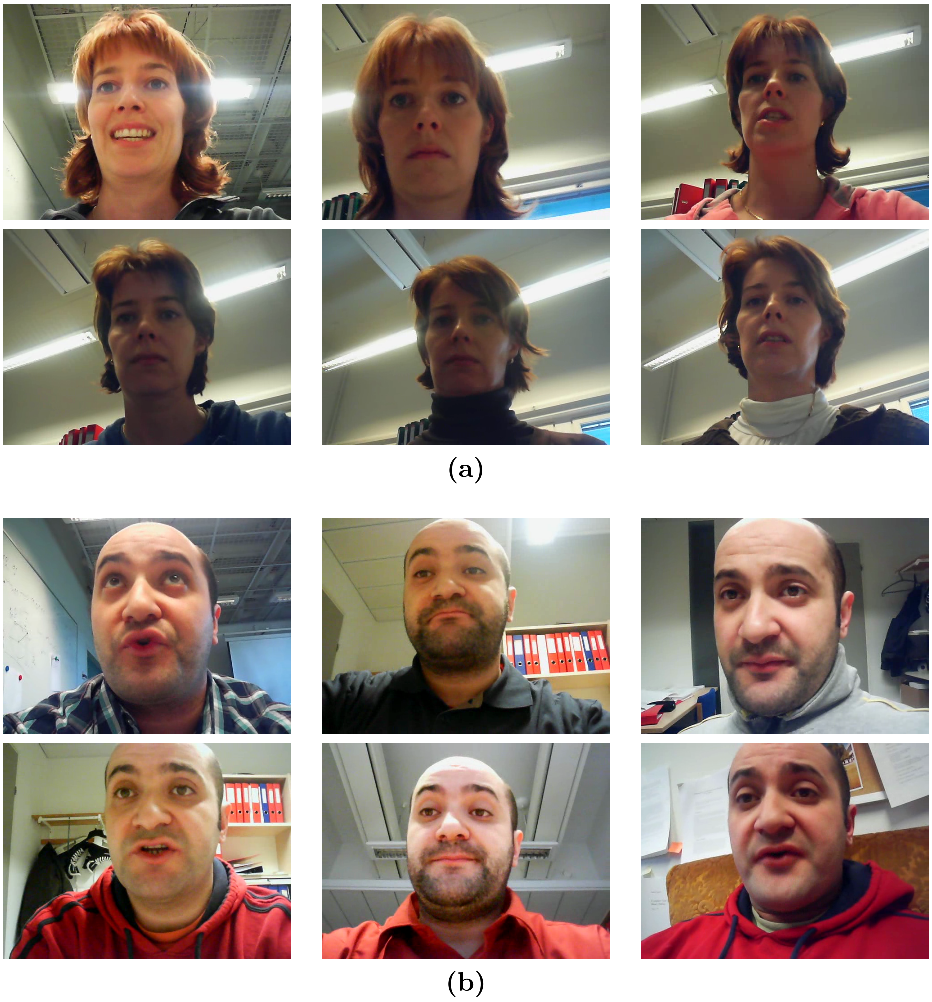
The MOBIO competition grants a propitious framework to develop novel solutions, evaluate them and assess them against other state-of-the-art methods. While working on this thesis, we considered it productive to take part in the competition. Our main interests were in the area of face recognition, as it offered an opportunity to complement and put together the work presented during the previous chapters.

The main reasons that motivated our participation in the MOBIO contest are summarized in the following:

- The evaluation of the MOBIO database is a perfect complement to the study carried out through this work. Because of the mobile environment provided, we gained the opportunity to experiment on realistic samples.

  The MOBIO database contains a number of challenging features, such as the uncontrolled pose of the face, the gesticulation that results from people's speech or severe changes in illumination, among others. These features are difficult to find in other face databases, as shown in Appendix B.

Figure 7.1 displays some examples of challenging images extracted from the MOBIO database.



(a)

(b)

Figure 7.1: Examples of two challenging identities in different sample videos from the MOBIO database.

- The intentional lack of groundtruth in the images about the location of faces or any specific facial landmarks requires the use of a face detection stage prior to extracting biometric information for face recognition. In this thesis, it became necessary to unify the eye location system presented in Chapter 5 with some of the face graph algorithms studied in Section 6 for face recognition.

  The study of the joint performance of both stages was a good complement to the experiments in previous chapters as it gave an overview of the system as a whole.

- The MOBIO database offered the chance of designing experimental setups when video samples are used. The majority of databases used for face recognition are based on still images.

Performing on videos enables a preselection of the faces with the best appearance from all the video frames, unlike other databases where only a few images per person are given. On the other hand, applying the traditional still-image algorithms to videos involves finding solutions to some additional problems: a mechanism is needed to select the best face in each frame. Each detection produces a score which allows the most suitable faces in the video to be selected. However, in the MOBIO database, each different identity is associated with one video, as the videos were recorded with one person alone (i.e., one face per frame).

- The MOBIO competition provided a framework for evaluating our joint system against other state-of-the-art approaches from other competitors.
- Thanks to this competition, we met our goal of disseminating new technologies, transmitting our approaches to other researchers and comparing results in a common scenario.

This chapter starts by analyzing the MOBIO database and the performance evaluation proposed by the organizers. Then, our contribution to the competition is detailed, with the results analyzed and compared to the most significant results from the rest of competitors. Finally, some conclusions about the competition are drawn.

7.3 The MOBIO database

The composition of the MOBIO database determines the evaluation and the experimental tests that can be performed on it. That is, the MOBIO competition provides not only the necessary samples, but also the evaluation protocol for determining the training, the development and the test sets, as well as the performance evaluation that has to be used for a fair comparison between all the participants.

As stated before, our main interest in the MOBIO contest was to validate our face analysis algorithms by using the video sequences and leaving out the audio samples. Therefore, the rest of this section is mainly focused on the description of the samples and the protocol established to deal with the video sequences for face recognition.

7.3.1 Composition of the MOBIO database

The capture of video and audio samples in the MOBIO database addressed several issues neglected by other public databases. Some of its main features are summarized in the following:

1. The database consisted of recordings which acquired consistent data over a period of time. This is important when studying the problem of model adaptation.
2. The videos were recorded directly from mobile devices and showed people talking, which leads to a constant change in facial expressions. Also, the recordings were acquired under different lighting conditions and with a

variety of poses. This covered the necessity of having video sequences captured in realistic scenarios.

3. The audio was captured on a mobile platform, producing a considerable variance in noise in each sample, meeting the need of realistic audio samples for the speaker recognition algorithms.

4. One of the goals of the database was to contain a large variety of scenarios and people from diverse origins. *Phase I* was recorded collaboratively in locations from the MOBIO consortium.

5. The samples of the database were acquired primarily using a regular mobile phone, with a camera resolution of 640 × 480 pixels and a rate of 25 *frames per second*. All the sequences had an approximate duration of between 10 seconds and 20 seconds, which amounts to 250 to 500 images per video.

6. The sound samples were extracted directly from the audio track associated with the video samples. Although the speech was limited to English, the MOBIO database included recordings from both native and non-native English speakers.

The *Phase I* of the the database contained 160 participants who completed six recording sessions. In each session, the participants were asked to answer a set of questions, classified as: i) set responses, ii) read speech from a paper, and iii) free speech. In all, each session consisted of 21 questions: 5 set response questions, 1 read speech question and 15 free speech questions. More details are given below:

- Set responses were given to the user. There were five such questions in total and fake responses were supplied. Five different questions were asked, and each question took approximately five seconds to answer (although this varies among users).

- Read speech was obtained by supplying each user with three written sentences. The sentences were the same for each session.

- Free speech was obtained by prompting each user with a random question. For five of these questions, the user was asked to speak for five seconds; and for ten questions, the user was asked to speak for ten seconds, giving a total of fifteen questions. The users were asked again to give false information and make up their answers.

7.3.2 The MOBIO evaluation protocol

The database was split into three different sets, following the format of supervised algorithms: one for training, one for development and one for testing. For each subset, the data was split in such a way that it came at least from two different institutions. This way, a total independence between sets was achieved, as the three sets for the evaluation were completely independent, with no information regarding individuals or the conditions shared between any of them.

The data contained in the training set was available for use by all participants in the competition. The data was intended to derive background models: for instance, training a world background model or a vectorial space to apply reductive techniques, among others. Table 7.1 lists the main features and the uses intended for the training set in the database.

TRAINING SPLIT

Usage	Background Training
Data to Use	All Data (non-mandatory)
Total Number of identities	53
Number of female identities	14
Number of male identities	39
Number of Recording Sessions	6
Number of Videos per Session	21
Total Number of Videos per Person	126

Table 7.1: Training split of the MOBIO database.

The development and the test splits were almost identical in their structure. They were divided into several sessions: one for the enrollment (gallery set), and the rest for the evaluation (probe set). Despite their similarities, each of the splits was designed with a very different purpose:

- The development split had to be used as a self-assessment set. The results obtained in this stage were not relevant to the public, but were very useful to adjust the proposed system to the actual scenario.

  In this set, the identity of the probe samples were known and the idea was that the participants were able to derive the parameters and thresholds to tune their systems. Also, it was allowed in this competition to determine any fusion parameters if the participants chose to do so.

  The video samples of the development split were recorded in six sessions: one for enrollment and five for testing. Table 7.2 summarizes the main features of this set.

- The test split was used to derive the final matching scores. In this set, the identity of the probe samples was not known by the participants. Following the evaluation protocol of the MOBIO competition, no parameters could be derived from this set to assist in the matching, only the final feature vectors.

  The organizers provided the test split to the participants in a second stage of the competition. The main rule was that no *a priori* knowledge about the probe set was allowed. To ensure this statement, the data was encoded so that the filename gave no clue to the identity of the user.

  The video samples of the test split were recorded in six sessions: one for enrollment and five for testing. For further details, see Table 7.3.

DEVELOPMENT SPLIT

Usage	Self-assessment
Data to Use	Set questions only
Total Number of identities	47
Number of female identities	20
Number of male identities	27
Number of Recording Sessions	6
Number of Videos in Enrollment Session	5
Number of Videos in Test Session	15
Total Number of Videos per Person	5 + 75

Table 7.2: Development Split of the MOBIO database.

TESTING SPLIT

Usage	Final Scores
Data to Use	Free Speech Only
Total Number of identities	47
Number of female identities	20
Number of male identities	27
Number of Recording Sessions	6
Number of Videos in Enrollment Session	5
Number of Videos in Test Session	15
Total Number of Videos per Person	5 + 75

Table 7.3: Testing Split of the MOBIO database.

In the development and testing splits, the first recording session was used for enrollment, while the remaining sessions were used as probe. This can be summarized as follows:

- Enrollment (Session 1): For each individual, the data consisted of five video recordings. All the individual parameters were derived from this set.

- Evaluation (Sessions 2-6): Data came from the free speech. Each video was treated as an individual probe observation, producing a *client* match and $C-1$ *impostor* scores, where C is the number of classes (individuals) in the enrollment set. In all, 15 videos were recorded from each session, producing a total of 75 videos per person.

Also, for a more extensive evaluation, *male* and *female* labeling was provided with the samples, so that the testing could be performed independently for each group. *A posteriori* mixed evaluation was also given (only the results are mixed, not the test sets).

7.4 HOG-EBGM with eye location

The system we proposed for the competition integrated a face detection with eye location stage with a face recognition stage based on HOG-EBGM (studied in Chapter 6). For face detection, we tried two different solutions: one based on the techniques described in Chapter 5, and one based on a commercial off-the-shelf algorithm.

The system we designed for the contest also included an additional stage between detection and recognition: *Best Faces Selection*. It was aimed at selecting the best faces of an individual, exploiting the advantages of using videos instead of still images. This stage was designed to solve two problems:

1. Reduce the information from the videos to specific face images in which to perform the identity matching. Given the high number of images extracted from each of the video samples, it was important to reduce it to a small set of face images, and therefore optimal for face recognition.

2. Tackle the problem of multiple faces detected in a single frame. In this case, only the faces with the best appearance (with higher reliability) were selected.

The structure of the MOBIO competition forced us to add an independent phase for the development and test splits:the *video feature extraction*. It was an off-line phase performed prior to the tests. Its main goal was to extract the main information from each video of the target and query sets. To that end, this phase made use of the *Best Faces Selection* stage.

The rest of this section is aimed at analyzing the particularities of our facial analysis system, going into detail during the face detection and normalization stage, the extraction of features and the enrollment phase.

7.4.1 Face Detection, Cropping and Normalization

The face detection stage has a great impact on the whole facial analysis system. For example, the reliability of the results given by the face recognition stage are highly dependent on the accuracy of the location of the detected face, as confirmed by the experiments in Section 6.8.

For the MOBIO competition, we proposed two different solutions, which only differed in the *Coarse Face Detection* step (as shown in Figure 5.1): the first solution used the face detection algorithm studied in Chapter 5, while the second used a commercial off-the-shelf solution.

The principal motivation for presenting two different solutions was that for some specific videos, we were not able to detect any faces with the OpenCV detector. As explained below, this was due to the inability of our implementation to deal with the problem of faces that exceed the borders of the image. This problem had a significant impact on the global recognition rate, since the number of enrolled people in MOBIO was rather small[2].

[2] See Table 7.2 for further details

Next, the two solutions proposed to detect faces are described:

- **Solution 1 - Boosting based**: This solution follows all the steps described in Chapter 5, in which the OpenCV AdaBoost classifier is used to detect frontal faces. This solution has already been proven to be effective for face detection (as can be seen in Table 5.2, with a high rate of hits and a low rate of false alarms).

  Unfortunately, this approach presents some major drawbacks regarding some of the samples given by the MOBIO database. Basically, in the samples where the face was not completely contained in the image, the face could be detected by any means. As explained before, this led to a great loss of information. Some examples of difficult images are shown in Figure 7.2.

- **Solution 2 -VeriLook SDK + Kalman Filter**: This solution adopted a commercial product for the detection of faces, the VeriLook SDK developed by Neurotechnologija [73]. The selection of this system was motivated by the results obtained for the eye location analysis performed in Chapter 5.

