Tesis de Máster

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Face recognition using Lattice Independent Component Analysis and Extreme Learning Machine

Ion Marqués Bailón

Donostia-San Sebastián, Septiembre 2011

Universidad del País Vasco / Euskal Herriko Unibertsitatea
Departamento de Ciencia de la Computación
e Inteligencia Artificial

Director: Manuel Graña Romay (UPV/EHU)

http://www.ccia-kzaa.ehu.es/

Face recognition with Lattice Independent Component Analysis and Extreme Learning Machines

Ion Marqués

Computational Intelligence Group
Universidad del País Vasco/Euskal Herriko Unibertsitatea
www.ehu.es/ccwintco

Abstract

We focus on two aspects of the face recognition: Feature extraction and classification. We propose a two component system, introducing Lattice Independent Component Analysis (LICA) for feature extraction and Extreme Learning Machines (ELM) for classification. In previous works we have proposed LICA for a variety of image processing tasks. The first step of LICA is to identify strong lattice independent components from the data. In the second step, the set of strong lattice independent vector are used for linear unmixing of the data, obtaining a vector of abundance coefficients. The resulting abundance values are used as features for classification, specifically for face recognition. Extreme Learning Machines are accurate and fast-learning innovative classification methods based on the random generation of the input-to-hidden-units weights followed by the resolution of the linear equations to obtain the hidden-to-output weights. The LICA-ELM system has been tested against state-of-the-art feature extraction methods and classifiers, outperforming them when performing cross-validation on four large unbalanced face databases.

1 Introduction

Face recognition [6] is one of the most relevant applications of image analysis. To build an automated system which equals human ability to recognize faces is still an open problem. There are many different industrial applications interested in it, mostly related to security and safety, attracting much attention and media coverage, such as entertainment systems and driving safety devices. Face recognition may be stated in two radically different ways. First it may consist in the authentication of a user, which is a binary decision problem. Second, it may consist in the search for the identification of a user in a large image database, which is a (large) multiclass problem. This initial problem can be extended to gaze, expression or mood recognition [53]. Taken as pattern recognition problem, face recognition provides a perfect benchmarking framework to test feature extraction techniques and classifiers.

In statistical learning approaches, each face image is viewed as a point (vector) in a d-dimensional space. The face images often belong to a low dimension manifold. The high dimensionality of the data imposes the need for feature extraction processes previous to face classification. Therefore, the goal is to choose and apply the right statistical tool for the extraction and analysis of the manifold where the face images lie in this high dimensional space. These tools must define the embedded face space in the image space and extract the basis functions from the face space. Ideally, patterns belonging to different classes (identities) will occupy disjoint and compact regions in the feature space, which will be easy to discriminate by means of statistical or bio-inspired classifier systems. In the best case a linear discriminant would be enough to obtain good classification performance results. The earliest approach applied Principal Component Analysis (PCA) for feature extraction [56], other approaches use the variations of the Linear Discriminant Analysis (LDA) [61, 45, 60, 46, 4], or the Locality Preserving Projections (LPP) [19]. Other successful statistic tools include Bayesian networks [37], bi-dimensional regression [29], generative models [20], and ensemble based and other boosting methods [33]. Here we propose Lattice Independent Component Analysis (LICA) [13]. This method uses a Lattice Computing [12] based Endmember Induction Algorithm (EIA) [57] to perform feature extraction and dimension reduction. This is a new approach to face recognition, although Lattice Computing approaches have been previously applied to fMRI imaging [14, 15], mobile robot localization [58] and hyperspectral image analysis [13, 48].

The classification system development process involves training a classifier from a data sample and testing the trained system on independent samples to guess the correct class. Translated into the face recognition paradigm, it means to train the system on a set of identified faces and then try to assign each new unknown face image to the correct identity. Extreme Learning Machine (ELM) constitute an innovative category of neural-network based classification and regression techniques [25]. Different kinds of ELM variations have been recently used in fields as diverse as sales forecasting [54], antiviral therapy [44], metal temperature prediction [55] or arrhythmia classification [30]. ELMs have been also applied in biometrics, specifically for on-line face detection [41] and fingerprint classification [34].

One of the main problems that a classification method must overcome is the unbalanced class distribution of the data set [26]. However, most face recognition algorithms and classifiers are tested over well balanced databases like ORL, Yalefaces or Multi-PIE. Under such ideal circumstances, most classifiers and feature extraction methods mentioned before work successfully [6]. It is reasonable to think that the environments or devices that require face recognition will not always provide such well balanced databases. Therefore, it is relevant to address the face recognition task in these unfavorable conditions. We have used Color FERET database [42, 43] to create 4 unbalanced experimental databases. We have tested LICA and other well known algorithms for feature extraction altogether with ELM. The performance of ELM has been compared with other classifiers. The aim of these experiments was to test the proficiency of both LICA and ELM in the recognition of faces of a complex and unbalanced database. Experimental results indicate that, among the tested methods, LICA is the most effective feature extraction algorithm for face recognition under high subject-per-class variability. Experimental results also reveal that ELM is the classifier less sensitive to high class-variation induced noise.

The remainder of the work is organized as follows: Section 2 introduces the LICA approach and the feature extraction algorithms with which it was compared. ELM and the rest of classifiers are presented in section 3. Section 4 gives a detailed description of our experimental design. Experimental results are presented in Section 5. Section 6 gives our empirical conclusions and further work directions.

2 Feature extraction algorithms

Feature extraction is the process of mapping the original data into a more effective feature space. The extracted features must preserve the best class separability possible in addition to dimension reduction. That is, if we have some data X, we find coefficients Y such that

$$X = A \cdot Y, \tag{1}$$

$$Y = A^{-1} \cdot X, \tag{2}$$

where A is the mixing matrix. The data X is therefore projected by its inverse A^{-1} into a more convenient feature space Y. We have tested some of the most widely used feature extraction algorithms: Principal Component Analysis (PCA) [56], Independent Component Analysis (ICA) [2, 39, 40, 9, 38, 35, 17] and Linear Discriminant Analysis (LDA) [1] along with Lattice Independent Component Analysis (LICA) [13]. The PCA and LDA both try to find orthogonal projection directions with greatest variance of the prejection coefficients. PCA is an unsupervised approach while LDA is supervised. ICA sources need not be orthogonal, because it maximizes the source statistical independence Finally, LICA is a Lattice Computing approach based on lattice independence. These algorithms are explained in more detail below.

2.1 Principal Component Analysis (PCA)

The PCA finds othogonal projection axes of the data in the order of decreasing projection variance. These directions are called principal components. Therefore, A^{-1} is formed by the principal components of the covariance matrix of X.

