Face Recognition Using SVM Based on LDA

Anissa Bouzalmat[1], Jamal Kharroubi[2] and Arsalane Zarghili[3]

[1] Department of Computer Science faculty of Science and Technology, Sidi Mohamed Ben Abdellah University, Route d'Imouzzer Fez, 2202/30000 Morocco

[2] Department of Computer Science faculty of Science and Technology, Sidi Mohamed Ben Abdellah University, Route d'Imouzzer Fez, 2202/30000 Morocco

[3] Department of Computer Science faculty of Science and Technology, Sidi Mohamed Ben Abdellah University, Route d'Imouzzer Fez, 2202/30000 Morocco

Abstract

We present a method for face recognition that investigate the overall performance of linear ,polynomial and RBF kernel of SVM for classification based on global approach and used images having different expression variations, pose and complex backgrounds. In the first we reduce dimensional feature vector by LDA method , the result of vectors feature propagates to a set of SVM classifier, we trained SVM classifier with linear and non linear kernel for each dataset (face94, face96, grimaces)[1,2,3] in the database . Experiments demonstrate that use the LDA method combined with SVM classifier and the choice of a suitable kernel function with optimal parameters can produce high classification accuracy compared to KNN classifier on a variety of images on different Database.

Keywords: *Face Recognition, SVM, LDA, PCA, KNN.*

1. Introduction

A face recognition system recognizes a face by matching the input image against images of all faces in a database and finding the best match.

It can be roughly divided into two main categories: local and global approaches. In local feature approaches a number of fiducially or control points are extracted and used for classification, while in global approaches the whole image serves as a feature vector, the techniques was developed this approach are Eigen faces ,Linear Discriminate Analysis (LDA)[4] that method outperforms PCA in terms of class discrimination, neural networks [5] and Support Vector Machines (SVM) [6], is considered easier to use and performs particularly well with high dimensional feature vectors and in case of lack of training data ,these factors which may significantly limit the performance of most neural networks [7].

The features constitute of the global image in case of the approach global are a high dimension, so it is difficult to use it without applying reduction method such high dimension of vectors have proven to be one of the biggest problems of face recognition systems then it is necessary to apply a method to reduce dimension of vectors that will perform the process for face recognition algorithm. As one of feature extraction methods for face recognition problem, linear discriminate analysis (LDA) were applied for the images this drastically reduced the number of attributes of feature vectors. By choosing the two of the most popular machine learning algorithms SVM method and K-nearest neighbors that is one of the simplest but effective in many cases in machine learning algorithms [8] then comparing the accuracy for face classification of these two classifiers, We implement a recognition system using SVM and KNN classifiers based on LDA, The procedure is described as follows: The outline of the paper is as follows: Section 2 Description of the proposed method. In section 3 contains experimental results. Section 4 concludes the paper.

2. The Proposed Method

The present study proposed in this paper (Fig 1) is designed for face recognition. The system consists of: a) The whole image serves as a feature vector and reduces it by linear discriminate analysis b) Transform theses feature vectors to the format of an SVM and scale them. Finally, the obtained feature vectors are used as input of classifier support vector machine with different kernel and K-nearest neighbor classifier.

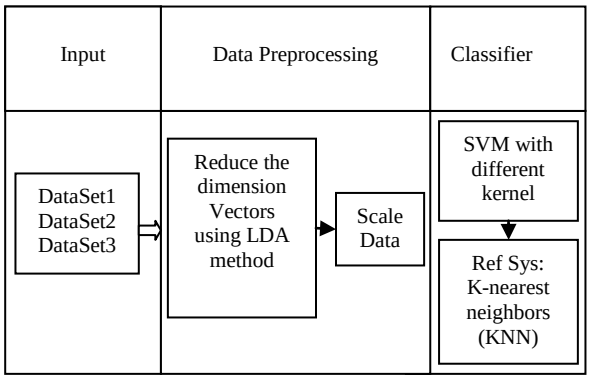


Fig. 1: Architecture description of the proposed approach.

2.1 Construction the features vectors

When the whole images of database were treated as features, we have a large features vector in this situation, the computational expense will increase, especially for non-linear classifiers so
It is important to reduce the dimensionality of features before classification by applying LDA method which is significantly better to perform dimensionality reduction while preserving as much of the class discriminatory information as possible [9] .