  Before setting up this solution, we also performed some experiments on some of the MOBIO sets and the accuracy of this software was similar to that obtained with the Boosting stage. The key difference was that the faces were detected in almost all the images.

  Additionally, we considered that taking advantage on the video properties would also improve the results. In this solution, we introduced a Kalman Filter [54] to track the eyes and reduce the eye detection noise. Tracking the position of the eyes throughout a sequence and making a prediction of it for each frame using a Kalman Filter offers two main contributions:

  1. For static or near-static faces, comparing the detected and the predicted coordinates can smooth the jittering in the location of the eyes (due to bad detections, noise, changes in illumination or eyes gestures, such as blinking).
  2. The faces that are in movement are usually displayed as blurred images, making a proper facial analysis difficult. These images can be easily discarded using the prediction error of the Kalman Filter.

  The contribution of the Kalman filtering step to improving the recognition results proved to be even more important than the change in the coarse face detection method.

For both solutions, the detected faces were normalized using the eye coordinates, following the stages proposed in Chapter 2. The location of the eyes was based on the extraction of HOG-features and preselected eye candidates, as shown in Chapter 5.

A diagram of the two solutions for face detection can be seen in Figure 7.3.

7.4.2 Face Graph Extraction

Regarding the extraction of face graphs for face recognition, we decided to use the HOG-EBGM algorithm, as described in Chapter 6, in our two solutions

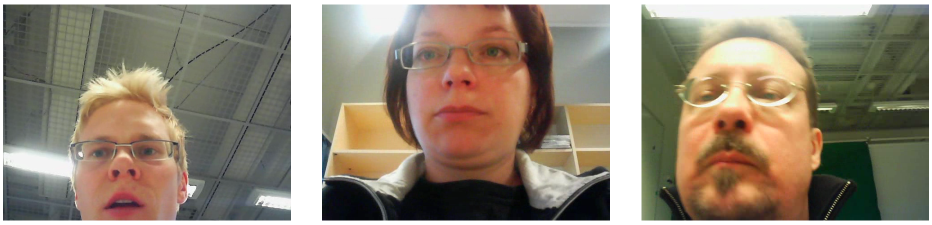
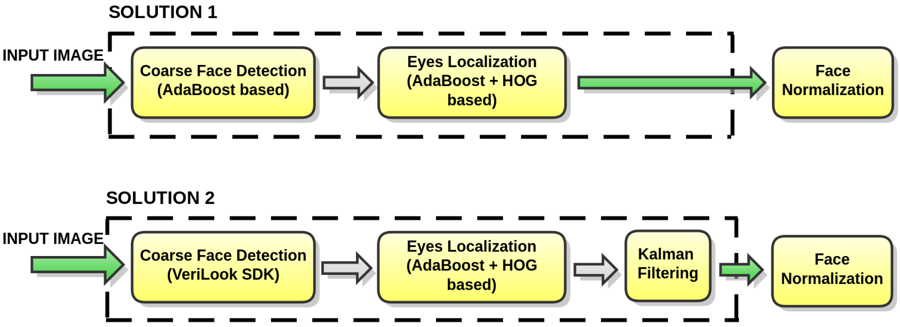


Figure 7.2: Examples of three challenging videos from the MOBIO database, with faces not completely contained in the images.



SOLUTION 1

INPUT IMAGE

Coarse Face Detection (AdaBoost based)

Eyes Localization (AdaBoost + HOG based)

Face Normalization

SOLUTION 2

INPUT IMAGE

Coarse Face Detection (VeriLook SDK)

Eyes Localization (AdaBoost + HOG based)

Kalman Filtering

Face Normalization

Figure 7.3: Diagram of the two solutions presented in this work for the MOBIO competition.

for the contest. In short, each face is represented by a feature vector resulting from the concatenation of the $N = 25$ facial landmarks. Each landmark is described by a HOG feature, so that the total dimensions of the feature vector are $d_{face} = 25 \times 128 = 3200$.

Since the dimensionality of this feature vector is too high, we applied some dimensionality reduction techniques. Following the results obtained in the experiments presented in Section 6.7, we decided to use the Kernel Fisher Analysis (KFA), which uses non-linear projection.

The system was trained using face images from the FERET database (600 images corresponding to 200 individuals)[3] and ten face images of each person in the MOBIO training split. Two additional experiments were performed, one using only the FERET database and one using only the MOBIO training set; the best results were always achieved when these two sets were combined together. This can be explained by the FERET images having included a higher number of different people. On the other hand, using images from the MOBIO training set is important, as they can better model the intra-person variability: more images per person are available, and also the context from the MOBIO training set is logically closer to the testing set than to the FERET database.

The KFA projecting subspace was learned using a polynomial kernel function of degree $n = 2$. The final number of features per face after dimensionality reduction was $d_{final} = 140$.

[3] See Appendix A

7.4.3 Video Feature Extraction

The protocol of the MOBIO competition included an enrollment phase that has to be performed prior to the testing itself. The *Video Feature Extraction* phase is an off-line stage aimed at condensing the information from the videos of each person so that an appropriate set of gallery images is generated. A block diagram of the steps needed in this phase is given in Figure 7.4.

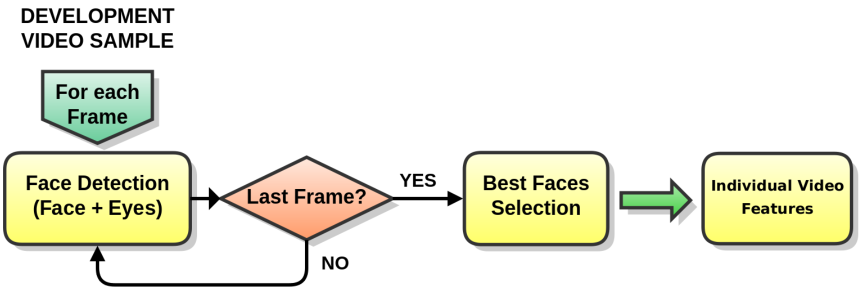


Figure 7.4: Diagram of the Video Feature Extraction phase.

The *Best Faces Selection* step in the figure is the key block of the Video Feature Extraction phase. Based on a certain criterion, it reduces the information extracted from the videos into N face images. Particularly, to extract the information of a new person, we only selected the N faces with the highest confidence from the corresponding videos, and stored these images (or, equivalently, the set of feature vectors from each of those faces) as a model for the person.

It is important to select an appropriate value for N, as a low number of face images would imply less information of the model enrolled. On the other hand, high values of N would greatly increase the complexity of the system. Therefore, during the development stage, we ran some experiments with different numbers of faces in each person model; we found that a number of $N = 10$ was a good trade-off between complexity and accuracy. In fact, no significant recognition improvements were achieved for values greater than N; this indicated that a good model of the person was already extracted with only ten face images.

Regarding the criterion for selecting the best faces from the videos, we used two different confidence values, one for each of the two solutions provided:

- **Solution 1 - Number of hits:** Almost every face detection system produces a number of hits around each real face, which are usually clustered into one detection. This number of hits is also referred to as the number of neighbors.

  We decided to use the number of hits produced after applying the OpenCV Adaboost face detector as a confidence measurement of the quality of the face. However, we found that with this confidence measurement, we were missing important information about the precision of the location of the eyes. In turn, it is very important to perform a good normalization of the face.

- **Solution 2 - Kalman Filter:** This solution is based on the commercial software VeriLook SDK for the detection of faces and our HOG-based eye location stage. We also included a simple Kalman Filter to track the location of the eyes in the video, giving us useful information.

  We decided to use the error of the Kalman filter predictions as a measurement of the face confidence. The smaller the distance between the actual and the predicted location of the eyes, the higher the confidence. This measurement allowed us to select faces with low head motion, which usually have better definition and lower noise in the location of the eyes.

The faces selected with the second criterion proved to be more representative (i.e., more useful for the extraction of the video features) than the faces obtained with the criterion for the number of hits.

Finally, the feature vector associated with each face image was extracted and stored with a label.

7.5 Discussion on the results in the MOBIO competition

The MOBIO competition not only provided a database of faces recorded in a mobile environment, but also defined the performance evaluation protocol the participants had to follow. With this test-bench, face recognition algorithms from diverse approaches could be tested and compared all together. In this section, the final evaluation of our solutions is given, performing a comparison with the results obtained by the rest of participants.

This section is divided into three parts: first, it provides the theory behind the performance evaluation proposed by the MOBIO competition; next, the experiments to obtain the performance results of our two solutions are explained, and finally, the analysis of the best results from all competitors is carried out.

7.5.1 Performance Evaluation Protocol

The performance evaluation protocol proposed within the framework of the MOBIO contest was based on common performance measurements, giving place to a fair comparison between different systems. All the performance methods used in this chapter are described in Chapter 3.

In order to measure the performance of the different verification systems, the MOBIO competition uses the Half Total Error Rate (HTER). The HTER operator was mandatory to evaluate each system in two scenarios, depending on the test videos included in the probe set: only the female or male video samples. A third scenario was derived from the fusion of these two.

The HTER operator might not have been enough, as it represents the results in just one operating point. For the MOBIO competition, the HTER was used for the working point where the error rates FAR and FRR were equal[4]. However, depending on the scenario, one may give more importance to one of both errors. For this reason, it became useful to compare the verification results using other

[4] See Section 3.4.2, p. 33

kinds of representations, such as the Receiver Operating Characteristic (ROC) curve or the Detection-Error Trade-off (DET) curve.

In the MOBIO competition, all comparisons between different algorithms were made by calculating the DET curves in the two scenarios described above: only female video samples and only male video samples.

Summarizing, the MOBIO evaluation performance was set to be given by the HTER operator for all the algorithms, choosing a suboptimal threshold value; for a more detailed comparative analysis between different algorithms, the results were given using the DET curves.

7.5.2 Evaluation of the two HOG-EBGM solutions proposed.

The results given by the two solutions provided for the MOBIO competition, and following the Performance Evaluation Protocol, are summarized in Table 7.4. In this table, the columns represent two scenarios: *only female* videos, *only male* videos. The results for the derived *joint* scenario are also given.

	Male	**Female**	**Joint**
UPV 1	23.74%	23.70%	23.72%
UPV 2	21.86%	23.84%	22.85%

Table 7.4: HTER results obtained by the UPV approach for the Test MOBIO set.

Both face recognition solutions, named *UPV 1* and *UPV 2* for the competition, performed well on the MOBIO data. Note that we did minimal tuning on the configuration of the tests, to improve the performance of the KFA technique. This tuning was done during the creation of the KFA space, where images from the FERET and the MOBIO training datasets were used. This particular tuning gave an improvement of about 2% in the equal error rate during the testing.

The difference in recognition performance between male samples and female samples is also statistically insignificant. This is consistent with the fact that our solutions were never designed to be gender dependent (using hair style features, for instance).

We did not observe any significant difference in the performance results using the development or the test sets. This shows that both datasets had similar levels of difficulty, and it also proves that our system was not tuned to any particular dataset.

Finally, a slight improvement in our second solution (UPV2) can be observed, our hypothesis being that it was produced by a better selection of useful faces using the Kalman tracker described before.

7.5.3 Comparative results of all the solutions in the MOBIO competition for face recognition

A total of nine universities, research institutes and companies participated in the competition: University of Surrey (UNIS), Visidon Ltd. (VISIDON), Instituto Tecnologico de Informatica de UPV (ITI), NICTA, National Taiwan University

(NTU), Idiap Research Institute (IDIAP), University of Nottingham (UON), Tecnologico de Monterrey (TEC) and ourselves, iTEAM Universidad Politecnica de Valencia (UPV).

It must be noted that most participants presented several solutions (two in our case). To simplify, in this section only the results corresponding to the solution that achieved better results for each of the competitors are shown. For a more extensive study of all the solutions and their results, the reader is referred to [71]. In our case, the algorithm shown corresponds to *UPV 2*. Another relevant aspect was the option to perform a normalization stage with the scores. Some of the participants employed score normalization techniques, and others (as was our case) did not. This favored an interesting discussion on the relevance of normalizing the scores.

A summary of the results of the best face verification systems for each participant can be found in Table 7.5. The columns in this table represent the three scenarios described above: *only female* videos, *only male* videos and the derived scenario under the title of *joint*. Also, notice that the columns of the table have been sorted according to the global results from the competition (which corresponds to the results in the last column).

	Male	**Female**	**Joint**
UNIS	9.75%	12.07%	10.91%
VISIDON	10.30%	14.95%	12.62%
ITI	16.92%	17.85%	17.38%
UPV - Our Solution	21.86%	23.84%	22.85%
NICTA	25.43%	20.83%	23.13%
NTU	20.50%	27.26%	23.88%
IDIAP	25.45%	24.39%	24.92%
UON	29.80%	23.89%	26.85%
TEC	31.36%	29.08%	30.22%

Table 7.5: HTER results of the best performing face verification systems by each participant of the MOBIO competition.

The first observation to be made from the table is that our solution was the fourth best performing, which put our algorithm in a good position. Also from the table, the approaches can be divided into three groups based on their performance:

- The first group is composed of the two systems with the best results. The best performance, with an HTER of 10.9%, was obtained by the University of Surrey, which fused multiple cues and post-processed the scores using score normalization. The same system without score normalization obtained an HTER of 12.9%. The second best performance was obtained by Visidon Ltd, with an HTER of 12.6%. It used local filtering with no score normalization. Interestingly, it should be noticed that these systems used a proprietary software for the task of face detection.

- The second group is composed of three systems: the Instituto Tecnologico de Informatica, UPV (ours) and NICTA (with score normalization). In this group, all the solutions achieved HTER values under 25% (except for the case of NICTA in *males*, which was slightly higher). The Instituto Tecnologico de Informatica also used a proprietary software for face detection (the same as the University of Surrey), *UPV Solution 2* used the VeriLook software and NICTA directly employed OpenCV's boosting solution. A particularity of NICTA was that the performance from the female test set was considerably better than from the male test set.

