Face Recognition using Bagging and Linear Regression

Manuel Montoya Catalá [1]

[1] Master in Multimedia and Communications
mmontoya@ing.uc3m.es

Abstract

In this paper, a comparison of methods for face recognition is performed. After the image preprocessing, which is constant throughout the experiments, several feature extraction, feature selection and classification techniques have been proposed and tested in order to obtain the best system for face detection. Feature extraction techniques used include PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), ICA (Independent Component Analysis), (Partial Least Squares) and kernel versions of these. Feature selection techniques include Random Forest, Linear SVC and Recursive Feature Elimination Cross Validation using SVM. Classification techniques include Linear Regression, LDA and SVM. The proposed methods were tested on Yale face database. Experimental results on this database demonstrated the effectiveness of the proposed method for face recognition with less misclassification in comparison with previous methods.

Index Terms: Face Recognition, PCA, LDA, ICA, PLS, Random Forest, SVC, SVM.

1. Introduction

In the last years, Face Recognition has become one of the most challenging tasks in the pattern recognition field. The recognition of faces is very important for many applications such as: video surveillance, retrieval of an identity from a data base for criminal investigations and forensic applications. Among various solutions to the problem, the most successful seems to be those appearance-based approaches. These methods extract features to optimally represent faces belonging to a class and separate faces from different classes.

The aim of this paper is to present a comparative study of three most popular appearance-based face recognition projection methods (PCA, ICA, LDA and PLS) in completely equal working conditions regarding preprocessing and algorithm implementation. Several classification algorithms have been used in order to reduce the bias of classification error of the different sets of features and to test the optimality of the different classifiers to the different features.

The first section describes the image preprocessing techniques used in order to normalize the images' size, align them and transform the pixel values. After that, several feature extraction techniques used for dimensionality reduction are exposed, introducing the basic concepts of the extraction and the eigenfaces they generate over the dataset. Then the different classification systems are briefly explained, several learners are used in order to have a better approximation of how relevant the different features are. Finally, the result of the different systems is shown, along with the conclusions and the future work.

2. Image preprocessing

The first task of almost every machine learning system is to preprocess the input data in order to normalize it and make it easier for the system to work with it, increasing the information extracted. The dataset used contains 165 images from 15 subjects in front view perspective. Several image preprocessing stages can be done in order to bring all images to a common framework, the first step performed consists in erasing the useless information of the pictures and normalize the size of them.

Since eyes are critical for face detection being a high reliable part of the face, eyes' location is used for aligning the images. Using the Viola-Jones detector, the position of the eyes of every individual image is computed. These positions are used for:

- Aligning all people images to the same eyes position.
- Rotating the images so that the imaginary line joining them is parallel to the X axis.

Then, a 64x64 pixel window crops de image both for normalizing the pixel size of the image and also for deleting background information which is not useful for the desired task. The images are cropped such that the position of the eyes is the same for every image.

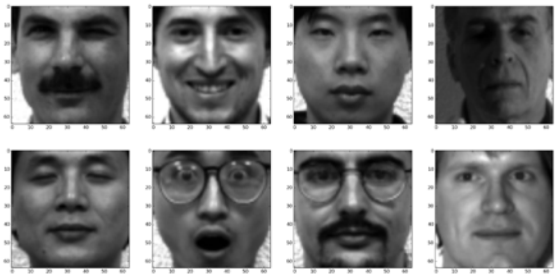


Figure 1 Postprocessed images

Once we have an aligned and standardized set of images, we perform histogram equalization of all input images in order to spread energy of all pixels inside the image and then normalize them to equalize amount of energy related to each face image. As a next step, we subtract the mean image from all the images of the dataset. The next figure shows the mean face image from the dataset. There is no need to rescale the feature vectors to [-1, 1] since they are already constrained to values 0 – 255.

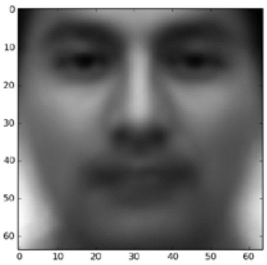


Figure 2. Mean face

Finally, all postprocessed face images are resized to a 4096 x 1 row vector where each component is a pixel if the image and represents a feature.