Let be a data-set composed of N images of n pixels, denoted by $X = \{\mathbf{x}_j; j = 1, \ldots, N\} \in \mathbb{R}^{n \times N}$, where each $\mathbf{x}_j$ is an image column vector. We center the data by subtracting the mean column. We want to find the eigenvectors $\mathbf{a}$ solving the eigen-problem:

$$\lambda \mathbf{a} = X \mathbf{a} \tag{3}$$

The Singular Value Decomposition of X given by $X = U \cdot S \cdot V^T$ where matrix U is the matrix of the eigenvectors of XX^T, S is the diagonal matrix of the eigenvalues. The data matrix X can be projected into a reduced spaced of dimensionality m by computing $Y = U_m^T X$, where U_m denotes the matrix composed of the first m columns of U.

2.2 Linear Discriminant Analysis (LDA)

PCA is unsupervised because it doesn't use the class information of data sample points. Linear Discriminant Analysis (LDA) searches for optimal class discrimination projections given data-set

$$X = \left\{\mathbf{x}_j^k; j = 1, \ldots, N; k = \{1, \ldots, C\}\right\} \in \mathbb{R}^{n\times N} \tag{4}$$

where data data samples are partitioned into C classes, $\mathbf{x}$ are n-dimensional vectors. Each class has m_k samples. Assume that the mean has been extracted from the samples, as in PCA. The objective function for the LDA can be defined [4] as

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a}}, \tag{5}$$

$$S_b = \sum_{k=1}^{c} m_k \boldsymbol{\mu}^k (\boldsymbol{\mu}^k)^T \tag{6}$$

$$= \sum_{k=1}^{c} \left(\frac{1}{m_k}(\sum_{i=1}^{m_k} \mathbf{x}_i^k)\right)\left(\frac{1}{N_k}(\sum_{i=1}^{m_k} \mathbf{x}_i^k)\right),^T \tag{7}$$

$$S_t = \sum_{i=1}^{m} \mathbf{x}_i (\mathbf{x}_i)^T, \tag{8}$$

where $\boldsymbol{\mu}$ is the total sample mean vector, $\boldsymbol{\mu}^k$ is the mean vector of the k-th class and $\mathbf{x}_i^k$ is the i-th sample in k-th class. The total scatter matrix S_t and between-class scatter matrix S_b can be expressed in matrix form, if the sample vectors of each class are grouped together:

$$S_b = XW_{NxN}X^T, \tag{9}$$

$$S_t = XX^T, \tag{10}$$

where W_{NxN} is a diagonal matrix defined as

$$W_{NxN} = \begin{bmatrix} W^1 & 0 & \ldots & 0 \\ 0 & W^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & W^c \end{bmatrix} \tag{11}$$

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

$$W^k = \begin{bmatrix} \frac{1}{m_k} & \frac{1}{m_k} & \ldots & \frac{1}{m_k} \\ \frac{1}{m_k} & \frac{1}{m_k} & \ldots & \frac{1}{m_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_k} & \frac{1}{m_k} & \ldots & \frac{1}{m_k} \end{bmatrix} \tag{12}$$

Finally, we can state LDA as the following eigenproblem:

$$S_b\mathbf{a} = \lambda S_t\mathbf{a}, \tag{13}$$

which is equivalent to

$$XW_{NxN}X^T(XX^T)^{-1}\mathbf{a} = \lambda\mathbf{a}. \tag{14}$$

The solution of this eigenproblem provides the eigenvectors needed to project the data in an analogous manner of PCA. When there are many variables, for instance if samples are images and observations are pixels, some previous dimensionality reduction must be performed.

2.3 Independent Component Analysis (ICA)

ICA is a generative model which aims to describe how the data is generated by mixing non-Gaussian, mutually statistically independent latent variables with and unknown mixing matrix [27]. Let us denote $\mathbf{x}$ the n-dimensional observed data vector and B the $n \times M$ mixing matrix. The mixing model is formulated for ICA as follows:

$$\mathbf{x} = B\mathbf{s}, \tag{15}$$

$$\mathbf{s} = V\mathbf{x}, \tag{16}$$

where $V = B^{-1}$ and $\mathbf{s}$ are the independent sources. If we consider the whole sample, the equation is rewritten as

$$S = VX \tag{17}$$

where $X = \{\mathbf{x}_j; j = 1, \dots, N\} \in \mathbb{R}^{n \times N}$, each $\mathbf{x}_j$ being a face image column vector.

It has been shown that the mixing model is completely identifiable, up to a permutation and scale of the sources, if the sources are statistically independent and at least $M - 1$ of them are non-Gaussian. In the case of M gaussian variables, the matrix B is not identifiable. It is also required that the number of sources is smaller than or equal to the number of available observations, i.e. $M \leq n$. The mixing and unmixing matrices can be estimated following three approaches: maximizing the nongaussianity, minimizing the mutual information and maximizing the likelihood. Quantitative measures of random variable nongaussianity are kurtosis, negentropy or approximations of negentropy. If the component are constrained to be uncorrelated, ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities. The constraint of uncorrelatedness simplifies the computations considerably. In the maximum likelihood estimation approach, the log-likelihood it's usually used, which is equivalent to entropy maximization, or “infomax”.

There are two possible ways of performing face recognition with ICA. We can treat the images as random variables and pixels as observation. This approach maximizes the independence of pixels It has been argued that it will produce better object recognition, since it implements recognition by parts [31]. Other approach is to treat pixels as variables and images and observations. Treating the face recognition problem from a wholistic approach, it has been demonstrated that it performs better [11]. In this work we chose the second option.

We have used the DTU:ICA toolbox developed by the Technical University of Denmark [10].

Mean-field ICA

This method estimates sources from the mean of their posterior distribution and the mixing matrix (and noise level) is estimated by maximum a posteriori (MAP) [28]. The latter requires the computation of a good approximation to the correlations between sources. For this purpose, [28] propose three increasingly advanced mean-field methods: the variational (also known as naive mean field) approach, linear response corrections, and an adaptive version of the Thouless, Anderson and Palmer (TAP) mean-field approach [39, 40].

We have empirically searched for the best of those approaches on our problem. The followed criteria was recognition accuracy, constrained to a feasible execution time. The selected method uses a constant prior mixing matrix and noise covariance as well as a non-analytic power law source prior. The Mean-field method used was linear response correction.

ICA Infomax

The “infomax” framework original purpose was to maximize the output entropy of a neural network with non-linear outputs [2]. It is closely connected to the maximum likelihood estimation. For a data matrix $X = \{\mathbf{x}_j ; j = 1, \ldots, N\} \in \mathbb{R}^{n\times N}$, the log-likelihood function has the form [27]

$$L = \sum_{i=1}^{t} \sum_{j=1}^{n} \log f_j(\mathbf{v}_j \mathbf{x}(i)) + t \cdot \log |\det V| \tag{18}$$

where $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \in \mathbb{R}^{t\times n}$ is the inverse of the source mixing matrix B. In our case, the function used is

$$L = t \cdot \log |\det V| - \sum_{i=1}^{t} \sum_{j=1}^{n} \log f_j(\mathbf{v}_j \mathbf{x}(i)) + N \cdot n \cdot \log(\pi) \tag{19}$$

where $f(x) = cosh(x)$.

