

GMM vs SVM for Face Recognition and Face Verification

Jesus Olivares-Mercado, Gualberto Aguilar-Torres, Karina Toscano-Medina,
Mariko Nakano-Miyatake and Hector Perez-Meana
*National Polytechnic Institute
Mexico*

1. Introduction

The security is a theme of active research in which the identification and verification identity of persons is one of the most fundamental aspects nowadays. Face recognition is emerging as one of the most suitable solutions to the demands of recognition of people. Face verification is a task of active research with many applications from the 80's. It is perhaps the biometric method easier to understand and non-invasive system because for us the face is the most direct way to identify people and because the data acquisition method consist basically on to take a picture. Doing this recognition method be very popular among most of the biometric systems users. Several face recognition algorithms have been proposed, which achieve recognition rates higher than 90% under desirable's condition (Chellapa et al., 2010; Hazem & Mastorakis, 2009; Jain et al., 2004; Zhao et al., 2003).

The recognition is a very complex task for the human brain without a concrete explanation. We can recognize thousands of faces learned throughout our lives and identify familiar faces at first sight even after several years of separation. For this reason, the Face Recognition is an active field of research which has different applications. There are several reasons for the recent increased interest in face recognition, including rising public concern for security, the need for identity verification in the digital world and the need for face analysis and modeling techniques in multimedia data management and computer entertainment. Recent advances in automated face analysis, pattern recognition, and machine learning have made it possible to develop automatic face recognition systems to address these applications (Duda et al., 2001). This chapter presents a performance evaluation of two widely used classifiers such as Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) for classification task in a face recognition system, but before beginning to explain about the classification stage it is necessary to explain with detail the different stages that make up a face recognition system in general, to understand the background before using the classifier, because the stages that precede it are very important for the proper operation of any type of classifier.

1.1 Face recognition system

To illustrate the general steps of a face recognition system consider the system shown in Fig. 1, which consists of 4 stages:



Fig. 1. General Structure of a face recognition system.

1.1.1 Capture

This stage is simple because it only needs a camera to take the face image to be processed. Due to this is not necessary to have a camera with special features, currently cell phones have a camera with high resolution which would serve or a conventional camera would be more than enough because the image can be pre-processed prior to extract the image features. Obviously, if the camera has a better resolution can be obtained clearer images for processing.

1.1.2 Pre-processing

In this stage basically apply some kind of cutting, filtering, or some method of image processing such as normalization, histogram equalization or histogram specification, among others. This is to get a better image for processing by eliminating information that is not useful in the case of cutting or improving the quality of the image as equalization. The pre-processing of the image is very important because with this is intended to improve the quality of the images making the system more robust for different scenarios such as lighting changes, possibly noise caused by background, among others.

1.1.3 Feature extraction

The feature extraction stage is one of the most important stages in the recognition systems because at this stage are extracted facial features in correct shape and size to give a good representation of the characteristic information of the person, that will serve to have a good training of the classification models.

Today exists great diversity of feature extraction algorithms, the following are listed some of them:

- Fisherfaces (Alvarado et al., 2006).
- Eigenfaces(Alvarado et al., 2006).
- Discrete Walsh Transform (Yoshida et al., 2003).
- Gabor Filters (Olivares et al., 2007).
- Discrete Wavelet Transform (Bai-Ling et al., 2004).
- Eigenphases (Savvides et al., 2004).

1.1.4 Classifiers

The goal of a classifier is to assign a name to a set of data for a particular object or entity. It defines a set of training as a set of elements, each being formed by a sequence of data for a specific object. A classifier is an algorithm to define a model for each class (object specific), so that the class to which it belongs an element can be calculated from the data values that define the object. Therefore, more practical goal for a classifier is to assign of most accurate form to new elements not previously studied a class. Usually also considered a test set that allows measure the accuracy of the model. The class of each set of test is known and is used to validate the model. Currently there are different ways of learning for classifiers among which are the supervised and unsupervised.

In supervised learning, a teacher provides a category label or cost for each pattern in a training set, and seeks to reduce the sum of the costs for these patterns. While in unsupervised learning or clustering there is no explicit teacher, and the system forms clusters or “natural groupings” of the input patterns. “Natural” is always defined explicitly or implicitly in the clustering system itself; and given a particular set of patterns or cost function, different clustering algorithms lead to different clusters.

Also, it is necessary to clarify the concept of identification and verification. In identification the system does not know who is the person that has captured the characteristics (the human face in this case) by which the system has to say who owns the data just processed. In verification the person tells the system which is their identity either by presenting an identification card or write a password key, the system captures the characteristic of the person (the human face in this case), and processes to create an electronic representation called live model. Finally, the classifier assumes an approximation of the live model with the reference model of the person who claimed to be. If the live model exceeds a threshold verifying is successful. If not, the verification is unsuccessful.

1.1.4.1 Classifiers types.

Exist different types of classifiers that can be used for a recognition system in order to choose one of these classifiers depends on the application for to will be used, it is very important to take in mind the selection of the classifier because this will depend the results of the system. The following describes some of the different types of classifiers exist.

Nearest neighbor. In the nearest-neighbor classification a local decision rule is constructed using the k data points nearest the estimation point. The k -nearest-neighbors decision rule classifies an object based on the class of the k data points nearest to the estimation point x_0 . The output is given by the class with the most representative within the k nearest neighbors. Nearness is most commonly measured using the Euclidean distance metric in x -space (Davies E. R., 1997; Vladimir & Filip, 1998).

Bayes' decision. Bayesian decision theory is a fundamental statistical approach to the problem of pattern recognition. This approach is based on quantifying the tradeoffs between various classification decisions using probability and the costs that accompany such decisions. It makes the assumption that the decision problem is posed in probabilistic terms, and that all of the relevant probability values are known (Duda et al., 2001).

Neural Networks. Artificial neural networks are an attempt at modeling the information processing capabilities of nervous systems. Some parameters modify the capabilities of the network and it is our task to find the best combination for the solution of a given problem. The adjustment of the parameters will be done through a learning algorithm, i.e., not through explicit programming but through an automatic adaptive method (Rojas R., 1996).

Gaussian Mixture Model. A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model (Reynolds D. A., 2008).

Support Vector Machine. The Support Vector Machine (SVM) is a universal constructive learning procedure based on the statistical learning theory. The term “universal” means

that the SVM can be used to learn a variety of representations, such as neural net (with the usual sigmoid activation), radial basis function, splines, and polynomial estimators. In more general sense the SVM provides a new form of parameterization of functions, and hence it can be applied outside predictive learning as well (Vladimir & Filip, 1998).

In This chapter presents only two classifiers, the Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) as it classifiers are two of the most frequently used on different pattern recognition systems, and then a detailed explanation and evaluation of the operation of these classifier is required.

