Face Recognition Algorithms

Proyecto Fin de Carrera

June 16, 2010

Ion Marqués

Supervisor:
Manuel Graña

eman ta zabal zazu

Universidad   Euskal Herriko
del País Vasco   Unibertsitatea

Acknowledgements

I would like to thank Manuel Graña Romay for his guidance and support. He encouraged me to write this proyecto fin de carrera. I am also grateful to the family and friends for putting up with me.

II

Contents

List of Figures

List of Tables

Abstract

The goal of this "proyecto fin de carrera" was to produce a review of the face detection and face recognition literature as comprehensive as possible. Face detection was included as a unavoidable preprocessing step for face recogntion, and as an issue by itself, because it presents its own difficulties and challenges, sometimes quite different from face recognition. We have soon recognized that the amount of published information is unmanageable for a short term effort, such as required of a PFC, so in agreement with the supervisor we have stopped at a reasonable time, having reviewed most conventional face detection and face recognition approaches, leaving advanced issues, such as video face recognition or expression invariances, for the future work in the framework of a doctoral research. I have tried to gather much of the mathematical foundations of the approaches reviewed aiming for a self contained work, which is, of course, rather difficult to produce. My supervisor encouraged me to follow formalism as close as possible, preparing this PFC report more like an academic report than an engineering project report.

Chapter 1

The Face Recognition Problem

Contents

1.1   Development through history

Face recognition is one of the most relevant applications of image analysis. It's a true challenge to build an automated system which equals human ability to recognize faces. Although humans are quite good identifying known faces, we are not very skilled when we must deal with a large amount of unknown faces. The computers, with an almost limitless memory and computational speed, should overcome humans limitations.

Face recognition remains as an unsolved problem and a demanded technology - see table 1.1. A simple search with the phrase "face recognition" in the IEEE Digital Library throws 9422 results. 1332 articles in only one year - 2009. There are many different industry areas interested in what it could offer. Some examples include video surveillance, human-machine interaction, photo cameras, virtual reality or law enforcement. This multidisciplinary interest pushes the research and attracts interest from diverse disciplines. Therefore, it's not a problem restricted to computer vision research. Face recognition is a relevant subject in pattern recognition, neural networks, computer graphics, image processing and psychology [125]. In fact, the earliest works on this subject were made in the 1950's in psychology [21]. They came attached to other issues like face expression, interpretation of emotion or perception of gestures.

Engineering started to show interest in face recognition in the 1960's. One of the first researches on this subject was Woodrow W. Bledsoe. In 1960, Bledsoe, along other researches, started Panoramic Research, Inc., in Palo Alto, California. The majority of the work done by this company involved AI-related contracts from the U.S. Department of Defense and various intelligence agencies [4]. During 1964 and 1965, Bledsoe, along with Helen Chan and Charles Bisson, worked on using computers to recognize human faces [14, 18, 15, 16, 17]. Because the funding of these researches was provided by an unnamed intelligence agency, little of the work was published. He continued later his researches at Stanford Research Institute [17]. Bledsoe designed and implemented a semi-automatic system. Some face coordinates were selected by a human operator, and then computers used this information for recognition. He described most of the problems that even 50 years later Face Recognition still suffers - variations in illumination, head rotation, facial expression, aging. Researches on this matter still continue, trying to measure subjective face features as ear size or between-eye distance. For instance, this approach was used in Bell Laboratories by A. Jay Goldstein, Leon D. Harmon and Ann B. Lesk [35]. They described a vector, containing 21 subjective features like ear protrusion, eyebrow weight or nose length, as the basis to recognize faces using pattern classification techniques. In 1973,

Fischler and Elschanger tried to measure similar features automatically [34]. Their algorithm used local template matching and a global measure of fit to find and measure facial features.

There were other approaches back on the 1970's. Some tried to define a face as a set of geometric parameters and then perform some pattern recognition based on those parameters. But the first one that developed a fully automated face recognition system was Kenade in 1973 [54]. He designed and implemented a face recognition program. It ran in a computer system designed for this purpose. The algorithm extracted sixteen facial parameters automatically. In he's work, Kenade compares this automated extraction to a human or manual extraction, showing only a small difference. He got a correct identification rate of 45-75%. He demonstrated that better results were obtained when irrelevant features were not used.

I the 1980's there were a diversity of approaches actively followed, most of them continuing with previous tendencies. Some works tried to improve the methods used measuring subjective features. For instance, Mark Nixon presented a geometric measurement for eye spacing [85]. The template matching approach was improved with strategies such as "deformable templates" . This decade also brought new approaches. Some researchers build face recognition algorithms using artificial neural networks [105].

The first mention to eigenfaces in image processing, a technique that would become the dominant approach in following years, was made by L. Sirovich and M. Kirby in 1986 [101]. Their methods were based on the Principal Component Analysis. Their goal was to represent an image in a lower dimension without losing much information, and then reconstructing it [56]. Their work would be later the foundation of the proposal of many new face recognition algorithms.

The 1990's saw the broad recognition of the mentioned eigenface approach as the basis for the state of the art and the first industrial applications. In 1992 Mathew Turk and Alex Pentland of the MIT presented a work which used eigenfaces for recognition [110]. Their algorithm was able to locate, track and classify a subject's head. Since the 1990's, face recognition area has received a lot of attention, with a noticeable increase in the number of publications. Many approaches have been taken which has lead to different algorithms. Some of the most relevant are PCA, ICA, LDA and their derivatives. Different approaches and algorithms will be discussed later in this work.

The technologies using face recognition techniques have also evolved through the years. The first companies to invest in such researches where law enforcement agencies - e.g. the Woodrow W. Bledsoe case. Nowadays diverse enterprises are using face recognition in their products. One good example

could be entertainment business. Products like Microsoft's Project Natal [31] or Sony's PlayStation Eye [75] will use face recognition. It will allow a new way to interact with the machine. The idea of detecting people and analyzing their gesture is also being used in automotive industry. Companies such as Toyota are developing sleep detectors to increase safety [74]. These and other applications are raising the interest on face recognition. It's narrow initial application area is being widened.

Areas	Applications
Information Security	Access security (OS, data bases)
	Data privacy (e.g. medical records)
	User authentication (trading, on line banking)
Access management	Secure access authentication (restricted facilities)
	Permission based systems
	Access log or audit trails
Biometrics	Person identification (national IDs, Passports,
	voter registrations, driver licenses)
	Automated identity verification (border controls)
Law Enforcement	Video surveillance
	Suspect identification
	Suspect tracking (investigation)
	Simulated aging
	Forensic Reconstruction of faces from remains
Personal security	Home video surveillance systems
	Expression interpretation (driver monitoring system)
Entertainment - Leisure	Home video game systems
	Photo camera applications

Table 1.1: Applications of face recognition.

1.2 Psychological inspiration in automated face recognition

The extended title of this chapter could be "The relevance of some features used in the human natural face recognition cognitive processes to the automated face recognition algorithms". In other words, to what extent are biologically relevant elements useful for artificial face recognition system design?

1.2.1 Psychology and Neurology in face recognition

Many researches tried to understand how humans recognize faces, most of them when the automatic face recognition problem arose, looking for design inspiration. It seems important to understand how we do this task, how we perceive humans [21]. Then this knowledge could be applied in automatic face recognition systems. However, many algorithms don't use this information, using just mathematical tools. Through these year some question have emerged: Are features relevant to our eyes important for automatic face recognition? Can human vision system teach us useful thinks in this regard? Could psychological studies enlight this problem in some way? In short, can the human face recognition ability help to develop a non-human face recognition system? This section will try to answer some relevant questions.

Is face recognition a dedicated process in the brain?

One early paper that answered this question was published by Diamond and Carey back in 1986 [29]. They presented four experiments. They tried to know if the difficulty of recognizing inverted faces was also common in other class of stimuli. At the same time, they tried to isolate the cause of this difficulty. They concluded that faces were no unique in the sense of being represented in memory in terms of special features. This may suggested that, consequently, face recognition has not a special spot in brain. This theory can be supported by the fact that patients with prosopagnosis -a neurological condition in which it's very hard to recognize familiar faces- had also difficulties recognizing other familiar pictures.

More recent studies demonstrated that face recognition is a dedicated process in our brains [7]. They demonstrated that recognizing human faces throw a negative ERP (event-related potential), N170. They also found that it reflects the activity of cells turned to exclusively recognize human faces or face components. The same was true for inverted pictures. They suggested that there is a special process in our brains, and a special part of it, dedicated to recognize human faces.

This question remains unanswered and it is still a much debated issue . The dedication of the fusiform face area (FFA) as a face processing module seems to be very strong. However, it may be responsible for performing subordinate or expert-level categorization of generic objects [100]. We can conclude that there is a huge possibility that humans have a specialized face recognition mechanism.

Are face and expression recognition separated systems?

It could be interesting to know if humans can extract facial expression independently from the identity of the subject and vice versa. Is facial expression an important constraint or condition in face recognition? Thus, can a biological implementation of a computerized face recognition system identify faces in spite of facial expression? Many studies propose that identity and expression processes separate early in the facial perception procedure [100]. Whether face recognition algorithm designers can find this information useful or not, that its another matter.

Is color an important factor in face recognition?

Many face recognition algorithms don't use color as a feature. However, it could be interesting to know if color play a key role in human face recognition process. How objects are stored in the brain is a subject of much debate. Moreover, it isn't known if color cues play an important role in object recognition or not.

    It is widely accepted that color cues do not provide diagnostic information for recognition, but they are not completely unrelated to face recognition systems. They could be nearly irrelevant when we try to recognize chromatically similar objects. On the other hand, it has been demonstrated that their contribution is essential under degraded conditions [120]. So, color cues play an important role especially when shape cues are degraded. This feature could be extrapolated to face recognition system design.

Does symmetry play an important role in face recognition?

From both neurological and computational point of view the answer is the same: yes. It has been demonstrated that an exceptional dimension reduction can be made by taking into account facial symmetry [102]. The cited study also concludes that there are less than 70 dimensions for human recognition system. This result is smaller than the previously proposed $\simeq$100 dimensions. The cause is the relevance of human face similarity.

1.2.2   Recognition algorithm design points of view

The most evident face features were used in the beginning of face recognition. It was a sensible approach to mimic human face recognition ability. There was an effort to try to measure the importance of certain intuitive features [20](mouth, eyes, cheeks) and geometric measures (between-eye distance [85], width-length ratio). Nowadays is still an relevant issue, mostly

because discarding certain facial features or parts of a face can lead to a better performance [24]. In other words, it's crucial to decide which facial features contribute to a good recognition and which ones are no better than added noise.

However, the introduction of abstract mathematical tools like eigenfaces [101, 56] created another approach to face recognition. It was possible to compute the similarities between faces obviating those human-relevant features. This new point of view enabled a new abstraction level, leaving the anthropocentric approach behind.

There are still some human-relevant features that are being taken into account. For example, skin color [99, 33] is an important feature for face detection. The location of certain features like mouth or eyes is also used to perform a normalization prior to the feature extraction step [125].

To sum up, a designer can apply to the algorithms the knowledge that psychology, neurology or simple observation provide. On the other hand, it's essential to perform abstractions and attack the problem from a pure mathematical or computational point of view.

1.3 Face recognition system structure

Face Recognition is a term that includes several sub-problems. There are different classifications of these problems in the bibliography. Some of them will be explained on this section. Finally, a general or unified classification will be proposed.

1.3.1 A generic face recognition system

The input of a face recognition system is always an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video. Some approaches [125] define a face recognition system as a three step process - see Figure 1.1. From this point of view, the Face Detection and Feature Extraction phases could run simultaneously.
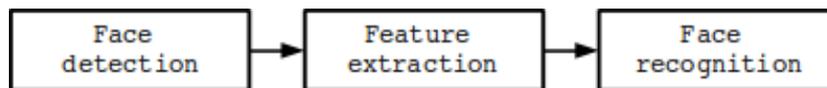
Figure 1.1: A generic face recognition system.

Face detection is defined as the process of extracting faces from scenes. So, the system positively identifies a certain image region as a face. This

procedure has many applications like face tracking, pose estimation or com-
pression. The next step -feature extraction- involves obtaining relevant facial
features from the data. These features could be certain face regions, varia-
tions, angles or measures, which can be human relevant (e.g. eyes spacing) or
not. This phase has other applications like facial feature tracking or emotion
recognition. Finally, the system does recognize the face. In an identifica-
tion task, the system would report an identity from a database. This phase
involves a comparison method, a classification algorithm and an accuracy
measure. This phase uses methods common to many other areas which also
do some classification process -sound engineering, data mining et al.

These phases can be merged, or new ones could be added. Therefore,
we could find many different engineering approaches to a face recognition
problem. Face detection and recognition could be performed in tandem, or
proceed to an expression analysis before normalizing the face [109].

1.4   Face detection

Nowadays some applications of Face Recognition don't require face detection.
In some cases, face images stored in the data bases are already normalized.
There is a standard image input format, so there is no need for a detection
step. An example of this could be a criminal data base. There, the law
enforcement agency stores faces of people with a criminal report. If there is
new subject and the police has his or her passport photograph, face detection
is not necessary. However, the conventional input image of computer vision
systems are not that suitable. They can contain many items or faces. In these
cases face detection is mandatory. It's also unavoidable if we want to develop
an automated face tracking system. For example, video surveillance systems
try to include face detection, tracking and recognizing. So, it's reasonable to
assume face detection as part of the more ample face recognition problem.

Face detection must deal with several well known challenges[117, 125].
They are usually present in images captured in uncontrolled environments,
such as surveillance video systems. These challenges can be attributed to
some factors:

- Pose variation. The ideal scenario for face detection would be one in
  which only frontal images were involved. But, as stated, this is very
  unlikely in general uncontrolled conditions. Moreover, the performance
  of face detection algorithms drops severely when there are large pose
  variations. It's a major research issue. Pose variation can happen due
  to subject's movements or camera's angle.