ICA with Molgedey and Schuster decorrelation algorithm

ICA with the Molgedey and Schuster decorrelation algorithm (ICA-MS) uses the decorrelation algorithm presented in [35] to uncorrelate a some superimposed sources X and X_{ts}, where ts stands for time-shifted. The problem was reduced to solve the eigenproblem of correlation matrices $X_{ts}X^T$ and XX^T. The solution is found by solving the eigenvalue problem of the quotient matrix $Q = X_{ts}X^T(XX^T)^{-1}$ [18]. The delay time is estimated using autocorrelation differences.

2.4 Lattice Independent Component Analysis (LICA)

Lattice Independent Component Analysis is based on the Lattice Independence discovered when dealing with noise robustness in Morphological Associative

Memories [49]. Works on finding lattice independent sources (aka endmembers) for linear unmixing started on hyperspectral image processing [13, 50]. Since then, it has been also proposed for functional MRI analysis [14, 15] or mobile robot location [58] among others.

Under the Linear Mixing Model (LMM) the design matrix is composed of endmembers which define a convex region covering the measured data. The linear coefficients are known as fractional abundance coefficients that give the contribution of each endmember to the observed data:

$$\mathbf{y} = \sum_{i=1}^{M} a_i \mathbf{s}_i + \mathbf{w} = \mathbf{S}\mathbf{a} + \mathbf{w}, \tag{20}$$

where $\mathbf{y}$ is the d-dimension measured vector, $\mathbf{S}$ is the $d \times M$ matrix whose columns are the d-dimension endmembers $\mathbf{s}_i, i = 1, .., M$, $\mathbf{a}$ is the M-dimension abundance vector, and $\mathbf{w}$ is the d-dimension additive observation noise vector. Under this generative model, two constraints on the abundance coefficients hold. First, to be physically meaningful, all abundance coefficients must be non-negative $a_i \geq 0, i = 1, .., M$, because the negative contribution is not possible in the physical sense. Second, to account for the entire composition, they must be fully additive $\sum_{i=1}^{M} a_i = 1$. As a side effect, there is a saturation condition $a_i \leq 1, i = 1, .., M$, because no isolate endmember can account for more than the observed material. From a geometrical point of view, these restrictions mean that we expect the endmembers in $\mathbf{S}$ to be an Affine Independent set of points, and that the convex region defined by them covers *all* the data points.

The *Lattice Independent Component Analysis* (LICA) approach assumes the LMM as expressed in equation 20. Moreover, the equivalence between Affine Independence and Strong Lattice Independence [48] is used to induce from the data the endmembers that compose the matrix $\mathbf{S}$. Briefly, LICA consists of two steps:

1. Use an Endmember Induction Algorithm (EIA) to induce from the data a set of Strongly Lattice Independent vectors. In our works we use the algorithm described in [13, 14]. These vectors are taken as a set of affine independent vectors that forms the matrix $\mathbf{S}$ of equation 20.

2. Apply the Least Squares estimation to obtain the abundance vector of the LMM.

The advantages of this approach are (1) that we are not imposing statistical assumptions to find the sources, (2) that the algorithm is one-pass and very fast because it only uses lattice operators and addition, (3) that it is unsupervised and incremental, and (4) that it can be tuned to detect the number of endmembers by adjusting a noise-filtering related parameter. When $M \ll d$ the computation of the abundance coefficients can be interpreted as a dimension reduction transformation, or a feature extraction process.

Our input is a matrix of face images in the form of column vectors. In the linear mixing model (LMM), we represent the a face image as a linear combination of endmember faces. The weight of each endmember face (abundance) is proportional to its fractional contribution to the construction of the observed face image. In other words, the induced SLI vectors (endmembers) are selected face images which define the convex polytope covering the data. A

Algorithm 1 LICA feature extraction for face recognition. $E^{\#}$ denotes the pseudo-inverse of the matrix E.

```
1. Build a training face image matrix $X_{TR} = \{\mathbf{x}_j; j = 1, \ldots, m\} \in \mathbb{R}^{N \times m}$.
   The testing image matrix is denoted $X_{TE} = \{\mathbf{x}_j; j = 1, \ldots, m/3\} \in \mathbb{R}^{N \times m/3}$.
2. Obtain a set of $k$ endmembers using an EIA over $X_{TR}$: $E = \{\mathbf{e}_j; j = 1, \ldots, k\}$ from $X_{TR}$.
   Varying EIA parameters will give different $E$ matrices. The algorithm has been tested with $\alpha$ values dependant on database size.
3. Unmix train and test data: $Y_{TR} = E^{\#} X_{TR}^T$ and $Y_{TE} = E^{\#} X_{TE}^T$.
```

face image is defined as a $A_{a \times b}$ matrix composed by $a \cdot b = N$ pixels. Images are stored like row-vectors. Therefore, column-wise the data-set is denoted by $Y = \{\mathbf{y}_j; j = 1, \ldots, N\} \in \mathbb{R}^{n \times N}$, where each $\mathbf{y}_j$ is a pixel vector. Firstly, the set of SLI $X = \{\mathbf{x}_1\} \in \mathbb{R}^{n \times K}$ is initialized with the maximum norm pixel (vector) in the input data-set Y. We chose to use the maximum norm vector as it showed experimentally to be the most successful approach. The method is summarized in algorithm 1.

The algorithm for endmember induction, the EIA, used is the one in [13] which has tolerance parameter α controlling the amount of endmembers detected. In the ensuing experiments we have varied this parameter in order to obtain varying numbers of endmembers on the same data.

3 Classification

One of the goals of this work is to compare the performance of Extreme Learning Machines (ELM) with other classifiers. We have chosen two competing state of the art classification algorithms. One is an ensemble classifier based on decision trees - Random Forest [3]. The other is a Support Vector Machine variant introduced in [51] called $\nu-$SMV. We have used the implementations of Random Forest and $\nu-$SMV provided in Weka [16, 5]. In the following subsections, we describe the classifiers in more detail. Additionally, we have also compared ELM with Feed-forward Neural Networks (FFNNs) trained with two standard learning algorithms as provided in Matlab.

3.1 Extreme Learning Machines

Standard Single Layer Feed-forward Neural Network (SLFNs) training is too slow because of: (1) Usual gradient-based learning algorithms are slow and (2) all the parameters of the networks are tuned iteratively by using such learning algorithms. An Extreme Learning Machine (ELM) is a learning method that aims to overcome these limitations by randomly choosing weights connecting input vectors to hidden nodes and threshold values of hidden nodes [24, 23].