3. Feature Extraction

Images are highly dimensional elements, each pixel is a low-level feature of the image, the dataset contains images of 64x64 pixels, meaning that every sample vector (image) contains 4096 features; on the other hand we only have 165 samples from 15 classes. Some technique is needed to reduce the dimensionality of the problem so that it is tractable. Several methods have been developed for this purpose, given an s-dimensional vector representation of each face in a training set of M images; our objective is to transform it into a t-dimensional vector with $t \ll s$. The methods have been implemented in Python using the sklearn library.

3.1. PCA (Principal Component Análisis)

PCA is an unsupervised method that finds the projections (hyperplanes) over the dataset that maximizes the variance of the projected data. A training matrix with N samples and D dimensions will form a matrix (NxD), being $N \ll D$, the matrix will have rank N. The N D-divisional vectors form a basis of the subspace where the images are. PCA obtains a new set of basis vectors which are orthogonal and whose projections have maximum variance.

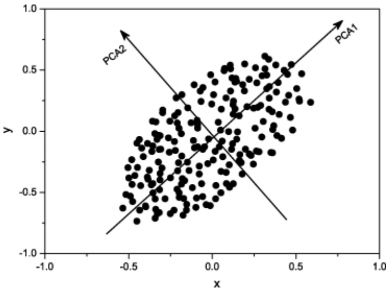


Figure 3 PCA projection

So our objective is to find a linear transformation matrix U so that we obtain a new set of "basis vectors" $\bar{\bar{X}}_{PCA}$:

$$\bar{\bar{X}}_{PCA} = \bar{\bar{\mathrm{U}}}_{(NxN)}\bar{\bar{\mathrm{X}}}_{(NxD)}$$

So that the covariance of the projections of the images over $\bar{\bar{\mathrm{X}}}_{\mathrm{PCA}}$ is maximized. The features are the projection over the new basis $\bar{\bar{X}}_{PCA}$, the $i-th$ feature is obtained as:

$$v_i = \left(\bar{X}_{PCA_i}\right)^T_{(Dx1)}\bar{\bar{\mathrm{X}}}_{img_{(1xD)}}$$

The "basis vectors" define a subspace of face images called "face space", these base vectors are just hyperplanes to which images are projected, and they are called eigenfaces. The projection of an image into one of this hyperplanes outputs a feature. All images of known faces are projected onto the "face space", obtaining a new set of features that contains all the variance of the original set. To identify an unknown image, that image is projected onto the same "face space" as well to obtain the corresponding features.

The following figure shows the first eigenvectors of the dataset transformation.

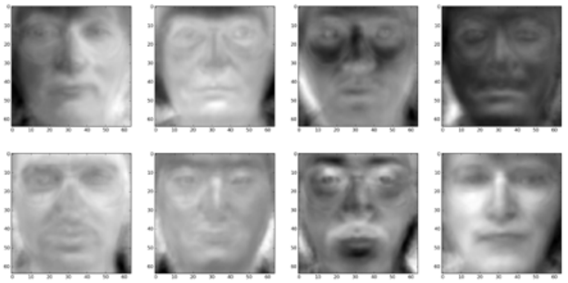


Figure 4 PCA eigenfaces

These eigenfaces also can give us an intuition of how discriminative the different parts of the head are for face recognition. For example, the sixth component may indicate whether the subject is Asian or not.

Since these eigenfaces are a basis of the images in the training set, we can express them as the linear combination of this base, where the coefficients of every eigenface is the projection value of the image in that eigenface.

$$Image = meanface + \sum_{i=1}^{N} v_i \bar{X}_{PCA_i}$$

The more eigenvectors $\bar{X}_{PCA_i}$ we use to reconstruct the image, the less error will be made. The following image shows the reconstruction of one image using different numbers of components to reconstruct it.