- Feature occlusion. The presence of elements like beards, glasses or hats introduces high variability. Faces can also be partially covered by objects or other faces.

- Facial expression. Facial features also vary greatly because of different facial gestures.

- Imaging conditions. Different cameras and ambiental conditions can affect the quality of an image, affecting the appearance of a face.

There are some problems closely related to face detection besides feature extraction and face classification. For instance, face location is a simplified approach of face detection. It's goal is to determine the location of a face in an image where there's only one face. We can differenciate between face detection and face location, since the latter is a simplified poblem of the former. Methods like locating head boundaries [59] were first used on this scenario and then exported to more complicated problems. Facial feature detection concerns detecting and locating some relelvant features, such as nose, eyebrow, lips, ears, etc. Some feature extraction algorithms are based on facial feature detection. There is much literature on this topic, which is discused later. Face tracking is other problem which sometimes is a consequence of face detection. Many system's goal is not only to detect a face, but to be able to locate this face in real time. Once again, video surveillance system is a good example.

1.4.1 Face detection problem structure

Face Detection is a concept that includes many sub-problems. Some systems detect and locate faces at the same time, others first perform a detection routine and then, if positive, they try to locate the face. Then, some tracking algorithms may be needed - see Figure 1.2.
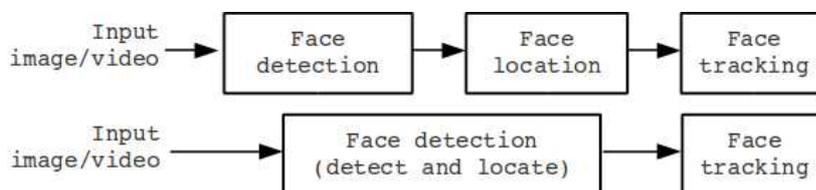
Figure 1.2: Face detection processes.

Face detection algorithms ussually share common steps. Firstly, some data dimension reduction is done, in order to achieve a admissible response

time. Some pre-processing could also be done to adapt the input image to the algorithm prerequisites. Then, some algorithms analize the image as it is, and some others try to extract certain relevant facial regions. The next phase usually involves extracting facial features or measurements. These will then be weighted, evaluated or compared to decide if there is a face and where is it. Finally, some algorithms have a learning routine and they include new data to their models.

Face detection is, therefore, a two class problem where we have to decide if there is a face or not in a picture. This approach can be seen as a simplified face recognition problem. Face recognition has to classify a given face, and there are as many classes as candidates. Consequently, many face detection methods are very similar to face recognition algorithms. Or put another way, techniques used in face detection are often used in face recognition.

1.4.2   Approaches to face detection

It's not easy to give a taxonomy of face detection methods. There isn't a globally accepted grouping criteria. They usually mix and overlap. In this section, two classification criteria will be presented. One of them differentiates between distinct scenarios. Depending on these scenarios different approaches may be needed. The other criteria divides the detection algorithms into four categories.

Detection depending on the scenario.

- Controlled environment. It's the most straightforward case. Photographs are taken under controlled light, background, etc. Simple edge detection techniques can be used to detect faces [70].

- Color images. The typical skin colors can be used to find faces. They can be weak if light conditions change. Moreover, human skin color changes a lot, from nearly white to almost black. But, several studies show that the major difference lies between their intensity, so chrominance is a good feature [117]. It's not easy to establish a solid human skin color representation. However, there are attempts to build robust face detection algorithms based on skin color [99].

- Images in motion. Real time video gives the chance to use motion detection to localize faces. Nowadays, most commercial systems must locate faces in videos. There is a continuing challenge to achieve the best detecting results with the best possible performance [82]. Another

approach based on motion is eye blink detection, which has many uses aside from face detection [30, 53].

Detection methods divided into categories

Yan, Kriegman and Ahuja presented a classifications that is well accepted [117]. Methods are divided into four categories. These categories may overlap, so an algorithm could belong to two or more categories. This classification can be made as follows:

- Knowledge-based methods. Ruled-based methods that encode our knowledge of human faces.

- Feature-invariant methods. Algorithms that try to find invariant features of a face despite it's angle or position.

- Template matching methods. These algorithms compare input images with stored patterns of faces or features.

- Appearance-based methods. A template matching method whose pattern database is learnt from a set of training images.

Let us examine them on detail:

Knowledge-based methods.

These are rule-based methods. They try to capture our knowledge of faces, and translate them into a set of rules. It's easy to guess some simple rules. For example, a face usually has two symmetric eyes, and the eye area is darker than the cheeks. Facial features could be the distance between eyes or the color intensity difference between the eye area and the lower zone. The big problem with these methods is the difficulty in building an appropriate set of rules. There could be many false positives if the rules were too general. On the other hand, there could be many false negatives if the rules were too detailed. A solution is to build hierarchical knowledge-based methods to overcome these problems. However, this approach alone is very limited. It's unable to find many faces in a complex image.

Other researches have tried to find some invariant features for face detection. The idea is to overcome the limits of our instinctive knowledge of faces. One early algorithm was developed by Han, Liao, Yu and Chen in 1997 [41]. The method is divided in several steps. Firstly, it tries to find eye-analogue pixels, so it removes unwanted pixels from the image. After performing the

segmentation process, they consider each eye-analogue segment as a candidate of one of the eyes. Then, a set of rule is executed to determinate the potential pair of eyes. Once the eyes are selected, the algorithms calculates the face area as a rectangle. The four vertexes of the face are determined by a set of functions. So, the potential faces are normalized to a fixed size and orientation. Then, the face regions are verificated using a back propagation neural network. Finally, they apply a cost function to make the final selection. They report a success rate of 94%, even in photographs with many faces. These methods show themselves efficient with simple inputs. But, what happens if a man is wearing glasses?

There are other features that can deal with that problem. For example, there are algorithms that detect face-like textures or the color of human skin. It is very important to choose the best color model to detect faces. Some recent researches use more than one color model. For example, RGB and HSV are used together successfully [112]. In that paper, the authors chose the following parameters

$$0.4 \leq r \leq 0.6, 0.22 \leq g \leq 0.33, r > g > (1-r)/2 \qquad (1.1)$$

$$0 \leq H \leq 0.2, 0.3 \leq S \leq 0.7, 0.22 \leq V \leq 0.8 \qquad (1.2)$$

Both conditions are used to detect skin color pixels. However, these methods alone are usually not enough to build a good face detection algorithm. Skin color can vary significantly if light conditions change. Therefore, skin color detection is used in combination with other methods, like local symmetry or structure and geometry.

Template matching

Template matching methods try to define a face as a function. We try to find a standard template of all the faces. Different features can be defined independently. For example, a face can be divided into eyes, face contour, nose and mouth. Also a face model can be built by edges. But these methods are limited to faces that are frontal and unoccluded. A face can also be represented as a silhouette. Other templates use the relation between face regions in terms of brightness and darkness. These standard patterns are compared to the input images to detect faces. This approach is simple to implement, but it's inadequate for face detection. It cannot achieve good results with variations in pose, scale and shape. However, deformable templates have been proposed to deal with these problems.

Appearance-based methods

The templates in appearance-based methods are learned from the examples in the images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. Some appearance-based methods work in a probabilistic network. An image or feature vector is a random variable with some probability of belonging to a face or not. Another approach is to to define a discriminant function between face and non-face classes. These methods are also used in feature extraction for face recognition and will be discussed later. Nevertheless, these are the most relevant methods or tools:

- Eigenface-based. Sirovich and Kirby [101, 56] developed a method for efficiently representing faces using PCA (Principal Component Analysis). Their goal of this approach is to represent a face as a coordinate system. The vectors that make up this coordinate system were referred to as eigenpictures. Later, Turk and Pentland used this approach to develop a eigenface-based algorithm for recognition [110].

- Distribution-based. These systems where first proposed for object and pattern detection by Sung [106]. The idea is collect a sufficiently large number of sample views for the pattern class we wish to detect, covering all possible sources of image variation we wish to handle. Then an appropriate feature space is chosen. It must represent the pattern class as a distribution of all its permissible image appearances. The system matches the candidate picture against the distribution-based canonical face model. Finally, there is a trained classifier which correctly identifies instances of the target pattern class from background image patterns, based on a set of distance measurements between the input pattern and the distribution-based class representation in the chosen feature space. Algorithms like PCA or Fisher's Discriminant can be used to define the subspace representing facial patterns.

- Neural Networks. Many pattern recognition problems like object recognition, character recognition, etc. have been faced successfully by neural networks. These systems can be used in face detection in different ways. Some early researches used neural networks to learn the face and non-face patterns [93]. They defined the detection problem as a two-class problem. The real challenge was to represent the "images not containing faces" class. Other approach is to use neural networks to find a discriminant function to classify patterns using distance measures [106]. Some approaches have tried to find an optimal boundary between face and non-face pictures using a constrained generative model [91].

- Support Vector Machines. SVMs are linear classifiers that maximize the margin between the decision hyperplane and the examples in the training set. So, an optimal hyperplane should minimize the classification error of the unseen test patterns. This classifier was first applied to face detection by Osuna et al. [87].

- Sparse Network of Winnows. SNoWs were first used for detection by Yang et al. [118]. They defined a sparse network of two linear units or target nodes, one representing face patterns and the other for the non-face patterns. The SNoW had a incrementally learned feature space. New labeled cases served as positive example for one target and as a negative example for the remaining target. The system proved to be effective at the time, and less time consuming.

- Naive Bayes Classifiers. Schneiderman and Kanade described an object recognition algorithm that modeled and estimated a Bayesian Classifier [96]. They computed the probability of a face to be present in the picture by counting the frequency of occurrence of a series of patterns over the training images. They emphasized on patterns like the intensity around the eyes. The classifier captured the joint statistics of local appearance and position of the face as well as the statistics of local appearance and position in the visual world. Overall, their algorithm showed good results on frontal face detection. Bayes Classifiers have also been used as a complementary part of other detection algorithms.

- Hidden Markov Model. This statistical model has been used for face detection. The challenge is to build a proper HMM, so that the output probability can be trusted. The states of the model would be the facial features, which are often defined as strips of pixels. The probabilistic transition between states are usually the boundaries between these pixel strips. As in the case of Bayesians, HMMs are commonly used along with other methods to build detection algorithms.

- Information-Theoretical Approach. Markov Random Fields (MRF) can be used to model contextual constraints of a face pattern and correlated features. The Markov process maximizes the discrimination between classes (an image has a face or not) using the Kullback–Leibler divergence. Therefore, this method can be applied in face detection.

- Inductive Learning. This approach has been used to detect faces. Algorithms like Quinlan's C4.5 or Mitchell's FIND-S have been used for this purpose [32, 45].

1.4.3   Face tracking

Many face recognition systems have a video sequence as the input. Those systems may require to be capable of not only detecting but tracking faces. Face tracking is essentially a motion estimation problem. Face tracking can be performed using many different methods, e.g., head tracking, feature tracking, image-based tracking, model-based tracking. These are different ways to classify these algorithms [125]:

- Head tracking/Individual feature tracking. The head can be tracked as a whole entity, or certain features tracked individually.

- 2D/3D. Two dimensional systems track a face and output an image space where the face is located. Three dimensional systems, on the other hand, perform a 3D modeling of the face. This approach allows to estimate pose or orientation variations.

The basic face tracking process seeks to locate a given image in a picture. Then, it has to compute the differences between frames to update the location of the face. There are many issues that must be faced: Partial occlusions, illumination changes, computational speed and facial deformations.

One example of a face tracking algorithm can be the one proposed by Baek et al. in [33]. The state vector of a face includes the center position, size of the rectangle containing the face, the average color of the face area and their first derivatives. The new candidate faces are evaluated by a Kalman estimator. In tracking mode, if the face is not new, the face from the previous frame is used as a template. The position of the face is evaluated by the Kalman estimator and the face region is searched around by a SSD algorithm using the mentioned template. When SSD finds the region, the color information is embeded into the Kalman estimator to exactly confine the face region. Then, the state vector of that face is updated. The result showed robust when some faces overlapped or when color changes happened.

1.5   Feature Extraction

Humans can recognize faces since we are 5 year old. It seems to be an automated and dedicated process in our brains [7], though it's a much debated issue [29, 21]. What it's clear is that we can recognize people we know, even when they are wearing glasses or hats. We can also recognize men who have grown a beard. It's not very difficult for us to see our grandma's wedding photo and recognize her, although she was 23 years old. All these processes seem trivial, but they represent a challenge to the computers. In fact, face
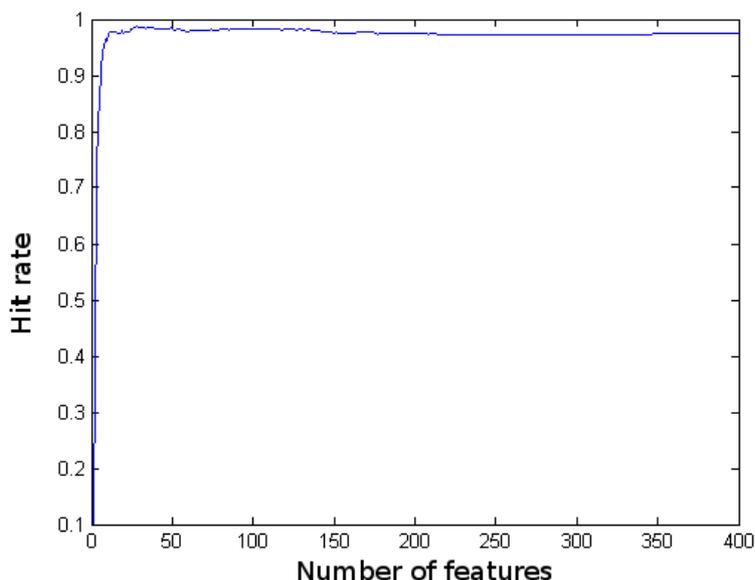
Figure 1.3: PCA algorithm performance

recognition's core problem is to extract information from photographs. This feature extraction process can be defined as the procedure of extracting relevant information from a face image. This information must be valuable to the later step of identifying the subject with an acceptable error rate. The feature extraction process must be efficient in terms of computing time and memory usage. The output should also be optimized for the classification step.