Given N arbitrary distinct samples $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$ are the data vectors and $\boldsymbol{t}_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \mathbb{R}^m$ are the target classes, a standard SLFN can be mathematically modeled as:

$$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g_i(\boldsymbol{w}_i\text{ï¿œ}\boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, \tag{21}$$

where $\boldsymbol{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes , b_i is the threshold of the ith node and $\tilde{N}$ is the number of hidden nodes. In matrix form:

$$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g_i(\boldsymbol{w}_i\text{ï¿œ}\boldsymbol{x}_j + b_i) = \boldsymbol{t}_j \longrightarrow \boldsymbol{H\beta = T}, \tag{22}$$

where these matrices are defined as

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{w}_1\boldsymbol{x}_1 + b_1 & \ldots & g(\boldsymbol{w}_{\tilde{N}}\boldsymbol{x}_j + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1\boldsymbol{x}_N + b_i & \ldots & g(\boldsymbol{w}_{\tilde{N}}\boldsymbol{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N\times\tilde{N}}, \tag{23}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_i^T \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}\times m} \quad \text{and } \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_i^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix}_{N\times m} \tag{24}$$

$\boldsymbol{H}$ is called the hidden layer output matrix. It's ith column is the ith hidden node output. For any SLFN, $\boldsymbol{H}$ is invertible and $\|\boldsymbol{H\beta - T} = 0\|$. There also exists an error $\varepsilon < \|\boldsymbol{H\beta - T}\|$ for a given $\tilde{N} \leq N$ [24]. The solution to the traditional SLFN would be: Find $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{b}}$ so that $\left\|\hat{\boldsymbol{H}}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{T}}\right\| = \min\limits_{\boldsymbol{w}_i, b_i, \boldsymbol{\beta}} \|\boldsymbol{H\beta - T}\|$.

The ELM learning approach proposes the following: For fixed input weights $\boldsymbol{w}_i$ and the hidden layer biases b_i, to train a SLFN is equivalent to finding least-squares solution $\hat{\boldsymbol{\beta}}$ of the linear system

$$\boldsymbol{H\beta = T.} \tag{25}$$

The smallest norm least-squares solution of the above system is

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H^\dagger T,} \tag{26}$$

where $\boldsymbol{H}^\dagger$ is the Moore–Penrose generalized inverse of $\boldsymbol{H}$. On a side note, $\boldsymbol{H}^\dagger$ can be calculated using Singular Value Decomposition or doing $(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$.

Finally, an ELM algorithm can be summarized as: Given training set of N $(\boldsymbol{x}_i, \boldsymbol{t}_i)$ samples, an activation function $g(x)$, and hidden node number $\tilde{N}$,

1. Randomly assign $\boldsymbol{w}_i$ and b_i.

2. Calculate $\boldsymbol{H}$.

3. Calculate $\boldsymbol{\beta} = \boldsymbol{H}^\dagger\boldsymbol{T}$.

The ELM described above is the basic ELM which was first proposed on [23]. Many more have been developed, in [25] - Random hidden layer feature mapping based ELM, Incremental ELM, etc.

The orthogonal projection method can be used to obtain $\boldsymbol{H}^{\dagger}$: $\boldsymbol{H}^{\dagger} = (\boldsymbol{H^T H})^{-1}\boldsymbol{H^T}$. In that case, we can add a ridge parameter $1/\lambda$ to the diagonal of $(\boldsymbol{H^T H})$. This regularization approach, known as ridge regression, stabilizes the solution [21]. Thus, the calculation of the output weights $\boldsymbol{\beta}$ is:

$$\boldsymbol{\beta} = \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H^T H}\right)^{-1} \boldsymbol{H^T T} \tag{27}$$

where $\boldsymbol{I}$ is an identity matrix the same size as $\boldsymbol{H}$. This variation of the basic ELM is called Random hidden layer feature mapping based ELM [25]. We will call it ELM-FM for convenience.

In addition to those described above, many more ELMs have been developed:, [25]: Kernel based ELM, sequential ELMs, incremental ELMs, etc.

3.2 Random Forest

A random forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_k, k = 1, ...$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\boldsymbol{x}$ [3]. Random Forest select inputs randomly. This randomness is chosen so that the correlation between two different members of the forest is minimized. A Random Tree is formed by selecting at random, at each node, a small group of input variables to split on. In our case, this number was set to $log_2 a + 1$, where a is the number of attributes. The tree grows using CART methodology to maximum size. Trees are not pruned.

3.3 Support Vector Machines

Support Vector Machines (SVMs) are linear or non-linear (with a kernel trick) non-probabilistic binary classifiers [8]. The class of SVM that we have used was introduced in [51]. When it is a regression method we call it SVR, when it is a classifier it's called SVC. The main idea behind SVMs is to build a hyperplane that best separates members of different classes. Let be $(\boldsymbol{x}_1, \boldsymbol{y}_1) \dots (\boldsymbol{x}_l, \boldsymbol{y}_l)$, our two-class labeled data set. It is said to be linearly separable if there exists a vector $\boldsymbol{w}$ and scalar b so that for all the elements of the training set

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1. \tag{28}$$

In the $\nu -$ SVM classification algorithm [51, 52], the optimization problem presented is to minimize

$$\tau(\boldsymbol{w}, \xi, \rho) = \frac{1}{2} \left\|\boldsymbol{w}\right\|^2 + \nu\rho \sum_{i=1}^{l} \xi_i \tag{29}$$

where $\|w\|^2$ is a term that characterizes the model complexity, the ξ are some variables and ν and ρ are two constants. This function is subject to the constraints

Tab. 1: Summary of the 4 databases used in our experiments.

	DB 1	DB 2	DB 3	DB 4
Number of samples	5169	3249	832	347
Number of classes	994	635	265	79
Mean (samples per class)	4.3924	3.1396	5.2835	5.2002
Standard deviation (samples per class)	5.8560	3.4498	4.9904	4.5012
Median (samples per class)	2	2	4	4
Mode (samples per class)	2	2	2	2

$$y_i\left((\boldsymbol{x}_i \boldsymbol{w}) + b\right) \geq \rho - \xi_i \tag{30}$$

$$\xi_i \geq 0\ ,\ \rho \geq 0. \tag{31}$$

The decision function, defining α_i that are $0 \leq \alpha_i \leq \frac{1}{l}$, and using a kernel k, takes the form

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b\right). \tag{32}$$

SVMs are binary classifiers, so we use one-against-one approach for multi-class classification. Details on the computation of b and ρ and justification of the preference of $\nu - \mathrm{SVM}$ over classic SVM are thoughtfully explained on [51].

4 Experimental design

We have performed two separate but related experiments. The goal was to obtain answers to two questions about ELMs:

1. Used as a preprocessing step for ELMs, is LICA a better than or comparable to other state-of-the-art feature extraction algorithms when dealing with big, unbalanced face databases? and

2. Can ELMs outperform state-of-the art classifiers in such experimental environment?

We based our experimental designs on the Color FERET database [42, 43]. Color FERET contains 10344 face images, varying in scale, rotation and lighting. There are also occlusions caused by glasses or hair. Some of the images are grayscale, but the vast majority are RGB. We chose frontal and mildly rotated images - with a rotation of 15 ,22.5 and 45 degrees. Representative face image samples can be seen in figure 1. This left us with 5175 facial photo candidates to build our experimental databases. Classes correspond to subject identities. These databases have a highly unbalanced class size distribution, as is illustrated in figure where we plot a histogram of the number of samples per class in the first selected database. Following the detection process described below, we made three additional face image subset selections, resulting in four experimental databases of 5169, 3249, 832 and 347 images respectively. Table 1 shows a summary of each database's main features.