2.1.1 Linear discriminate analysis (LDA)

Linear discriminate analysis (LDA) is powerful tools used to reduce dimension of feature vectors without loss of information and used as a feature extraction step before classification [4] ,it is generally believed that algorithms based on LDA are superior to those based on PCA in the lower dimensional subspace [10].
LDA try to maximize class seperability witch determines a subspace in which the between-class scatter (extra personal variability) is as large as possible, while the within-class scatter (intrapersonal variability) is kept constant. In this sense, the subspace obtained by LDA optimally discriminates the classes-faces. The Objective of LDA seeks to reduce dimensionality while preserving as much of the class discriminatory information as possible.
We have a set of C-class and D-dimensional samples $\left\{x^{(1},x^{(2},...x^{(N}\right\}$, N1 of which belong to class w_1, N_2 to class w_2 and N_c to class w_c, in order to find a good discrimination of these classes we need to define a measure of separation, we define a measure of the within-class scatter by Eq. (1):

$$S_w = \sum_{i=1}^{c} S_i$$

$$S_i = \sum_{x \in w_i} (x - \mu_i)(x - \mu_i)^T \qquad (1)$$

$$\mu_i = \frac{1}{N_i} \sum_{x \in w_i} x_i$$

And the between-class scatter Eq. (2) becomes:

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T \qquad (2)$$

$$\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^{C} N_i \mu_i$$

Matrix $S_T = S_B + S_W$ is called the total scatter similarly; we define the mean vector and scatter matrices for the projected samples as:

$$\tilde{S}_W = \sum_{i=1}^{c} \sum_{y \in w_i} (y - \tilde{\mu}_i)(y - \tilde{\mu}_i)^T$$

$$\tilde{S}_B = \sum_{i=1}^{c} N_i (\tilde{\mu}_i - \tilde{\mu})(\tilde{\mu}_i - \tilde{\mu})^T, \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in w_i} y, \quad \tilde{\mu} = \frac{1}{N} \sum_{\forall y} y$$

From our derivation for the two-class problem, we can write: $\begin{cases} \tilde{S}_B = W^T S_B W \\ \tilde{S}_W = W^T S_W W \end{cases}$

Recall that we are looking for a projection that maximizes the ratio of between-class to within-class scatter. Since the projection is no longer a scalar (it has C−1 dimensions), we use the determinant of the scatter matrices to obtain a scalar objective function Eq. (3):

$$J(W) = \frac{\left|\tilde{S}_B\right|}{\left|\tilde{S}_W\right|} = \frac{W^T S_B W}{W^T S_W W} \qquad (3)$$

And we will seek the projection matrix w* that maximizes this ratio It can be shown that the optimal projection matrix w * is the one whose columns are the eigenvectors corresponding to the largest eigen values of the following generalized eigen value problem Eq. (4):

$$w^* = \left[w_1^* \middle| w_2^* \middle| .. w_{c-1}^* \right] = \arg\max \frac{\left|W^T S_B W\right|}{\left\|W^T S_W W\right\|}$$

$$\Rightarrow (S_B - \lambda_i S_W) W_i^* \qquad (4)$$

S_B is the sum of C matrices of rank ≤1 and the mean vectors are constrained by: $\frac{1}{c}\sum_{i=1}^{C}\mu_i = \mu$

Therefore, S_B will be of rank (C−1) or less and this means that only (C−1) of the eigen values λi will be non-zero. The projections with maximum class separability information are the eigenvectors corresponding to the largest eigen values of $S_W^{-1}S_B$ we seek (C−1) projections [y_1, $_2$,…y_{c-1}] by means of (c−1) projection vectors w_i arranged by columns into a projection matrix W=[w_1|w_2|…|w_{c-1}] :

$y_i = w_i Tx \Rightarrow y=WTx$

The LDA be useful in order to reduce dimensionality and speed up the classifier for training.

2.2 Support Vector Machines (SVM)

SVMs (Support Vector Machines) are a useful technique for data classification and are still under intensive research [11],[12]. Although SVM is considered easier to use than Neural Networks, In addition, its true potential is highlighted when the classification of non-linearly separable data becomes possible with the use of a kernel function, which maps the input space into a possibly higher dimensional feature space in order to transform the non-linear decision boundary into a linear one. There exists a range of kernel functions, where a particular function may perform better for certain applications. For more details we describe the theory of support vector machines as a combination of two main concepts: kernel functions and maximal margin hyperplanes.