Feature extraction involves several steps - dimensionality reduction, feature extraction and feature selection. This steps may overlap, and dimensionality reduction could be seen as a consequence of the feature extraction and selection algorithms. Both algorithms could also be defined as cases of dimensionality reduction.

Dimensionality reduction is an essential task in any pattern recognition system. The performance of a classifier depends on the amount of sample images, number of features and classifier complexity. One could think that the false positive ratio of a classifier does not increase as the number of features increases. However, added features may degrade the performance of a classification algorithm - see Figure 1.3. This may happen when the number of training samples is small relative to the number the features.

This problem is called "curse of dimensionality" or "peaking phenomenon". A generally accepted method of avoiding this phenomenon is to use at least

ten times as many training samples per class as the number of features. This
requirement should be satisfied when building a classifier. The more complex
the classifier, the larger should be the mentioned ratio [48]. This "curse" is
one of the reasons why it's important to keep the number of features as small
as possible. The other main reason is the speed. The classifier will be faster
and will use less memory. Moreover, a large set of features can result in a
false positive when these features are redundant. Ultimately, the number of
features must be carefully chosen. Too less or redundant features can lead
to a loss of accuracy of the recognition system.

We can make a distinction between feature extraction and feature selec-
tion. Both terms are usually used interchangeably. Nevertheless, it is rec-
ommendable to make a distinction. A feature extraction algorithm extracts
features from the data. It creates those new features based on transforma-
tions or combinations of the original data. In other words, it transforms
or combines the data in order to select a proper subspace in the original
feature space. On the other hand, a feature selection algorithm selects the
best subset of the input feature set. It discards non-relevant features. Fea-
ture selection is often performed after feature extraction. So, features are
extracted from the face images, then a optimum subset of these features is
selected. The dimensionality reduction process can be embedded in some of
these steps, or performed before them. This is arguably the most broadly
accepted feature extraction process approach - see figure 1.4.
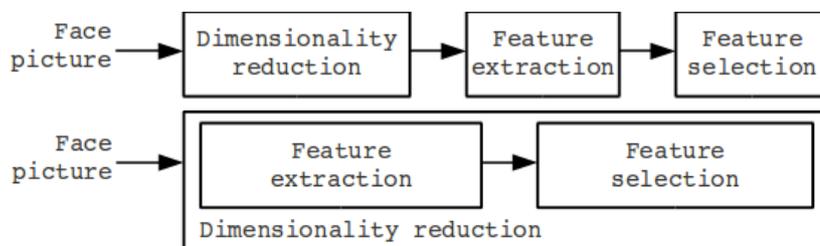
Figure 1.4: Feature extraction processes.

1.5.1   Feature extraction methods

There are many feature extraction algorithms. They will be discussed later
on this paper. Most of them are used in other areas than face recognition.
Researchers in face recognition have used, modified and adapted many algo-
rithms and methods to their purpose. For example, PCA was invented by
Karl Pearson in 1901[88], but proposed for pattern recognition 64 years later
[113]. Finally, it was applied to face representation and recognition in the

early 90's [101, 56, 110]. See table 1.2 for a list of some feature extraction algorithms used in face recognition

Method	Notes
Principal Component Analysis (PCA)	Eigenvector-based, linear map
Kernel PCA	Eigenvector-based , non-linear map, uses kernel methods
Weighted PCA	PCA using weighted coefficients
Linear Discriminant Analysis (LDA)	Eigenvector-based, supervised linear map
Kernel LDA	LDA-based, uses kernel methods
Semi-supervised Discriminant Analysis (SDA)	Semi-supervised adaptation of LDA
Independent Component Analysis (ICA)	Linear map, separates non-Gaussian distributed features
Neural Network based methods	Diverse neural networks using PCA, etc.
Multidimensional Scaling (MDS)	Nonlinear map, sample size limited, noise sensitive.
Self-organizing map (SOM)	Nonlinear, based on a grid of neurons in the feature space
Active Shape Models (ASM)	Statistical method, searches boundaries
Active Appearance Models (AAM)	Evolution of ASM, uses shape and texture
Gavor wavelet transforms	Biologically motivated, linear filter
Discrete Cosine Transform (DCT)	Linear function, Fourier-related transform, usually used 2D-DCT
MMSD, SMSD	Methods using maximum scatter difference criterion.

Table 1.2: Feature extraction algorithms

1.5.2 Feature selection methods

Feature selection algorithm's aim is to select a subset of the extracted features that cause the smallest classification error. The importance of this error is what makes feature selection dependent to the classification method used. The most straightforward approach to this problem would be to examine every possible subset and choose the one that fulfills the criterion function. However, this can become an unaffordable task in terms of computational time. Some effective approaches to this problem are based on algorithms like branch and bound algorithms. See table 1.3 for selection methods proposed in [48].

Method	Definition	Comments
Exhaustive search	Evaluate all possible subsets of features.	Optimal, but too complex.
Branch and bound	Use branch and bound algorithm.	Can be optimal. Complexity of max $O(2^n)$.
Best individual features	Evaluate and select features individually.	Not very effective. Simple algorithm.
Sequential Forward Selection (SFS)	Evaluate growing feature sets (starts with best feature).	Retained features can't be discarded. Faster than SBS.
Sequential Backward Selection (SBS)	Evaluate shrinking feature sets (starts with all the features).	Deleted features can't be reevaluated.
"Plus l-take away r" selection	First do SFS then SBS.	Must choose l and r values.
Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS)	Like "Plus l-take away r", but l and r values automatic pick and dynamic update.	Close to optimal. Affordable computational cost.

Table 1.3: Feature selection methods

Recently more feature selection algorithms have been proposed. Feature selection is a NP-hard problem, so researchers make an afford towards a satisfactory algorithm, rather than an optimum one. The idea is to create an algorithm that selects the most satisfying feature subset, minimizing the dimensionality and complexity. Some approaches have used resemblance coefficient [121] or satisfactory rate [122]as a criterion and quantum genetic algorithm (QGA).

1.6 Face classification

Once the features are extracted and selected, the next step is to classify the image. Appearance-based face recognition algorithms use a wide variety of classification methods. Sometimes two or more classifiers are combined to achieve better results. On the other hand, most model-based algorithms match the samples with the model or template. Then, a learning method is can be used to improve the algorithm. One way or another, classifiers have a big impact in face recognition. Classification methods are used in many areas like data mining, finance, signal decoding, voice recognition, natural language processing or medicine. Therefore, there is many bibliography regarding this

subject. Here classifiers will be addressed from a general pattern recognition point of view.

Classification algorithms usually involve some learning - supervised, unsupervised or semi-supervised. Unsupervised learning is the most difficult approach, as there are no tagged examples. However, many face recognition applications include a tagged set of subjects. Consequently, most face recognition systems implement supervised learning methods. There are also cases where the labeled data set is small. Sometimes, the acquisition of new tagged samples can be infeasible. Therefore, semi-supervised learning is required.

1.6.1   Classifiers

According to Jain, Duin and Mao [48], there are three concepts that are key in building a classifier - similarity, probability and decision boundaries. We will present the classifiers from that point of view.

Similarity

This approach is intuitive and simple. Patterns that are similar should belong to the same class. This approach have been used in the face recognition algorithms implemented later. The idea is to establish a metric that defines similarity and a representation of the same-class samples. For example, the metric can be the euclidean distance. The representation of a class can be the mean vector of all the patterns belonging to this class. The 1-NN decision rule can be used with this parameters. It's classification performance is usually good. This approach is similar to a k-means clustering algorithm in unsupervised learning. There are other techniques that can be used. For example, Vector Quantization, Learning Vector Quantization or Self-Organizing Maps - see 1.4. Other example of this approach is template matching. Researches classify face recognition algorithm based on different criteria. Some publications defined Template Matching as a kind or category of face recognition algorithms [20]. However, we can see template matching just as another classification method, where unlabeled samples are compared to stored patterns.

Probability

Some classifiers are build based on a probabilistic approach. Bayes decision rule is often used. The rule can be modified to take into account different factors that could lead to miss-classification. Bayesian decision rules can give

Method	Notes
Template matching	Assign sample to most similar template. Templates must be normalized.
Nearest Mean	Assign pattern to nearest class mean.
Subspace Method	Assign pattern to nearest class subspace.
1-NN	Assign pattern to nearest pattern's class
k-NN	Like 1-NN, but assign to the majority of k nearest patterns.
(Learning) Vector Quantization methods	Assign pattern to nearest centroid. There are various learning methods.
Self-Organizing Maps (SOM)	Assign pattern to nearest node, then update nodes pulling them closer to input pattern

Table 1.4: Similarity-based classifiers

an optimal classifier, and the Bayes error can be the best criterion to evaluate features. Therefore, a posteriori probability functions can be optimal.

There are different Bayesian approaches. One is to define a Maximum A Posteriori (MAP) decision rule [66]:

$$p(Z|w_i)P(w_i) = max\left\{p(Z|w_j)P(w_j)\right\} Z \in w_i \qquad (1.3)$$

where w_i are the face classes and Z an image in a reduced PCA space. Then, the within class densities (probability density functions) must be modeled, which is

$$p(Z|w_i) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma_i|^{\frac{1}{2}}}exp\left\{-\frac{1}{2}(Z-M_i)^t\sum_i^{-1}(Z-M_i)\right\} \qquad (1.4)$$

where Σ_i and M_i are the covariance and mean matrices of class w_i. The covariance matrices are identical and diagonal, getting the components by sample variance in the PCA subspace. Other option is to get the within class covariance matrix by diagonalizing the within class scatter matrix using singular value decomposition (SVD).

There are other approaches to Bayesian classifiers. Moghaddam et al. proposed on [78] an alternative to the MAP - the maximum likelihood (ML). They proposed a non-euclidean measure similarity measure, and two classes of facial image variations: Differences between images from the same individual (Ω_I, interpersonal) and variations between different individuals (Ω_E). They define the a ML similarity matching as

$$S' = p(\Delta|w_i) = \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_i|^{\frac{1}{2}}} exp \left\{ -\frac{1}{2} \|i_j - i_k\|^2 \right\} \qquad (1.5)$$

where Δ is the difference vector between to samples, i_j and i_k are images stored as a vector with *whitened* subspace interpersonal coefficients. The idea is to pre-process these images offline, so the algorithm is much faster when performs the face recognition.

This algorithms estimate the densities instead of using the true densities. These density estimates can be either parametric or nonparametric. Commonly used parametric models in face recognition are multivariate Gaussian distributions, as in [78]. Two well-known non-parametric estimates are the k-NN rule and the Parzen classifier. They both have one parameter to be set, the number of neighbors k, or the smoothing parameter (bandwidth) of the Parzen kernel, both of which can be optimized. Moreover, both these classifiers require the computation of the distances between a test pattern and all the samples in the training set. These large numbers of computations can be avoided by vector quantization techniques, branch-and-bound and other techniques. A summary of classifiers with a probabilistic approach can be seen in table 1.5.

Method	Notes
Bayesian	Assign pattern to the class with the highest estimated posterior probability.
Logistic Classifier	Predicts probability using logistic curve method.
Parzen Classifier	Bayesian classifier with Parzen density estimates.

Table 1.5: Probabilistic classifiers

Decision boundaries

This approach can become equivalent to a Bayesian classifier. It depends on the chosen metric. The main idea behind this approach is to minimize a criterion (a measurement of error) between the candidate pattern and the testing patterns. One example is the Fisher's Linear Discriminant (often FLD and LDA are used interchangeably). It's closely related to PCA. FLD attempts to model the difference between the classes of data, and can be used to minimize the mean square error or the mean absolute error. Other algorithms use neural networks. Multilayer perceptron is one of them. They allow non-linear decision boundaries. However, neural networks can be trained in many different ways, so they can lead to diverse classifiers. They can also provide

a confidence in classification, which can give an approximation of the posterior probabilities. Assuming the use of an euclidean distance criterion, the classifier could make use of the three classification concepts here explained.

A special type of classifier is the decision tree. It is trained by an iterative selection of individual features that are most salient at each node of the tree. During classification, just the needed features for classification are evaluated, so feature selection is implicitly built-in. The decision boundary is built iteratively. There are well known decision trees like the C4.5 or CART available. See table 1.6 for some decision boundary-based methods, including the ones proposed in [48]:

Method	Notes
Fisher Linear Discriminant (FLD)	Linear classifier. Can use MSE optimization
Binary Decision Tree	Nodes are features. Can use FLD. Could need pruning.
Perceptron	Iterative optimization of a classifier (e.g. FLD)
Multi-layer Perceptron	Two or more layers. Uses sigmoid transfer functions.
Radial Basis Network	Optimization of a Multi-layer perceptron. One layer at least uses Gaussian transfer functions.
Support Vector Machines	Maximizes margin between two classes.