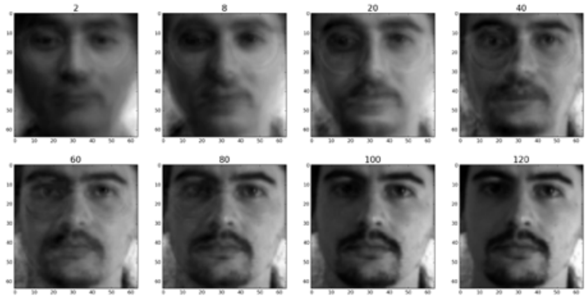


Figure 5 Image reconstruction

It can be seen that with only a few eigenfaces, we can reconstruct the original image almost entirely, thus reducing even more the dimensionality of the problem.

3.2. LDA (Principal Component Análisis)

LDA is a supervised method that fits a Multivariate Gaussian density to each class, assuming that all classes share the same covariance matrix. Making that assumption, the decision boundary is a hyperplane.

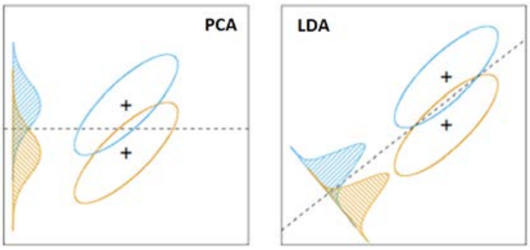


Figure 6. PCA and LDA projections

LDA finds the direction of minimum overlap among classes. It computes the D-dimensional mean vectors for the different classes:

$$\bar{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N_j}\bar{x}_i$$

The mean values $\bar{\mu}_j$ of the first 8 subject can be seen in the following figure.

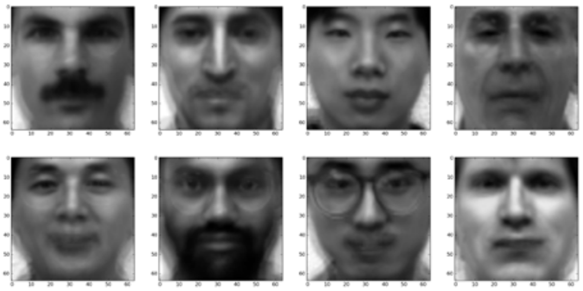


Figure 7 Mean faces of the classes

It computes the scatter matrix of the classes, first it assumes that every class have the same covariance matrix (D dimensional), it is computed as the average of the covariance matrix of the classes:

$$\bar{\bar{S}}_X = \frac{1}{C}\sum_{j=1}^{C}\bar{\bar{S}}_{Xj} = \frac{1}{C}\sum_{j=1}^{C}\left[\sum_{i=1}^{N_j}(\bar{x}_i-\bar{\mu}_j)(\bar{x}_i-\bar{\mu}_j)^T\right]$$

We will assume every class has this covariance matrix. There is also the inter-class scatter matrix which is as follows:

$$\bar{\bar{S}}_B = \sum_{j=1}^{C}(\bar{\mu}_j-\bar{\mu})(\bar{\mu}_j-\bar{\mu})^T$$

Where $\bar{\mu}$ is the mean of all samples. The class separation in a direction $\bar{w}$ in this case will be given by:

$$S = \frac{\bar{w}^T\bar{\bar{S}}_X\bar{w}}{\bar{w}^T\bar{\bar{S}}_B\bar{w}}$$

This means that when $\bar{w}$ is an eigenvector $\bar{\bar{S}}_X^{-1}\bar{\bar{S}}_B$ the separation will be equal to the corresponding eigenvalue, thus maximizing the separation.

The hyperplanes $\bar{w}$ separating the classes compose the eigenfaces of the LDA algorithm.