- Finally, the third group is composed of the remaining systems, which obtained HTER values of more than 25%. The majority of these systems used an OpenCV-based face detection scheme and all seem to have similar performances.

An extension of these results can be seen in the DET plots shown in Figure 7.5. The two sets of curves correspond to the same algorithms from Table 7.5, performed on the male and female trials.

The curves confirm the conclusions extracted from the table of results. The only difference of note is with the NICTA algorithm, which apparently performed better than shown in the HTER results.

Summarizing, two conclusions can be drawn from the results:

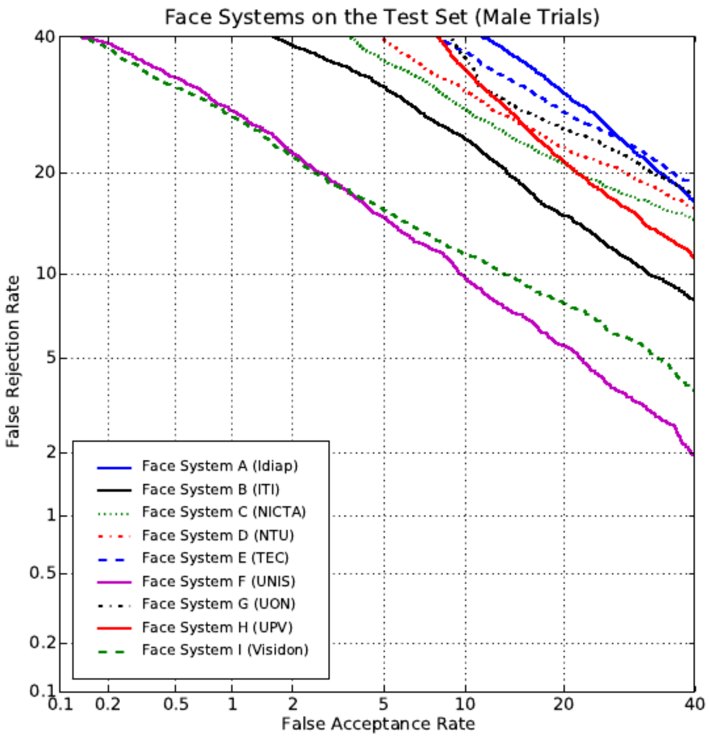
1. **The accuracy of the face detection system can have an important impact on the face verification performance**: The impact of the face detection algorithm can be seen when examining our two solutions. The commercial solution performed slightly better than the one based on OpenCV, by a difference of 1% in the HTER results.

   However, the same behavior was seen in other solutions. For instance, the Instituto Tecnologico de Informatica presented two systems differing only in the face detection technique used. They used the frontal OpenCV face detector in one system, and in the second, Affinity SDK from OmniPerception [74]. The difference in the stage of face detection led to an absolute improvement in the average HTER of more than 4%.

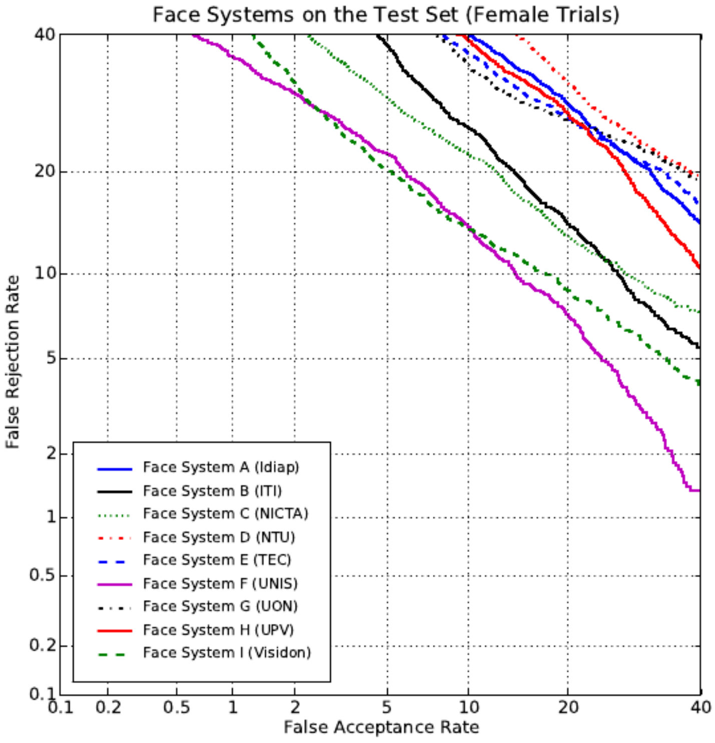
   This reasserts our hypothesis that one of the biggest challenges for video-based face recognition (and face recognition in general) is the problem of accurate face detection.

2. **It is difficult to define the influence of the score normalization on the performance**: A second interesting conclusion is that score normalization can be difficult to apply to face recognition. This can be seen by examining the performances of the systems from the University of Surrey and NICTA. The NICTA results show that score normalization provides a minor but noticeable improvement in performance. However, the University of Surrey systems provide conflicting results; score normalization in some cases degraded performance whereas score normalization in other cases improved performance.

   The only conclusion that can be drawn from this is that further research is necessary to successfully apply score normalization to face verification.



(a)



(b)

Figure 7.5: DET plot of face verification systems on the test set in: **a)** male individuals, **b)** female individuals.

7.6 Conclusions

This work contributed to the MOBIO competition using a completely automatic facial analysis system, using face detection with eye location and face recognition and based on face graph algorithms.

Participating in the MOBIO competition gave us the opportunity to:

- Evaluate our joint systems in a mobile and realistic environment, with an appropriate evaluation protocol to analyze the data.
- Integrate the two stages studied in this work: face detection with eye location based on HOG-features, and a face recognition algorithm using HOG-EBGM.
- Study the advantages and inconveniences of using videos instead of still images in our facial analysis algorithms.
- Compare our solutions to other state-of-the-art algorithms.

The first conclusion is that our solution achieved quite good results, finishing among the top participants. This means that it can be suitable for realistic scenarios with minor adaptations. Note that our solution did not introduce some mechanisms (such as the score normalization), which in some cases helped to improve the results.

This competition also highlighted the necessity of having an accurate face detection stage before face recognition. All the algorithms with the highest performances in the MOBIO competition had commercial solutions for the detection stage. Our first solution (UPV1), was the best in the ranking if we consider only those solutions using a non-proprietary face detection approach.

Finally, participating in the MOBIO contest confirmed that working on videos rather than still images can produce more advantages than inconveniences. In a video sequence, the high number of face images per person allowed us to preselect images. This enabled us to perform the facial analysis only on the images that best fit our needs. To select the best images, some techniques should be studied. In our case, the Kalman filtering was the most suitable.

Chapter 8

Conclusions, dissemination and future lines of research

The primary goal of this work was to develop a fully automatic approach for face detection and recognition. The solution proposed tackles this topic with a two-faceted approach: first, the detection stage locates the eyes with a high degree of precision, and second, the recognition approach extracts local features, from the eye locations information. In order to find a final system, several approaches were tested for each of the modules.

This last chapter summarizes the achievements of this thesis. It discusses the principal results and their limitations. An overview of the scientific contributions of the present work, as well as its dissemination and possible applications, is likewise included. This chapter concludes with various suggestions for future lines of research.

8.1 Conclusions

Discussion of the conclusions derived from the work in this thesis is divided into three sections:

- Eye Location: Conclusions following the evaluation of the novel eye location algorithm based on local features, compared to other state-of-the-art approaches.

- Face Recognition: Conclusions extracted from the study of the HOG-EBGM feature-based algorithm for the extraction of graphs, compared to other approaches. Also in this section, some conclusions are drawn concerning the use of color features versus grayscale features.

- Integrated System: Conclusions extracted from the integration of detection and recognition subsystems for the MOBIO contest.

Subsequently, each of these categories are discussed.

8.1.1 Eye Location

One primary goal in this work was to increase the accuracy of face detection by locating the eyes. A fully automatic eye location algorithm was developed, targeting grayscale frontal faces in semi-controlled scenarios where the two eyes are visible and open or semi-opened. The novelty of this approach was to combine a preliminary stage of fast and robust boosting classifiers to detect face-regions and some eye-candidates, with a second series of steps where HOG local features were extracted from the candidates and a SVM classifier was applied to select the optimal eye-pair.

The main conclusions drawn from the experiments for eye location can be summarized in the following:

- **Boosting Stage**: After testing on different datasets, it can be determined that the boosting classifiers can extract true eye candidates with high confidence. In the tests performed on FRGCv2, with more than 36000 images, the extraction of true candidates was close to 97%. Moreover, the results for the negative candidates showed that, on average, the classifiers produce a reduced number of false alarms per eye, which attests to the reliability of the stage.

  Additionally, the performance results attained by using two different classifiers, each trained to detect the left and right eyes, respectively, are higher than when a single classifier is used.

- **Local Descriptors**: The use of HOG local descriptors in combination with a binary SVM classifier leads to a robust selection of the best eye-pair from a set of candidates. For a Radial Basis Function kernel on the SVM, with the variance parameter set to $\sigma = 3$, the detection rate values achieved were approximately $TP = 96\%$, with a false alarm rate of approximately $FP = 1\%$ during the validation and test of the classifier. These results outperform those achieved using a polynomial kernel approach.

- **Multi-Resolution Approach**: Locating the eyes with our multi-resolution design provides more precision than other state-of-the-art approaches. Compared to three referent works [52, 24, 114], as well as to *VeriLook*, a commercial software product created by Neurotechnologija [73], our algorithm achieved promising results with two extensive datasets, FERET and FRGC. With an inter-ocular distance error of $Nerror < 5\%$, the approach here surpassed the results of the other solutions compared, with a hit rate in FERET of $HR_{FERET} \simeq 80\%$ and $HR_{FRGC} \simeq 98\%$ in FRGC.

  In regard to execution time, our approach takes $200ms$ to locate the eyes in a face[1]. However, there are some approaches, such as that of Jin *et al.* [52] that perform faster. This may be due to the fact that Jin *et al.* directly classify eye-pairs using low-level features (such as borders), while in our approach each of the eyes is independently described using a boosting classifier. The majority of our system execution time, around 70%, is dedicated to the two boosting stages. Despite the simplicity of these stages, their intensive scanning of the face region for each possible location and scale still renders them the most inefficient of all the stages.

[1] This results were achieved using a 1.85GHz dual core CPU, without parallel computing and non-fully optimized

8.1.2 Face Recognition

This work addressed the study and design of a face recognition method to deal with partially uncontrolled scenarios. The primary targets of this stage were frontal faces, and the approach relies on the use of local texture features to extract descriptive information about the facial elements.

The present study focused on three different feature-based algorithms, each employing HOG local descriptors, with our principal contribution being made in the HOG-EBGM approach. Several analyses were performed to optimize both the dimensionality reduction methods and the similarity measures, followed by a study of the performance results when color cues were introduced.

The main conclusions drawn from the experiments are summarized below:

- **HOG-EBGM**: The experiments with EBGM focused on comparing differences in performance in landmark locations and in descriptive power when HOG features substituted Gabor Filters.

  Regarding face graph location, the inclusion of HOG descriptors in the original EBGM leads to a more precise set of keypoints. For most of the landmarks, the location error achieved by using HOG features was between 5% and 15% lower than the error using Gabor Filters. On the other hand, the location accuracy of the HOG-EBGM was higher than that of the AAM algorithm in only for the inner landmarks (like the eyes or the mouth). For the borders of the face, AAM achieved higher accuracy.

  Regarding descriptive power, the graphs generated using HOG features revealed better results than those using Gabor Filters. Using an extensive dataset of images (FERET), the recognition results obtained with HOG-EBGM outperformed those obtained with Gabor-EBGM, increasing the rate of hits between 20% and 40%. Additionally, the performance of HOG-EBGM surpasses other off-the-shelf face recognition holistic approaches, such as PCA, LDA or Bayesian.

- **Feature-based algorithms using HOG descriptors**: In this work, two additional Face Graph Algorithms were analyzed: HOG-Grids (Square Rigid Grids) and HOG-AAM. The two algorithms were also compared to our approach, HOG-EBGM. The main conclusions drawn for each of these algorithms and their resultant comparisons are summarized here. The results are valid for both face identification and verification.

  - *HOG-Grids Conclusions*: The experiments on Rigid Square Grids reveal that grids with a higher number of points, N_p clearly tend to achieve higher recognition results. A denser grid contains higher quantity of information. However, when the overlapping of the HOG windows exceeds 50%, increases in the number of points on the grid register only a negligible increase in the final results. The best grid size is therefore one with an overlap of approximately 50%.
  - *HOG-AAM Conclusions*: The experiments carried out with different AAM configurations focused on employing models with varying numbers of keypoints. The results show that, in the case of AAM, incrementing the number of keypoints does not have a direct influence on the discrimination power of the graphs that are extracted.

This is an indicator that many of the keypoints in the graph model proposed by Milborrow [78] contain redundant information. Therefore, the best AAM configuration tested was that in which the model keypoints coincided with the landmarks extracted with an EBGM approach.

  - *Comparison of the Three FGAs*: When comparing the three HOG-based FGAs, it becomes apparent that the three algorithms present similar results. However, it was surprising to find that in many experiments, the best performing method was the HOG-Grid, the simplest of of the three. This can be explained by the fact that the images from the FRGC dataset were completely frontal, so the rigid location of landmarks was more coincident than the iterative extraction performed by HOG-EBGM and HOG-AAM. HOG-EBGM and HOG-AAM got similar performances, as the only difference between them was on the location of the landmarks, which in both cases is quite accurate.
  - *Similarity Measures*: In the majority of experiments performed, the cosine distance outperformed the Euclidean distance, independent of the reduction method performed.
  - *Dimensionality Reduction Methods*: The reduction methods that best address the problem of face recognition are in essence the Orthogonal LDA –a variation of LDA– and the Kernel Fisher Analysis using a polynomial kernel.