2. Gaussian Mixture Model

2.1 Introduction.

Gaussian Mixture Models can be used to represent probability density functions complex, from the marginalization of joint distribution between observed variables and hidden variables. Gaussian mixture model is based on the fact that a significant number of probability distributions can be approximated by a weighted sum of Gaussian functions as shown in Fig. 2. Use of this classifier has excelled in the speaker's recognition with very good results (Reynolds & Rose, 1995; Reynolds D. A., 2008).

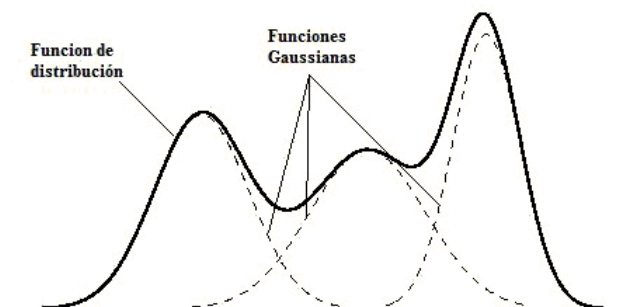


Fig. 2. Approximation of a probability distribution function by a weighted sum of Gaussian functions.

To carry out the development of Gaussian Mixture Model must consider 3 very important points:

- Model initialization.
- Model development.
- Model evaluation.

2.1.1 Model initialization

Gaussian mixture models allow grouping data. The K-means algorithm is an algorithm that corresponds to a non-probabilistic limit, particular of the maximum likelihood estimation applied to Gaussian mixtures.

The problem is to identify data groups in a multidimensional space. It involves a set x_1, \dots, x_N of a random variable of D -dimensions in a Euclidean space. A group can be thought of as a data set whose distance between them is small compared to the distance to points outside the group.

Introducing a set of D -dimensional vectors μ_k , with $k = 1, 2, \dots, K$ where μ_k is the prototype associated with the k -th group. The goal is to find an assignment of the observed data to the groups, as well as a set of vectors μ_k as to minimize the sum of the squares of the distances between each point to its nearest vector μ_k .

For example, initially select the first M feature vectors as the initial centers, as shown in Figure 3, ie:

$$\mu_i = X_i \tag{1}$$

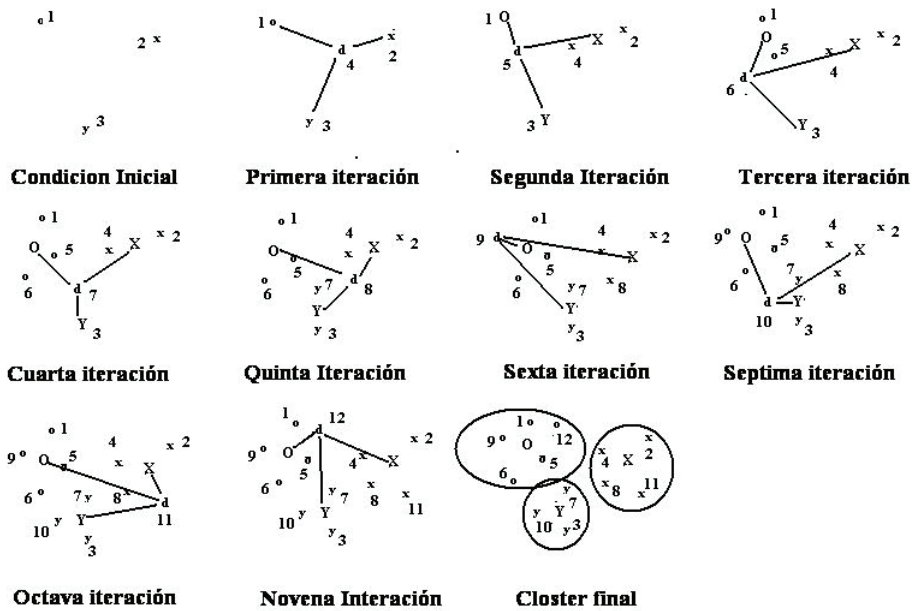


Fig. 3. Illustration of K-Means algorithm for M3.

Then is added a vector more and get the distance between the new vector and M centers, determining that the new vector belongs to the center with which the distance is the lowest. Subsequently the new center is calculated by averaging the items belonging to the center. Thus denoting by $X_{i,j}$ the characteristic vectors belonging to the center $\mu = k$, the new center is given by:

$$\mu_k = \frac{1}{N} \sum_{j=1}^N X_{k,j} \tag{2}$$

This process is repeated until the distance between the $k - th$ center on the iteration n and $n + 1$ is less than a given constant.

Figure 3 shows that the first three vectors are used as initial centers. Then insert the fourth vector which has the shortest distance from the center x . Subsequently the new center is

calculated by averaging the two vectors belonging to the center X . Then is analyzed the vector 5, which has a shorter distance from the center O , which is modified by using the vectors 1 and 5, as shown in the second iteration in Figure 3. Then is analyzed the vector 6, which has a minimum distance to the center O . Here the new center O vectors are modified using 1, 5 and 6. The process continues until ninth iteration, where the center O is calculated using the vectors 1, 5, 6, 9, 12; the center X is calculated using the vector 2, 4, 8, 11, while the Y is obtained 3, 7, 10 from the vectors.

After obtaining the centers, the variance of each center is obtained using the relationship:

$$\sigma_k = \frac{1}{N} \sum_{j=1}^{N_k} (\mu_k - X_{k,j})^2 \quad (3)$$

2.1.2 Model development

Gaussian Mixture Models(GMM) are statistical modeling methods while a model is defined as a mixture of a certain numbers of Gaussian functions for the feature vectors (Jin et al., 2004). A Gaussian mixture density is a weighted sum of M component densities, this is shown in Figure 4 and obtained by the following equation:

$$p(x|\lambda) = \sum_{i=1}^M p_i b_i(x) \quad (4)$$

Where x is a N -dimensional vector, $b_i(\vec{x}), i = 1, 2, \dots, M$, are the components of density and $p_i, i = 1, 2, \dots, M$, are weights of the mixtures. Each component density is a D -Gaussian variation of the form:

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (x - \mu_i)' \sigma_i^{-1} (x - \mu_i) \right\} \quad (5)$$

Where $()'$ denotes the transposed vector, $\mu - i$ denotes the average value of N dimensions and σ_i covariance matrix which is diagonal, and p_i the distribution of weights which satisfy the relationship:

$$\sum_{i=1}^M p_i = 1 \quad (6)$$

So the distribution model is determined by the mean vector, covariance matrix and the weights of the distribution with which the model is represented as:

$$\lambda = p_i, \mu_i, \sigma_i, \quad i = 1, 2, \dots, M \quad (7)$$

The estimation of system parameters using the ML algorithm (Maximum Likelihood) seeks to find the parameters to approximate the best possible distribution of the characteristics of the face under analysis and will seek to find the parameters of λ to maximize distribution. For a sequence of T training vectors $X = x_1, \dots, x_T$, the GMM likelihood can be written as:

$$p(X|\lambda) = \prod_{t=1}^T p(X_t|\lambda) \quad (8)$$

Unfortunately, Equation 8 is nonlinear in relation to the parameters of λ , so to is possible to maximize directly, so it must use an iterative algorithm called Baum-Welch. Baum-Welch