Table 1.6: Classifiers using decision boundaries

Other method widely used is the support vector classifier. It is a two-class classifier, although it has been expanded to be multiclass. The optimization criterion is the width of the margin between the classes, which is the distance between the hyperplane and the support vectors. These support vectors define the classification function. Support Vector Machines (SVM) are originally two-class classifiers. That's why there must be a method that allows solving multiclass problems. There are two main strategies [43]:

1. On-vs-all approach. A SVM per class is trained. Each one separates a single class from the others.

2. Pairwise approach. Each SVM separates two classes. A bottom-up decision tree can be used, each tree node representing a SVM [39]. The coming face's class will appear on top of the tree.

Other problem is how to face non-linear decision boundaries. A solution is to map the samples to a high-dimensional feature space using a kernel function [49].

1.6.2   Classifier combination

The classifier combination problem can be defined as a problem of finding the combination function accepting N-dimensional score vectors from M-classifiers and outputting N final classification scores [108]. There can be several reasons to combine classifiers in face recognition:

- The designer has some classifiers, each developed with a different approach. For example, there can be a classifier designed to recognize faces using eyebrow templates. We could combine it with another classifier that uses other recognition approach. This could lead to a better recognition performance.

- There can be different training sets, collected in different conditions and representing different features. Each training set could be well suited for a certain classifier. Those classifiers could be combined.

- One single training set can show different results when using different classifiers. A combination of classifiers can be used to achieve the best results.

- Some classifiers differ on their performance depending on certain initializations. Instead of choosing one classifier, we can combine some of them.

There are different combination schemes. They may differ from each other in their architectures and the selection of the combiner. Combiner in pattern recognition usually use a fixed amount of classifiers. This allows to take advantage of the strengths of each classifier. The common approach is to design certain function that weights each classifier's output "score". Then, there must be a decision boundary to take a decision based on that function. Combination methods can also be grouped based on the stage at which they operate. A combiner could operate at feature level. The features of all classifiers are combined to form a new feature vector. Then a new classification is made. The other option is to operate at score level, as stated before. This approach separates the classification knowledge and the combiner. This type of combiners are popular due to that abstraction level. However, combiners can be different depending on the nature of the classifier's output. The output can be a simple class or group of classes (abstract information level). Other more exact output can be an ordered list of candidate classes (rank level). A classifier could have a more informative output by including some weight or confidence measure to each class (measurement level). If the combination involves very specialized classifiers, each of them usually has a

different output. Combining different output scales and confidence measures can be a tricky problem. However, they will have a similar output if all the classifiers use the same architecture.

Combiners can be grouped in three categories according to their architecture:

- Parallel. All classifiers are executed independently. The combiner is then applied.

- Serial. Classifiers run one after another. Each classifier polishes previous results.

- Hierarchical. Classifiers are combined into a tree-like structure.

Combiner functions can be very simple ore complex. A low complexity combination could require only one function to be trained, whose input is the scores of a single class. The highest complexity can be achieved by defining multiple functions, one for each class. They take as parameters all scores. So, more information is used for combination. Higher complexity classifiers can potentially produce better results. The complexity level is limited by the amount of training samples and the computing time. Thus it's very important to choose a complexity level that best complies these requirements and restrictions. Some combiners can also be trainable. The trainable combiners can lead to better results at the cost of requiring additional training data. See table 1.7 for a list of combination schemes proposed in [108] and [48].

1.7 Face recognition: Different approaches

Face recognition is an evolving area, changing and improving constantly. Many research areas affect face recognition - computer vision, optics, pattern recognition, neural networks, machine learning, psycology, etcetera. Previous sections explain the different steps of a face recognition process. However, these steps can overlap or change depending on the bibliography consulted. There is not a consensus on that regard. All these factors hinder the development of a unified face recognition algorithm classification scheme. This section explains the most cited criteria.

1.7.1 Geometric/Template Based approaches

Face recognition algorithms can be classified as either geometry based or template based algorithms [107, 38]. The template based methods compare the input image with a set of templates. The set of templates can be costructed

Scheme	Architecture	Trainable	Info-level
Voting	Parallel	No	Abstract
Sum, mean, median	Parallel	No	Confidence
Product, min, max	Parallel	No	Confidence
Generalized ensemble	Parallel	Yes	Confidence
Adaptive weighting	Parallel	Yes	Confidence
Stacking	Parallel	Yes	Confidence
Borda count	Parallel	Yes	Rank
Behavior Knowledge Space	Parallel	Yes	Abstract
Logistic regression	Parallel	Yes	Rank
Class set reduction	Parallel/Cascading	Yes	Rank
Dempster-Shafer rules	Parallel	Yes	Rank
Fuzzy integrals	Parallel	Yes	Confidence
Mixture of Local Experts	Parallel	Yes	Confidence
Hierarchical MLE	Hierarchical	Yes	Confidence
Associative switch	Parallel	Yes	Abstract
Random subspace	Parallel	Yes	Confidence
Bagging	Parallel	Yes	Confidence
Boosting	Hierarchical	Yes	Abstract
Neural tree	Hierarchical	Yes	Confidence

Table 1.7: Classifiers combination schemes

using statistical tools like Suppor Vector Machines (SVM) [39, 49, 43], Principal Component Analysis (PCA) [80, 109, 110], Linear Discriminant Ananlysis (LDA) [6], Independent Component Analysis (ICA) [5, 65, 67], Kernel Methods [127, 3, 97, 116], or Trace Transforms [51, 104, 103].

The geometry feature-based methods analyze local facial features and their geometric relationshps. This approach is sometimes collad featurebased approach [20]. Examples of this approach are some Elastic Bunch Graph Matching algorithms [114, 115]. This apporach is less used nowadays [107]. There are algorithms developed using both approaches. For instance, a 3D morphable model approach can use feature points or texture as well as PCA to build a recognition system [13].

1.7.2 Piecemeal/Wholistic approaches

Faces can often be identified from little information. Some algorithms follow this idea, processing facila features independently. In other words, the relation between the features or the relation of a feature with the whole face is

not taken into account. Many early researchers followed this approach, trying to deduce the most relevant features. Some approaches tried to use the eyes [85], a combination of features [20], and so on. Some Hidden Markov Model (HMM) methods also fall in thuis category [84].Although feature processing is very imporant in face recognition, relation between features (configural processing) is also important. In fact, facial features are processed wholistically [100]. That's why nowadays most algorithms follow a holistic approach.

1.7.3   Appeareance-based/Model-based approaches

Facial recoginition methods can be divided into appeareance-based or model-based algorithms. The diferential element of these methods is the representation of the face. Appeareance-based methods represent a face in terms of several raw intensity images. An image is considered as a high-dimensional vector. Then satistical techiques are usually used to derive a feature space from the image distribution. The sample image is compared to the training set. On the other hand, the model-based approach tries to model a human face. The new sample is fitted to the model, and the parametres of the fitten model used to recognize the image. Appeareance methods can be classified as linear or non-linear, while model-based methods can be 2D or 3D [72].

Linear appeareance-based methods perform a linear dimension reduction. The face vectors are projected to the basis vectors, the projection coefficients are used as the feature representation of each face image. Examples of this approach are PCA, LDA or ICA. Non-linear appeareance methods are more complicate. In fact, linear subspace analysis is an approximation of a non-linear manifold. KernelPCA (KPCA) is a method widely used [77].

Model-based approaches can be 2-Dimensional or 3-Dimensional. These algorithms try to build a model of a human face. These models are often morphable. A morphable model allows to classify faces even when pose changes are present. 3D models are more complicate, as they try to capture the three dimensional nature of human faces. Examples of this approach are Elastic Bunch Graph Matching [114] or 3D Morphable Models [13, 12, 46, 62, 79].

1.7.4   Template/statistical/neural network approaches

A similar separation of pattern recognition algorithms into four groups is proposed by Jain and colleges in [48]. We can grop face recognition methods into three main groups. The following approaches are proposed:

- Template matching. Patterns are represented by samples, models, pix-
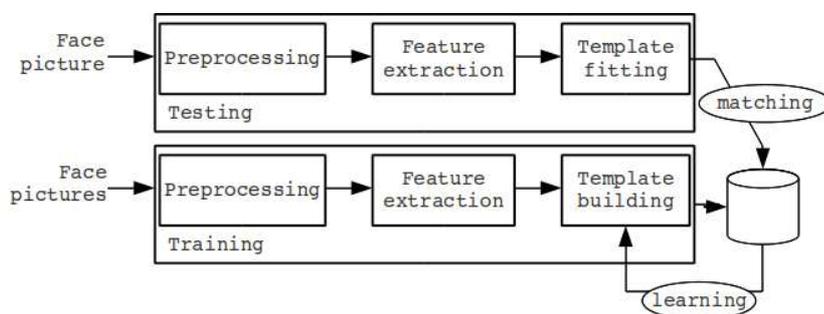
Figure 1.5: Template-matching algorithm diagram

els, curves, textures. The recognition function is usually a correlation or distance measure.

- Statistical approach. Patterns are represented as features. The recognition function is a discriminant function.

- Neural networks. The representation may vary. There is a network function in some point.

Note that many algorithms, mostly current complex algorithms, may fall into more than one of these categories. The most relevant face recognition algorithms will be discussed later under this classification.

1.8 Template matching face recognition methods

Many face recognition algorithms include some template matching techniques. A template matching process uses pixels, samples, models or textures as pattern. The recognition function computes the differences between these features and the stored templates. It uses correlation or distance measures. Although the matching of 2D images was the early trend, nowadays 3D templates are more common. The 2D approaches are very sensitive to orientation or illumination changes. One way of addressing this problem is using Elastic Bunch Graphs to represent images [114]. Each subject has a bunch graph for each of it's possible poses. Facial features are extracted from the test image to form an image graph. This image graph can be compared to the model graphs, matching the right class.

The introduction of 3D models is motivated by the potential ability of three dimensional patterns to be unaffected by those two factors. The problem is that 3D data should be acquired doing 3D scans, under controlled

conditions. Moreover, in most cases requires the collaboration of the subject to be recognized. Therefore, in applications such as surveillance systems, this kind of 3D data may not be available during the recognition process. This is why there is tendency to build training sets using 3D models, but gathering 2D images for recognition. Techniques that construct 3D models from 2D data are being developed in this context.

Blanz and Vetter state in [13] that there are different ways of separating shape and orientation of a face in 3D models: To match feature vertexes to image positions and then interpolate deformations of the surface or to use restricted class-specific deformations, defined manually or automatically, from non textured or textured head scans. Separation between texture and illumination is achieved using models of illumination that consider illumination direction and intensity from Lambertian or non-Lambertian reflectance. The initial constraint of systems like the cited one is that the database of faces is obtained via 3D scans. So, there a need of building a solid 3D model database. Other constraint is that it requires to manually define some feature points. The recognition process is done by building a 3D model of the subject. Then, this 3D model is compared with the stored patterns using two parameters -shape and texture. Their algorithm achieves a performance of around 95.5%. Therefore, 3D models are a powerful representation of human faces for recognition purposes. They have huge potential towards pose and illumination invariant face recognition.

This solid representation of faces has been used in other algorithms for recognition purposes. However, most current algorithms take advantage of statistical tools like PCA [19], computational models and classifiers. Although pure sample-model matching systems are not viable, face templates are a tool widely used in face recognition.

1.8.1 Example: Adaptative Appeareance Models

One interesting tool is the Adaptative Appearance Model [26]. Its goal is to minimize the difference between the model and the input image, by varying the model parameters, c.

There is a parameter controlling shape, $x = \bar{x} + Q_s c$, where $\bar{x}$ is the mean shape and Q_s is the matrix defining the variation possibilities. The transformation function S_t is typically described by a scaling, $(s \cos \theta - 1, s \sin \theta)$, an in-plane rotation θ and a translation (t_x, t_y). The pose parameter vector $t = (s \cos \theta - 1, s \sin \theta, t_x, t_y)^T$ is then zero for an identity transformation and $S_{t+\delta t}(x) \approx S_t(S_{\delta t}(x))$.

There is also a texture parameter $g = \bar{g} + Q_g c$, where $\bar{g}$ is the mean texture in a mean shaped path and Q_g is the matrix describing the modes of variation.

The texture in the image is defined as $g_{im} = T_u(g) = (u_1 + 1)g_{im} + u_2$, where u is the transformation parameter vector. It is zero for an identity transformation and $T_{u+\delta u}(x) \approx T_u(T_{\delta u}(x))$.

The parameters c and t define the position of the model points. During matching we sample the pixels and project into the texture model frame. The current difference between the model and the image (measured in the normalized texture frame) is

$$r(p) = T^{-1}(g_{im}) - (\bar{g} + Q_g c) \tag{1.6}$$

where p are the parameters of the model, $p^T = (c^T | t^T | u^T)$. Then, we can perform expansions and derivation in order to minimize the differences between models and input images. It is interesting to precompute all the parameters available, so that the following searches can be quicker.

1.9 Statistical approach for recognition algorithms

Images of faces, represented as high-dimensional pixel arrays, often belong to a manifold of lower dimension. In statistical approach, each image is represented in terms of d features. So, it's viewed as a point (vector) in a d-dimensional space. The dimensionality -number of coordinates needed to specify a data point- of this data is too high. Therefore, the goal is to choose and apply the right statistical tool for extraction and analysis of the underlaying manifold. These tools must define the embedded face space in the image space and extract the basis functions from the face space. This would permit patterns belonging to different classes to occupy disjoint and compacted regions in the feature space. Consequently, we would be able to define a line, curve, plane or hyperplane that separates faces belonging to different classes.

Many of these statistical tools are not used alone. They are modified or extended by researchers in order to get better results. Some of them are embedded into bigger systems, or they are just a part of a recognition algorithm. Many of them can be found along classification methods like a DCT embedded in a Bayesian Network [83] or a Gabor Wavelet used with a Fuzzy Multilayer Perceptron [9].
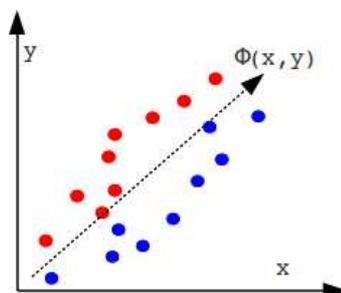
Figure 1.6: PCA. x and y are the original basis. ϕ is the first principal component.