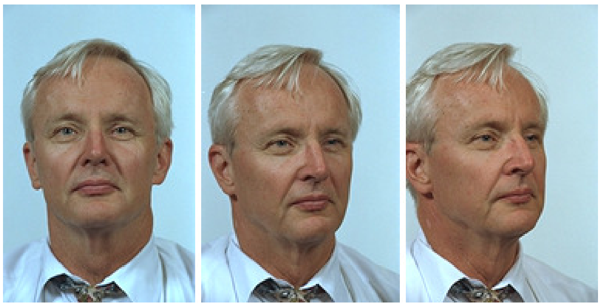

Fig. 1: Example of the rotation that we allowed. Images from Color FERET database [42].

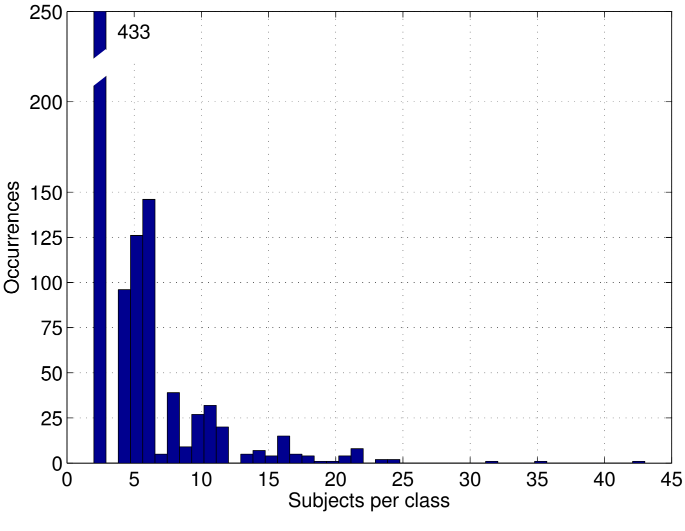

Fig. 2: Histogram showing the class distribution of the DB 1 database.

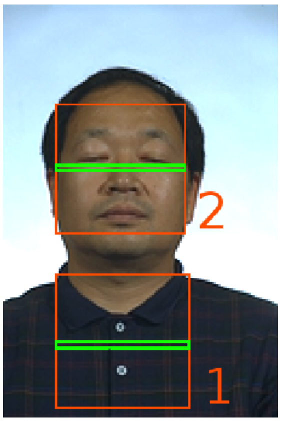

Fig. 3: Detection example. Orange squares show the first and second candidates. First candidate's middle row's RGB values are R=41.95 G=41.97 B=46.60. Second candidate's are R=133.03 G=106.84 U=79.49.

The faces were not suitable for recognition, because of the noise produced by different backgrounds and the differences in scale. Therefore, we used the detection algorithm developed in [59, 32] and available in Scilab SIVP. The algorithm usually detects several faces in a photography of a single subject. We added a face selection process based firstly on candidate's size. A second step checked if in the middle row's average color composition the red channel was predominant. This method works well under average lighting conditions and regardless of skin color. We did not modify the face area selected by the algorithm. We allowed a partial occlusion of the faces, up to a 20% of the face area. There were 18 detection failures. We also removed 6 detected faces because the provided ground-truth deviated from reality. Overall, this method achieved a success rate of 99.65%. The process is illustrated in figure 3. The next step was to scale images to 100x100 pixels using bicubic resampling. Then we needed to do a conversion from RGB to grayscale prior to feature extraction. We used a $Gr = 0.85 \cdot R + 0.10 \cdot G + 0.05 \cdot B$ conversion method which is reported to be the optimal grayscale conversion formula for face recognition [7].

Feature extraction was performed using the algorithms mentioned on section 2. PCA has no parameter whatsoever. LDA usually needs a previous dimension reduction phase. We performed Singular Value Decomposition (SVD) over the data retaining the maximum amount of eigenvectors. Both ICA Infomax and ICA-MS also require a the same preprocess. Mean-field ICA has several parameters, like prior mixing matrix, noise covariance, etc. We found that constant mixing matrix and noise covariance, as well as power law tail source prior. This method showed empirically the best results in a reasonable time.

Classifiers were also empirically tuned. The parameter of the ELMs was the number of hidden nodes, in addition to the ridge parameter λ in the case of ELM-FM. Random Forest only required to fix the number of trees. In the case of SVMs, we chose $\nu - \mathrm{SVM}$ because it showed better recognition rate that C-SVM. The ν parameter was also set empirically. Both the $\nu - \mathrm{SVM}$ kernel function and the ELM activation function were sigmoidal. We also tested two

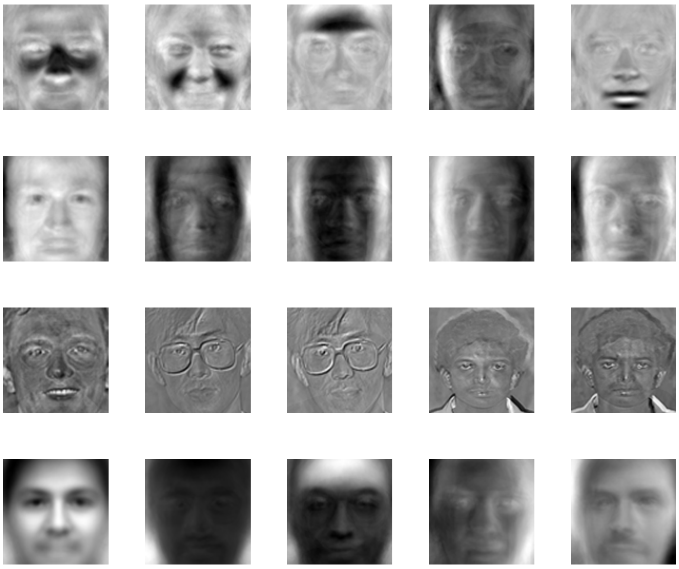

Fig. 4: An instance of the first 5 independent components (ICA Infomax and ICA MS), endmembers (LICA) and eigenvectors (PCA)

FFNNs with Backpropagation algorithm. One uses Resilient Backpropagation Algorithm (RPROP) [47] and the other Scaled Conjugate Gradient Algorithm (SCG) [36]. The five classifiers were tested with the four experimental databases described above, tuning their parameters to obtain the best accuracy possible. We performed 2-fold cross-validation. The recognition results are obtained based on 20 repetitions. In other words, in each of the 20 trials we randomly choose the 50% of the members of each class, having both testing and training set a similar size (not equal, because some classes contain an odd number of images).