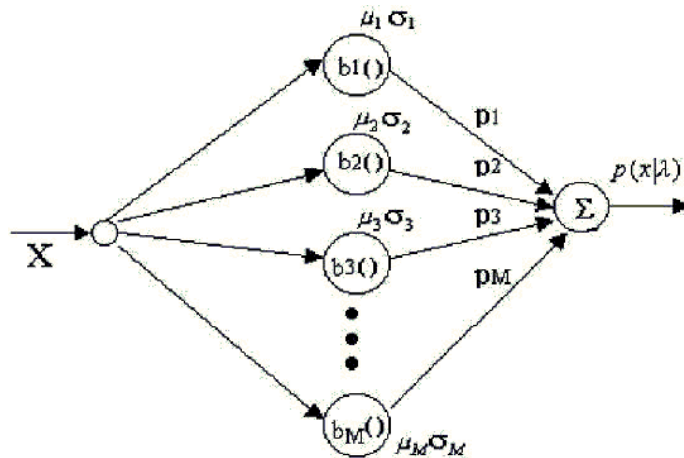


Fig. 4. Gaussian Mixture Model, GMM

algorithm is used by HMM algorithm to estimate its parameters and has the same basic principle of the algorithm of Expectation Maximization (EM Expectation-Maximization), which is part of an initial set of parameters $\lambda(r - 1)$ and a new model is estimated $\lambda(r)$, where r denotes the $r - th$ iteration, so to:

$$p(X|\lambda(r)) \geq P(X|\lambda(r - 1)) \tag{9}$$

Thus, this new model $(\vec{\lambda})$, becomes the initial model for the next iteration. Each T elements must update the model parameters as follows:

Pesos de la mezcla

$$p_i = \frac{1}{T} \sum_{t=1}^T p(i|X_{t+k}, \lambda) \tag{10}$$

Media

$$\mu_i = \frac{\sum_{t=1}^T p(i|X_{t+k}, \lambda) X_{t+k}}{\sum_{t=1}^T p(i|X_{t+k}, \lambda)} \tag{11}$$

Covarianza

$$\sigma_i = \frac{\sum_{t=1}^T p(i|X_{t+k}, \lambda) (X_{t+k} - \sigma_i)^2}{\sum_{t=1}^T p(i|X_{t+k}, \lambda)} \tag{12}$$

To calculate the posterior probability is obtained by:

$$p(i|X_{t+k}, \lambda) = \frac{p_i b_i(X_{t+k})}{\sum_{j=1}^M p_j b_j(X_{t+k})} \tag{13}$$

2.1.3 Model evaluation

In order to carry out the evaluation of the model considers that the system will be used to identify R people, which are represented by models $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_R$. The aim is then to find the model with maximum posterior probability for a given observation sequence. Formally, the person identified is one that satisfies the relation:

$$\hat{R} = \arg \max Pr(\lambda_k|X), \quad k = 1, 2, \dots, R \quad (14)$$

Using Bayes theorem, equation 14 can be expressed as:

$$\hat{R} = \arg \max \frac{p(X|\lambda_k)Pr(\lambda_k)}{p(X)}, \quad k = 1, 2, \dots, R \quad (15)$$

Assuming to the probability of each person is equally likely, then $Pr(\lambda_k) = \frac{1}{R}$ and taking into account to $P(X)$ is the same for all models of speakers, equation 15 simplifies to:

$$\hat{R} = \arg \max p(X|\lambda_k), \quad k = 1, 2, \dots, R \quad (16)$$

Replacing $p(X|\lambda_k)$,

$$p(X|\lambda) = \prod_{t=1}^T p(X_t|\lambda_k) \quad (17)$$

in equation 16 yields:

$$\hat{R} = \arg \max \prod_{t=1}^T p(X_t|\lambda_k), \quad k = 1, 2, \dots, R \quad (18)$$

Finally using logarithms have:

$$\hat{R} = \arg \max \sum_{t=1}^T \log_{10}(p(X_t|\lambda_k)), \quad k = 1, 2, \dots, R \quad (19)$$

where $p(X_t|\lambda_k)$ is given by the equation 4, that is by the output of the system shown in Figure 4.

3. Support Vector Machine

The Support Vector Machine (SVM) (Vladimir & Filip, 1998) is a universal constructive learning procedure based on the statistical learning theory. Unlike conventional statistical and neural network methods, the SVM approach does not attempt to control model complexity by keeping the number of features small. Instead, with SVM the dimensionality of z-space can be very large because the model complexity is controlled independently of its dimensionality. The SVM overcomes two problems in its design: The *conceptual problem* is how to control the complexity of the set of approximating functions in a high-dimensional space in order to provide good generalization ability. This problem is solved by using penalized linear estimators with a large number of basis functions. The *computational problem* is how to perform numerical optimization in a high-dimensional space. This problem is solved by taking advantage of the dual kernel representation of linear functions.

The SVM combines four distinct concepts:

1. *New implementation of the SRM inductive principle.* The SVM use a special structure that keeps the value of the empirical risk fixed for all approximating functions but minimizes the confidence interval.
2. *Input samples mapped onto a very high-dimensional space using a set of nonlinear basis functions defined a priori.* It is common in pattern recognition applications to map the input vectors into a set of new variables which are selected according to a priori assumptions about the

learning problem. For the support vector machine, complexity is controlled independently of the dimensionality of the feature space (z-space).

3. *Linear functions with constraints on complexity used to approximate or discriminate the input samples in the high-dimensional space.* The support vector machine uses a linear estimator to perform approximation. As it, nonlinear estimators potentially can provide a more compact representation of the approximation function; however, they suffer from two serious drawbacks: lack of complexity measures and lack of optimization approaches which provide a globally optimal solution.
4. *duality theory of optimization used to make estimation of model parameters in a high-dimensional feature space computationally tractable.* For the SVM, a quadratic optimization problem must be solved to determine the parameters of a linear basis function expansion. For high-dimensional feature spaces, the large number of parameters makes this problem intractable. However, in its dual form this problem is practical to solve, since it scales in size with the number of training samples. The linear approximating function corresponding to the solution of the dual is given in the kernel representation rather than in the typical basis function representation. The solution in the kernel representation is written as a weighted sum of the support vectors.