2.2.1 Scaling Data

After the low-dimensional feature vector were obtained we scale attributes of training data to the range [-1, 1] linearly, then scale the attributes of the test data using the same scaling function of the training data that is very important which is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation.

2.2.2 Kernels functions

The simplest form of SVM is introduced as a hyper plane maximizing the margin of separation on linear separable data sets, in many real-world problems, noisy data will render linear separation impossible so no feasible solution to the margin maximization problem can be found, in this case we consider the non linear classifiers which can be overcome by an approach called kernel technique, it was introduced as the method of potential functions [13] .The general idea of the kernel is to map the input data to a high dimensional space and separate it there by a linear classifier. This will result in a classifier nonlinear in input space.

The mapping $\Phi: \quad R^P \rightarrow F$ is applied to each example before training and the optimal separating hyper plane is constructed in the feature space F (Fig 2).

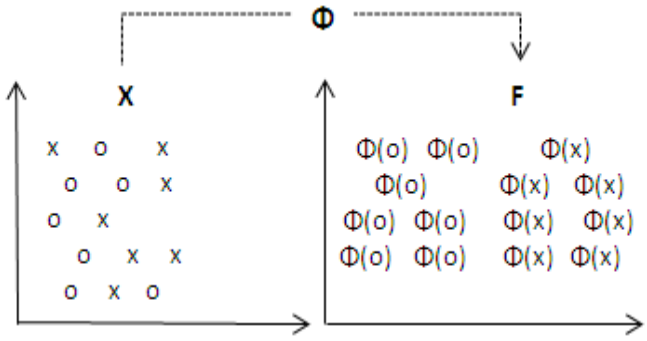


Fig. 2 : The mapping of features to feature space F.

There are several kernels are being proposed by researchers, we lists the kernel expressions and corresponding parameters. The three basic kernels as follow: linear, polynomial, radial basis function (RBF).

The Linear kernel (5) is the simplest kernel function. It is given by the inner product. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts.

$$k(x_i, x_j) = < x_i, x_j > \qquad (5)$$

Note that < xi, xj> represents dot product, where xi and xj denote two arbitrary feature vectors.
For the Polynomial kernels (6) are a non-stationary kernels. They are well suited for problems where all the training data is normalized.

$$k(x_i, x_j) = (gamma \quad < x_i, x_j > + coef0)^{degree} \qquad (6)$$

degree,gamma,coef0 respectively denote degree of the polynomial, coefficient of the polynomial function, and the coadditive constant.

The Gaussian kernel (7) is an example of radial basis function kernel.

$$k(x_i, x_j) = exp(-\gamma \left\| x_i - x_j \right\|^2) \quad (7)$$

The parameter γ denotes the width of the Gaussian radial basis function where: $\gamma = \frac{1}{2\sigma^2}$

The parameter σ plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand. If over estimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its non-linear power. In the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noise in training data.

2.2.3 Maximal Margin Hyperplanes

After we change the representation of the training examples by mapping the data to a features space F where the optimal separating hyper plane (OSH) is constructed Fig 3, we limited our study to the case of two-class discrimination [14] and we consider the training data S a set of l vectors features each vector has n dimension, where each point x_i belongs to one of two classes identified by the label -1 or 1 Eq 8.

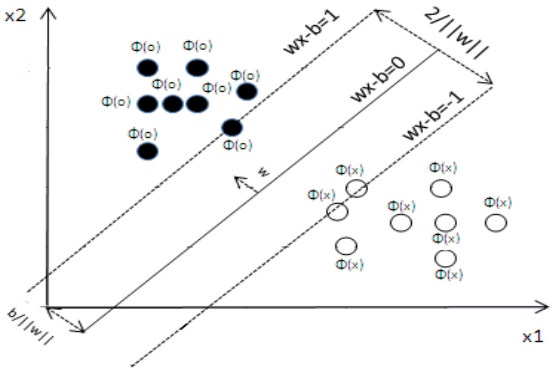


Fig. 3: Maximum-margin hyper plane for SVM trained with samples from two classes.

$$S = \left\{ (x_i, y_i) \middle| x_i \in R^n, y_i \in \{-1,1\} \right\}_{i=1}^{l} \quad (8)$$