5 Experimental results

Experiments were run on a Intel i5 2400 processor and 8 GB of RAM memory. Random Forest is resource greedy, and it's performance is limited by the amount of trees that computer's memory allows to grow. Other classification and feature extraction processes do not pose any computational resource-related problem. The following two subsections describe the results obtained, each corresponding to one of the two questions raised earlier in the section 4.

5.1 Results of LICA using Extreme Learning Machines

The computational experiments covered systematic dimensionality reduction up to 86, 107, 32 and 21 dimensions for databases DB 1, DB 2, DB 3 and DB

4, respectively. Working with dimensions above those limits did not show any increase in the accuracy of the algorithms. For ICA and PCA selecting the target dimension reduction was immediately accomplished selecting the desired sources and eigenvectors, respectively. For LICA that exploration implies varying the value of the α parameter and observing the number of endmembers detected. All feature extraction methods were evaluated in a wrapper scheme using an ELM for classification. The average number of hidden nodes was 1290 for DB 1, 870 for DB 2, 275 for DB 3 and 142 for DB 4.

Figure 5 shows the recognition rate for the smallest database DB 4. The database has high average number of images per class (5.2002) with a standard deviation of 4.5012. Most classes have 2 samples. The results show that LDA and PCA converge quickly to their maximum hit-rate. This small database with high class size variability seems to be unsuitable for some ICA methods, such as the Mean field ICA and the M&S ICA. Although showing worst results than LDA in 0 to 5 dimension space, LICA based classification obtains the best recognition rate for dimensions above 5. Notice that LDA is a supervised dimension reduction algorithm, so that the remaining algorithms have a strong handicap against LDA. Figure 6 provides the recognition results for the next bigger database DB 3. LICA is also the best feature extraction algorithm in this case, improving PCA and LDA. The ICA algorithms perform badly in this database. The change from DB 3 to DB 2, as shown in table 1, lies in the addition of much more classes with few samples. This makes the DB 2 database even more unbalanced and complex than DB 3 and DB 4. The performance of all feature extraction algorithms drops heavily. Nevertheless, LICA continues to offer the best results, followed by LDA, as seen in figure 7. The change from DB 2 to DB 1 is different. DB 1 has many more subjects with more than two samples, thus rising both the sample-per-class mean and standard deviation. The most sensitive algorithm to the cited change is LICA. While the other methods see a 10-20% drop in their hit-rate at most, LICA drops about a 40%. The most efficient algorithms when testing DB 1 are LICA and LDA, as shown on figure 5. We must remind the reader that LDA is a supervised feature extraction method, while LICA is unsupervised. The main conclusion of this collection of computational experiments is that LICA-ELM outperforms the remaining feature extraction algorithms.

5.2 Results of ELM compared to other classifiers

In order to evaluate the resilience of ELMs to unbalanced datasets such as those in face recognition problems, we extracted the LICA features from all the databases and tested the five classifiers described in section 4. Other algorithms Naive-Bayes, Multinomial Naive-Bayes, Radial Basis Function Networks or Multilayer Perceptrons were discarded after pilot experiments on the DB1 database that resulted in very low recognition (below 1%). The recognition results are summarized in table 2. We report the mean and standard deviation test accuracy over the databases, for all LICA feature dimensions.

The figure 9 plots the obtained results. The FFNNs, Random Forest and $\nu-$SVM obtain systematically decrease their accuracy results as the size of the database increases. When testing the two small databases, $\nu-$SVM improves Random Forest. The FFNN SCG algorithm reports better results that FFNN RPROP. It is interesting that ELM obtains the worst accuracy result in the

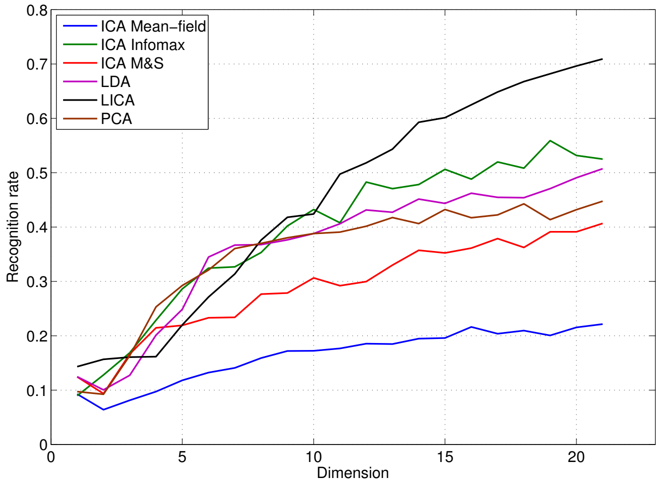

Fig. 5: ELM recognition rate on DB 4 (347 subjects).

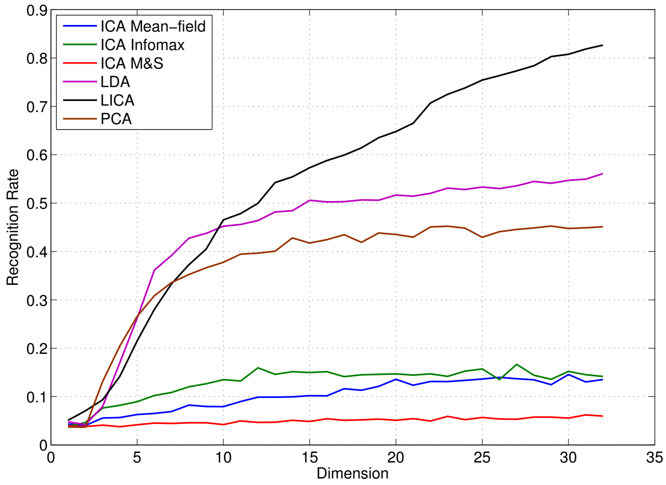

Fig. 6: ELM recognition rate on DB 3 (832 subjects).

Tab. 2: Testing accuracy average (variance) for 4 Color FERET database subsets on features computed by the LICA feature extraction algorithm.