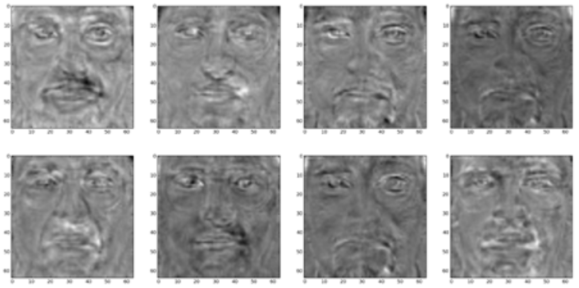


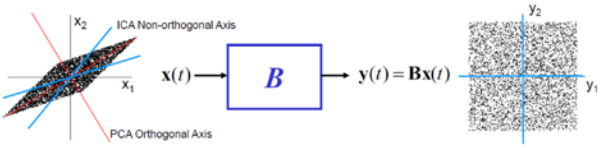
Figure 8 LDA hyperplanes

3.3. ICA (Independent Component Análisis)

Given a set of linearly dependent variables $\{x_1, x_2 \dots, x_D\}$, the goal of ICA is to find the transformation matrix

$$\bar{\bar{X}}_{ICA} = \bar{\bar{U}}_{(CxD)}\bar{\bar{X}}_{(DxN)}$$

That transform them in a set of C random variables $\{y_1, y_2 \dots, y_C\}$ that linearly independent. It is an unsupervised method in which the N D-divisional vectors form a basis of the subspace where the images are whitened and decorrelated.



We used the FastICA algorithm implemented in the sklearn library in Python. The number of components chosen is 15, as many as classes. The transformation matrix $\bar{\bar{U}}_{(CxD)}$ contains the transformation of the D-dimensional vectors to the C uncorrelated random variables obtained. The next figure shows the projection of the first 8 classes, out of the total 15. Although we could have chosen any number of classes C, the number 15 was chosen so that the features obtained might contain the differentiation among the 15 sujects.

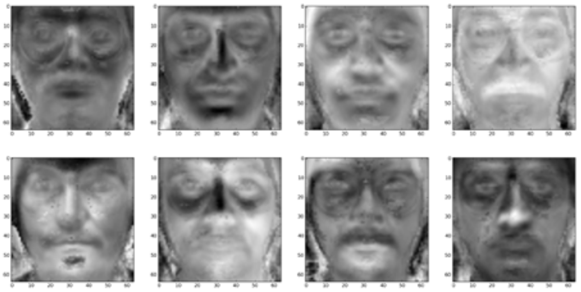


Figure 9 ICA transformation matrix

3.4. PLS (Partial Least Squares)

PLS a supervised technique that finds the projections of the input and output data with maximum covariance. The first 8 hyperplane projections of the PLS decomposition are shown in the image below.

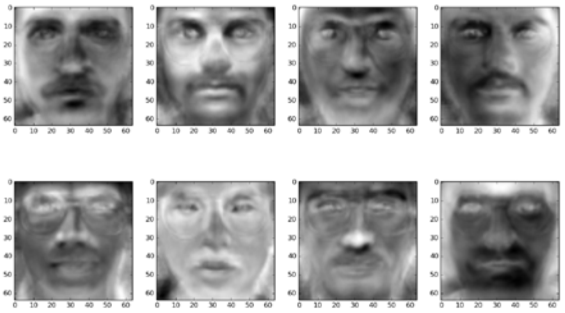


Figure 10 PLS hyperplanes

4. Feature Selection

When the dimension of sample vectors is too high, some of them same redundant information while others do not contain information to solve the desired task. In these cases it is needed to select a subset of these features that have relevant and needed information, discarding the ones that are useless and impoverish the system both in computation complexity and precision. The feature selection method hereby used is a Recursive Feature Elimination Crossvalidation using the SVM as a base learner. This method initially uses the base learner, the SVM in our case, and then it will undergo an iterative process in which it will train the base classifier with all the remaining features except 1. At every iteration, the algorithm removes the feature that makes the system to improve the most, or to impoverish the least; until we are left with the desired number of features. The algorithm implemented performs crossvalidation at every stage for selecting the feature to remove.