We have solving a quadratic optimization problem with linear constraints that can be interpreted in terms of the Lagrange multipliers calculated by quadratic programming Eq: 9

$$\left\{ \begin{array}{c} \max(\alpha_i) : \tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ (\text{for any } i = 1,..,n) \quad 0 \le \alpha i \le c \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{array} \right\} \quad (9)$$

α_i are the Lagrange multipliers parameters to be adjusted , c is the penalty parameter of the classification error term it must be adjusted because the data are rarely completely separable, the x_i are the training examples.
The solution of the optimization problem will be a vector w € F, that can be written as a linear combination of the training inputs Eq: 10

$$w = \sum \alpha_i y_i x_i \quad (10)$$

(w,b) define the hyperplane and b is the bias Eq:11

$$OSH = \{ x : w.x + b = 0 \} \quad (11)$$

we use the separating (OSH), once we have trained it on the training set, The (OSH) divides the R^n into two regions: one where w.xi + b ≥ 0 and one where w.xi + b ≤ 0. To use the maximal margin classifier, we determine on which side the test vector lies and assign the corresponding class label. Hence, the predicted class of a test point x is the output of the decision function Eq 12.

$$d(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(x_i, x) + b \right) \quad (12)$$

2.2.4 Multiclass Classification

In real world problems we often have to deal with d ≥ 2 classes. Our training set will consist of pairs (xi; yi), where xi € Rn and yi € {1,….,d} ; i = 1.. l. for solving this problem we construct decision functions of the form Eq:13

$$x \to \arg \max_{c \in \{1,..,d\}} \left[< w_c, \phi(x) > + b_c \right] \quad (13)$$

Here, $\Phi : x \to H, \Phi = k(x,.)$, is a feature map into a reproducing kernel Hilbert space H with corresponding kernel k, and w1,..,wd € H are class-wise weight vectors.

Two basic strategies to extend SVMs to multi-category classification can be distinguished. One approach is to combine separately trained binary SVM classifiers after training as done in the prominent one-versus-all method [15,16] .In the second method, a single optimization problem considering all classes is derived and solved at once, in [17] all-in-one approach perform significantly better than the one-vs-all method.

Crammer & Singer [18] proposed an alternative multi-class SVM. They also take all class relations into account at once and solve a single optimization problem, however, with fewer slack variables. The main reason for this modification of the Weston & Watkins approach [19] primal problem was to speed-up the training, because the Weston & Watkins approach turned out to be too slow for many applications. The Crammer & Singer classifier is trained by solving the primal problem Eq:14

$$\min_{w_c} \frac{1}{2}\sum_{c=1}^{d} < w_c, w_c > + \quad C\sum_{n=1}^{l} \xi_n \qquad (14)$$

Subject to Eq 15:

$$\begin{cases} \forall n \in \{1,..,l\}, \forall c \in \{1,..,d\} \setminus \{y_n\} \\ \quad < w_{yn} - w_c, \phi(x_n) > \geq 1 - \xi_n \qquad (15) \\ \quad and \quad \forall n \in \{1,..,l\} : \xi_n \geq 0 \end{cases}$$

For learning structured data, Crammer & Singer's method is usually the SVM algorithm of choice because is the one sometimes denoted as simply multi-class SVM.

2.2.5 K-Fold Cross-validation

In order to find the optimal models it has been performed a k-fold cross validation on the data set. In particular, we performed the parameters optimization on the Experimental training set using Cross-validation method that consists to divide the training set into k subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining k-1 subsets. Thus; each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. The advantage of k-fold Cross validation is that all the examples in the Dataset are eventually used for both training and testing.

2.2.6 Selection parameters of SVM model

The process of determining the SVM model is greatly influenced by the selection of kernel. First of all, we need to choose our kernel. This is a parameter in itself. Each kernel has a different set of parameters, and will perform differently, so in order to compare kernels you will have to optimize each kernel's parameters.

The search for selected these parameters was conducted using a grid-search method [20] for all three kernel functions, the grid search technique is not always an exhaustive method (depending on the grid resolution) such that is the stable technique than adaptive search which can save not so much time and lead to some strange results, other automatic methods, such as [21,22,23], exist but are iterative and can be computationally expensive.

the selection of optimal parameters for different kernel type (linear, RBF, polynomial) and find the best model selection is an important research issue in the data mining area .For each kernel chosen we search the optimal parameters and we vary these parameters around values intervals and calculate the classification accuracy.