- **Inclusion of color cues**: The experiments that analyzed four different color spaces to extract CHOG features using EBGM revealed that color cues lead to better performances than the grayscale HOG-EBGM. Indeed, the best results were achieved using the Opponent Color Space along with the Discriminant Color Space. The results obtained using color were approximately 10% higher than those yielded from the FRGC dataset.

8.1.3 Integrated System

An integration for automated face recognition based in HOG-EBGM with eye location was tested in a challenging scenario: a realistic mobile-based scenario for video sequences. This verification was carried out as part of the MOBIO competition.

The experiments show that the recognition performance obtained with an EBGM algorithm depends closely on the initial location of the eyes. The recognition results obtained from our proposed eye location subsystem increased up to 10% regarding a random location with displacements of 5% to 10% of the inter-ocular distance.

Regarding the results of the MOBIO experiments, some conclusions can be drawn:

- **Selection of the best faces in video sequences**: To select the best faces to extract biometric information, the spatial redundancies over time can be exploited. In a video sequence, the high number of face images per person enables that the facial analysis may be performed only on those images that best fit the requirements of the face graph algorithms.

Kalman filtering can be helpful in obtaining a more stable location of the eyes, thereby rendering more accurate faces. Recognition results using this method surpassed those in which a simple eye location was performed, as the faces selected to describe individuals were more representative.

- **Our system compared to other MOBIO participants**: The solution proposed in the present work achieved quite good results in the MOBIO competition, placing among the top participants. The use of Kalman filtering to stabilize the location of the eyes in video-sequences improved the results. Also, let's note that our solution did not perform score normalization, which in some cases helped improve results.

In all, it has been proved that our integrated system based on the use of HOG local descriptors is on par with other face recognition algorithms. The fusion of the information provided by the two subsystems has been successfully resolved, with the location of the eyes serving as an appropriate starting point for the recognition subsystem.

8.2 Dissemination and Applications

The research carried out in this work has been received positively by the scientific community, thereby giving rise to a number of scientific publications, dissertations in international conferences and research projects. Moreover, collaboration with various firms has also generated an industrial interest in our developments. This sections is dedicated to describing the dissemination of this research, as well as the industrial applications derived from it.

8.2.1 Scientific Publications and International Conferences

Dissemination of findings related to the investigation in the present thesis has extended to various publications in scientific journals and presentations in international conferences. The list of publications is as follows:

International Journals

- David Monzo, Alberto Albiol, Jorge Sastre, and Antonio Albiol. **Precise eye localization using HOG descriptors**. Machine Vision and Applications, pp 1-10, 2010. [83]

- Alberto Albiol, David Monzo, Antoine Martin, Jorge Sastre, and Antonio Albiol. **Face recognition using HOG-EBGM**. Pattern Recognition Letters, 29(10):1537-1543, July 2008. [7]

International Conferences

- David Monzo, Alberto Albiol, Antonio Albiol, and Jose M Mossi. **Color HOG-EBGM for face recognition**. In 2011 18th IEEE International Conference on Image Processing (ICIP), pages 785-788, September 2011. [81]

- David Monzo, Alberto Albiol, Antonio Albiol, and Jose Manuel Mossi. **A Comparative Study of Facial Landmark Localization Methods for Face Recognition Using HOG descriptors**. In Proceedings of the International Conference on Pattern Recognition, Istanbul, August 2010. [80]

- David Monzo, Alberto Albiol, Jorge Sastre, and Antonio Albiol. **HOG-EBGM vs. Gabor-EBGM**. In IEEE International Conference on Image Processing (ICIP), San Diego, CA, USA, pages 1636-1639, October 2008. [82]

- S. Marcel, D. Monzo, A. Albiol *et al.* **On the results of the first mobile biometry (mobio) face and speaker verification evaluation**. Recognizing Patterns in Signals, Speech, Images and Videos, volume 6388 of Lecture Notes in Computer Science, pages 210-225, Springer Berlin Heidelberg, 2010. [71]

Public Dissemination

- K. Baker, J. Kantorovitch, D. Monzo, J. Vandenabeele, C. Sandoval, **Guarantee: Active Safety Products, Architecture and Business Opportunities**, in eChallenges e-2011 Conference Proceedings, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation Ltd 2011.

8.2.2 Research Projects

The work carried out in this thesis has formed the basis for various research projects at the local and European level. Two principal projects incorporating our research are described below, with the specific research contributions highlighted.

International: Guarantee Project

The novel face recognition system presented here has been used in European research projects involving computer vision. This is the case of the Guarantee Project of the ITEA2 commission which researches ICT solutions incorporating advanced technology, safety and human behavioral understanding.

The Guarantee research project has selected a set of 40 cases where safety could be improved around the home and public places by incorporating an understanding of human behavior, and studied how this knowledge can be applied to advanced technologies. These use cases were reduced to four demonstrators, which show how current home communication networks, robotics, care terminals and vision systems can be used jointly to improve safety.

Our approach formed the core of one of the demonstrators, the *Smart Door* scenario. This scenario was aimed at assisting elderly people who spend long periods alone at home. When a visitor arrives, the lack of information about this person can lead to unexpected reactions from the elderly. The project posited that a set of technological subsystems and sensors might be combined in this scenario to produce a synergy capable of revealing the identities of the actors, and the viewers resultant reactions and behaviors, thereby facilitating or denying the callers entrance. The HOG-EBGM algorithm was used to detect and recognize potential visitors at the door, and manage a database of friends/relatives of the elderly resident.

Figure 8.1 shows an example of the set-up in Guarantee. The HOG-EBGM sends the recognition results from a visitor at the main door to an interactive IPTV, where the interface with the elderly resident is shown.

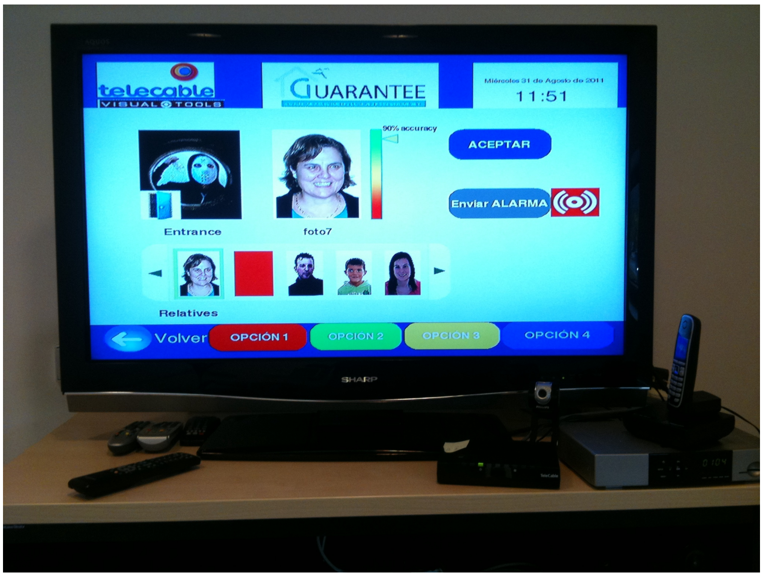


Figure 8.1: Picture of the set-up in the Guarantee Project: an interactive IPTV shows the results of face recognition (using HOG-EBGM) from a peep-hole camera at the main entrance of a home.

As a result of this project, a scientific publication was included in the *eChallenges e-2011 Conference*.

Spain: PATRICIA Project

Our face recognition system has likewise formed part of local research projects. One example is project PATRICIA, developed as part of the IMPIVA Program (Comunidad Valenciana, Spain). Project PATRICIA (Activity Patterns in Retail using Advanced Computer Vision) is aimed at providing the retail sector computer vision solutions beyond uses normally associated with security . This project was carried out by a technological firm, **Visual Tools S.A.**, in collaboration with the optics institute, AIDO.

The object of PATRICIA was to assist shop managers in defining or refining their marketing strategies through an analysis of customer behavior. This behavior is inferred from objective information retrieved by a multi-camera system installed in a shopping area. In this research project, two different use cases were analyzed: an estimation of crowd movements using heat maps, and a statistical estimation of a customer average time inside a shopping area.

Our face recognition system formed the core of the latter. Specifically, two cameras were installed in a supermarket, one monitoring the entrance and one monitoring the exit, both capturing frontal face images of the customers. Both cameras were synchronized: when a face from a customer leaving the supermarket was identified with a face of a customer that had previously come in, then the time spent inside the shop could be determined. This identification process was designed with the HOG-EBGM as a base.

Figure 8.2 shows two pictures from the set-up used in the PATRICIA project for estimation of stay time of the clients, and two images recorded from the testing footage of these cameras.

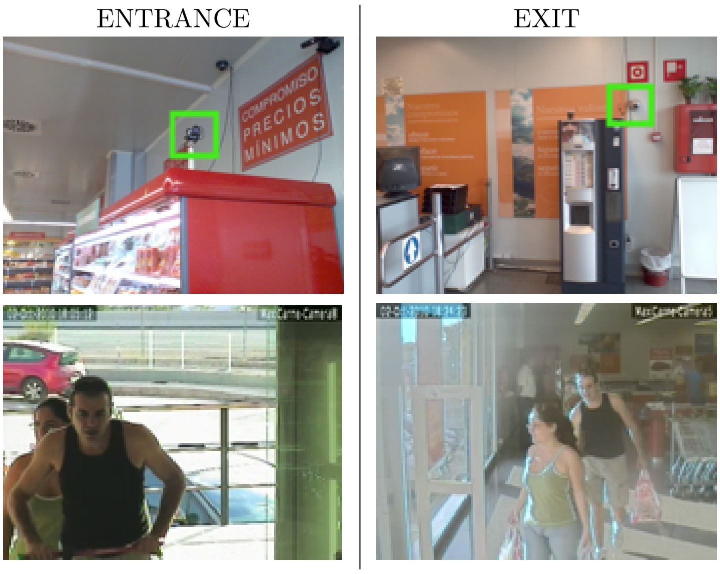


Figure 8.2: Set-up in the project PATRICIA, along with two images retrieved using the system.

The pictures in the figure show that in this use-case the light conditions in the images recorded were sometimes extreme, rendering them invalid for their original purpose.

8.2.3 Industrial Applications

Escritorio Movistar: Telefónica I+D

During its initial stages, this thesis was granted by **Telefonica I+D**, a pioneer company in Research and Development. Collaboration with Telefonica I+D took place in the framework of the research project *Codificacion y Aplicaciones Multimedia*. The main goal of this project was to improve a software package developed by Telefonica I+D: Escritorio Movistar.

Escritorio Movistar is a free software that provides smart management of network connections and enables mobile access to data services such as Internet, corporate Intranets or E-mail and other Movistar services. All services are designed to work across different platforms, thus allowing the user to use it with multiple terminals and operating systems.

The project *Codificacion y Aplicaciones Multimedia* aspired to, among other things, incorporate an automated face recognition system in the main program, for use as a non-intrusive biometric password.

This collaboration facilitated the development of the early prototypes for joint face detection and recognition system presented in this work. In Figure 8.3, an example is given of a recognition match process.

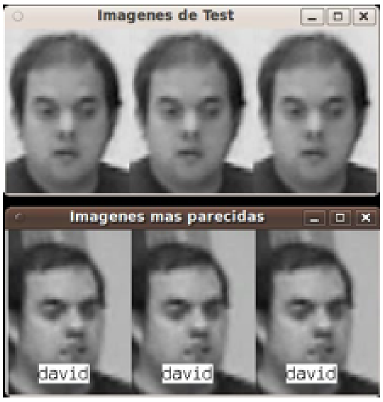


Figure 8.3: Example of a sequence for face recognition with test images matched to labeled images.

Security Applications: Visual Tools S.A.

Our work has also left its mark on the area of security and surveillance. The technology firm **Visual Tools S.A.** is closely involved with the field of computer vision, particularly in its application to security and surveillance. Our group collaborated with the company to produce various commercial applications based on the HOG-EBGM face recognition algorithm.

This collaboration yielded developments of various prototype systems which offered the opportunity to test our algorithms in realistic scenarios. In 2010, a banking security system was designed for a major Spanish bank[2]. In this case, real images of criminals entering banking offices were matched to a database of suspects. The scenario proved to be highly challenging: the cameras at the banking offices were not positioned to capture frontal face-images of the suspects and the conditions were highly uncontrolled.

Figure 8.4 presents an example of the interface employed by our system to identify criminals and manage the database of suspects. Due to privacy issues, all faces have been blurred in this figure.

8.3 Future Work

In this section, some outlines about pending issues and future directions for research are briefly offered.

The face detector based on eye location offers good performance and significant robustness to semi-controlled variations. However, its usability is restricted by some inherent limitations: to detect the faces, they have to be frontal or

[2] Due to privacy concerns, details and non-blurred images regarding this project cannot be included.

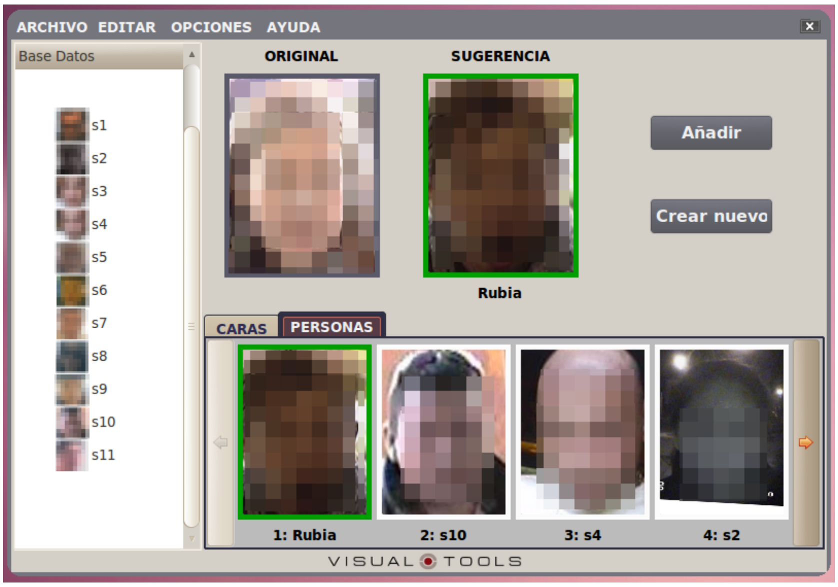


Figure 8.4: Snapshot of the interface used to identify people in the banking security application (faces are blurred).

quasi-frontal, and both eyes must also be perfectly visible. In images where the eyes are not entirely visible (because of the face angle or partial occlusion by elements such as glasses frames, light reflections on the glasses or dense fringes), the performance markedly decreases.