1.9.1 Principal Component Analysis

One of the most used and cited statistical method is the Principal Component Analysis (PCA) [101, 56, 110]. It is a mathematical procedure that performs a dimensionality reduction by extracting the *principal components* of the multi-dimensional data. The first principal component is the linear combination of the original dimensions that has the highest variability. The n-th principal component is the linear combination with the maximum variability, being orthogonal to the *n-1* first principal components. The idea of PCA is illustrated in figure 1.7.

The greatest variance of any projection of the data lies in the first coordinate. The n-st coordinate will be the direction of the n-th maximum variance - the n-th principal component.

Usually the mean $\overline{x}$ is extracted from the data, so that PCA is equivalent to Karhunen-Loeve Transform (KLT). So, let X_{nxm} be the the data matrix where $x_1, ..., x_m$ are the image vectors (vector columns) and n is the number of pixels per image. The KLT basis is obtained by solving the eigenvalue problem

$$C_x = \Phi \Lambda \Phi^T \tag{1.7}$$

where C_x is the covariance matrix of the data

$$C_x = \frac{1}{m} \sum_{i=1}^{m} x_i x_i^T \tag{1.8}$$

$\Phi = [\phi_1, ..., \phi_n]$ is the eigenvector matrix of C_x. Λ is a diagonal matrix, the eigenvalues $\lambda_1, ..., \lambda_n$ of C_x are located on its main diagonal. λ_i is the variance of the data projected on ϕ_i.

PCA can be computed using Singular Value Decomposition (SVD). The SVD of the data matrix X_{nxm} is

Figure 1.7: Face image and its DCT

$$X = UDV^T \tag{1.9}$$

It is known that $U = \Phi$. This method allows efficient implementation of PCA without having to compute the data covariance matrix C_x -knowing that $C_x = U^T X$. The embedding is done by $y_i = U^T x_i$, thus obtaining the mapped points $y_1, ..., y_m$.

1.9.2 Discrete Cosine Transform

The Discrete Cosine Transform [2] DCT-II standard (often called simply DCT) expresses a sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. It has strong energy compaction properties. Therefore, it can be used to transform images, compacting the variations, allowing an effective dimensionality reduction. They have been widely used for data compression. The DCT is based on the Fourier discrete transform, but using only real numbers.

When a DCT is performed over an image, the energy is compacted in the upper-left corner. An example can be found in image 1.8. The face has been taken from the ORL database[95], and a DCT performed over it.

Let B be the DCT of an input image A_{NxM}:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \tag{1.10}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, p=0 \\ \sqrt{2/M}, 1 \le p \le M-1 \end{cases} \quad \alpha_q = \begin{cases} 1/\sqrt{N}, q=0 \\ \sqrt{2/N}, 1 \le q \le N-1 \end{cases}$$

where M is the row size and N is the column size of A. We can truncate the matrix B, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem.

1.9.3 Linear Discriminant Analysis

LDA is widely used to find linear combinations of features while preserving class separability. Unlike PCA, LDA tries to model the differences between classes. Classic LDA is designed to take into account only two classes. Specifically, it requires data points for different classes to be far from each other, while point from the same class are close. Consequently, LDA obtains differenced projection vectors for each class. Multi-class LDA algorithms which can manage more than two classes are more used.

Suppose we have m samples $x_1,...,$x$_m$ belonging to c classes; each class has m_k elements. We assume that the mean has been extracted from the samples, as in PCA. The objective function of the LDA can be defined [22] as

$$a_{opt} = argmax \frac{a^T S_b a}{a^T S_t a} \tag{1.11}$$

$$S_b = \sum_{k=1}^{c} m_k \mu^{(k)} (\mu^{(k)})^T = \sum_{k=1}^{c} \left(\frac{1}{m_k} (\sum_{i=1}^{m_k} x_i^{(k)}) \right) \left(\frac{1}{l_k} (\sum_{i=1}^{m_k} x_i^{(k)}) \right)^T = X W_{mxm} X^T \tag{1.12}$$

$$S_t = \sum_{i=1}^{m} x_i (x_i)^T = X X^T \tag{1.13}$$

where W_{mxm} is a diagonal matrix defined as

$$W_{mxm} = \begin{bmatrix} W^1 & 0 & \dots & 0 \\ 0 & W^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W^c \end{bmatrix} \tag{1.14}$$

and W^k is a $m_k \times m_k$ matrix

$$W^k = \begin{bmatrix} \frac{1}{m_k} & \frac{1}{m_k} & \cdots & \frac{1}{m_k} \\ \frac{1}{m_k} & \frac{1}{m_k} & \cdots & \frac{1}{m_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_k} & \frac{1}{m_k} & \cdots & \frac{1}{m_k} \end{bmatrix} \quad (1.15)$$

Finally, we can write the eigenproblem:

$$S_b a = \lambda S_t a \rightarrow S_t^{-1} S_b a = \lambda a \rightarrow X W_{lxl} X^T (XX^T)^{-1} a = \lambda a \quad (1.16)$$

The solution of this eigenproblem provides the eigenvectors; the embedding is done like the PCA algorithms does.

1.9.4   Locality Preserving Projections

The Locality Preserving Projections (LPP) was introduced by He and Niyogi [42]. It's an alternative to PCA, designed to preserve locality structure. Pattern recognition algorithms usually make a search for the nearest pattern or neighbors. Therefore, the locality preserving quality of LPP can quicken the recognition.

Let m be the number of points that we want to map. In our case, those points correspond to images. The LPP algorithm has four steps:

- Constructing the adjacency map: A graph G with m nodes is built using, for example, k-NN algorithm.

- Choosing the weights: Being W_{ij} a weight matrix, we can build it using a Heat kernel of parameter t-if nodes i and j are connected, put

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}} \quad (1.17)$$

- Solving the eigenproblem. D is a diagonal matrix where it's elements are defined as $d_{ii} = \sum_j w_{ij}$, and $L=D$-W is the Laplacian matrix. The following eigenproblem must be solved:

$$\lambda a = XDX^T (XLX^T)^{-1} \quad (1.18)$$

The embedding process and the PCA's embedding process are analogous.
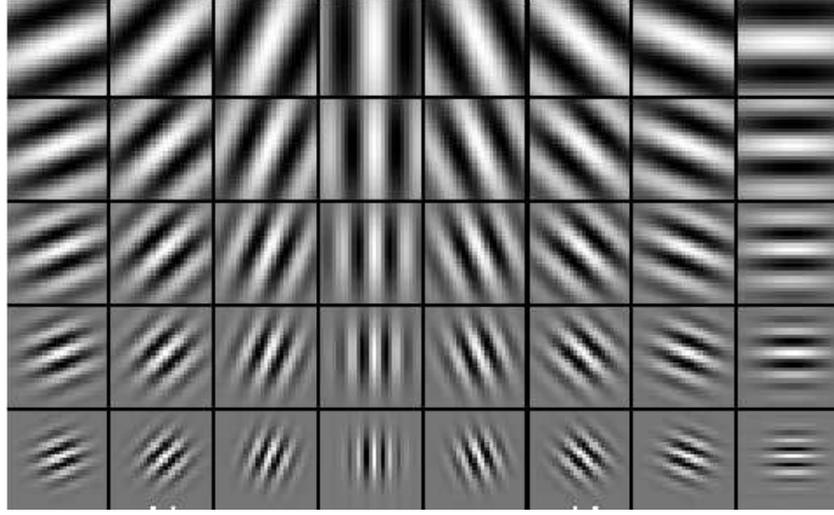
Figure 1.8: Gabor filters.

1.9.5 Gabor Wavelet

Neurophysiological evidence from the visual cortex of mammalian brains suggests that simple cells in the visual cortex can be viewed as a family of self-similar 2D Gabor wavelets. The Gabor functions proposed by Daugman [28, 63] are local spatial bandpass filters that achieve the theoretical limit for conjoint resolution of information in the 2D spatial and 2D Fourier domains. Daugman generalized the Gabor function to the following 2D form :

$$\Psi_i(\overrightarrow{x}) = \frac{\|\overrightarrow{k_i}\|^2}{\sigma^2} e^{\frac{\|\overrightarrow{k_j}\|^2 \|\overrightarrow{x}\|^2}{2\sigma^2}} \left[e^{j\overrightarrow{k_i}\overrightarrow{x}} - e^{-\frac{\sigma^2}{2}} \right] \tag{1.19}$$

Each Ψ_i is a plane wave characterized by the vector k_i enveloped by a Gaussian function, where σ is the standard deviation of this Gaussian. The center frequency of i-th filter is given by the characteristic wave vector,

$$\overrightarrow{k_i} = \begin{pmatrix} k_{ix} \\ k_{iy} \end{pmatrix} = \begin{pmatrix} k_v cos\theta_\alpha \\ k_v sin\theta_\alpha \end{pmatrix}; k_v = \pi 2^{-\frac{v+2}{2}}; \theta_\alpha = \alpha \frac{\pi}{8} \tag{1.20}$$

where the scale and orientation is given by (k_v, θ_α), being v the spatial frequency number and α the orientation.

An image is represented by the Gabor wavelet transform in four dimensions, two are the spatial dimensions, and the other two represent spatial frequency structure and spatial relations or orientation. So, processing the face image with Gabor filters with 5 spatial frequency ($v = 0, .., 4$) and 8 orientation ($\alpha = 0, .., 7$) captures the whole frequency spectrum - see image

1.9. So, we have 40 wavelets. The amplitude of Gabor filters are used for recognition.

Once the transformation has been performed, different techniques can be applied to extract the relevant features, like high-energized points comparisons [55].

1.9.6   Independent Component Analysis

Independent Component Analysis aims to transform the data as linear combinations of statistically independent data points. Therefore, its goal is to provide an independent rather that uncorrelated image representation. ICA is an alternative to PCA which provides a more powerful data representation [67]. It's a discriminant analysis criterion, which can be used to enhance PCA.

The ICA algorithm is performed as follows [25]. Let c_x be the covariance matrix of an image sample X. The ICA of X factorizes the covariance matrix c_x into the following form: $c_x = F \Delta F^T$ where Δ is diagonal real positive and F transforms the original data into Z ($X = FZ$). The components of Z will be the most independent possible. To derive the ICA transformation F,

$$X = \Phi \Lambda^{\frac{1}{2}} U \tag{1.21}$$

where X and Λ are derived solving the following eigenproblem:

$$c_x = \Phi \Lambda \Phi^T \tag{1.22}$$

Then, there are rotation operations which derive independent components minimizing mutual information. Finally, a normalization is carried out.

1.9.7   Kernel PCA

The use of Kernel functions for performing nonlinear PCA was introduced by Scholkopf et al [97]. Its basic methodology is to apply a non-linear mapping to the input ($\Psi(x) : R^N \to R^L$) and then solve a linear PCA in the resulting feature subspace. The mapping of $\Psi(x)$ is made implicitly using kernel functions

$$k(x_i, x_j) = (\Psi(x_i) \cdot \Psi(x_j)), \tag{1.23}$$

where n the input space correspond to dot- products in the higher dimensional feature space. Assuming that the projection of the data has been centered, the covariance is given by $C_x = < \Psi(x_i), \Psi(x_i)^T >$, with the resulting eigenproblem:

$$\lambda V = C_x V \tag{1.24}$$

where there must exist some coefficients w_i so that

$$V = \sum_{i=1}^{M} w_i \Psi(x_i) \tag{1.25}$$

The operations performed in [97] lead to an equivalent eigenproblem:

$$M\lambda w = Kw \tag{1.26}$$

The kernel matrix K is then diagonalized by PCA. This leads to the conclusion[98] that the n-th principal component y_n of x is given by

$$y_n = V_n \cdot \Psi(x) = \sum_{i=1}^{M} w_i^n k(x, x_i) \tag{1.27}$$

where V_n is the n-th eigenvector of the feature space defined by Ψ.

The selection of an optimal kernel is another engineering problem. Typical kernels include gussians

$$k(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \tag{1.28}$$

polynomial kernels

$$k(x, y) = (x \cdot y + 1)^d, \tag{1.29}$$

or Neural Network type kernels

$$k(x, y) = \tanh((x \cdot y) + b). \tag{1.30}$$

1.9.8   Other methods

Other algorithms are worth mentioning. Fore example, genetic algorithms have been used, and proved more accurate (but more resource-consuming) than PCA or LDA [65, 68]. Other successful statistic tools include Bayesian networks [83, 78, 66], bi-dimensional regression [52] and ensemble-based and other boosting methods [71, 76, 111].

1.10   Neural Network approach

Artificial neural networks are a popular tool in face recognition. They have been used in pattern recognition and classification. Kohonen [57] was the first to demonstrate that a neuron network could be used to recognize aligned and normalized faces. Many different methods based on neural network have been proposed since then. Some of these methods use neural networks just for classification. One approach is to use decision-based neural networks, which classifies pre-processed and sub sampled face images [58].

There are methods which perform feature extraction using neural networks. For example, Intrator et. al proposed a hybrid or semi-supervised method [47]. They combined unsupervised methods for extracting features and supervised methods for finding features able to reduce classification error. They used feed-forward neural networks (FFNN) for classification. They also tested their algorithm using additional bias constraints, obtaining better results. They also demonstrated that they could decrease the error rate training several neural networks and averaging over their outputs, although it is more time-consuming that the simple method.