The parameters that could be varied are: C is the parameter for SVM, regardless of kernel chosen; it has nothing to do with the kernel. It determines the tradeoff between a wide margin and classifier error it was varied over Eq: 16

$$C \in \{2^{-5}, 2^{-3}, .., 2^{15}\} \qquad (16)$$

For the polynomial kernel, gamma serves as inner product coefficient in the polynomial Eq:17.

$$\mathrm{k}(\mathrm{x}_i, \mathrm{x}_j) = (gamma \quad < x_i, x_j > + coef\,0)^{\mathrm{degree}}$$
$$gamma \in \{1, 2, ...\} \qquad (17)$$

In the case of the RBF kernel, γ determines the RBF width. In both cases , γ and gamma was varied over Eq:18:

$$\mathrm{k}(\mathrm{x}_i, \mathrm{x}_j) = exp(-\gamma \left\| x_i - x_j \right\|^2)$$
$$\gamma \in \{2^{-15}, 2^{-13}, .., 2^{-5}, 2^{3}\} \qquad (18)$$

The degree is the main parameter of polynomial kernel, but we can also vary gamma and coef0 to make the kernel non-symmetric Eq :19.

$$\mathrm{deg}\,ree \in \{1, 2, 3, ...\} \qquad coef \in \{1, 2, 3...\} \qquad (19)$$

We take the parameters of classifier to minimize the generalization error of that classification problem, we try small and large values for these parameters then we decide which are better for the data by cross validation, the parameters are tuned using 5-fold cross-validation estimation performance. We select the optimal parameters by a two-step grid search.

In linear kernel we vary the parameter C, in the case of polynomial kernel we change the two parameters degree and gamma while keeping C and coef constant, and C and coef while keeping degree and gamma constant and for RBF kernel we vary the parameters C and γ.

First we do a coarse grid search for each of the kernel using the following sets of these values,

Thus combinations of parameters are tried in this step, An optimal pair (C_o,γ_0, $gamma_0$,$degree_0$,$coef_0$) is selected from this coarse grid search. In the second step, a fine grid search is conducted around (C_0,γ_0,$gamma_0$,$degree_0$,$coef_0$) with Eq:20

$$C \in \{0.1c_0, 0.2c_0, ...6c_0, 8c_0\}$$
$$\gamma \in \{0.1\gamma_0, 0.2\gamma_0, ...6\gamma_0, 8\gamma_0\}$$
$$coef \in \{0.1coef_0, 0.2coef_0, ...6coef_0, 8coef_0\} \quad (20)$$
$$\deg ree \in \{0.1\deg ree_0, 0.2\deg ree_0, ...0.6\deg ree_0, 0.8\deg ree_0\}$$
$$gamma \in \{0.1gamma_0, 0.2gamma_0, ...6gamma_0, 8gamma_0\}$$

All together combinations of C , γ , gamma ,degree , coef are tried in this step and we selected the ones providing the best accuracy .The final optimal hyperparameter is selected from this fine search. After those parameters are found, the whole training set is trained again to generate the final classifier.

2.2.7 K-nearest neighbors (KNN)

K-Nearest Neighbor algorithm (KNN) is part of supervised learning and it is one of the simplest using machine learning algorithms. Besides it simplicity, KNN is a widely used technique,being successfully applied in a many applications in the field of data mining, statistical pattern recognition and many others [8, 24].

An object is classified by the "distance" from its neighbors, with the object being assigned to the class most common among its k distance-nearest neighbors. It is usual to use the Euclidean distance, though other distance measures such as the Manhattan distance can be used. To calculate distance between two vector positions in the multidimensional space.

The training process for KNN consists only of storing the feature vectors and class labels of the training samples and requires a parameter K, which is the number of near neighbors to consider. K-selection is the important problem we should take into account is how to choose a suitable K for making the classification more successful . Generally, according to Shakhnarovish et.al [25], larger values of k reduce the effect of noise on the classification, but make boundaries between classed less distinct. For the purpose of this assignment the parameter is searched by sequentially trying all of the possible values (1,..,N ,N = size of the dataset). This process is computationally expansive since N classification of the full dataset has to be made in order to find the value that gives less error as possible.