To deal with the non-frontality of the of the face, several approaches could be addressed, such as the introduction of 3D models of the face or adaptation of face graphs to use only the information for the parts that are visible. In order to deal with the partial or total occlusion of the eyes, two possible approaches arise. One solution would be to improve the training datasets to include more examples of difficult images as positive samples. This would generate more accurate models and could help to minimize the classification errors. Another possible solution requires providing more robustness to the location algorithm, including a third facial landmark. With three landmarks, when just one is missing, the remaining pair still offer information to normalize the representation. In this case, it would be advisable to take the tip of the nose as a landmark as this key-point is visible most of the times and is also less apt to be contorted by facial gestures.

Regarding the face graphs extracted with EBGM, this thesis has tackled with the problem of frontal faces. However, when the face is not frontal, the graph gets distorted and the data from many of the nodes introduces more noise than biometric information. An analysis of the head pose could move our algorithm to a new stage to determine which landmarks have higher contributions and statistically model the useful information. This way, some of the landmarks could even be neglected, while the addition of pose-specific landmarks could also be considered.

The main texture descriptor used in this thesis has been local gradients. The HOG features have proven to offer good performance in facial analysis, nevertheless some open lines of research could still be derived from them. For a start, a mixed approach combining holistic and feature-based unique HOG size was used in this work to describe multiple facial elements. However, its size was optimized for the description of the eyes. Considering that each facial landmark (the eyes, the nose, the mouth, etc.) have a different size, one could logically predict that the descriptive power of the face graph could likewise be optimized by varying the size of the descriptor for each landmark. In the same vein, it would be of great interest to combine this component-based approach with holistic HOG descriptors extracted globally from the face.

Finally, this line of research should not be limited to the use of a single descriptor, the HOG feature. Other state-of-the-art descriptors, such as the Local Binary Patterns [89], should be studied/examined. These descriptors have been proven to be quite robust in describing faces in uncontrolled conditions.

Appendix A

Dimension Reduction Methods

A.1 The need for dimension reduction in feature-based face analysis

In Machine Learning, it is common to find the problem of methods breaking down because they cannot manage the number of random variables (dimensions) of the samples measured on an observation [36]. When working with high-dimensional sets of samples, quite often we find that the majority of the variables do not have a significant contribution to solve a specific machine learning problem.

This gives raise to the interest to reduce the dimension of the original data, $x = (x_1, \ldots, x_p)^T$, into a lower dimensional representation, $s = (s_1, \ldots, s_k)^T, k \leq p$, used to model a machine learning problem.

The study of the reductive techniques in this thesis has been motivated by the high dimensional features obtained using the feature-based algorithms described in Chapter 5 and Chapter 6. The current appendix is focused on the theoretical study of some key approaches: Principal Component Analysis (PCA) [91, 107], the Linear Discriminant Analysis [37, 12], some algorithms directly derived from the LDA analysis, such as the PCA over Null-Spaces [46], Orthogonal LDA [122] (OLDA) and the Regularized LDA [59], and finally the Kernel Fisher Analysis (KFA) [64].

This process of dimensionality reduction is shown in the diagram displayed in Figure A.1.

A.2 Principal Component Analysis - PCA

The Principal Component Analysis (PCA) is one of the most extended statistical methods, as it is the best linear dimension reduction technique in terms of reducing the mean-square error. PCA is mainly used in two ways: for the representation of samples and for the reduction of dimensions.

This supervised method is trained producing an orthogonal lineal transformation over the samples of a set of training data. A subspace is derived where

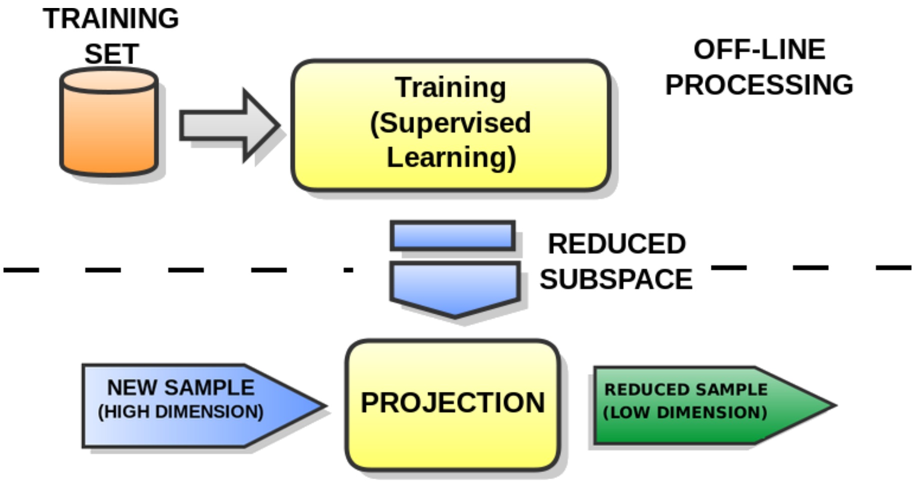

TRAINING SET
Training (Supervised Learning)
OFF-LINE PROCESSING
REDUCED SUBSPACE
NEW SAMPLE (HIGH DIMENSION)
PROJECTION
REDUCED SAMPLE (LOW DIMENSION)

Figure A.1: Diagram of the steps needed for the dimension reduction stage.

the main axis are related with the directions of maximum variation of the training samples; using this coordinates change, the correlation between the training samples can be highly reduced. The directions where the samples have less variation are rejected, giving place to a lower final number of dimensions in the subspace.

From a mathematical perspective, the set of directions with maximum variation, Φ, can be found by solving an eigenproblem such like $S_t\lambda = \lambda\phi$, where S_t is the scatter matrix of the training data, directly related to the correlation matrix.

Let's define the sample matrix, $X = [x_1, \ldots, x_N] \in \mathbb{R}^{m\times n}$, where each row corresponds to a vectorized training image. The set of training images has been previously normalized to have zero mean. From this data, the scatter matrix is defined as $S_t = XX^T$.

Solving the equation:

$$S_t\Lambda = \Lambda\Phi, \tag{A.1}$$

the main directions of the new subspace are extracted, $\Phi = [\phi_1, \ldots, \phi_n] \in \mathbb{R}^{m\times n}$, where ϕ_1 is the direction of maximum variation of the data, ϕ_2 is the following direction of maximum variation and so on, and the initial dimension of the samples is n.

The vector $\Lambda = [\lambda_1, \ldots, \lambda_n] \in \mathbb{R}^{m\times n}$ contains the eigenvalues associated to the vectors of directions ϕ_i, and they are indicators of the quantity of energy of the data contained in each of the directions in the subspace. The directions with higher variance correspond to eigenvalues of higher energy.

Once the PCA subspace is learned, projecting a single sample into the new subspace is done by simply solving the following transformation:

$$Y = \Phi^T X_{new} \tag{A.2}$$

Given a new set of samples, $\hat{X}_{new}$, the procedure to project them is the following:

1. calculate the normalized zero-mean version of $\hat{X}_{new}$ by subtracting the mean vector from the training set, x_{mean} to each of the samples, such that $x^i_{new} = \hat{x}^i_{new} - x_{mean}$.

   A PCA analysis is optimum for the statistical representation of the samples in terms of mean square error. Normalizing X to have zero mean is critical to find the new base that minimizes the mean square error of the projected data.

2. project X_{new} using the transformation matrix Φ as shown in equation A.2. Removing the directions ϕ_i associated to eigenvalues λ_i with no energy o energy close to zero leads to a reduction of the dimensions without losing relevant information.

   In many works, the criterion selected for the reduction is to conserve a number of eigenvectors with a high rate of total energy, e_{ratio}. Specifically, the energy contained on the first i eigenvectors associated to the largest i eigenvalues is defined as,

   $$e_i = \frac{\sum_{k=0}^{i} \lambda_k}{\sum_{k=0}^{n} \lambda_k} \tag{A.3}$$

   Using this criterion, the final projection matrix, Φ' consists of $\Phi' = [\phi_1, \ldots, \phi_i]$, such that $e_i < e_{ratio}$ and $e_{i+1} > e_{ratio}$. Selecting the first i eigenvectors, the dimensionality of the samples is reduced from $dim_{inicio} = n$ to $dim_{final} = i$.

A.3 Linear Discriminant Analysis - LDA

The Linear Discriminant Analysis (LDA) is one of the most popular reduction and classification methods. The outline behind this algorithm is similar to the PCA technique, as the goal is to find a new base of vectors conforming a subspace in which the new samples will be projected.

LDA generates a subspace where the distance between all the samples belonging to the same class is minimized, while at the same time the distance between samples belonging to different classes is maximized.

In mathematical terms, given a set of training samples X, having a mean sample $\bar{m}$, and with all the samples labeled into C different classes, $\{L_1, \ldots, L_C\}$, let's define a *between classes* scatter matrix, S_b, and a *within class* scatter matrix S_w, such that:

$$S_b = \sum_{i=1}^{C} N_i (m_i - \bar{m})(m_i - \bar{m})^T \tag{A.4}$$

$$S_w = \sum_{i=1}^{C} \sum_{x_j \in L_i} (x_i - m_i)(x_i - m_i)^T \tag{A.5}$$

where m_i is the mean of all the samples belonging to the class $i, x_j \in L_i$. Let's also remark that the scatter matrix S_b represents the variations between the classes represented by their means, and the scatter matrix S_w represents the the variations between samples of the same class. These two matrices are directly related with the total scatter matrix, S_t, which is the scatter matrix of the whole set of samples. This relation is showed in the following expression:

$$S_t = S_b + S_w \tag{A.6}$$

To generate a final subspace in which the maximum separation of samples from different classes is reached, it is necessary to maximize the so called Fisher Criterion:

$$\Phi_{opt} = arg \max_{\Phi} \frac{\|\Phi^T S_b \Phi\|}{\|\Phi^T S_w \Phi\|} \tag{A.7}$$

where Φ is a matrix for linear transformation. In an equivalent way, the criterion can be expressed as

$$J_F(\Phi) = \max_{\Phi} \frac{trace(\Phi^T S_b \Phi)}{trace(\Phi^T S_w \Phi)} = \max_{\Phi} \{trace((\Phi^T S_w \Phi)^- 1(\Phi^T S_b \Phi))\} \tag{A.8}$$

The maximization of the J_F criterion is given after solving the following generalized eigen-problem

$$S_w^{-1} S_b \Lambda = \Lambda \Phi \tag{A.9}$$

where the eigenvectors that are associated to the eigenvalues with higher values form the base of the new subspace. Let's note that upper limit regarding the dimensions of this subspace is given by $l = C - 1$.

Assuming a normal distribution of the samples for each of the classes, it has been proved [43] that applying the LDA technique is equivalent to perform a maximum likelihood classification. Even if this initial assumption was not completely true, the effectiveness of the LDA has been also proved, mainly due to the fact that the lineal models are robust against noise and tend not to suffer the effects of *overfitting*.

When the LDA is used on samples that are images for facial analysis, it is quite common to come across the situation where the number of face images used to learn the new subspace is much lower than the original dimension of their feature vectors. This is known as the *undersampling* problem, and has a direct effect on the calculus of the Fisher Criterion A.8. In such cases the within scatter matrix, S_w, is singular and consequently it has no inverse S_w^{-1}. This leads to a generalized eigen-problem with no solution.

To avoid the *undersampling* problem, many authors have adopted the solution of including a preprocessing stage using PCA, prior to the LDA technique (PCA-LDA). This solution is aimed for a reduction of the dimensions of the original data X in the first stage, projecting them to the PCA subspace. The final dimension of the PCA subspace is chosen such that the within-class scatter matrix of the projected data, S'_w, is non-singular, being then possible to solve the generalized eigenproblem in A.9.

Other solutions to the *undersampling* problem are also possible. Next, some of them are described.

A.4 Techniques deriving from the LDA algorithm

The technique of the PCA-LDA exposed before is not always the best solution for the *undersampling* problem.

When a preprocessing PCA stage is applied, there is not a unique criterion to select the number of dimensions dim_{PCA} that should be kept to make S'_w non-singular without loss of information. Trying to avoid the singularity problem of S_w, after applying the PCA stage, some eigenvectors associated to non-null eigenvalues are removed to form the base of the PCA subspace. However, this vectors usually correspond to directions that contain pieces of discriminative information, which will be completely lost during the LDA stage.

To overcome this problem, some LDA-based variants have been developed. Specifically, in this work, three different algorithms derived from the original LDA are analyzed: the PCA on null-spaces [46], the Orthogonal LDA [122] and the Adaptive Power Method LDA [99, 59], also known as the *Lanczos algorithm.*

A.4.1 PCA on null-spaces

Huang et al. [46] developed an alternative to the *undersampling* problem in LDA by studying the behavior of the subspaces spanned by the scatter matrices S_t, S_b y S_w previously described. From Equation A.6 it can be seen that the three scatter matrices are linked, and therefore, there should also exist a direct relation between the subspaces spanned by each of them.

From other works, it is known that the null space generated by S_t is the subspace spanned by the eigenvectors associated to null eigenvalues. This subspace is the intersection of the null subspaces of S_b and S_w, and it does not contain discriminative information. It can be removed without losing significant information.