3.1 Optimal separating hyperplane

A separating hyperplane is a linear function that is capable of separating the training data without error (see Fig. 5). Suppose that the training data consisting of n samples $(x_1, y_1), \dots, (x_n, y_n), x \in \mathbb{R}^d, y \in \{+1, -1\}$ can be separated by the hyperplane decision function

$$D(x) = (w \cdot x) + w_0 \quad (20)$$

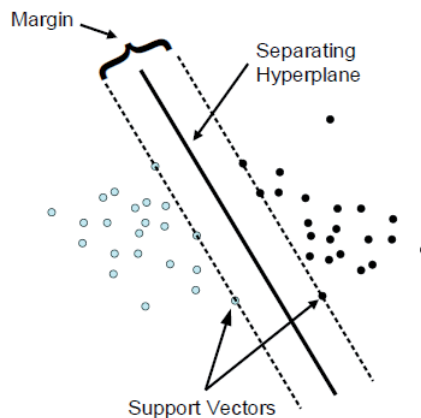


Fig. 5. Classification (linear separable case)

with appropriate coefficients w and w_0 . A separating hyperplane satisfies the constraints that define the separation of the data samples:

$$\begin{aligned} (w \cdot x) + x_0 &\geq +1 && \text{if } y_i = +1 \\ (w \cdot x) + x_0 &\leq -1 && \text{if } y_i = -1, i = 1, \dots, n \end{aligned} \quad (21)$$

For a given training data set, all possible separating hyperplanes can be represented in the form of equation 21.

The minimal distance from the separating hyperplane to the closest data point is called the *margin* and will be denoted by τ . A separating hyperplane is called *optimal* if the margin is the maximum size. The distance between the separating hyperplane and a sample x' is $|D(x')|/||w||$, assuming that a margin τ exists, all training patterns obey the inequality:

$$\frac{y_k D(x_k)}{||w||} \geq \tau, \quad k = 1, \dots, n \quad (22)$$

where $y_k \in -1, 1$.

The problem of finding the optimal hyperplane is that of finding the w that maximizes the margin τ . Note that there are an infinite number of solutions that differ only in scaling of w . To limit solutions, fix the scale on the product of τ and norm of w ,

$$\tau ||w|| = 1 \quad (23)$$

Thus maximizing the margin τ is equivalent to minimizing the norm of w . An optimal separating hyperplane is one that satisfies condition (21) above and additionally minimizes

$$\eta(w) = ||w||^2 \quad (24)$$

with respect to both w and w_0 . The margin relates directly to the generalization ability of the separating hyperplane. The data points that exist at margin are called the *support vectors* (Fig. 5). Since the support vectors are data points closest to the decision surface, conceptually they are the samples that are the most difficult to classify and therefore define the location of the decision surface.

The generalization ability of the optimal separating hyperplane can be directly related to the number of support vectors.

$$E_n[\text{Errorrate}] \leq \frac{E_n[\text{Number of support vectors}]}{n} \quad (25)$$

The operator E_n denotes expectation over all training sets of size n . This bound is independent of the dimensionality of the space. Since the hyperplane will be employed to develop the support vector machine, its VC-dimension must be determined in order to build a nested structure of approximating functions.

For the hyperplane functions (21) satisfying the constraint $||w||^2 \leq c$, the VC-dimension is bounded by

$$h \leq \min(r^2 c, d) + 1 \quad (26)$$

where r is the radius of the smallest sphere that contains the training input vectors (x_1, \dots, x_n) . The factor r provides a scale in terms of the training data for c . With this measure of the VC-dimension, it is now possible to construct a structure on the set of hyperplanes according to increasing complexity by controlling the norm of the weights $||w||^2$:

$$S_k = (w \cdot x) + w_0 : ||w||^2 \leq c_k, c_1 < c_2 < c_3 \dots \quad (27)$$

The structural risk minimization principle prescribes that the function that minimizes the guaranteed risk should be selected in order to provide good generalization ability.

By definition, the separating hyperplane always has zero empirical risk, so the guaranteed risk is minimized by *minimizing the confidence interval*. The confidence interval is minimized

by minimizing the VC-dimension h , which according to (26) corresponds to minimizing the norm of the weights $\|w\|^2$. Finding an optimal hyperplane for the separable case is a quadratic optimization problem with linear constraints, as formally stated next.

Determine the w and w_0 that minimize the functional

$$\eta(w) = \frac{1}{2} \|w\|^2 \quad (28)$$

subject to the constraints

$$y_i[(w \cdot x) + w_0] \geq 1, \quad i = 1, \dots, n \quad (29)$$

given the training data $(x_i, y_i), i = 1, \dots, n, x \in \mathbb{R}^d$. The solution to this problem consists of $d + 1$ parameters. For data of moderate dimension d , this problem can be solved using quadratic programming.

For training data that cannot be separated without error, it would be desirable to separate the data with a minimal number of errors. In the hyperplane formulation, a data point is nonseparable when it does not satisfy equation (21). This corresponds to a data point that falls within the margin or on the wrong side of the decision boundary. Positive slack variables $\xi_i, i = 1, \dots, n$, can be introduced to quantify the nonseparable data in the defining condition of the hyperplane:

$$y_i[(w \cdot x) + w_0] \geq 1 - \xi_i \quad (30)$$

For a training sample x_i , the slack variable ξ_i is the deviation from the margin border corresponding to the class of y_i ; see Fig. 6. According to our definition, slack variables greater than zero correspond to misclassified samples. Therefore the number of nonseparable samples is

$$Q(w) = \sum_{i=1}^n I(\xi_i > 0) \quad (31)$$

Numerically minimizing this functional is a difficult combinatorial optimization problem because of the nonlinear indicator function. However, minimizing (31) is equivalent to minimizing the functional

$$Q(\xi) = \sum_{i=1}^n \xi_i^p \quad (32)$$

where p is a small positive constant. In general, this minimization problem is NP-complete. To make the problem tractable, p will be set to one.

3.2 Inner product kernel

The inner product kernel (H) is known a priori and used to form a set of approximating functions, this is determined by the sum

$$H(x, x') = \sum_{j=1}^m g_j(x)g_j(x') \quad (33)$$

where m may be infinite.

Notice that in the form (33), the evaluation of the inner products between the feature vectors in a high-dimensional feature space is done indirectly via the evaluation of the kernel H between support vectors and vectors in the input space. The selection of the type of kernel function

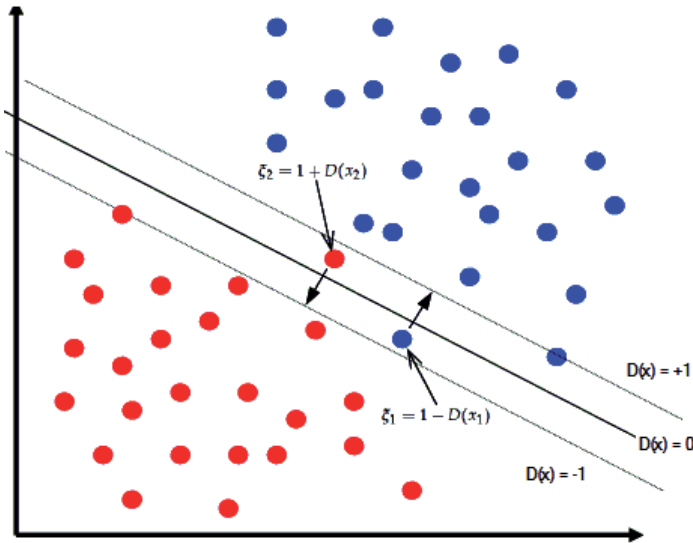


Fig. 6. Nonseparable case

corresponds to the selection of the class of functions used for feature construction. The general expression for an inner product in Hilbert space is

$$(z \cdot z') = H(x, x') \tag{34}$$

where the vectors z and z' are the image in the m -dimensional feature space and vectors x and x' are in the input space.