Lawrence et. al [60] used self-organizing map neural network and convolutional networks. Self-organizing maps (SOM) are used to project the data in a lower dimensional space and a convolutional neural network (CNN) for partial translation and deformation invariance. Their method is evaluated, by substituting the SOM with PCA and the CNN with a multi-layer perceptron (MLP) and comparing the results. They conclude that a convolutional network is preferable over a MPL without previous knowledge incorporation. The SOM seems to be computationally costly and can be substituted by a PCA without loss of accuracy.

Overall, FFNN and CNN classification methods are not optimal in terms of computational time and complexity [9]. Their classification performance is bounded above by that of the eigenface but is more costly to implement in practice.

Zhang and Fulcher presented and artificial neural network Group-based Adaptive Tolerance (GAT) Tree model for translation-invariant face recognition in 1996 [123]. Their algorithm was developed with the idea of implementing it in an airport surveillance system. The algorithm's input were passport photographs. This method builds a binary tree whose nodes are neural network group-based nodes. So, each node is a complex classifier, being a MLP the basic neural network for each group-based node.

Other authors used probabilistic decision based neural networks (PDBNN). Lin et al. developed a face detection and recognition algorithm using this kind of network [64]. They applied it to face detection, feature extraction
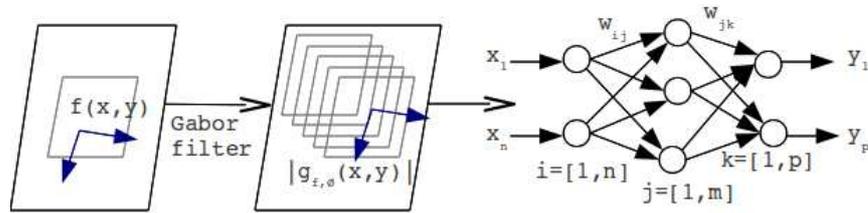
Figure 1.9: Neural networks with Gabor filters.

and classification. This network deployed one sub-net for each class, approximating the decision region of each class locally. The inclusion of probability constraints lowered false acceptance and false rejection rates.

1.10.1 Neural networks with Gabor filters

Bhuiyan et al. proposed in 2007 a neural network method combined with Gabor filter [10]. Their algorithm achieves face recognition by implementing a multilayer perceptron with back-propagation algorithm. Firstly, there is a pre-processing step. Every image is normalized in terms of contrast and illumination. Noise is reduce by a "fuzzily skewed" filter. It works by applying fuzzy membership to the neighbor pixels of the target pixel. It uses the median value as the 1 value membership, and reduces the extreme values, taking advantage from median filter and average filter.

Then, each image is processed through a Gabor filter. The filter is represented as a complex sinusoidal signal modulated by a Gaussian kernel function. The Gabor filter has five orientation parameters and three spatial frequencies, so there are 15 Gabor wavelets. The architecture of the neural network is illustrated in figure 1.10.

To each face image, the outputs are 15 Gabor-images which record the variations measured by the Gabor filters. The first layer receives the Gabor features. The number of nodes is equal to the dimension of the feature vector containing the Gabor features. The output of the network is the number of images the system must recognize. The training of the network, the back-propagation algorithm, follows this procedure:

1. Initialization of the weights and threshold values.

2. Iterative process until termination condition is fulfilled:

    (a) Activate, applying input and desired outputs. Calculate actual outputs of neurons in hidden and output layers, using sigmoid activation function.
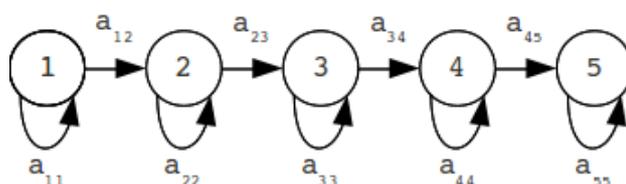
Figure 1.10: 1D-HDD states.

    (b) Update wights, propagating backwards the errors.

    (c) Increase iteration value

Although the algorithms main purpose is to face illumination variations, it shows a useful neural network application for face recognition. It could be useful with some improvements in order to deal with pose and occlusion problems.

1.10.2 Neural networks and Hidden Markov Models

Hidden Markov Models (HMM) are a statistical tool used in face recognition. They has been used in conjunction with neural networks. Bevilacqua et. al have developed a neural network that trains pseudo two-dimensional HMM [8].

    The composition of a 1-dimensional HMM is illustrated in figure 1.11. A HMM λ can be defined as $\lambda = (A, B, \Pi)$:

- $A = [a_{ij}]$ is a state transition probability matrix, where a_{ij} is the probability that the state i becomes the state j.

- $B = [b_j(k)]$ is a state transition probability matrix, where $b_j(k)$ is the probability to have the observation k when the state is j.

- $\Pi = \{\pi_1, \ldots, \pi_n\}$ is the initial state distribution, where π_i is the probability associated to state i.

They propose a pseudo 2D-HMM, defining superstates formed by states, being the 3-6-6-6-3 the most successful configuration, as shown in figure 1.12.

    The input of this 2D-HMM process is the output of the artificial neural network (ANN). The ANN provides the algorithm with the proper dimensionality reduction. So, the input to the 2D-HMM are images compressed into observation vectors of binary elements. The ANN, using error backpropagation algorithm, extracts main features and stores them in a 50 bit
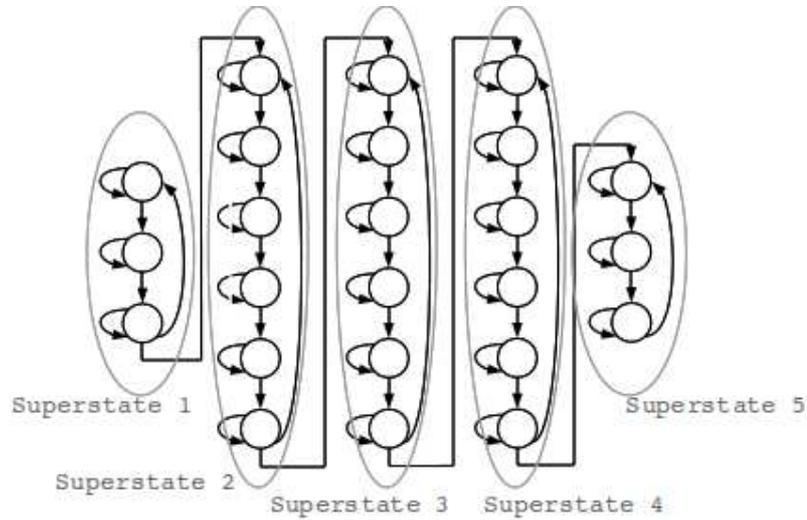
Figure 1.11: 2D-HDD superstates configuration.

sequence. The input face image is divided into 103 segments of 920 pixels, and each segment is divided into four 230 pixel features. So, the first and last layers are formed by 230 neurons each. The hidden layer is formed by 50 nodes. So a section of 920 pixels is compressed in four sub windows of 50 binary values each. The training function is iterated 200 times for each photo. Finally, the ANN is tested with images similar to the input image, doing this process for each image. This method showed promising results, achieving a 100% accuracy with ORL database.

1.10.3 Fuzzy neural networks

The introduction of fuzzy mathematics in neural networks for face recognition is another approach. Bhattacharjee et al. developed in 2009 a face recognition system using a fuzzy multilayer perceptron (MLP) [9]. The idea behind this approach is to capture decision surfaces in non-linear manifolds, a task that a simple MLP can hardly complete.

The feature vectors are obtained using Gabor wavelet transforms. The method used is similar to the one presented in point 10.1. Then, the output vectors obtained from that step must be fuzzified. This process is simple: The more a feature vector approaches towards the class mean vector, the higher is the fuzzy value. When the difference between both vectors increases, the fuzzy value approaches towards 0.

The selected neural network is a MLP using back-propagation. There is a network for each class. The examples of this class are class-one, and

the examples of the other classes form the class-two. Thus, is a two-class classification problem. The fuzzification of the neural network is based on the following idea: The patterns whose class is less certain should have lesser role in wight adjustment. So, for a two-class (i and j) fuzzy partition $\phi_i(x_k)$ $k = 1, \ldots, p$ of a set of p input vectors,

$$\varphi_i = 0.5 + \frac{e^{c(d_j - d_i)/d} - e^{-c}}{2(e^c - e^{-c})} \tag{1.31}$$

$$\varphi_j = 1 - \varphi_i(x_k) \tag{1.32}$$

where d_i is the distance of the vector from the mean of class i. The constant c controls the rate at which fuzzy membership decreases towards 0.5.

The contribution of x_k in weight update is given by $|\phi_1(x_k) - \phi_2(x_k)|^m$, where m is a constant, and the rest of the process follows a usual MLP procedure. The results of the algorithm show a 2.125 error rate using ORL database.

Chapter 2

Conclusions

Contents

Face recognition is a complex and mutable subject. In addition to algorithms, there are more things to think about. The study of face recognition reveals many conclusions, issues and thoughts. This chapter aims to explain and sort them, hopefully to be useful is future research.

2.1 The problems of face recognition.

This work has presented the face recognition area, explaining different approaches, methods, tools and algorithms used since the 60's. Some algorithms are better, some are less accurate, some of the are more versatile and others are too computationally costly. Despite this variety, face recognition faces some issues inherent to the problem definition, environmental conditions and hardware constraints. Some specific face detection problems are explained in previous chapter. In fact, some of these issues are common to other face recognition related subjects. Nevertheless, those and some more will be detailed in this section.

2.1.1 Illumination

Many algorithms rely on color information to recognize faces. Features are extracted from color images, although some of them may be gray-scale. The color that we perceive from a given surface depends not only on the surface's nature, but also on the light upon it. In fact, color derives from the perception of our light receptors of the spectrum of light -distribution of light energy versus wavelength. There can be relevant illumination variations on images taken under uncontrolled environment. That said, the chromacity is an essential factor in face recognition. The intensity of the color in a pixel can vary greatly depending on the lighting conditions.

Is not only the sole value of the pixels what varies with light changes. The relation or variations between pixels may also vary. As many feature extraction methods relay on color/intensity variability measures between pixels to obtain relevant data, they show an important dependency on lighting changes. Keep in mind that, not only light sources can vary, but also light intensities may increase or decrease, new light sources added. Entire face regions be obscured or in shadow, and also feature extraction can become impossible because of solarization. The big problem is that two faces of the same subject but with illumination variations may show more differences between them than compared to another subject. Summing up, illumination is one of the big challenges of automated face recognition systems. Thus, there is much literature on the subject. However, it has been demonstrated

that humans can generalize representations of a face under radically different illumination conditions, although human recognition of faces is sensitive to illumination direction [100].

Zhao et al. [124] illustrated this problem plotting the variation of eigenspace projection coefficient vectors due to differences in class label (2.1 a) along with variations due to illumination changes of the same class (2.1 b).
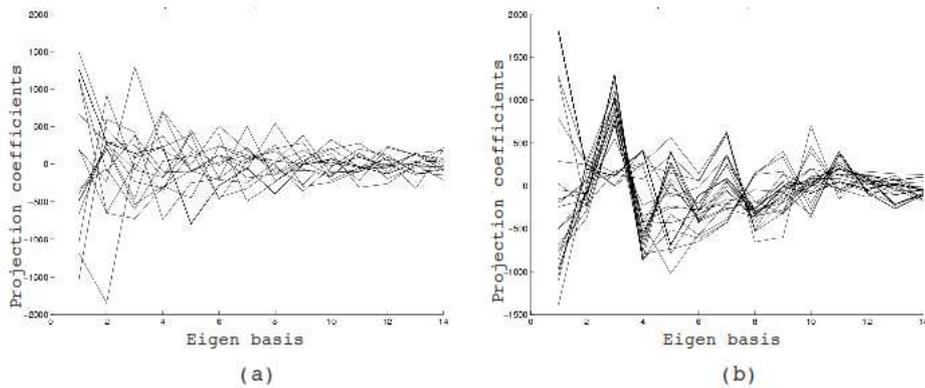
Figure 2.1: Variability due to class and illumination difference.

It has been demonstrated that the most popular image representations such as edge maps or Gabor-filtered images are not capable of overcoming illumination changes [1]. This illumination problem can be faced employing different approaches:

Heuristic approach

The observation of face recognition algorithms behavior can provide relevant clues in order to prevent illumination problems.

It has been suggested that, within eigenspace domain, three most significant principal component can be discarded [124]. However, we must maintain the system's performance with normally illuminated face images. Thus, we must assume that the first three principal components capture the variations only due to illumination.

An approach based on symmetry is proposed by Sirovich et. al in 2009 [102]. Their method is based on the natural symmetry of human faces. They conclude that odd eigenfaces are caused by illumination artifacts. Therefore, they discard them from their syntactic face construction procedure. This algorithm shows a nearly perfect accuracy recognizing frontal face images under different lighting conditions.

Statistical approach

Statistical methods for feature extraction can offer better or worse recognition rates. Moreover, there is extensive literature about those methods. Some papers test these algorithms in terms of illumination invariance, showing interesting results [98, 37].

The class separability that provides LDA shows better performance that PCA, a method very sensitive to lighting changes. Bayesian methods allow to define intra-class variations such as illumination variations, so these methods show a better performance than LDA. However, all this linear analysis algorithms do not capture satisfactorily illumination variations.

Non-linear methods can deal with illumination changes better than linear methods. Kernel PCA and non-linear SVD show better performance than previous linear methods. Moreover, Kernel LDA deals better with lighting variations that KPCA and non-linear SVD. However, it seems that although choosing the right statistical model can help dealing with illumination, further techniques may be required.

Light-modeling approach

Some recognition methods try to model a lighting template in order to build illumination invariant algorithms. There are several methods, like building a 3D illumination subspace from 3 images taken under different lighting conditions, developing an illumination cone or using quotient shape-invariant images [124].