However, let's keep in mind that the part of the null spaces of S_b and S_w which do not intersect do contain discriminative power. Taking advance on this feature, the method that is proposed in [46], solves the LDA problem following some steps:

1. First, the intersected null-space from S_b and S_w –equivalent to the null-subspace generated by S_t–, is removed. To achieve this, a PCA technique is applied on thel data matrix X, obtaining a transformation matrix, V_{PCA}. The columns corresponding to the eigenvectors associated with null eigenvalues are removed, resulting in the transformation matrix Φ'_{PCA}.

   Then, the matrices S_b and S_w are projected on the subspace generated by Φ'_{PCA}, such that $S'_b = {\Phi'_{PCA}}^T S_b \Phi'_{PCA}$ and $S'_w = {\Phi'_{PCA}}^T S_w \Phi'_{PCA}$.

2. The matrix S'_b is projected over the null-space spanned by S'_w, which is known to have the greatest discriminative power [127]. To that end, PCA is again applied on S'_w to obtain a projection matrix Φ_{S_W}.The null-space of the matrix corresponds to the eigenvectors associated to the null eigenvalues; this is spanned by the projection matrix $\Phi_{S_W\perp}$. Projecting S'_b with $\Phi_{S_W\perp}$, a new between-class scatter matrix, $S''_b = {\Phi_{S_W\perp}}^T S'_b \Phi_{S_W\perp}$, is obtained.

3. The last step consists of removing the null-space in Sb'', which ideally does not contain discriminative information to separate the samples. Again, PCA is applied on this scatter matrix, obtaining a transformation matrix Φ_{S_b}, from which the eigenvectors associated to the null eigenvalues are removed. The result is the projection matrix $\Phi'_{S_b} = \Phi_{NULL}$.

The projection matrix Φ_{NULL} maximizes a modified version of the Fisher Criterion A.7:

$$\Phi_{NULL} = arg \max_{\Phi} \|\Phi^T S_b \Phi\| \tag{A.10}$$

and thus is a valid alternative to the LDA.

Contrary to PCA-LDA, this technique does not remove discriminative information when the PCA is used. However, it has to be remarked that the modified Fisher Criterion has lower accuracy compared to the original criterion A.8, as it does not take into account the minimization of the scatter matrix of the samples belonging to the same class, and it can lead to also inaccurate distribution of the samples in the final subspace.

A.4.2 Orthogonal Linear Discriminant Analysis - OLDA

Among the multiple LDA variants that are proposed in the literature to avoid the undersampling problem, one that is not much extended but has been proved to be quite effective is the one exposed by Ye in [122]. The key idea of this variant lies in the discriminant vectors that form the basis for the LDA subspace, which in [122] are all orthogonal. In other words, the columns of the projection matrix derived by this technique are orthogonal, giving its name to the method, *Orthogonal LDA*.

Ye proposes an efficient method to calculate the projection matrix of the OLDA. Next, in this section, we describe the mathematics that are needed to extract the orthogonal basis from a training data set X, in which the samples belong to C different classes. Due to the complexity of the mathematics involved in this technique, which are out of the scope of this thesis, only a simplification of the main steps is provided here:

1. Similar to the steps performed in null-space PCA, OLDA removes the null space of the total scatter matrix S_t, applying PCA on it. The result is the projection matrix Φ_t associated to the eigenvalues Λ_t:

$$S_t = \Phi_t \Lambda_t \Phi_t^T = [\Phi_1, \Phi_2] \begin{pmatrix} \Sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix} [\Phi_1, \Phi_2]^T \tag{A.11}$$

   where Φ_1 corresponds to the eigenvectors associated to non-null eigenvalues Σ_t^2.

2. A matrix H_b is defined such that it has in its columns the centered means that represent each of the classes, $H_b = [\sqrt{N_1}(m_1 - m), \ldots, \sqrt{N_C}(m_C - m)]$. It is easy to check that $S_b = H_b H_b^T$.

   H_b is projected on the non-null space spanned by S_t:

$$H_b' = \Sigma_t^{-1} \Phi_1^t H_b \tag{A.12}$$

   and afterwards, H_b' is decomposed using Singular Value Decomposition,

$$H_b' = P \hat{\Sigma} Q^T \tag{A.13}$$

   obtaining a projection matrix Φ_{diag} that, as Ye proves in [122], simultaneously diagonalizes the three scatter matrices S_b, S_w and S_t:

$$\Phi_{diag} = \Phi_t \begin{pmatrix} \Sigma_t^{-1} P & 0_m \\ 0_k & I_m \end{pmatrix} \tag{A.14}$$

3. Given that usually the range of the within scatter matrix is $\mathrm{rank}(S_b) = C - 1 = q$, the next step consists of selecting the first q columns of Φ_{diag} to obtain a new projection matrix Φ_q.

   In [122], Ye also proves that the Fisher Criterion A.8 can be maximized from the matrix Φ_q, such that $\Phi = \Phi_q M$, where $M \in \mathbb{R}^{q \times q}$ is a non-singular arbitrary matrix. The degree of freedom to select the matrix M gives place to select an orthogonal projection matrix, which is the object of this reduction technique.

   To that end, first the QR decomposition is performed on the matrix Φ_q, such that $\Phi_q = \hat{Q}\hat{R}$. Because of the properties of the QR decomposition, the columns that compose the matrix $\hat{Q}$ are orthogonal. Thus, an arbitrary matrix can be selected to fulfil $M = \hat{R}^{-1}$, making $\Phi_q M = \hat{Q}\hat{R}\hat{R}^{-1} = \hat{Q}$, which is orthogonal.

   Solving the problem this way, the final projection matrix for the Orthogonal-LDA is:

$$\Phi_{OLDA} = \Phi_q \hat{R}^{-1} = \hat{Q} \tag{A.15}$$

Ye proves in [122] that the removal of $\hat{R}$ during the QR decomposition reduces the noise inherent to the classification of the samples, and thus it provides them with a higher discriminative power.

A.4.3 Regularized LDA

One of the most extended methods for solving the undersampling problem is the Regularized LDA (RLDA). These techniques are focused on avoiding the *undersampling* problem inducing an artificial conditioning to the total scatter matrix. The most widespread regularizing method is the following:

$$\hat{S_t} = \alpha S_t + (1 - \alpha)I, 0 \leq \alpha \leq 1 \tag{A.16}$$

where I is the identity matrix and α is the so called arbitrary regularization constant. Nevertheless, in [59] the authors use a simplified (and also quite extended) variant of this regularization technique, defined as:

$$\hat{S_t} = S_t + \Gamma I \tag{A.17}$$

where Γ is a vector of elements that modify the principal diagonal of the scatter matrix S_t, so that the matrix $\hat{S_t}$ is invertible.

In this thesis, we use RLDA approach based on the work of De la Torre *et al.* [59] in the context of Oriented Component Analysis (ROCA). This approach uses a variant of the Power Method technique to solve generalized eigengproblems. Specifically, following the formulation of the *Lanczos algorithm* [99], it solves the following variant of the Fisher Criterion:

$$J_F(\Phi) = \max_{\Phi} \frac{trace(\Phi^T S_b \Phi)}{trace(\Phi^T S_t \Phi)} \tag{A.18}$$

where the matrix that is minimized is not the within-class scatter matrix, S_w, which represents the variation of the samples from the class, but the total scatter matrix, S_t, which represents the variation of all the samples in the training set.

Given that S_t and S_w are related such that $S_t = S_w + Sb$, both variants of the Fisher Criterion produce equivalent results.

Starting from a simple regularization, based on the addition of an arbitrary constant γ to the principal diagonal of the total scatter matrix, $\hat{S_t} = S_t + \gamma I$, the iterative method proposed by De La Torre *et al.* is solved following the next steps:

- **Step 0**: Generate an initialization for the projection matrix $\Phi^{(k=0)} \in \mathbb{R}^{d \times q}$. In this case, the initialization is not arbitrary. To reduce the number of iterations, the matrix formed with the eigenvectors of S_b associated to the non-null eigenvalues is taken as the seed for the iterations.

- **Step 1**: The iteration $k = k + 1$ starts.

- **Step 2**: The linear system $\hat{\Phi}^{(k+1)} = \hat{S_t}^{-1} S_b \Phi^{(k)}$ is solved.

- **Step 3**: The matrices $\hat{\Phi}^{(k+1)}, \hat{S_t}, S_b$ are normalized, such that:

$$\begin{aligned} \hat{\Phi'}^{(k+1)} &= \frac{\hat{\Phi}^{(k+1)}}{max(\Phi^{(\hat{k+1})})} \\ S'_b &= \hat{\Phi'}^{(k+1)T} S_b \hat{\Phi'}^{(k+1)} \\ \hat{S_t}' &= \hat{\Phi'}^{(k+1)T} \hat{S_t} \hat{\Phi'}^{(k+1)} \end{aligned} \tag{A.19}$$

- **Step 4**: The projection matrix $\Phi^{(k+1)}$ is determined after solving the generalized eigen-problem $S'_b W = \hat{S_t}' W \Delta$, and the projection $\Phi^{(k+1)} = \hat{\Phi'}^{(k+1)} W$.

- **step 5**: Repeat the sequence **Step 1-Step 4** until the condition $\frac{|\sigma_i^{(k+1)} - \sigma_i^{(k)}|}{\sigma_i^{(k+1)}} < \epsilon, \forall \epsilon$ is reached, where σ_i are the eigenvalues extracted in **Step 4**.

A.5 Kernel Fisher Analysis - KFA

The last reductive technique studied in this work is the one developed by Liu in [64], which is the Fisher analysis based in kernel functions, also known as Kernel Fisher Analysis (KFA). This technique is designed to be applied on sets of labelled samples belonging to multiple classes and make use of different kernel functions, such us the polynomial or the radial basis, among others.

Sometimes, the linear discrimination is not enough to solve specific machine learning problems; Liu proposes to exploit the non-linear dependencies that exist between the samples and their sets. The idea that supports KFA is to find a mapping using a proper kernel function that projects the original samples to a vectorial space of higher dimensions than the feature space. This new space is called the *mapping space*. The goal of projecting to this space is that in it a lineal separation of the samples can be performed. However, let's remark that even though the separation of samples in the mapping space is linear, when such mapping is reversed the separation in the feature space is not linear. This is because of the non-linearity of the Kernel functions.

There are other algorithms, like the Generalized Discriminant Analysis (GDA), proposed by Baudat et al. [10], which also use the non-linear approach to improve the results of the LDA. In this work, we have studied KFA, as a derivation of the GDA, because unlike other methods, it gives a unique solution to the calculus of the final subspace, being thus an optimal solution for an specific Kernel function.

Following the steps of the kernel-based techniques, the key to classify the training samples using non-linear discrimination using a space of higher dimensions is in the use of dot products, such that $kernel(x, y) = kernel(x) \cdot kernel(y)$. Given a kernel function ρ, the training set is mapped such that $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m \times n}$. The mapped set is $\mathcal{D} = [\rho(x_1), \ldots, \rho(x_n)] \in \mathbb{R}^{q \times n}, q > m$ and it can be arranged to have zero mean. Defining a correlation matrix of the samples in the mapped space, we have that:

$$\mathcal{K} = \mathcal{D}\mathcal{D}^T \tag{A.20}$$

Now, following the steps of the LDA, a total scatter matrix and a between-class scatter matrix can be defined from the mapped set:

$$\bar{S_t} = \frac{1}{n}\mathcal{D}\mathcal{D}^T \tag{A.21}$$

$$\bar{S_b} = \frac{1}{n}\mathcal{D}W\mathcal{D}^T \tag{A.22}$$

where n is the total number of samples, and $W \in \mathbb{R}^{n \times n}$ is a block diagonal matrix, $W = diag[W_1, \ldots, W_C]$, such that each element is $W_j \in \mathbb{R}^{N_j \times N_j}$, with a constant value in its diagonal: $\frac{1}{N_j}, 1 \leq j \leq C$, being C the number of classes and N_j the number of elements in class j.

Using these matrices, the Fisher Criterion A.7 can be formulated to derive a projection matrix. This projects the samples to a subspace of less dimensions than the original feature space. To do this, the generalized eigen problem $\bar{S_b}\Phi = \lambda\bar{S_t}\Phi$ is solved:

$$\Phi = \sum_{i=1}^{m} c_i\rho(x_i) = \mathcal{D}\alpha \tag{A.23}$$

where $\alpha = [c_1, \ldots, c_m] \in \mathbb{R}^m$. The generalized eigen-problem can be reformulated using Equation A.20 and Equation A.23, and applying a regularization step to the total scatter matrix of the mapping space, as it is shown in Equation A.17:

$$\mathcal{K}W\mathcal{K}\alpha = \lambda\mathcal{K}\mathcal{K}\alpha \tag{A.24}$$

with $\alpha = [\alpha_1, \ldots, \alpha_n]$, and where each α is defined as: α_i, $\|\Phi_i\|^2 = \alpha_i^T\mathcal{K}\alpha_i = 1$.

The projection matrix $\Phi = [\Phi_i, \ldots, \Phi_n]$ can be directly solved as

$$\Phi = \mathcal{D}\alpha \tag{A.25}$$

In PCA, LDA and the LDA-derived techniques, to project test samples to the learned subspace it is only necessary to use the projection matrix and maybe the mean training vector to have zero mean data. Let's notice however that in KFA, after learning the final subspace, the projection of a new data sample,

X_{new}, is done as $Y_{new} = \Phi^T \rho(X_{new}) = \alpha\mathcal{B}$, where $\mathcal{B}$ is defined from the dot products of each of the training samples with regard to X_{new}, such that

$$\mathcal{B} = [\rho(X_1) \cdot \rho(X_{new}), \ldots, \rho(X_1) \cdot \rho(X_{new})] \tag{A.26}$$

This implies that in this case we have to store not only the information about the projection matrix, but also the original training samples and the kernel function that was used for the first mapping.

Appendix B

Public Face Databases and Evaluation Protocols

This appendix describes the face databases and their related evaluation protocols used for the experiments in this thesis. These datasets are public and available for researchers, and all of them contain frontal face images in controlled or semi-controlled scenarios. Also, they contain more than one sample per individual, providing an good context for supervised learning.