Below are several common classes of multivariate approximating functions and their inner product kernels:

Polynomials of degree q have inner product kernel

$$H(x, x') = [(x \cdot x') + 1]^q \tag{35}$$

Radial basis functions of the form

$$f(x) = \text{sign} \left(\sum_{i=1}^n \alpha_i \exp \left\{ -\frac{|x - x_i|^2}{\sigma^2} \right\} \right) \tag{36}$$

where σ defines the width have the inner product kernel

$$H(x, x') = \exp \left\{ -\frac{|x - x'|^2}{\sigma^2} \right\} \tag{37}$$

Fourier expansion

$$f(x) = v_0 + \sum_{j=1}^q (v_j \cos(jx) + w_j \sin(jx)) \tag{38}$$

has a kernel

$$H(x, x') = \frac{\sin(q + \frac{1}{2})(x - x')}{\sin(x - x')/2} \tag{39}$$

4. Evaluation results

Here are some results with both classifiers, GMM and SVM combined with some feature extraction methods mentioned above, like Gabor, Wavelet and Eigenphases. The results shown were performed using the database "The AR Face Database" (Martinez, 1998) is used, which has a total of 9,360 face images of 120 people (65 men and 55 women) that includes face images with several different illuminations, facial expression and partial occluded face images with sunglasses and scarf. Two different training set are used, the first one consists on images without occlusion, in which only illumination and expressions variations are included. On the other hand the second image set consist of images with and without occlusions, as well as illumination and expressions variations. Here the occlusions are a result of using sunglasses and scarf. These images sets and the remaining images of the AR face database are used for testing.

Tables 1 and 2 shows the recognition performance using the GMM as a classifier. The recognition performance obtained using the Gabor filters-based, the wavelet transform-based and eigenphases features extraction methods are shown form comparison. Table 1 shows that when the training set 1 is used for training, with a GMM as classifier, the identification performance decrease in comparison with the performance obtained using the training set 2. This is because the training set 1 consists only of images without occlusion and then system cannot identify several images with occlusion due to the lack of information about the occlusion effects. However when the training set 2 is used the performance of all of them increase, because the identification system already have information about the occlusion effects.

	Image set 1	Image set 2
Gabor	71.43 %	91.53 %
Wavelet	71.30 %	92.51 %
Eigenphases	60.63 %	87.24 %

Table 1. Recognition using GMM

Average	Image set 1		Image set 2	
	False acceptance	False reject	False acceptance	False reject
Gabor	4.74 %	7.26 %	1.98 %	4.13 %
Wavelet	6.69 %	6.64 %	1.92 %	5.25 %
Eigenphases	37.70 %	14.83 %	21.39 %	21.46 %

Table 2. Verification using GMM

Tables 3 and 4 show the obtained results with Gabor filters, Wavelet and Eigenphases as features extractors methods in combination with SVM for identification and verification task. Also shows the same characteristics like GMM when the training set 1 and training set 2 are used for training.

Figs. 7 and 8 shows the ranking performance evaluation of Gabor, Wavelets and eigenphases feature extractions methods, using GMM for identification and Figs. 9 and 10 shows the ranking performance evaluation of Gabor, Wavelet and Eigenphases with the Support Vector Machine for identification.

In Figs. 11-13 shows the evaluation of the GMM as verifier using different thresholds for acceptance in these graphs shows the performance of both the false acceptance and the false rejection. Showing the moment when both have the same percentage, depending on the needs

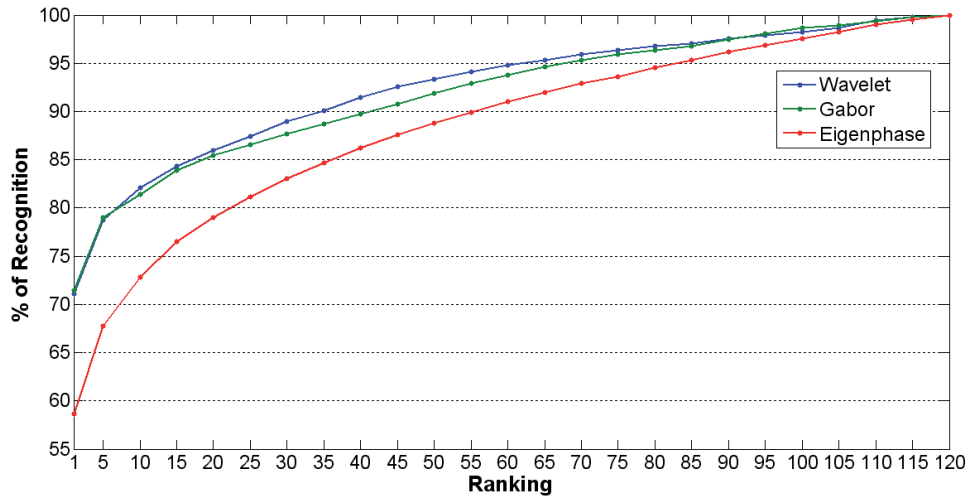


Fig. 7. Ranking performance evaluation using GMM and Training set 1.

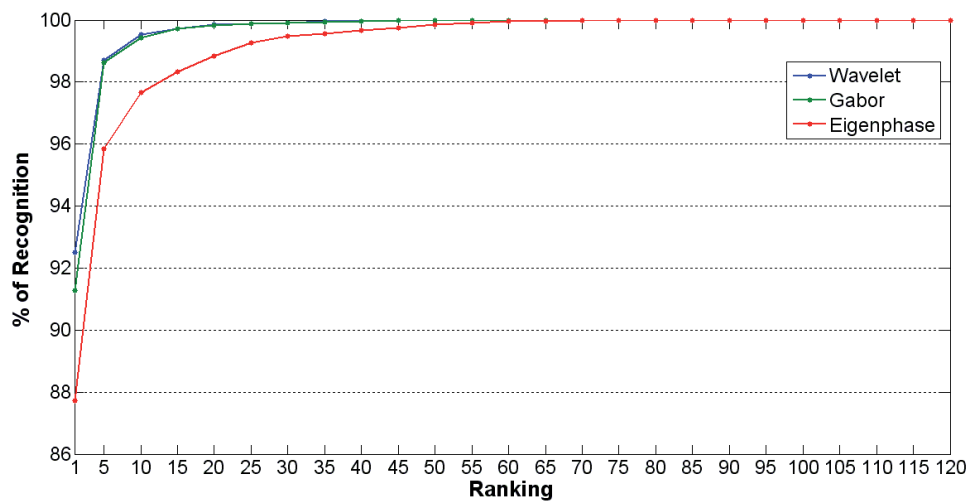


Fig. 8. Ranking performance evaluation using GMM and Training set 2.