Some other methods try to model the illumination in order to detect and extract the lighting variations from the picture. Gross et. al develop a Bayesian sub-region method that regards images as a product of the reflectance and the luminance of each point. This characterization of the illumination allows the extraction of lighting variations, i.e. the enhancement of local contrast. This illumination variation removing process enhances the recognition accuracy of their algorithms in a 6.7%.

Model-based approach

The most recent model-based approaches try to build 3D models. The idea is to make intrinsic shape and texture fully independent from extrinsic parameters like light variations. Usually, a 3D head set is needed, which can be captured from a multiple-camera rig or a laser-scanned face [79]. The 3D heads can be used to build a morphable model, which is used to fit the input images [13]. Light directions and cast shadows can be estimated automatically.

This approaches can show good performance with data sets that have many illumination variations like CMU-PIE and FERET [13].

Multi-spectral imaging approach

Multi-spectral images (MSI) are those that capture image data at specific wavelengths. The wavelengths can be separated by filters or other instruments sensitive to particular wavelets.

Chang et. al developed a novel face recognition approach using fused MSIs [23]. They chose MSIs for face recognition not only because MSIs carry more information than conventional images, but also because it enables the separation of spectral information of illumination from other spectral information.

They proposed different multi-spectral fusion algorithms:

- Physics-based weighted fusion. They define the camera response of band i in the following manner

$$\rho_{\lambda_i} = \int_{\lambda_{max}}^{\lambda_{min}} R(\lambda)L(\lambda)S_{\lambda_i}(\lambda)T_{\lambda_i}(\lambda)d(\lambda) \qquad (2.1)$$

where the centered wavelength λ_i has range λ_{min} to λ_{max}. $R(\lambda)$ is the spectral reflectance of the object, $L(\lambda)$ is the spectral distribution of the illumination, $S_{\lambda_i}(\lambda)$ is the spectral response of the camera and $T_{\lambda_i}(\lambda)$ is the spectral transmittance of the Liquid crystal tunable filter.

The pixel values of the weighted fusion, p_w can be represented as

$$p_w = \frac{1}{C}\sum_{i=1}^{N} w_{\lambda_i}\rho_{\lambda_i} \qquad (2.2)$$

where C is the sum of the wighted values of λ. The weights can be set to compensate intensity differences.

- Illumination adjustment via data fusion. In this case, the camera response has a direct relationship to the incident illumination. This fusion technique allows to distinguish different light sources, like day-light and fluorescent tubes. They can convert an image taken under the sun into one taken under a fluorescent tube light operating with the $L(\lambda)$ values of both images.

- Wavelet fusion. Wavelet transform is a data analysis tool that provides a multi-resolution decomposition of an image. Given two registered images I_1 and I_2 of the same object in two sets of probes, two dimensional

discrete wavelet decomposition is performed on I_1 and I_2, to obtain the wavelet approximation coefficients a_1, a_2 and detail coefficients d_1, d_1. Then, wavelet approximation and detail coefficients of the fused image, a_f and d_f are calculated. The two-dimensional discrete wavelet inverse transform is then performed to obtain the fused image.

- Rank-based decision level fusion. A voting decision fusion strategy is presented. The decision fusion is based on rank values of individuals, hence the the "rank-based" name.

Their experiment tested the image sets under different lighting conditions. They demonstrated that face images created by fusion of continuous spectral images improve recognition rates compared to conventional images under different illumination. Rank-based decision level fusion provided the highest recognition rate among the proposed fusion algorithms.

2.1.2 Pose

Pose variation and illumination are the two main problems face by face recognition researchers. The vast majority of face recognition methods are based on frontal face images. These set of images can provide a solid research base. It can be mentioned that maybe the 90% of papers referenced on this work use these kind of databases. Image representation methods, dimension reduction algorithms, basic recognition techniques, illumination invariant methods and many other subjects are well tested using frontal faces.

On the other hand, recognition algorithms must implement the constraints of the recognition applications. Many of them, like video surveillance, video security systems or augmented reality home entertainment systems take input data from uncontrolled environment. The uncontrolled environment constraint involves several obstacles for face recognition: The aforementioned problem of lighting is one of them. Pose variation is another one.

There are several approaches used to face pose variations. Most of them have already been detailed, so this section won't go over them again. However, it's worth mentioning the most relevant approaches to pose problem solutions:

Multi-image based approaches

These methods require multiple images for training. The idea behind this approach is to make templates of all the possible pose variations [124]. So, when an input image must be classified, it is aligned to the images corresponding to a single pose. This process is repeated for each stored pose,

until the correct one is detected. The restrictions of such methods are firstly, that many images are needed for each person. Secondly, this systems perform a pure texture mapping, so expression and illumination variations are an added issue. Finally, the iterative nature of the algorithm increases the computational cost.

Multi-linear SVDs are another example of this approach [98]. Other approaches include view-based eigenface methods, which construct an individual eigenface for each pose [124].

Single-model based approaches

This approach uses several data of a subject on training, but only one image at recognition. There are several examples of this approach, from which 3D morphable model methods are a self-explanatory example. Data may be collected from many different sources, like multiple-camera rigs [13] or laser-scanners [61]. The goal is to include every pose variation in a single image. The image, on this case, would be a three dimensional face model. Theoretically, if a perfect 3D model could be built, pose variations would become a trivial issue. However, the recording of sufficient data for model building is one problem. Many face recognition applications don't provide the commodities needed to build such models from people. Other methods mentioned in [124] are low-level feature based methods and invariant feature based methods.

Some researchers have tried an hybrid approach, trying to use a few images and model the rotation operations, so that every single pose can be deducted from just one frontal photo and another profile image. This method is called Active Appearance Model (AAM) in [27].

Geometric approaches

There are approaches that try to build a sub-layer of pose-invariant information of faces. The input images are transformed depending on geometric measures on those models. The most common method is to build a graph which can link features to nodes and define the transformation needed to mimic face rotations. Elastic Bunch Graphic Matching (EBGM) algorithms have been used for this purpose [114, 115].

2.1.3   Other problems related to face recognition

There are other problems that automated face recognition research must face sooner or later. Some of them have no relation between them. They are

explained along this section, in no particular order:

Occlusion

We understand occlusion as the state of being obstructed. In the face recognition context, it involves that some parts of the face can't be obtained. For example, a face photograph taken from a surveillance camera could be partially hidden behind a column. The recognition process can rely heavily on the availability of a full input face. Therefore, the absence of some parts of the face may lead to a bad classification. This problem speaks in favor of a piecemeal approach to feature extraction, which doesn't depend on the whole face.

There are also objects that can occlude facial features -glasses, hats, beards, certain hair cuts, etc.

Optical technology

A face recognition system should be aware of the format in which the input images are provided. There are different cameras, with different features, different weaknesses and problems. Usually, most of the recognition processes involve a preprocessing step that deals with this problem.

Expression

Facial expression is another variability provider. However, it isn't as strong as illumination or pose. Several algorithms don't deal with this problem in a explicit way, but they show a good performance when different facial expressions are present.

On the other hand, the addition of expression variability to pose and illumination problems can become a real impediment for accurate face recognition.

Algorithm evaluation

It's not easy to evaluate the effectiveness of a recognition algorithm. Several core factors are unavoidable:

- Hit ratio.

- Error rate.

- Computational speed.

- Memory usage.

Then, depending on the system requirements, there could be other important factors.

- Illumination/occlusion/expression/pose invariability.

- Scalability.

- Adaptability (to variable input image formats, sizes, etc.).

- Automatism (unsupervised processes).

- Portability.

The most used face recognition algorithm testing standards evaluate the pattern recognition related accuracy and the computational cost. The most popular evaluation systems are the FERET Protocol [92, 89] and the XM2VTS Protocol [50]. For further information on FERET and XM2VTS refer to [125].

There are many face data bases available. some of them are free to use, some other require an authorization. Table 2.1 shows the most used ones.

Database	Description
MIT [110]	16 people, 27 images each. Variable illumination, scale, orientation.
FERET [89]	Large collection.
UMIST [36]	20 people, 28 images each. Pose variations.
University of Bern	30 people, 15 images each. Frontal and profile.
Yale [6]	15 people, 11 images each. High variability.
AT&T (ORL) [94]	40 people, 10 images each. Low variability
Harvard [40]	Large, variable set. Diverse lighting conditions.
M2VTS [90]	Multimodal. Image sequences.
Purdue AR [73]	130 people, 26 images each. Illumination and occlusion variability.
PICS (Sterling) [86]	Large psychology-oriented DB
BIOID [11]	23 subjects, 66 images each. Variable backgrounds
Labeled faces in the wild [44]	5749 people, 1680 of them with two or more pictures. 13233 images.

Table 2.1: Face recognition DBs

2.2 Conclussion

This work has presented a survey about face recognition. It isn't a trivial task, and today remains unresolved. These are the current lines of research:

- New or extended feature extraction methods. There is much literature about extending or improving well known algorithms. For example, including weighting procedures to PCA [69], developing new kernel-based methods [126] or turning methods like LDA into semi-supervised algorithms [22].

- Feature extraction method combination. Many algorithms are being built around this idea. As many strong feature extraction techniques have been developed, the challenge is to combine them. Fore example, LDA can be combined with SVD to overcome problems derived from small sample sizes [81].

- Classifier and feature extraction method combinations. It's a common approach to face recognition. For instance, there are recent works that combine different extraction methods with adaptive local hyperplane (ALH) classification methods [119].

- Classifier combination. There are strong classifiers that have achieved good performance in face recognition problems. Nowadays, there is a trend to combine different classifiers in order to get the best performance[108].

- Data gathering techniques. There are some novel methods to gather visual information. The idea is to obtain more information that that provided by simple images. Examples of this approach include 3D scans [61] and continuous spectral images [23].

- Works on biologically inspired techniques. The most common techniques that fall into this category are genetic algorithms [68] and, above all, artificial neural networks [8, 71].

- Boosting techniques. Diverse boosting techniques are being used and successfully applied to face recognition. One example is AdaBoost in the well known detection method developed by Viola and Jones [111].

Face recognition is also resulting in other dares, like expression recognition or body motion recognition. Overall, face recognition techniques and the emerging methods can see use in other areas. Therefore, it isn't just a unresolved problem but also the source of new applications and challenges.

References

[1] Y. Adini, Y. Moses, and S. Ullma. Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721–732, 1997.

[2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 23:90–93, 1974.

[3] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[4] M. Ballantyne, R. S. Boyer, and L. Hines. Woody bledsoe: His life and legacy. *AI Magazine*, 17(1):7–20, 1996.

[5] M. Bartlett, J. Movellan, and T. Sejnowski. Face recognition by independent component analysis. *IEEE Trans. on Neural Networks*, 13(6):1450–1464, November 2002.

[6] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[7] S. Bentin, T. Allison, A. Puce, E. Perez, and G. McCarthy. Electrophysiological studies of face perception in humans. *Journal of Cognitive Neuroscience*, 8(6):551–565, 1996.

[8] V. Bevilacqua, L. Cariello, G. Carro, D. Daleno, and G. Mastronardi. A face recognition system based on pseudo 2d hmm applied to neural network coefficient. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12(7):615–621, 2007.

[9] D. Bhattacharjee, D. K. Basu, M. Nasipuri, and M. Kundu. Human face recognition using fuzzy multilayer perceptron. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 14(6):559–570, April 2009.

[10] A.-A. Bhuiyan and C. H. Liu. On face recognition using gabor filters. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 22, pages 51–56, 2007.

[11] BIOID. Bioid face database. http://www.bioid.com/support/download s/software/bioid-face-database.html.

[12] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of the SIGGRAPH'99*, pages 187–194, Los Angeles, USA, August 1999.

[13] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, September 2003.

[14] W. W. Bledsoe. The model method in facial recognition. Technical report pri 15, Panoramic Research, Inc., Palo Alto, California, 1964.

[15] W. W. Bledsoe. Man-machine facial recognition: Report on a large-scale experiment. Technical report pri 22, Panoramic Research, Inc., Palo Alto, California, 1966.

[16] W. W. Bledsoe. Some results on multicategory patten recognition. *Journal of the Association for Computing Machinery*, 13(2):304–316, 1966.

[17] W. W. Bledsoe. Semiautomatic facial recognition. Technical report sri project 6693, Stanford Research Institute, Menlo Park, California, 1968.

[18] W. W. Bledsoe and H. Chan. A man-machine facial recognition system-some preliminary results. Technical report pri 19a, Panoramic Research, Inc., Palo Alto, California, 1965.

[19] E. M. Bronstein, M. M. Bronstein, R. Kimmel, and A. Spira. 3d face recognition without facial surface reconstruction, 2003.

[20] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, October 1993.

[21] J. S. Bruner and R. Tagiuri. The percepton of people. *Handbook of Social Psycology*, 2(17), 1954.

[22] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. *Computer Vision. IEEE 11th International Conference on Computer Vision*, 14:1–7, 2007.

[23] H. Chang, A. Koschan, B. Abidi, and M. Abidi. Fusing continuous spectral images for face recognition under indoor and outdoor illuminants. *Machine Vision and Applications*, 21(2):201–215, February 2010.

[24] L.-F. Chen, H.-Y. Liao, J.-C. Lin, and C.-C. Han. Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof. *Pattern Recognition*, 34(5):1393–1403, 2001.

[25] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[26] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, March 2004.

[27] T. Cootes, K. Walker, and C. Taylor. View-based active apearance models. In *Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 227–232, Grenoble, France, March 2000.

[28] J. G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847 – 856, 1980.

[29] R. Diamond and S. Carey. Why faces are and are not special. an effect of expertise. *Journal of Experimental Psychology: General*, 115(2):107–117, 1986.