B.1 BioID Database

The BioID Face Database [47] was recorded and published to give all researchers working in the area of face detection the possibility to compare the quality of their face detection algorithms with others. It comprises 1521 gray level images from 23 different subjects. The images show the frontal view of the persons and were acquired under a large variety of illumination, background and face sizes. The scenarios are semi-controlled indoors and groundtruth data of the position of the eyes (along with other facial landmarks) is also provided.

Some sample images of the BioID database can be seen in Figure B.1.

Database	Images	Individuals	Color	Metadata	Eval. Protocol
BioID	1521	23	–	–	–
Yale	165	15	–	X	–
CVL	114	7	X	–	–
AR	~ 4000	126	X	X	–
FERET	3365	~ 1000	–	X	X
FRGC	36818	16028	X	X	X

Table B.1: Summary of the main attributes of the face databases used in this thesis.

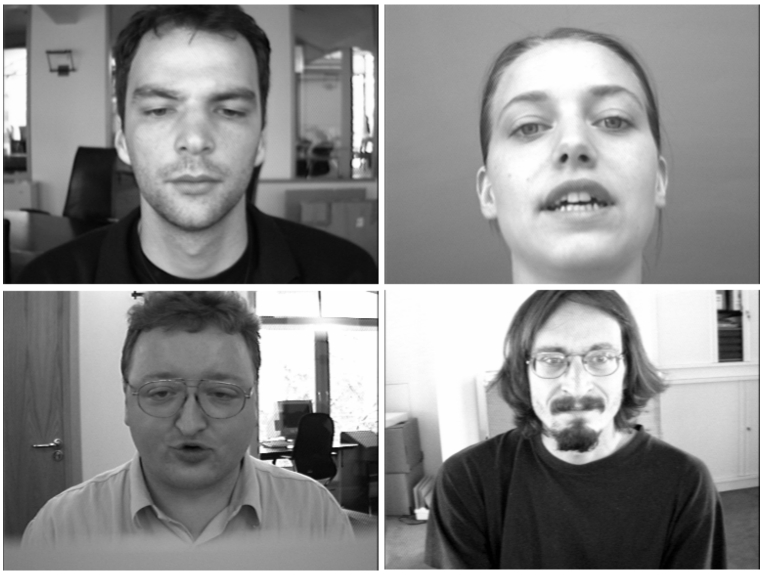


Figure B.1: Image samples from BioID.

B.2 Yale Database

The Yale Face Database [2] contains 165 grayscale images of 15 individuals. For all the subjects, a total of 11 images per subject were recorded. Each of these images corresponds to a different predetermined facial expression or scenario configuration: center-light, glasses, happy, left-light, no glasses, normal, right-light, sad, sleepy, surprised, and wink. This makes the Yale database useful to test the robustness any classification algorithm against changes in illumination and facial gestures.

Some sample images of the Yale database can be seen in Figure B.2.

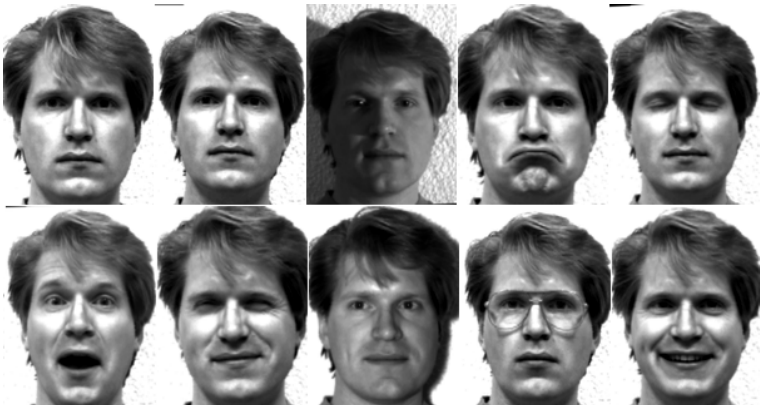


Figure B.2: Image samples from Yale.

B.3 AR Database

The AR face database was in the Computer Vision Center (CVC) at the U.A.B. It contains around 4,000 color images corresponding to 126 individuals. Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf) in controlled indoor scenarios. Each individual participated in two sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions.

Some images of the AR database can be seen in Figure B.3.

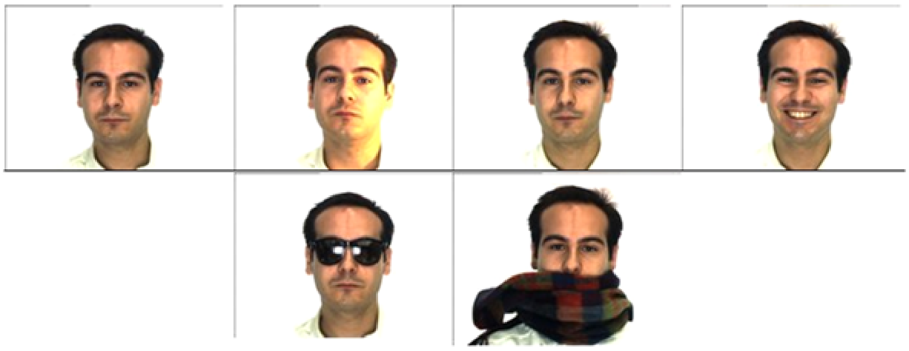


Figure B.3: Image samples from AR.

B.4 CVL Database

The CVL database [92] contains a set of 114 individuals with 7 images per each. The images were taken under uniform illumination and with projection screen in the background. Due to the different head orientations acquired, many works make only use of the images corresponding to frontal faces (3 images per subject).

Some sample images of the CVL database can be seen in Figure B.4.

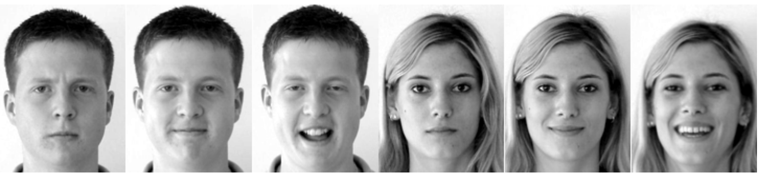


Figure B.4: Image samples from CVL.

B.5 FERET Database

The FERET database is one of the most widely adopted databases for benchmarking face recognition algorithms. It contains 3365 grayscale frontal facial images of almost 1000 different subjects.

Along with the face images, FERET also provides a face recognition evaluation protocol. In FERET, all frontal faces are divided into five categories: *fa*, *fb*, *fc*, *dup1*, and *dup2*. The pictures in *fa* and *fb* were taken on the same day, with the same camera and illumination condition. The pictures in *fc* were taken the same day that *fa* and *fb*, but using a different camera and changing illumination conditions. The categories *dup1* and *dup2* correspond to images of the same subjects, acquired several months later. In the case of *dup1* pictures were taken within the same year than *fa*. In *dup2*, the pictures were taken at least one year later than *fa* pictures.

In the FERET tests, 1196 *fa* pictures are used as the gallery set, while the categories *fb* (1195 images), *fc* (194 images), *dup1* (722 images), and *dup2* (234 images) constitute different probe sets. The gallery set contains only one image per person (i.e. each image defines a class label). Also, there is a set with 736 training samples.

In the *Face Identification Evaluation System* proposed by [17], the cumulative match curve is used in the FERET tests to compare the performance of different algorithms.

Some sample images of the FERET database, classified by their category subset, can be seen in Figure B.5.

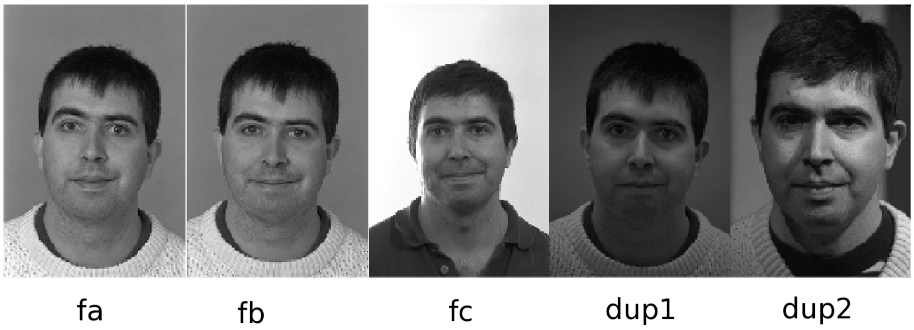


Figure B.5: Image samples from FERET.

B.6 FRGCv2 Database

The Face Recognition Grand Challenge (FRGCv2) face database [94] arose from the necessity of reducing the error rate in face recognition systems by an order of magnitude regarding older achievements liker the ones obtained with the Face Recognition Vendor Test (FRVT) 2002.

Three aspects of the FRGC dataset has projected it to be a benchmark for researchers: first is the size of the FRGC in terms of data. The FRGC data set contains up to $50,000$ recordings from around $4,000$ subjects; second is its complexity. FRGC consists of three modalities of samples: high resolution still images, 3D images, and multi-images of a person; third is the infrastructure. The infrastructure for FRGC is provided by the Biometric Experimentation Environment (BEE), which allows the description and distribution of experiments in a common format.

The FRGC distribution was organized into six experiments. In experiment 1, the gallery consists of a single controlled still image of a person and each probe consists of a single controlled still image. Experiment 2 studies the effect of using multiple still images of a person on performance. Experiment 3 measures the performance of 3D face recognition. Experiment 4 measures recognition performance from uncontrolled images. In experiment 4, the gallery consists of a single controlled still image, and the probe set consists of a single uncontrolled still image. Experiments 5 and 6 examine comparing 3D and 2D images.

Related to the current work, the most interesting subset of samples is Experiment 4. As it is explained in [94], the images belonging to this subset are considered the most challenging of the FRGCv2 database, due to the mix of controlled and uncontrolled conditions of acquisition, although all of them are indoor images. Experiment 4 has a total of $12,776$ training images, $16,028$ target images and $8,014$ query images.

Also, for Experiment 4, the Biometric Experimentation Environment (BEE) defines three possible selections of datasets, which give place to three different sets ROC curves during the evaluation: *ROC I*, *ROC II* and *ROC III*. Each of these variants corresponds different target–query subsets, defined in [94]. Specifically, in ROC I, all the data are within semesters, in ROC II, they are within a year, while in ROC III, the samples are between semesters. These experiments are of of increasing difficulty.

Some sample images of FRGCv2 can be seen in Figure B.6.

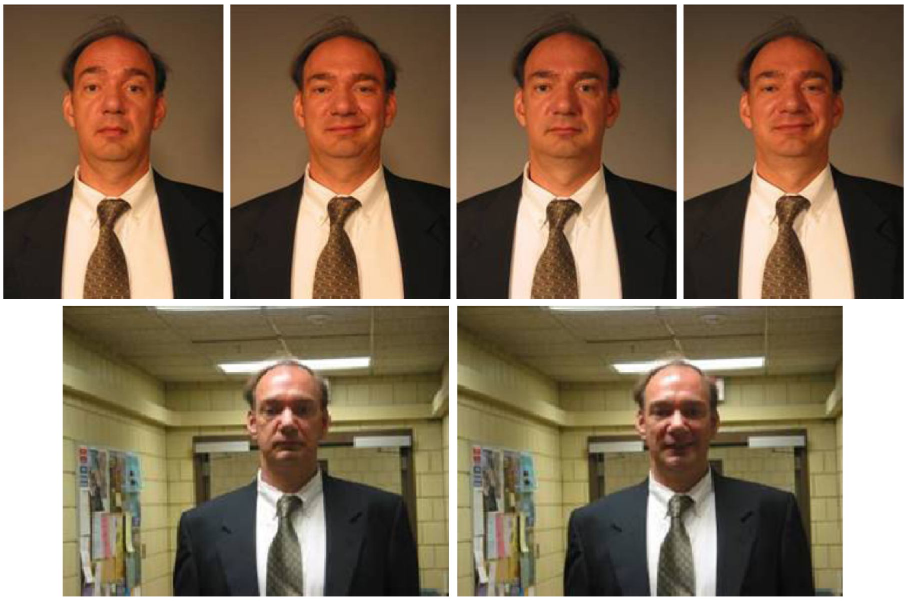


Figure B.6: Image samples from FRGCv2.

List of Figures

List of Tables

Bibliography

[1] National Institute of Standards and Technology. http://www.nist.gov.

[2] Yale database. Technical report, Yale University, http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[3] *Feature extraction from faces using deformable templates*, 1989.

[4] *CSIFT: A SIFT Descriptor with Color Invariant Characteristics*, volume 2, 2006.

[5] E Acosta, L Torres, A Albiol, and E Delp. An automatic face detection and recognition system for video indexing applications. *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 4:IV–3644– IV–3647, 2002.

[6] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 469–481. Springer Berlin / Heidelberg, 2004.

[7] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol. Face recognition using hog-ebgm. *Pattern Recognition Letters*, 29(10):1537–1543, July 2008.

[8] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *The Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974.

[9] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal on Machine Learning Research*, 3:1–48, March 2003.

[10] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computing*, 12:2385–2404, October 2000.

[11] S. Behnke. Face localization and tracking in the neural abstraction pyramid. *Neural Computer Applications*, 14(2):97–103, 2005.

[12] P.N. Belhumeur, J.P. Hespanh, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Proceedings of the 4th European Conference on Computer Vision*, pages 45–58, Cambridge, UK, April 1996.

[13] G. Van Belle, P. De Graef, K. Verfaillie, T. Busigny, and B. Rossion. Whole not hole: Expert face recognition requires holistic perception. *Neuropsychologia*, 48(9):2620–2629, 2010.

[14] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of SIFT features for face authentication. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, page 35, New York, June 2006.

[15] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, September 2003.

[16] W. W. Bledsoe. The model method in facial recognition. Technical report, Panoramic Res. Inc., 1966.

[17] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B.A. Draper. The CSU face identification evaluation system: Its purpose, features, and structure. In *Proceedings of the International Conference on Computer Vision Systems*, pages 304–313, Graz, Austria, February 2003.