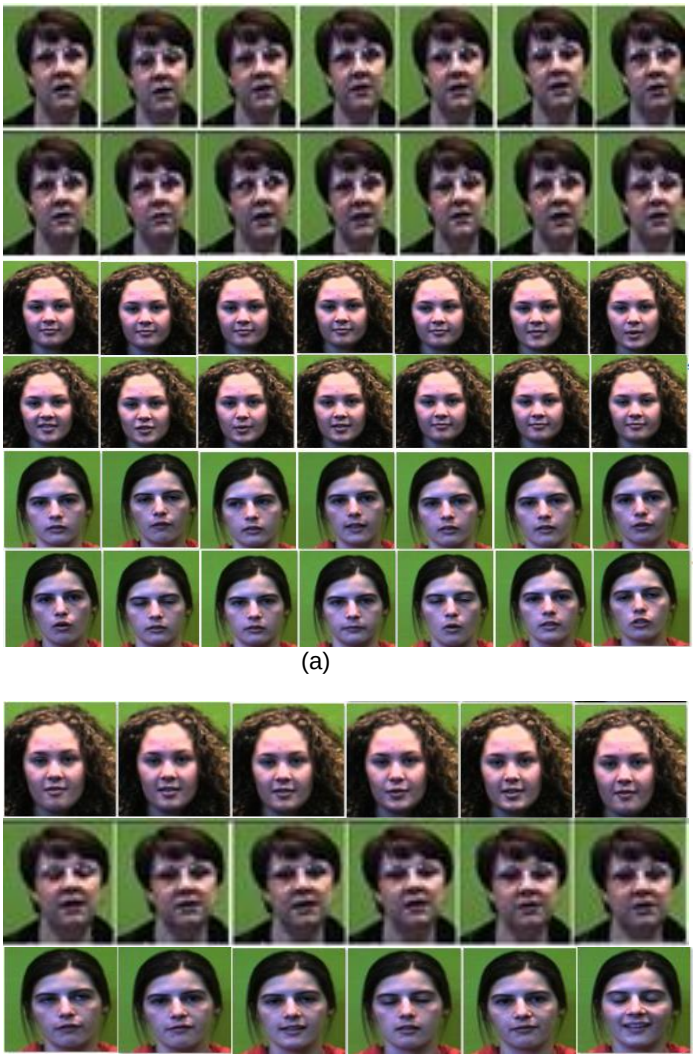
The only heuristic used to reduce the computation time is the assumption that the best value of K will be at most half of the size of the dataset (1<K<N/2).

Another important assumption was made about the classifier's parameter: the K that performs the best on the training set is likely to perform well on the test set.

3. Experimentation and Results

Our experiments were performed on three face databases face94, face96 and grimaces database. The datasets have both male and female subjects, and have representatives from 4 different races.

We split each DataSet on the training and testing sets both contains 10 classes, each class have 14 images for training and 6 images for testing Fig 4.The number of features is the dimension of image width*hight=180*200.

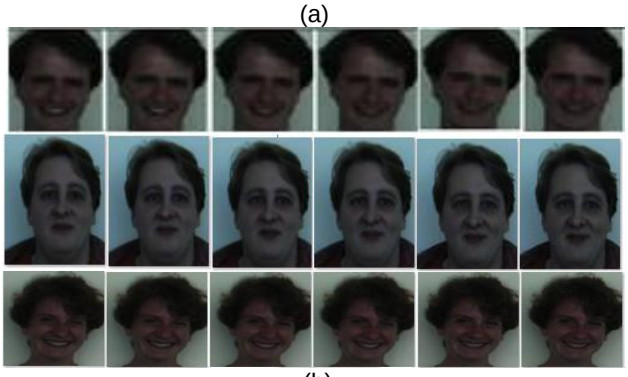


(a)

(b)

Fig. 4: Examples of (a) training and (b) test images (of database Face94) which used for classification by SVM kernels and KNN

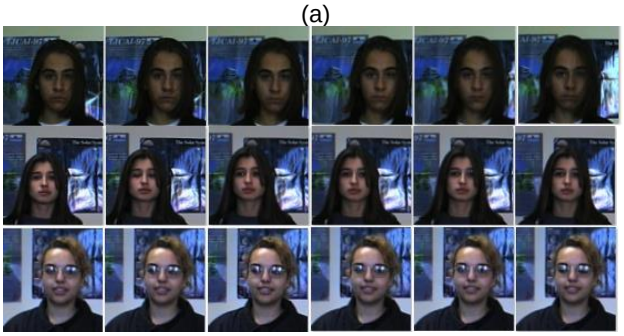
(a)