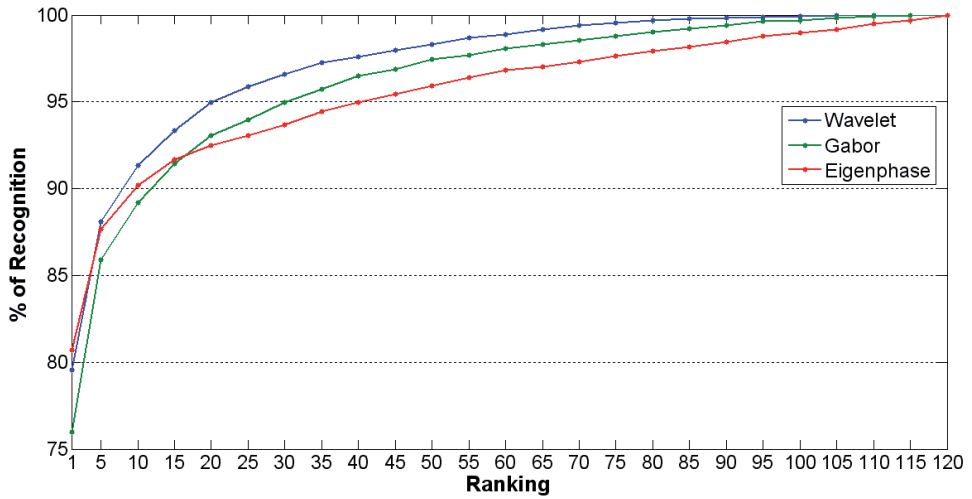


Fig. 9. Ranking performance evaluation using SVM and Training set 1.

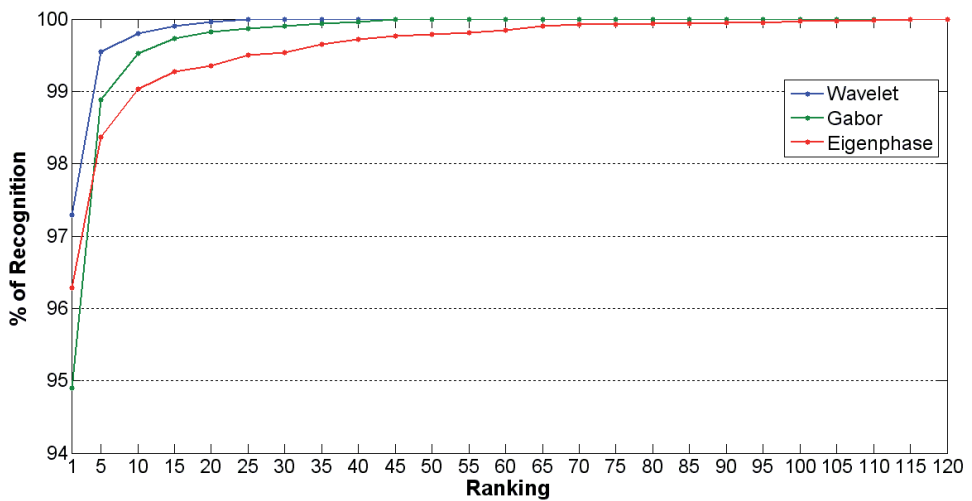


Fig. 10. Ranking performance evaluation using SVM and Training set 2.

	Image set 1	Image set 2
Gabor	75.98 %	94.90 %
Wavelet	79.54 %	97.29 %
Local 3	84.44 %	97.67 %
Local 6	81.05 %	97.29 %
Local fourier 3	85.92 %	97.92 %
Local fourier 6	85.59 %	97.85 %
Eigenphases	80.63 %	96.28 %

Table 3. Recognition using SVM

Average	Image set 1		Image set 2	
	False acceptance	False reject	False acceptance	False reject
Gabor	0.43 %	22.65 %	0.12 %	8.38 %
Wavelet	0.17 %	22.27 %	0.04 %	4.64 %
Eigenphases	0.001 %	34.74 %	0.002 %	16.04 %

Table 4. Verification using SVM

will have to choose a threshold. In Figs. 14-16 shows the evaluation of the SVM as verifier using also different thresholds.

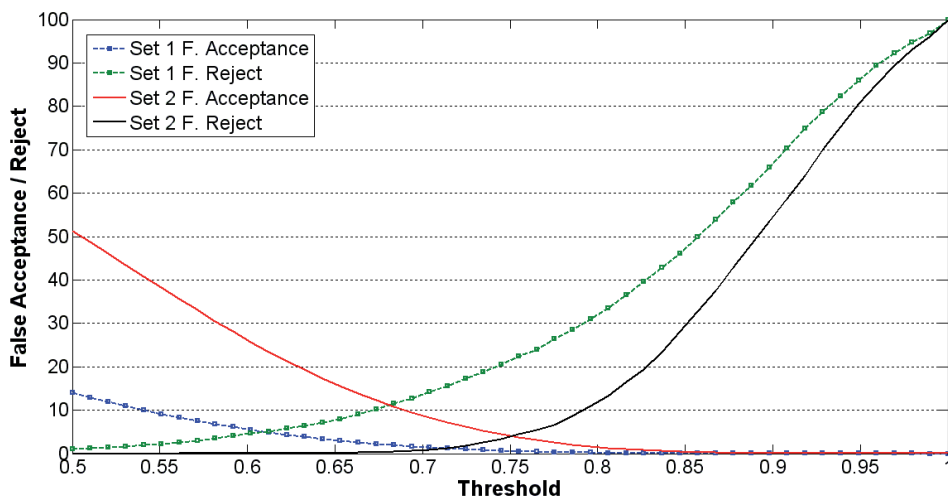


Fig. 11. Verification performance of Gabor-based feature extraction method, for several threshold values using GMM.