	DB 4	DB 3	DB 2	DB 1
ELM [23]	0.7093 (0.0385)	0.8782 (0.0199)	0.5834 (0.0126)	0.4735 (0.0061)
ELM-FM [22]	0.9035 (0.0237)	0.8721 (0.0153)	0.5834 (0.0143)	0.4830 (0.0056)
Random Forest [3]	0.7719 (0.0100)	0.7506 (0.0489)	0.3457 (0.0135)	0.2431 (0.0126)
$\nu -$ SVM [51]	0.8713 (0.0012)	0.8509 (0.0334)	0.3572 (0.0148)	0.2111 (0.0094)
FFNN RPROP [47]	0.8494 (0.0217)	0.7800 (0.0201)	0.1448 (0.0084)	0.3719 (0.0228)
FFNN SCG [36]	0.8692 (0.0198)	0.8166 (0.0244)	0.1205 (0.0024)	0.2110 (0.0338)

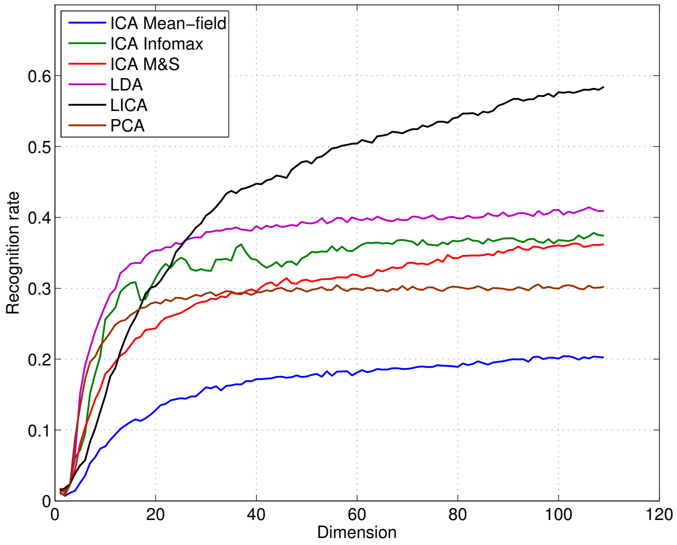

Fig. 7: ELM recognition rate on DB 2 (3249 subjects).

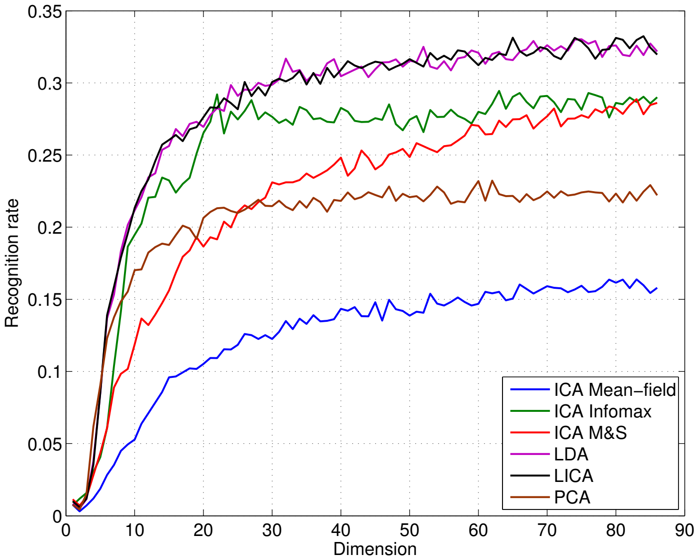

Fig. 8: ELM recognition rate on DB 1 (5169 subjects).

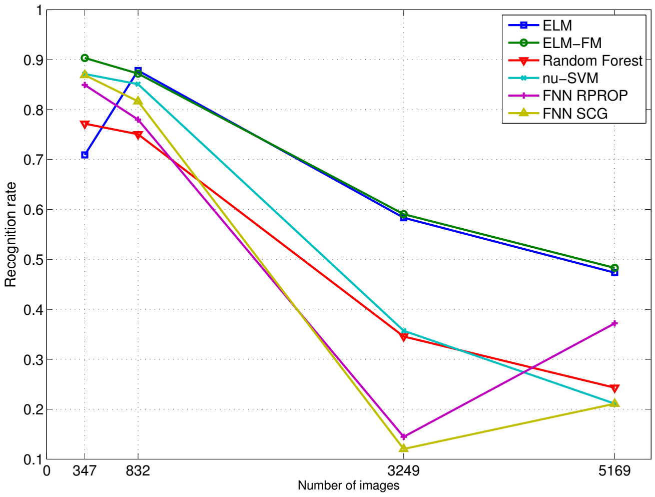

Fig. 9: Recognition rate on the 4 databases using ELM, Randon Forest, $\nu-$SVM, FFNN BPROP and FFNN SCG on features extracted with LICA.

DB 4 case but the best one in the remaining databases. ELM-FM algorithm, adding a regularization method, overcomes this disadvantage. ELM-FM obtains the best results in the small database and similar results than those of ELM in the other databases. ELMs systematically are more robust against introducing more classes and samples while maintaining the samples per class ratio. The experiments with DB 2 and DB 1 represent a big rise on complexity and database size. ELM is the algorithm that best deals under these circumstances. Specially in the DB 1 scenario, where it doubles the other algorithm's recognition rate. It's also noticeable that standard FFNNs perform poorly in those big complex databases. Particularly, FFNN SCG seems unable to train properly DB 2 and DB 1.Besides, we can assert that ELM's total time of training and testing was several magnitudes smaller.

6 Discussion

We have applied LICA and five well known feature extraction procedures to recognize faces on four subsets of a well known face database. We have also tried ELM and two widely used classifiers. The databases on which the experiments have been performed were unbalanced, large and complex. We draw the following conclusions from the obtained results:

- LICA is a better feature extraction algorithm for face recognition under the mentioned circumstances. It shows a better recognition rate in conjunction with ELM classifier than the rest of methods. LICA also is less likely to drop its effectiveness when we use smaller databases with high subject to class ratio variability. ICA methods depend highly on the number of samples of the database. LDA's results are more consistent, specially when dealing with the biggest database and high subject per class standard deviation. Overall, Lattice Computing-based LICA algorithm its approach to feature extraction is effective, being more competitive with large unbalanced databases, such as those common in face recognition applications.

- The joint use of LICA and ELM has retrieved the best recognition results. We can suggest that Lattice-based Endmember Induction Algorithms could be best fitted to work with ELMs than other statistical tools (PCA, LDA) or independent component extraction algorithms (ICA Infomax, ICA M&S, Mean-field ICA).

- It is stated in [26] that Naive-Bayes is more robust to higher levels of class noise then Random Forest and C-SVM. However, we have found that when dealing with large unbalanced face databases, Naive-Bayes is far outperformed by ELM, $\nu-$SVM and Random Forest. The same applies to Multinomial Naive-Bayes, Radial Basis Function Networks or Multilayer Perceptrons. Results were so bad that do not deserve publication here. There is no implementation bias as far as we applied the standard implementation found in Weka.

- Of all tested classifiers, ELM and ELM-FM are the most robust methods for large databases with high class-variation induced noise. It shows similar results than Random Forest or $\nu-$SVM when the databases are small. When the size is increased, ELM show an improvement of 124% and 95% over the results of $\nu-$SVM and Random Forest respectively. Furthermore, FFNNs with standard learning algorithms show worse performance than the rest of the classifiers. It is noteworthy that the regularization step added by ELM-FM to the basic ELM greatly increases the recognition accuracy in the smallest database.