5. Classification Systems

Once we have a feature subset of reasonable size, a classification system is trained with the transformed images in order to be able to classify a new image. The following training algorithms have been used:

5.1. LR (Logistic Regression)

Logistic regression is a classification technique that assumes the classes have all Gaussian distributions with the same covariance matrix, thus being the decision regions separated by hyperplanes. The probability of a vector $\bar{X}$ belonging to a class H is:

$$P\left(H_j|\bar{X}\right) = \frac{\exp\left(\bar{w}_j^T\bar{X}\right)}{1+\sum_{l=1}^{J}\exp(\bar{w}_l^T\bar{X})}$$

Where $\bar{w}_l$ is the hyperplane that separates the class l from the rest of the classes. Logistic Regression fits the model adjusting $\bar{w}$ by maximum ML to the training data (Newton method):

$$l(\bar{w}) = -\sum_{l=1}^{J}\log\left(1+exp\left(-y_i\left(\bar{w}_j^T\bar{X}_i\right)\right)\right)$$

The class H_j assigned to vector $\bar{X}$ is the one with the highest probability $P\left(H_j|\bar{X}\right)$.

5.2. SVM (Support Vector Machine)

The SVM is a binary classifier that aims to find the hyperplane over the feature space that maximizes the margin. The margin is the distance of a given sample to the hyperplane boundary multiplied by the label of the sample. If the problem is linearly separable and we establish that the nearest sample from each class has distance 1 to the hyperplane (we can do that since a hyperplane multiplied by a scalar is the same hyperplane, we just remove that degree of freedom). Then the problem can be written as:

$$\hat{w} = \arg\min_{w\in R^m}\frac{1}{2}|\bar{w}|^2$$

Subject to that the margin of every sample to the hyperplane is bigger than 1, since we already established that premise, mathematically it is express as:

$$y_i\ \langle\bar{x}_i,\bar{w}\rangle \geq 1\ \ \forall\ i = 1,2,\dots,N$$

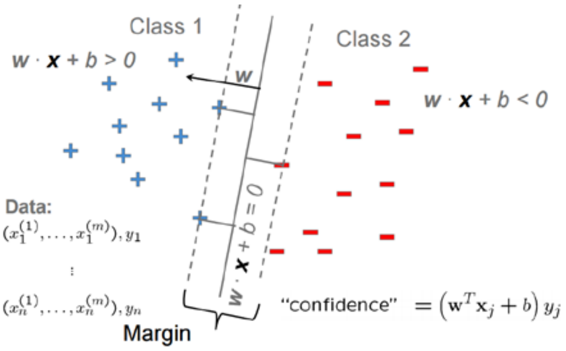

Class 1
Class 2
$w \cdot \boldsymbol{x} + b > 0$
$w \cdot \boldsymbol{x} + b < 0$
w
$w \cdot \boldsymbol{x} + b = 0$
Data:
$(x_1^{(1)},\dots,x_1^{(m)}), y_1$
⋮
$(x_n^{(1)},\dots,x_n^{(m)}), y_n$
Margin
"confidence" $= \left(\mathbf{w}^T\mathbf{x}_j + b\right) y_j$

Figure 11 SVM margin

If the classes are not linearly separable, we allow some outliers by introducing penalization term C, the bigger the parameter C, the more importance is given to the misclassified data.

We can also transform the input feature space into another one that is more linearly separable by applying a function to them such a polynomic extension or a Gaussian kernel. These kernel extensions allow using SVM in non-linearly separable datasets but also causing overfitting. They also have hyperparameters; the polynomic extension needs to know the degree of the polynomial and the Gaussian kernel, the variance and transformation of distance. Crossvalidation with $K = 5$ has been used in order to tune these hyperparameters.

5.3. LDA (Linear Discriminant Analysis)

As it was explained in the previous section LDA is a supervised method that fits a Multivariate Gaussian density to each class, assuming that all classes share the same covariance matrix. It maximizes the inter-class separation and minimizes the intra-class separation by means of maximizing the function:

$$S = \frac{\overline{w}^T \bar{\bar{S}}_X \overline{w}}{\overline{w}^T \bar{\bar{S}}_B \overline{w}}$$

The optimal value of the hyperplane decision boundary $\overline{w}$ is to be the eigenvector of $\bar{\bar{S}}_X^{-1} \bar{\bar{S}}_B$.