[30] M. Divjak and H. Bischof. Real-time video-based eye blink analysis for detection of low blink-rate during computer use. In *1st International Workshop on Tracking Humans for the Evaluation of their Motion in Image Secuences (THEMIS)*, pages 99–107, Leeds, UK, September 2008.

[31] B. Dudley. "e3: New info on microsoft's natal – how it works, multiplayer and pc versions". *The Seattle Times*, June 3 2009.

[32] N. Duta and A. Jain. Learning the human face concept from black and white pictures. In *Proc. International Conf. Pattern Recognition*, pages 1365–1367, 1998.

[33] K. B. et al. *AI 2005*, chapter Multiple Face Tracking Using Kalman Estimator Based Color SSD Algorithm, pages 1229–1232. LNAI 3809. Springer-Verlag Heidelberg, 2005.

[34] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67 – 92, 1973.

[35] A. Golstein, L. Harmon, and A. Lest. Identification of human faces. *Proceedings of the IEEE*, 59:748–760, 1971.

[36] D. Graham and N. Allinson. *Face recognition: From Theory to Applications*, chapter Characterizing Virtual Eigensignatures for General Purpose Face Recognition, pages 446–465. 1998.

[37] R. Gross, S. Baker, I. Matthews, and T. Kanade. Face recognition across pose and illumination. In S. Z. Li and A. K. Jain, editors, *Handbook of Face Recognition*. Springer-Verlag, June 2004.

[38] R. Gross, J. Shi, and J. Cohn. Quo vadis face recognition? - the current state of the art in face recognition. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, June 2001.

[39] G. Guo, S. Li, and K. Chan. Face recognition by support vector machines. In *Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 196–201, Grenoble, France, March 2000.

[40] P. Hallinan. *A Deformable Model for Face Recognition Under Arbitrary Lighting Conditions*. PhD thesis, Univesity of Harvard, 1995.

[41] C.-C. Han, H.-Y. M. Liao, K. chung Yu, and L.-H. Chen. *Lecture Notes in Computer Science*, volume 1311, chapter Fast face detection via morphology-based pre-processing, pages 469–476. Springer Berlin / Heidelberg, 1997.

[42] X. He and P. Niyogi. Locality preserving projections. In *Proceedings of the Conference on Advances in Nerual Information Processing Systems*, 2003.

[43] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *Proc. of the Eighth IEEE International Conference on Computer Vision, ICCV 2001*, volume 2, pages 688–694, Vancouver, Canada, July 2001.

[44] G. B. Huang, M. Ramesh, T. Berg, and E. 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.

[45] J. Huang, S. Gutta, and H. Wechsler. Detection of human faces using decision trees. In *Proc. Second International Conf. Automatic Face and Gesture Recognition*, pages 248–252, 1996.

[46] J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. In *roc. of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA*, pages 27–34, Guildford, UK, June 2003.

[47] N. Intrator, D. Reisfeld, and Y. Yeshurun. Face recognition using a hybrid supervised/unsupervised neural network. *Pattern Recognition Letters*, 17:67–76, 1995.

[48] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[49] K. Jonsson, J. Matas, J. Kittler, and Y. Li. Learning support vectors for face verification and recognition. In *Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 208–213, Grenoble, France, March 2000.

[50] M. H. Jonsson, J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, J. Bigun, N. Capdevielle, W. Gerstner, S. Ben-yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the xm2vts database. In *Proceedings of the 15th ICPR*, pages 858–863. IEEE Computer Society Press, 2000.

[51] A. Kadyrov and M. Petrou. The trace transform and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):811–823, August 2001.

[52] S. Kare, A. Samal, and D. Marx. Using bidimensional regression to assess face similarity. *Machine Vision and Applications*, 21(3):261–274, 2008.

[53] S. Kawato and N. Tetsutan. Detection and tracking of eyes for gaze-camera control. In *Proc. 15th Int. Conf. on Vision Interface*, pages 348–353, 2002.

[54] T. Kenade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, November 1973.

[55] B. Kepenekci. *Face recognition using Gabor Wavelet Transformation*. PhD thesis, Te Middle East Technical University, September 2001.

[56] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[57] T. Kohonen. *Self-organization and associative memory*. Springer-Verlag, Berlin, 1989.

[58] S. Kung and J. Taur. Decision-based neural networks with signal/image classification applications. *IEEE Transactions on Neural Networks*, 6(1):170–181, 1995.

[59] K.-M. Lam and H. Yan. Fast algorithm for locating head boundaries. *Journal of Electronic Imaging*, 03(04):351–359, October 1994.

[60] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8:98–113, 1997.

[61] J. Lee, J. Lee, B. Moghaddam, B. Moghaddam, H. Pfister, H. Pfister, R. Machiraju, and R. Machiraju. Finding optimal views for 3d face shape modeling. In *Proceedings International Conference on Automatic Face and Gesture Recognition*, pages 31–36, 2004.

[62] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. inding optimal views for 3d face shape modeling. In *Proc. of the International Conference on Automatic Face and Gesture Recognition, FGR2004*, pages 31–36, Seoul, Korea, May 2004.

[63] T. Lee. Image representation using 2d wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[64] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions on Neural Networks*, 8:114–132, 1997.

[65] C. Liu and H. Wechsler. Face recognition using evolutionary pursuit. In *Proceedings of the Fifth European Conference on Computer Vision, ECCV'98*, volume 2, pages 596–612, Freiburg, Germany, 1998.

[66] C. Liu and H. Wechsler. A unified bayesian framework for face recognition. In *roc. of the 1998 IEEE International Conference on Image Processing, ICIP'98*, pages 151–155, Chicago, USA, October 1998.

[67] C. Liu and H. Wechsler. Comparative assessment of independent component analysis (ica) for face recognition. In *Proc. of the Second International Conference on Audio- and Video-based Biometric Person Authentication, AVBPA'99*, Washington D.C., USA, March 1999.

[68] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):570–582, June 2000.

[69] F. Liu, Z. liang Wang, L. Wang, and X. yan Meng. *Affective Computing and Intelligent Interaction*, chapter Facial Expression Recognition Using HLAC Features and WPCA, pages 88–94. Springer Berlin / Heidelberg, 2005.

[70] R. Louban. *Image Processing of Edge and Surface Defects Theoretical Basis of Adaptive Algorithms with Numerous Practical Applications*, volume 123, chapter Edge Detection, pages 9–29. Springer Berlin Heidelberg, 2009.

[71] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Li. Ensemble-based discriminant learning with boosting for face recognition. *IEEE Transactions on Neural Networks*, 17(1):166–178, January 2006.

[72] X. Lu. Image analysis for face recognition. personal notes, May 2003.

[73] K. V. Mardia and I. L. Dryden. The ar face database. Technical report, Purdue University, 1998.

[74] K. Massy. "toyota develops eyelid-monitoring system". *Cnet reviews*, January 22 2008.

[75] M. McWhertor. "sony spills more ps3 motion controller details to devs". *Kotaku. Gawker Media.*, June 19 2009. http://kotaku.com/5297265/sony-spills-more-ps3-motion-controller-details-to-devs.

[76] R. Meir and G. Raetsch. An introduction to boosting and leveraging. In I. S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, pages 118–183. Springer, 2003.

[77] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, June 2002.

[78] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, November 2000.

[79] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju. Model-based 3d face capture with shape-from-silhouettes. In *Proc. of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, AMFG*, pages 20–27, Nice, France, October 2003.

[80] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.

[81] N. Nain, P. Gour, N. Agarwal, R. P. Talawar, and S. Chandra. Face recognition using pca and lda with singular value decomposition(svd) using 2dlda. In *Proceedings of the World Congress on Engineering*, volume 1, 2008.

[82] M. C. Nechyba, L. Brandy, and H. Schneiderman. *Lecture Notes in Computer Science*, volume 4625/2009, chapter PittPatt Face Detection and Tracking for the CLEAR 2007 Evaluation, pages 126–137. Springer Berlin / Heidelberg, 2009.

[83] A. Nefian. Embedded bayesian networks for face recognition. In *Proc. of the IEEE International Conference on Multimedia and Expo*, volume 2, pages 133–136, Lusanne, Switzerland, August 2002.

[84] A. Nefian and M. Hayes. Hidden markov models for face recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, volume 5, pages 2721–2724, Washington, USA, May 1998.

[85] M. Nixon. Eye spacing measurement for facial recognition. *Proceedings of the Society of Photo-Optical Instrument Engineers, SPIE*, 575(37):279–285, August 1985.

[86] U. of Stirling. Psychological image collection at stirling (pics). http://pics.psych.stir.ac.uk/.

[87] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. *Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition*, pages 130–136, June 1997.

[88] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[89] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, October 2000.

[90] S. Pigeon and L. Vandendrope. The m2vts multimodal face database,o. In *Proceedings First International Conferenbce on Audio- and Video-Based Biometric Person Authentication*, 1997.

[91] F. Raphael, B. Olivier, and C. Daniel. A constrained generative model applied to face detection. *Neural Processing Letters*, 5(2):11–19, April 1997.

[92] S. A. Rizvi, P. J. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms, 1999.

[93] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE trans. Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[94] F. Samaria. *Face Recognition Using Hidden Markov Models*. PhD thesis, University of Cambridge, 1997.

[95] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, USA, December 1994. DB abailable at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[96] H. Schneiderman and T. Kenade. Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 45–51, 1998.

[97] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max-Planck-Institut fur biologische Kybernetik, 1996.

[98] G. Shakhnarovich and B. Moghaddam. Face recognition in subspaces. In S. Z. Li and A. K. Jain, editors, *Handbook of Face Recognition*, page 35. Springer-Verlag, December 2004.

[99] S. K. Singh, D. S. Chauhan, M. Vatsa, and R. Singh. A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, 6(4):227–234, 2003.

[100] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: 19 results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, November November 2006.

[101] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A - Optics, Image Science and Vision*, 4(3):519–524, March 1987.

[102] L. Sirovich and M. Meytlis. Symmetry, probability, and recognition in face space. *PNAS - Proceedings of the National Academy of Sciences*, 106(17):6895–6899, April 2009.

[103] S. Srisuk and W. Kurutach. Face recognition using a new texture representation of face images. In *Proceedings of Electrical Engineering Conference*, pages 1097–1102, Cha-am, Thailand, November 2003.

[104] S. Srisuk, M. Petrou, W. Kurutach, and A. Kadyrov. Face authentication using the trace transform. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'03*, pages 305–312, Madison, Wisconsin, USA, June 2003.

[105] T. J. Stonham. Practical face recognition and verification with wisard. In H. D. Ellis, editor, *Aspects of face processing*. Kluwer Academic Publishers, 1986.

[106] K.-K. Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, Massachusetts Institute of Technology, 1996.

[107] L. Torres. Is there any hope for face recognition? In *Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS*, Lisboa, Portugal, 21-23 April 2004.

[108] S. Tulyakov. Review of classifier combination methods. *Studies in Computational Intelligence (SCI)*, 90:361–386, 2008.

[109] M. Turk. A random walk through eigenspace. *IEICE Transactions on Information and Systems*, E84-D(12):1586–1595, December 2001.

[110] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neurosicence*, 3(1):71–86, 1991.

[111] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[112] K. Wan-zeng and Z. Shan-an. Multi-face detection based on downsampling and modified subtractive clustering for color images. *Journal of Zhejiang University SCIENCE A*, 8(1):72–78, January 2007.

[113] S. Watanabe. Karhunen-loeve expansion and factor analysis theoretical remarks and applications. In *Proc. 4th Prague Conference on Information Theory*, 1965.

[114] L. Wiskott, J.-M. Fellous, N. Krueuger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[115] L. Wiskott, J.-M. Fellous, N. Krueuger, and C. von der Malsburg. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter Face Recognition by Elastic Bunch Graph Matching, pages 355–396. CRC Press, 1999.

[116] M.-H. Yang. Face recognition using kernel methods. *Advances in Neural Information Processing Systems*, 14:8, 2002.

[117] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, January 2002.

[118] M.-H. Yang, D. Roth, and N. Ahuja. A snow-based face detector. *Advances in Neural Information Processing Systems 12*, pages 855–861, 2000.

[119] T. Yang and V. Kecman. Face recognition with adaptive local hyperplane algorithm. *Pattern Analysis Applications*, 13:79–83, 2010.

[120] A. W. Yip and P. Sinhao. Contribution of color to face recognition. *Perception*, 31:995–1003, 2002.

[121] G. Zhang, L. Hu, and W. Jin. *Discovery Science*, volume 3245/2004 of *LNCS*, chapter Resemblance Coefficient and a Quantum Genetic Algorithm for Feature Selection, pages 125–153. Springer Berlin / Heidelberg, 2004.

[122] G. Zhang, W. Jin, and L. Hu1. *CIS*, volume 3314/2005 of *LNCS*, chapter A Novel Feature Selection Approach and Its Application, pages 665–671. Springer-Verlag Berlin Heidelberg, 2004.

[123] M. Zhang and J. Fulcher. Face recognition using artificial neural network group-based adaptive tolerance (gat) trees. *IEEE Transactions on Neural Networks*, 7(3):555–567, 1996.

[124] W. Zhao and R. Chellappa. Image-based face recognition: Issues and methods. In M. D. B. Javidi, editor, *Image Recognition and Classification*, pages 375–402. Marcel Dekker, Inc., 2002.

[125] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, pages 399–458, 2003.

[126] D. Zhou and Z. Tang. Kernel-based improved discriminant analysis and its application to face recognition. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 14(2):103–111, 2009.

[127] S. Zhou, R. Chellappa, and B. Moghaddam. Intra-personal kernel space for face recognition. In *Proc. of the 6th International Conference on Automatic Face and Gesture Recognition, FGR2004*, pages 235–240, Seoul, Korea, May 2004.