[18] LLC Books. *Film Techniques: Film Editing, Sound Effect, Tracking Shot, Shot Reverse Shot, Establishing Shot, L Cut, Point of View Shot, Medium Shot.* General Books LLC, May 2010.

[19] A. Bosch, A. Zisserman, and X. Munoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4), 2008.

[20] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1):188–197, January 2007.

[21] E. M. Bronstein, M. M. Bronstein, A. Spira, and R. Kimmel. Face recognition from facial surface metric. In *Proceedings of the ECCV*, pages 225–237. Springer, 2004.

[22] R. Brunelli and T. Poggio. Face recognition: Features vs. templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[23] A. M. Burton, S. Wilson, M. Cowan, and V. Bruce. Face recognition in poor-quality video: Evidence from security surveillance. *Psychological Science*, 10(3):243–248, 1999.

[24] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *Proceedings of the BMVC*, pages 187–196, 2006.

[25] P. Campadelli, R. Lanzarotti, and G. Lipori. Eye localization: A survey. *The fundamentals of verbal and nonverbal communication and their biometrical issues. NATO Science Series*, 18:234–245, 2007.

[26] S. Carey, R. Diamond, and B. Woods. Development of face recognition: A maturational component? *Developmental Psychology*, 16(4):257–269, 1980.

[27] T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 227 –232, 2000.

[28] T.F. Cootes, C.J. Taylor, and Manchester M Pt. Statistical models of appearance for computer vision, report, 2000.

[29] C. Cortes and Vl. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[30] T.F. Cotes, G. J. Edwards, and C. J. Taylor. Active appareance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[31] T. Cover and P. Hart. Nearest neighbor pattern recognition. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[32] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Journal of Pattern Recognition*, (10):3054–3067, 2008.

[33] George R. Cross and Anil K. Jain. Markov random field texture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(1):25 –39, January 1983.

[34] John G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics Speech and Signal Processing*, 36(7):1169–1179, 1988.

[35] M. Davis, S. Popa, and C. Surlea. *Real-Time Face Recognition from Surveillance Video*, chapter 9. Number 332 in Studies in Computational Intelligence. Springer-Verlag, Berlin, 2010.

[36] Imola K Fodor. A survey of dimension reduction techniques. *Library*, 18(1):1–18, 2002.

[37] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[38] V. Govindaraju, D.B. Sher, R. K. Srihari, and S. N. Srihari. Locating human faces in newspaper photographs. *CVPR*, 1989.

[39] Y. Guan. Robust eye detection from facial image based on multi-cue facial information. *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 1775–1778, June 2007.

[40] M. Hamouz, J. Kittler, J. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas. Feature-based affine-invariant localization of faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(9):1490–1495, 2005.

[41] J. A. Hanley and B. J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, September 1983.

[42] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786 – 804, May 1979.

[43] Trevor Hastie, Andreas Buja, and Robert Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, 3(1):73–102, 1995.

[44] R. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:696–706, 2002.

[45] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[46] R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of lda. In *Proceedings of the 16 th International Conference on Pattern Recognition*, volume 3, pages 29–32. IEEE Computer Society, 2002.

[47] Humanscan. BioID database. http://www.bioid.com.

[48] Leo M. Hurvich and Dorothea Jameson. An opponent-process theory of color vision. *Psychological Review*, 64(6):384–404, 1957.

[49] K. Irie, A.E. McKinnon, K. Unsworth, and I.M. Woodhead. A technique for evaluation of CCD Video-Camera noise. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(2):280 –284, February 2008.

[50] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing systems*, 31(3):264–323, September 1999.

[51] O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 90–95, London, UK, 2001. Springer-Verlag.

[52] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia. A hybrid classifier for precise and robust eye detection. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 731–735, 2006.

[53] R. Joseph. *Neuroscience: Neuropsychology, Neuropsychiatry, Behavioral Neurology, Brain and Mind.* University Press Science Publishers, 4th edition edition, dec. 2011.

[54] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[55] I. Kemelmacher and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):394–405, February 2011.

[56] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 1137–1143, 1995.

[57] T. Kohonen. *Self-Organization and Associative Memory.* Springer Verlag, 1984.

[58] B. Kroon, A. Hanjalic, and S. M. P. Maas. Eye localization for face matching: is it always useful and under what conditions? In *CIVR*, pages 379–388, 2008.

[59] F. De la Torre, R. Gross, S. Baker, and B. V. Kumar. Representational oriented component analysis (roca) for face recognition with one sample image per training class. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:266–273, 2005.

[60] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R. P. WÃ$\frac{1}{4}$rtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.

[61] R. Lanzarotti. *Facial feature detection and description.* PhD thesis, UniversitÃ degli Studi di Milano, Milano, Italia, 2003.

[62] X. Li, L. Wang, and E. Sung. Improving adaboost for classification on small training sample sets with active learning. In *The Sixth Asian Conference on Computer Vision (ACCV), Korea*, 2004.

[63] C. Liu and H. Wechsler. Comparative assessment of independent component analysis (ica) for face recognition. In *International Conference on Audio and Video Based Biometric Person Authentication*, pages 22–24, 1999.

[64] Chengjun Liu. Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. *IEEE TPAMI*, 28:725–737, 2006.

[65] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of Computer Vision*, 60(2):91–110, November 2004.

[66] D. G. Lowe. Perceptual organization an visual recognition. 1986.

[67] H. Lu, W. Zhang, and D. Yang. Eye detection based on rectangle features and pixel-pattern-based texture features. *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 746–749, December 2007.

[68] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[69] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:339, 2004.

[70] Christopher D. Manning and Heinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, 1999.

[71] S. Marcel, C. McCool, P. MatÄ›jka, T. Ahonen, J. ÄŒernockÃ½, S. Chakraborty, V. Balasubramanian, S. Panchanathan, C. Chano, J. Kittler, N. Poh, B. Fauve, O. Glembek, O. Plchot, Z. Jancik, A. Larcher, C. LÃ©vye, D. Matrouf, J. F. Bonastre, P. H. Lee, J. Y. Hung, S. W. Wu, Y. P. Hung, L. Machlica, J. Mason, S. Mau, C. Sanderson, Monzo D, A. Albiol, H. Nguyen, L. Bai, Y. Wang, M. Niskanen, M. Turtinen, J. A. Nolazco-Flores, L. P. Garcia-Perera, R. Aceves-Lopez, M. Villegas, and R. Paredes. On the results of the first mobile biometry (mobio) face and speaker verification evaluation. In Devrim Ãœnay, Zehra Ã‡ataltepe, and Selim Aksoy, editors, *Recognizing Patterns in Signals, Speech, Images and Videos*, volume 6388 of *Lecture Notes in Computer Science*, pages 210–225. Springer Berlin Heidelberg, 2010.

[72] J. D. Markel, B. T. Oshika, and A. H. Gray. Long-term feature averaging for speech recognition. *IEEE Transactions on Acoustic and Speech Signal Processing*, 25:1304–1312, 1977.

[73] Neurotechnologija, Biometrical and Artificial Intelligence Technologies. Verilook SDK. http://www.neurotechnologija.com.

[74] OmniPerception Technlogies. Affinity SDK. http://www.omniperception.com/products/affinity-sdk.

[75] S. McKenna and S. Gong. Recognising moving faces. In *Face Recognition: From Theory to Applications*, NATO ASI Series F. Springer-Verlag, 1998.

[76] P. Menezes, J. C. Barreto, and J. Dias. Face tracking based on haar-like features and eigenfaces. In *IN IAV2004*, pages 5–7, 2004.

[77] K. Messer, J. Kittler, M. Sadeghi, M. Hamouz, A. Kostin, S. Marcel, S. Bengio, Cardinaux, C. Sanderson, N. Poh, Y. Rodriguez, K. Kryszczuk, J. Czyz, L. Vandendorpe, J. Ng, H. Cheung, and B. Tang. Face authentication competition on the banca database. In *Proceedings of International Conference for Pattern Recognition*, volume 4, pages 8–15, 2004.

[78] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, pages 504–513. Springer-Verlag, 2008.

[79] B. Moghaddam, J. Lee, H. Pfister, and M. Raghu. Model-based 3d face capture with shape-from-silhouettes. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 20–27, October 2003.

[80] D. Monzo, A. Albiol, A. Albiol, and J. M. Mossi. A comparative study of facial landmark localization methods for face recognition using hog descriptors. In *Proceedings of the International Conference on Pattern Recognition*, Istanbul, August 2010.

[81] D. Monzo, A. Albiol, A. Albiol, and J. M. Mossi. Color HOG-EBGM for face recognition. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 785–788. IEEE, Sep. 2011.

[82] D. Monzo, A. Albiol, J. Sastre, and A. Albiol. Hog-ebgm vs. gabor-ebgm. In *IEEE Internacional Conference on Image Processing (ICIP)*, pages 1636–1639, San Diego, CA, USA, October 2008.

[83] D. Monzo, A. Albiol, J. Sastre, and A. Albiol. Precise eye localization using hog descriptors. *Machine Vision and Applications*, pages 1–10, 2010.

[84] Ch Morimoto, D Koons, A Amir, and M Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, 18(4):331–335, 2000.

[85] A. V. Nefian. Statistical approaches to face recognition. Technical report, 1996.

[86] A. V. Nefian and M. H. Hayes. Hidden markov models for face recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 2721–2724, May 1998.

[87] Son H. Nguyen and Andrzej Skowron. Quantization of real value attributes - rough set and boolean reasoning approach. In *Proc. of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, Sept 28 - Oct 1*, pages 34–37, 1995.

[88] Massachusetts Institute of Technology Work Group in the Biology of Language and E. Walker. *Explorations in the biology of language.* Series in higher mental processes. Bradford Books, 1978.

[89] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[90] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 555 –562, January 1998.

[91] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[92] P. Peer. CVL Face database,University of Ljubjana. http://www.fri.uni-lj.si/en.

[93] J. P. Phillips, H. Moon, S. Rizv, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[94] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, C. Jin, K. Hoffman, J. Marques, M. Jaesik, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 947–954, San Diego, CA, June 2005.

[95] W. K. Pratt. *Digital Image Processing: PIKS Inside.* John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2001.

[96] M. Rahman and N. Kehtarnavaz. Real-time face-priority auto focus for digital and cell-phone cameras. *Consumer Electronics, IEEE Transactions on*, 54(4):1506–1513, November 2008.

[97] E. Rentzeperis, A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. Impact of face registration errors on recognition. pages 187–194, 2006.

[98] Yann Rodriguez, Fabien Cardinaux, Samy Bengio, and Johnny Mariéthoz. Measuring the performance of face localization systems. *Image and Vision Computing*, 24(8):882–893, 2006.

[99] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halstead Press, New York, 1992.

[100] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang. Curse of mis-alignment in face recognition: Problem and a novel mis-alignment learning solution. In *FGR*, pages 314–320, 2004.

[101] L. Silva, O. R. P. Bellon, and K. L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):762–776, May 2005.

[102] Gita Sukthankar. Face recognition: A critical look at biologically-inspired approaches. Technical report, Robotics Institute, Pittsburgh, PA, January 2000.

[103] X. Tang, Z. Ou, T. Su, H. Sun, and P. Zhao. Robust precise eye location by adaboost and svm techniques. In *Lecture Notes in Computer Science*, pages 93–98. Springer Berlin / Heidelberg, 2005.

[104] D. W. Thompson and J. I. Mundy. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 208–220, Raleigh, NC, 1987.

[105] L. Torres. Is there any hope for face recognition? In *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Portugal, 2004.

[106] M. Tuceryan and A. K. Jain. *Texture Analysis*, volume 304, pages 1–41. World Scientific Publishing Co., 1998.

[107] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[108] S. Ullman. Cognition. 32:193–254, 1989.

[109] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[110] Joost van de Weijer, Theo Gevers, and Andrew D. Bagdanov. Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:150–156, 2006.

[111] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[112] M. Viola, M. J. Jones, and P. Viola. Fast multi-view face detection. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.

[113] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, Hawaii, December 2001.

[114] P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 3, pages 164–171, San Diego, CA, June 2005.

[115] S. Wang, X. Xiong, Y. Xu, C. Wang, W. Zhang, X. Dai, and D. Zhang. Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1097–1106, New York, NY, USA, 2006. ACM Press.

[116] L. K. Westin. *Receiver operating characteristic (ROC) analysis.* Department of Computing Science in Umea University, Umea, 2001.

[117] L. Wiskott, J. M. Fellous, N. Kruger, and C. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1996.

[118] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems).* Morgan Kaufmann, June 2005.

[119] J. D. Woodward. *Biometrics: a look at facial recognition.* RAND, Santa Monica, Calif :, 2003.

[120] Jian Yang and Chengjun Liu. A discriminant color space method for face representation and verification on a large-scale database. In *ICPR*, pages 1–4, 2008.

[121] M. Yang and N. Ahuja. Detecting human faces in color images. pages 127–130, 1998.

[122] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *The Journal of Machine Learning Research*, 6:483–502, December 2005.

[123] Andrew Yip and Pawan Sinha. Contribution of color to face recognition. *Perception*, 31(5):995–1003, 2002.

[124] H. B. Yu and S. Yu. A novel image preprocessing scheme based on face detection. In *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, volume 2, pages 1021–1026, 2003.

[125] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research, 2010.

[126] L. Zhang, D. Samaras, D. Tomasi, N. Volkow, and R. Goldstein. Machine learning for clinical diagnosis from functional magnetic resonance imaging, 2005.

[127] S. Zhang and T. Sim. Discriminant subspace analysis: A fukunaga-koontz approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1732 –1745, October 2007.

[128] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.

[129] X. Zhao, E. Dellandrea, and L. Chen. A people counting system based on face detection and tracking in a video. In *6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 67–72, 2009.

[130] Zhi-Hua Zhou and Xin Geng. Projection functions for eye detection. *Pattern Recognition*, 37(5):1049–1056, 2004.