6. Experimental Setup and Results

The system proposed performs face recognition over the Yale dataset [1]. The dataset used contains 165 front images from 15 subjects with different expressions and wearing different complements. The images are preprocessed in order to normalize their size and position respect to the eyes. Also illumination has been corrected through histogram equalization.

The dataset is divided in two sets, 20% for testing and 80 for training, although different ratios have been used with similar results, this ratio seems to be reasonable. As already stated, different feature extraction, selection and classification techniques have been used, in the following figures, results of these techniques will be illustrated. The performance measure chosen is the accuracy of the classification.

6.1. PCA features

The first 50 eigenvectors of PCA decomposition have been used for classification; they explain 88% of the variance of the dataset. The results of the classification for different learners are shown in the following graphs.

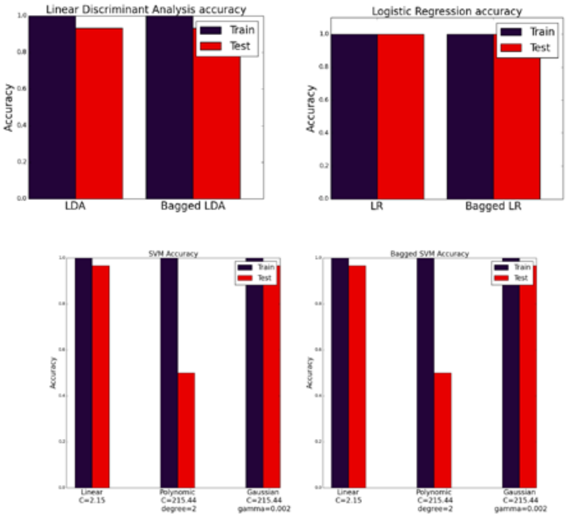


Figure 12 Classification Results PCA

6.2. LDA features

All 14 projections of the LDA decomposition have been used for prediction of the classes. The following graphs show the training and testing results of the classification for different learners.

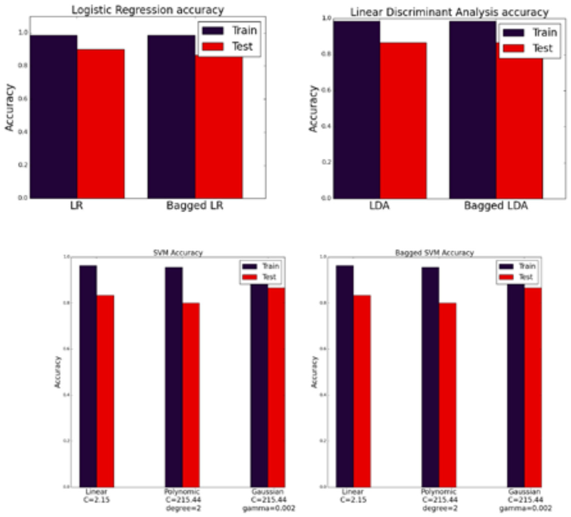


Figure 13 Classification results LDA features

6.3. ICA features

The initial feature space was decomposed into 15 uncorrelated random variables used as features. The following graphs show the training and testing results of the classification for different learners.

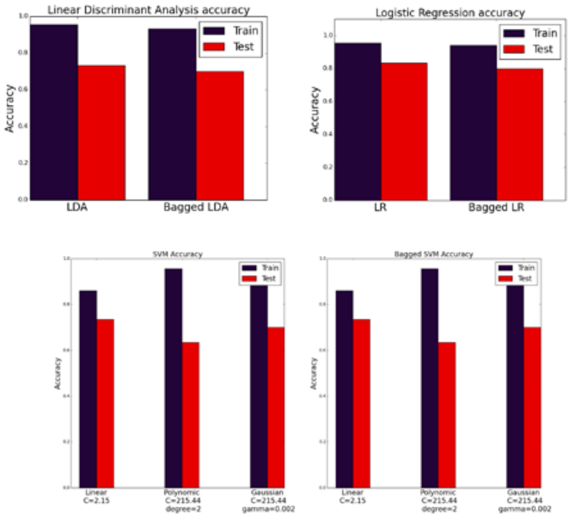


Figure 14 Classification results ICA features

6.4. PLS features

The first 50 eigenvectors of PLS decomposition have been used for classification. The results of the classification for different learners are shown in the following graphs.