5. Conclusion

In this chapter presented two classifiers that can be used for face recognition, and shown some evaluation results where the GMM and SVM are used for identification and verification tasks. Two different image sets were used for training. One contains images with occlusion and the

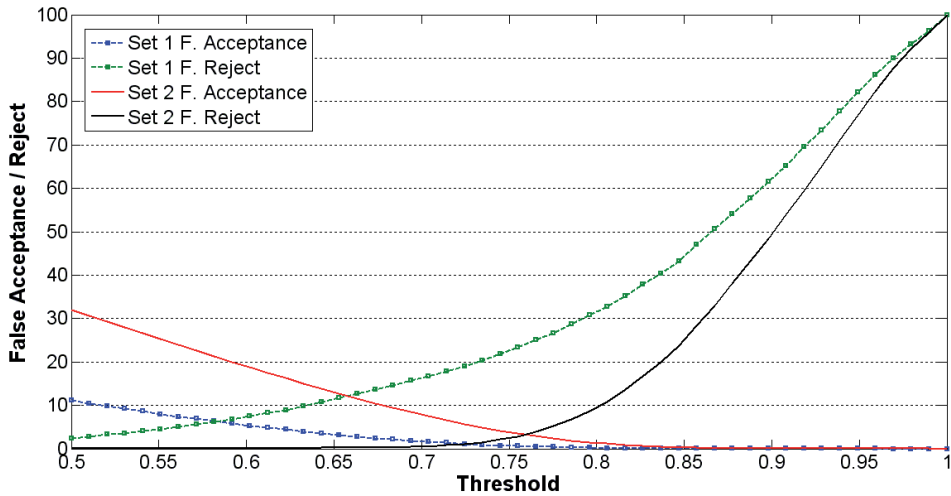


Fig. 12. Verification performance of Wavelet-based feature extraction method, for several threshold values using GMM.

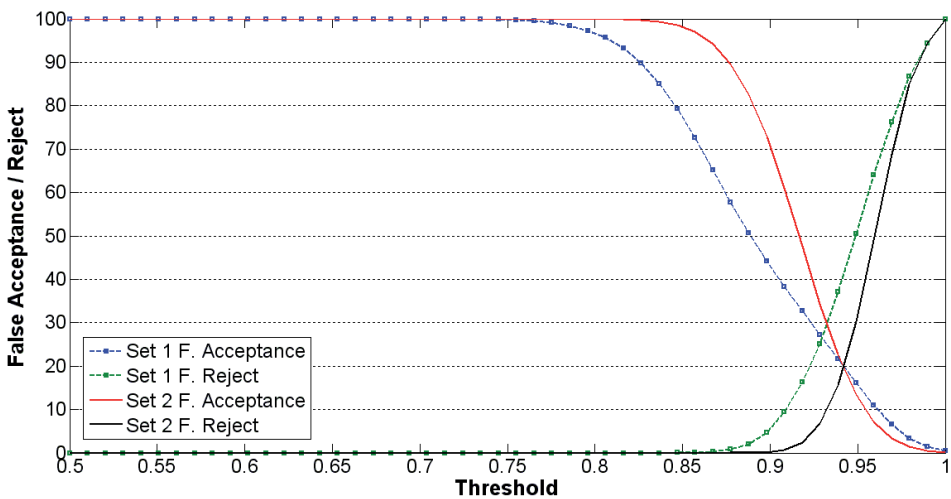


Fig. 13. Verification performance of Eigenphases feature extraction method, for several threshold values using GMM.

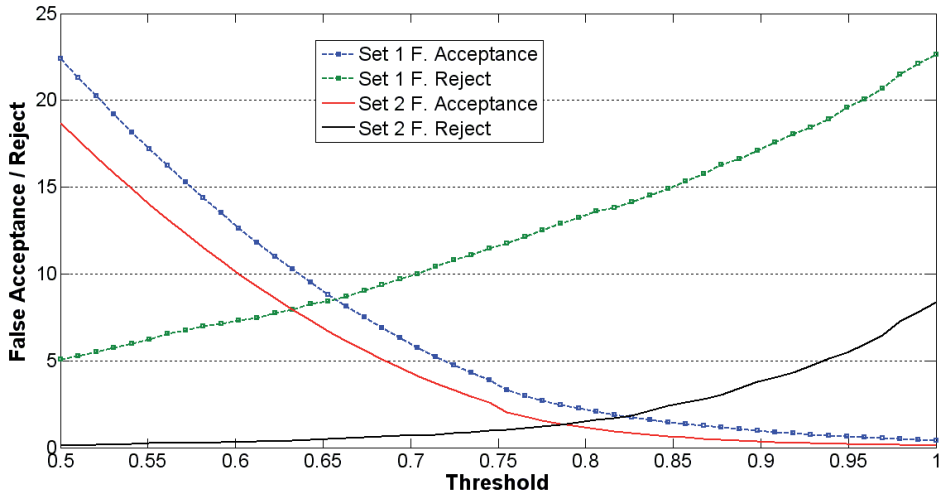


Fig. 14. Verification performance of Gabor-based feature extraction method, for several threshold values using SVM.

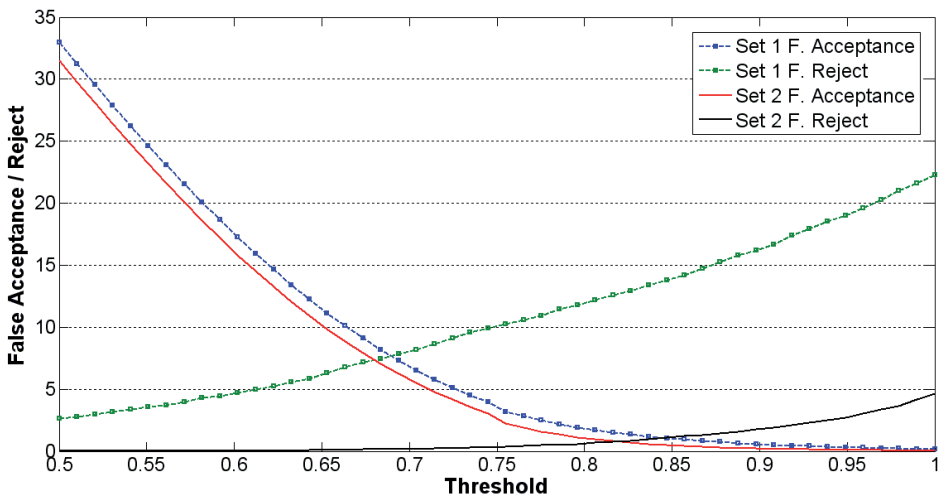


Fig. 15. Verification performance of Wavelet-based feature extraction method, for several threshold values using SVM.

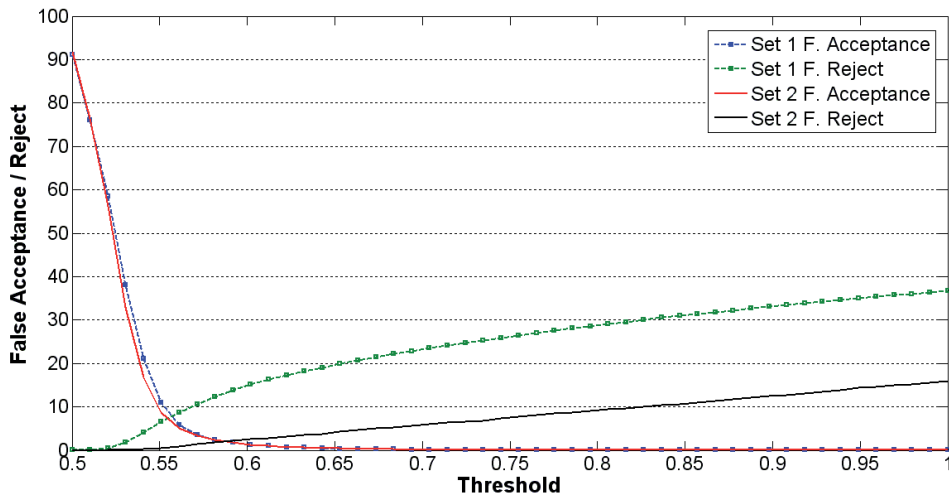


Fig. 16. Verification performance of Eigenphases feature extraction method, for several threshold values using SVM.

other one contains images without occlusions. The performance of this classifiers are shown using the Gabor-based, Wavelet-based and Eigenphases method for feature extraction.