(b)

Fig. 5 : Examples of (a) training and (b) test images (of database Grimace) which used for classification by SVM kernels and KNN

(a)



(b)

Fig. 6 : Examples of (a) training and (b) test images (of database Face96) which used for classification by SVM kernels and KNN

In the first the whole feature data set was reduced by LDA then we train the classifiers SVM and KNN .for SVM adopting different kernel we train the data and a grid search method with 5-fold cross-validation exercise was applied for selecting the best parameters in the final the models SVM was created. In the KNN the training process through 5 -fold cross-validations consists of searching the parameter k that gives the best performance on the training set.

In table 1,2,3,4, for each Datasets(Face94,Face96 and Grimace) it can be seen the optimal parameters of models SVM constructed in the training stage, each model requires the selection of optimal parameters for different kernel type and the best neighbors K for KNN.

Table 1: The optimal parameters for linear kernel SVM in different Database.

DataSet	***Linear***	
Face94	Parameter : c	Cross validation accuracy
	c = 0.011	90%
Face96	c =0.0312	90%
Grimace	c =0.217	91.42%

Table 2: The optimal parameters for polynomial kernel SVM in different Database.

DataSet	***Polynomial***	
Face94	Parameters: c, gamma, degree, coef	Cross validation accuracy
	(c =2) , (gamma = 1) , (degree= 2),(coef = 1)	73.4%
Face96	(c = 1) , (gamma = 3) , (degree= 3),(coef = 9)	60.4%
Grimace	(c = 2.25) , (gamma = 11) (degree= 3.3),(coef = 13)	80.4%

Table 3: The optimal parameters for RBF kernel SVM in different Database.

DataSet	***RBF***	
Face94	Parameters: c, gamma	Cross validation accuracy
	(c=0,26), (gamma=0.25)	97.14%
Face96	(c=2), (gamma=0.4525)	95.1%
Grimace	(c =2), (gamma =0. 16)	100%

Table 4: the best K neighbors of KNN in different Database.

DataSet	***Ref System : KNN***	
Face94	Best neighbors K	Cross validation accuracy
	(k=41)	82.08%
Face96	(k =19)	80.3%
Grimace	(k =33)	91.24%

In order to test the performance of classifiers SVM and KNN on the different Datasets, we compare the prediction accuracies of SVM adopting different type of kernel function with these selected parameters with the prediction accuracies of KNN.

Table 5 shows the classification accuracy on Testing Set using the classifiers SVM with different kernel and KNN on different dataset face94, face96, Grimace.

Table 5: The accuracy values obtained on three test Datasets while changing the SVM kernel (linear, polynomial, RBF) Vs Ref Sys (KNN).

DataSet	***Linear***	***Polynomial***	***RBF***	***Ref System : KNN***
Face94	78.34	73.2	90.8	57.29 (with k=41)
Face96	74.4	71.6	88.24	50.6 (with k =19)
Grimace	85.25	82.25	93.7	64.8 (with k =33)

For a given dataset, the classification accuracy varies significantly among different kernels function. This indicates that the classification accuracy vary with the SVM kernel .The testing accuracy of the RBF kernel is more efficient compared to the other two kernels, that may be due to a number of reasons: it can determine a non-linear decision boundary (not possible for a linear kernel), it has fewer parameters than the polynomial kernel and is consequently simpler to tune, and it faces less numerical difficulties (polynomial kernel value may go to infinity). and the other side the SVM classifier performs overall better than the KNN classifier ,The best overall performance was 93.7% obtained by SVM with RBF kernel and 64.8% by KNN so the SVM classifier outperforms the KNN when using the suitable kernel with the best parameters.

4. Conclusion

The proposed method is performed with less amount of memory which includes the efficient techniques as the use of LDA method gives a real performance on reducing data and the computational time involved in the training stage of cross-validation and grid search process which can improve SVM accuracy a little. And on the other hand the SVM classification is powerful for classification comparing with KNN, It was shown that considerably high classification accuracy can be achieved by selecting optimal set of parameters for the RBF kernel .In the future work will look into efficient technique using the Fisher kernel of SVM and automatic selection of optimal kernel parameters using a large dataset.