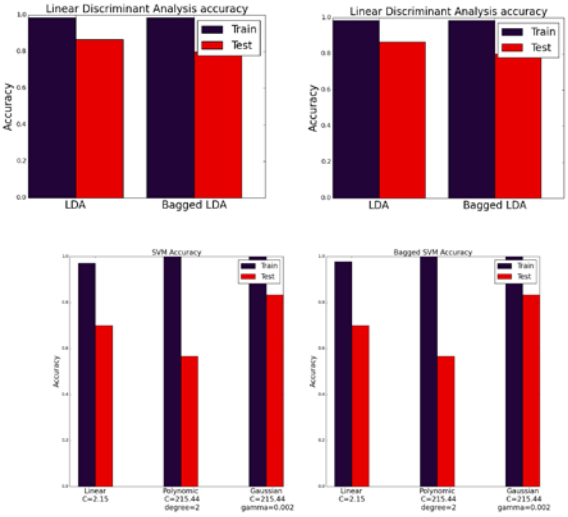


Figure 15 Classification results PLS features

7. Evaluation, Conclusions and Future Work

In order to evaluate the performance of the proposed system, we use the accuracy measure; this measure indicates the probability of misclassification of the system. From the results obtained using different classifiers we can see that the problem is highly separable, most of the systems achieved 100% of accuracy on training and a similar score for testing, thus, the best system is that one that generalizes the best, not causing overfitting.

The system that performed the best is a bagging ensemble of Linear Regression classifiers using PCA features, which performed perfect classification over the training and testing datasets. The SVM with Gaussian and linear kernel performed nearly as good achieving 97% of accuracy on test, but the polynomic kernel seems to overfit since it obtains 100% accuracy on training and only 50% on testing.

For the LDA features, all classification systems achieve a score of about 82% on test; no overfitting is caused by the SVM with polynomic kernel. PLS features obtaines similar results for Linear Regression and LDA, but the SVM seems to overfit a lot over these features. Finally, the ICA parameters scored the worst overall result with around 75% accuracy.

There are many future works to be done with this project concerning all of their stages:

- Combine the different features and perform feature extraction and selection over the whole set.
- Use different number of classes in the ICA algorithm.
- Try out all the systems described over a new dataset to evaluate if the system is optimal for general face recognition or just for this dataset.
- Use a random forest as a classifier.
- Vary the size and alignment of the post processed images.

8. References

[1] Lab Session 2 – Prediction of Epileptic Seizures from C4.278.8931-1 APPLICATIONS OF SIGNAL PROCESSING 14/15-S1.

[2] J. R. Solar, P. Navarreto, " Eigen space-based face recognition: a comparative study of different approaches, IEEE Tran. , Systems man and Cybernetics- part c: Applications, Vol. 35, No. 3, 2005.

[3] Yale face dataset. http://vision.ucsd.edu/content/yale-face-database.

[4] Discriminant Analysis of Principal Components for Face Recognition. Wenyi Xhao, Arvindh Krishnaswamy.

[5] A New Face Recognition Method using PCA, LDA and Neural Network. A. Hossein Sahoolizadeh, B. Zargham Heidari, and C. Hamid Dehghani. International Journal of Computer Science and Engineering 2:4 2008

[6] Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set. Kresimir Delac, Mislav Grgic, Sonja Grgic.

[7] PCA versus LDA. Aleix M. MartõÂnez, Member, IEEE, and Avinash C. Kak.

[8] Support Vector Machines for face authentication. K. Jonsson, J. Kittler, Y.P. Li, J. Matas. Image and Vision Computing 20 (2002).