It is important to mention, at the verification task it is very important to keep the false acceptance rate as low as possible, without much increase of the false rejection rate. To find a compromise between both errors, evaluation results of both errors with different thresholds are provided. To evaluate the performance of proposed schemes when they are required to perform an identification task the rank(N) evaluation was also estimated.

Evaluation results show that, in general, the SVM performs better than the GMM, specially, when the training set is relatively small. This is because the SVM uses a supervised training algorithm and then it requires less training patterns to estimate a good model of the person under analysis. However it requires to jointly estimating the models of all persons in the database and then when a new person is added, the all previously estimated models must be computed again. This fact may be an important limitation when the database changes with the time, as well as, when huge databases must be used, as in banking applications. On the other hand, because the GMM is uses a non-supervised training algorithm, it requires a larger number of training patterns to achieve a good estimation of the person under analysis and then its convergence is slower than those of SVM, however the GMM estimated the model of each person independently of that of the other persons in the database. It is a very important feature when large number of persons must be identified and the number of persons grows with the time because, using the GMM, when a new person is added, only the model of the new person must be added, remaining unchanged the previously estimated ones. Thus the GMM is suitable for applications when large databases must be handed and they change with the time, as in banking operations. Thus in summary, the SVM is more suitable when the size of databases under analysis is almost constant and it is not so large, while the GMM is more suitable for applications in which the databases size is large and it changes with the time.

6. References

- Alvarado G.; Pedrycz W.; Reformat M. & Kwak K. C. (2006). Deterioration of visual information in face classification using Eigenfaces and Fisherfaces. *Machine Vision and Applications*, Vol. 17, No. 1, April 2006, pp. 68-82, ISSN: 0932-8092.
- Bai-Ling Z.; Haihong Z. & Shuzhi S. G. (2004). Face recognition by applying wavelet subband representation and kernel associative memory. *Neural Networks, IEEE Transactions on*, Vol. 15, Issue:1, January. 2004, pp. 166 - 177, ISSN: 1045-9227.
- Chellapa R.; Sinha P. & Phillips P. J.(2010), Face recognition by computers and humans. *Computer Magazine*, Vol. 43, Issue 2, february 2010, pp. 46-55, ISSN: 0018-9162.
- Davies E. R. (1997). *Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, ISBN: 0-12-206092-X, San Diego, California, USA.
- Duda O. R; Hart E. P & Stork G. D. (2001). *Pattern Classification*, Wiley-Interscience, ISBN: 0-471-05669-3, United States of America.
- Hazem. M. El-Bakry & Mastorakis N. (2009). Personal identification through biometric technology, *AIC'09 Proceedings of the 9th WSEAS international conference on Applied informatics and communications*, pp. 325-340, ISBN: 978-960-474-107-6, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.
- Jain, A.K.; Ross, A. & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 14, Jan. 2004, pp. 4-20, ISSN: 1051-8215.
- Jin Y. K.; Dae Y. K. & Seung Y. N. (2004), Implementation and enhancement of GMM face recognition systems using flatness measure. *IEEE Robot and Human Interactive Communication*, Sept. 2004, pp. 247 - 251, ISBN: 0-7803-8570-5.
- Martinez A. M. & Benavente R. (1998). The AR Face Database. CVC Technical Report No. 24, June 1998.
- Olivares M. J.; Sanchez P. G.; Nakano M. M. & Perez M. H. (2007). Feature Extraction and Face Verification Using Gabor and Gaussian Mixture Models. *MICAI 2007: Advances in Artificial Intelligence*, Gelbukh A. & Kuri M. A., pp. 769-778, Springer Berlin / Heidelberg.
- Reynolds D.A. & Rose R.C. (1995). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, Vol. 3, Issue 1, Jan. 1995, pp. 72-83, ISSN: 1063-6676.
- Reynolds D.A. (2008). Gaussian Mixture Models, *Encyclopedia of Biometric Recognition*, Springer, Feb 2008, , ISBN 978-0-387-73002-8.
- Rojas R. (1995). *Neural networks: a systematic introduction*, Springer-Verlag, ISBN: 3-540-60505-3, New York.
- Savvides, M.; Kumar, B.V.K.V. & Khosla, P.K. (2004). Eigenphases vs eigenfaces. *Pro-ceedings of the 17th International Conference on Pattern Recognition*, Vol. 3, Aug. 2004, pp. 810-813, ISSN: 1051-4651.
- Vladimir C. & Filip M. (1998). *Learning From Data: Concepts, Theory, and Methods*, Wiley Inter-Science, ISBN: 0-471-15493-8, USA.
- Yoshida M.; Kamio T. & Asai H. (2003). Face Image Recognition by 2-Dimensional Discrete Walsh Transform and Multi-Layer Neural Network, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer.*, Vol. E86-A, No. 10, October 2003, pp.2623-2627, ISSN: 0916-8508.
- Zhao W.; Chellappa R.; Phillips P. J. & RosenfeldA. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, Vol. 35, Issue 4, December 2003, pp. 399-459.



Reviews, Refinements and New Ideas in Face Recognition

Edited by Dr. Peter Corcoran

ISBN 978-953-307-368-2

Hard cover, 328 pages

Publisher InTech

Published online 27, July, 2011

Published in print edition July, 2011

As a baby one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section on Statistical Face Models and Classifiers presents reviews and refinements of some well-known statistical models. The next section presents two articles exploring the use of Infrared imaging techniques and is followed by few articles devoted to refinements of classical methods. New approaches to improve the robustness of face analysis techniques are followed by two articles dealing with real-time challenges in video sequences. A final article explores human perceptual issues of face recognition.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jesus Olivares-Mercado, Gualberto Aguilar, Karina Toscano-Medina, Mariko Nakano and Hector Perez Meana (2011). GMM vs SVM for Face Recognition and Face Verification, Reviews, Refinements and New Ideas in Face Recognition, Dr. Peter Corcoran (Ed.), ISBN: 978-953-307-368-2, InTech, Available from: <http://www.intechopen.com/books/reviews-refinements-and-new-ideas-in-face-recognition/gmm-vs-svm-for-face-recognition-and-face-verification>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.