References

[1]Computer Vision Science Research ml, 2007. Projects,http://cswww.essex.ac.uk/mv/allfaces/faces94.ht

[2] Computer Vision Science Research ml, 2007. Projects, http://cswww.essex.ac.uk/mv/allfaces/faces96.html.

[3] Computer Vision Science Research ml, 2007. Projects, http://cswww.essex.ac.uk/mv/allfaces/grimace.html

[4] K. Etemad, R. Chellappa, Discriminant Analysis for Recognition of Human Face Images, Journal of the Optical Society of America A, Vol. 14, No. 8, August 1997, pp. 1724-1733.

[5] Lawrence, S., Giles, C.L., Tsoi, A.C., and Back, A.D. (1998).Face Recognition: A Convolutional Neural Network Approach IEEE Transactions on Neural Networks, 8(1):98-113.

[6] G. Guo, S.Z. Li, K. Chan, Face Recognition by Support Vector Machines, Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, 26-30 March 2000, Grenoble, France, pp. 196-201.

[7] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River,NJ, USA, 1994. ISBN 0-13-106360-X.

[8] D. Hand, H. Mannila, P. Smyth.: Principles of Data Mining. The MIT Press. (2001).
[9] A. Martinez, A. Kak, "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, 2001.
[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7,pp. 711–720, 1997.
[11] Vapnik, V., 1995: The Nature of Statistical Learning Theory. Springer, N.Y.
[12] Vapnik, V., 1998: Statistical Learning Theory. Wiley, N.Y.
[13] Aizerman, M. A., Braverman, E. M., and Rozonoer, L. I., 1964: Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning.Autom. Remote Control 25.
[14] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.
[15] V. Vapnik. Statistical Learning Theory. Wiley, 1998.
[16] R. Rifkin and A. Klautau. In defense of one-vs-all classification. Journal of Machine Learning Research, 5:101-141, 2004.
[17] C. Demirkesen and H. Cherifi. A comparison of multiclass SVM methods for realworld natural scenes. In J. Blanc-Talon and other, editors, Advanced Concepts for Intelligent Vision Systems (ACIVS 2008), volume 5259 of LNCS, pages 763-763. Springer, 2008.
[18]K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel based vector machines. Journal of Machine Learning Research, 2:265-292, 2002.
[19] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition.In M. Verleysen, editor, Seventh European Symposium on Artificial Neural Networks (ESANN), pages 219-224, 1999.
[20] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector machines.Technical report, Department of Computer Science & Information Engineering,National Taiwan University, July 2003.
[21] C. Staelin. Parameter selection for support vector machines. Technical report, HP Labs, Israel, 2002.
[22] O. Chapelle and V. Vapnik. Model selection for support vector machines. In Proc.Advances in Neural Information Processing Systems (NIPS), 1999.
[23] J.H. Lee and C.J. Lin. Automatic model selection for SVM. Technical report,Department of Computer Science & Information Engineering, National Taiwan University, November 2000.
[24] LI LiLi, ZHANG YanXia and ZHAO YongHeng, "K-Nearest Neighbors for automated classification of celestial objects," Science in China Series G-Phys Mech Astron, Vol.51, no.7, July 2008, pp. 916-922.
[25] Shakhnarrovish, Darrell and Indyk, Nearest Neighbor Methods in learning and vision, MIT-press, 2005.

Anissa Bouzalmat is a PhD student at Sidi Mohammed Ben Abdellah University, Laboratory Intelligent Systems and Applications ISA in Morocco. He received his Master in Computer Science from the University of Sidi Mohammed Ben Abdellah in 2008. His current research interests are face recognition and detection.

Jamal Kharroubi is a professor at Sidi Mohammed Ben Abdellah University , Laboratory Intelligent Systems and Applications ISA (Communication Systems and Knowledge Processing group) in Morocco. His current research interests are Biometric, face recognition and detection..etc.

Arsalane Zarghili is a professor at Sidi Mohammed Ben Abdellah University ,Laboratory Intelligent Systems and Applications ISA (Artificial Vision & Embedded Systems group) in Morocco. His current research interests are Biometric, face recognition and detection..etc.