The composition of LICA feature extraction and ELM classification show promising results in the domain of face recognition. More experiments over highly unbalanced databases could be performed on future works. It would also be valuable to test the various ELM algorithms, apart from basic-ELM available in the literature. We think that it would be interesting to explore further the interplay between Lattice Computing-based feature extraction methods and Extreme Learning Machines.

References

[1] 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*, 19(7):711–720, July 1997.

[2] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[3] Leo Breiman and E. Schapire. Random forests. In *Machine Learning*, volume 45, pages 5–32, 2001.

[4] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *in Proc. of the IEEE Int. Conf. on Comp. Vision (ICCV)*, Rio De Janeiro, 2007.

[5] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] Rama Chellappa, Pawan Sinha, and P. Jonathon Phillips. Face recognition by computers and humans. *IEEE Computer*, 43(2):46–55, 2010.

[7] Jae Y. Choi, Yong M. Ro, and Konstantinos N. Plataniotis. A comparative study of preprocessing mismatch effects in color image based face recognition. *Pattern Recognition*, 44(2):412–430, 2011.

[8] Corinna Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] Lehel Csato, Manfred Opper, and Ole Winther. Tap gibbs free energy, belief propagation and sparsity. In *Advances in Neural Information Processing Systems*. MIT Press, 2001.

[10] Danmarks-Tekniske-Universitet. Ica:dtu toolbox, 2002. http://cogsys.imm.dtu.dk/toolbox/.

[11] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. Recognizing faces with pca and ica. *Computer Vision and Image Understanding*, 91(1-2):115 – 137, 2003. ISSN 1077-3142. Special Issue on Face Recognition.

[12] M. Graña. A brief review of lattice computing. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1777 –1781, June 2008.

[13] M. Graña, I. Villaverde, J.O. Maldonado, and C. Hernandez. Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. *Neurocomputing*, 72(10-12):2111–2120, 2009.

[14] M. Graña, A. Manhaes-Savio, M. García-Sebastián, and E. Fernandez. A lattice computing approach for on-line fmri analysis. *Image and Vision Computing*, 28(7):1155–1161, 2010.

[15] Manuel Graña, Darya Chyzhyk, Maite García-Sebastián, and Carmen HernÃ¡ndez. Lattice independent component analysis for functional magnetic resonance imaging. *Information Sciences*, 181(10):1910 – 1928, 2011. ISSN 0020-0255. Special Issue on Information Engineering Applications Based on Lattices.

[16] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorer Newsetter.*, 11(1):10–18, November 2009.

[17] L. K. Hansen, J. Larsen, and T. Kolenda. *On Independent Component Analysis for Multimedia Signals*. CRC Press, 2000.

[18] L.K. Hansen, J. Larsen, and T. Kolenda. Blind detection of independent dynamic components. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 5, pages 3197 –3200, 2001.

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

[20] Guillaume Heusch and Sebastien Marcel. A novel statistical generative model dedicated to face recognition. *Image and Vision Computing*, 28(1): 101 – 110, 2010. ISSN 0262-8856.

[21] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, February 1970.

[22] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multi-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted, 2010.

[23] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985 – 990, july 2004.

[24] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501, 2006.

[25] Guang-Bin Huang, Dianhui Wang, and Yuan Lana. Extreme learning machines: A survey. *International Journal of Machine Leaning and Cybernetics*, in Press:107–122, April 2011.

[26] Jason Van Hulse and Taghi Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513 – 1542, 2009. ISSN 0169-023X. Including Special Section: 21st IEEE International Symposium on Computer-Based Medical Systems (IEEE CBMS 2008) - Seven selected and extended papers on Biomedical Data Mining.

[27] Aapo Hyvarinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[28] Pedro A.d.F.R. HÃžjen-SÃžrensen, Ole Winther, and Lars Kai Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 14(4):889–918, 2002.

[29] Sarvani Kare, Ashok Samal, and David Marx. Using bidimensional regression to assess face similarity. *Machine Vision and Applications*, 21(3): 261–274, 2008.

[30] Jinkwon Kim, Hang Sik Shin, Kwangsoo Shin, and Myoungho Lee. Robust algorithm for arrhythmia classification in ecg using extreme learning machine. *Biomedical Enineering Online*, 8:article–number 31, 2009.

[31] DD Lee and HS Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. ISSN 0028-0836.

[32] R Lienhart and J Maydt. An extended set of haar-like features for rapid object detection. In *2002 International Conference On Image Processing, Proceedings*, volume 1, pages 900–903. IEEE Signal Proc Soc, 2002.

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

[34] Abdul A. Mohammed, Q. M. Jonathan Wu, and Maher A. Sid-Ahmed. Application of wave atoms decomposition and extreme learning machine for fingerprint classification. In *Image Analysis and Recognition, 2010, PT II, Proceedings*, volume 6112 of *Lecture Notes in Computer Science*, pages 246–255. Springer-Verlag, 2010.

[35] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72:3634–3637, 1994.

[36] Martin F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993.

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

[38] H.B. Nielsen. Ucminf - an algorithm for unconstrained, nonlinear optimization. Technical Report IMM-TEC-0019, IMM, Technical University of Denmark, 2001.

[39] Manfred Opper and Ole Winther. Tractable approximations for probabilistic models: The adaptive thouless-anderson-palmer mean field approach. *Phys. Rev. Lett.*, 86(17):3695–3699, Apr 2001.

[40] Manfred Opper and Ole Winther. Adaptive and self-averaging thouless-anderson-palmer mean-field theory for probabilistic modeling. *Phys. Rev. E*, 64(5):056131, Oct 2001.

[41] Yaozhang Pan, Shuzhi Sam Ge, Hongsheng He, and Lei Chen. Real-time face detection for human robot interaction. In *RO-MAN 2009: The 18th IEEE International Symposium On Robot And Human Interactive Communication*, volume 1 and 2, pages 15–20, 2009.

[42] P.J. Phillips, H. Wechsler, and P. Rauss J. Huang. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[43] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[44] Mattia C. F. Prosperi, Andre Altmann, Michal Rosen-Zvi, Ehud Aharoni, Gabor Borgulya, Fulop Bazso, Anders Sonnerborg, Eugen Schuelter, Daniel Struck, Giovanni Ulivi, Anne-Mieke Vandamme, Jurgen Vercauteren, Maurizio Zazzi, and EuResist Virolab Study Grp. Investigation of expert rule bases, logistic regression, and non-linear machine learning techniques for predicting response to antiretroviral treatment. *Antiviral Therapy*, 14(14): 433–442, 2009.

[45] Lishan Qiao, Songcan Chen, and Xiaoyang Tan. Sparsity preserving discriminant analysis for single training image face recognition. *Pattern Recognition Letters*, 31(5):422 – 429, 2010. ISSN 0167-8655.

[46] Chuan-Xian Ren and Dao-Qing Dai. Incremental learning of bidirectional principal components for face recognition. *Pattern Recognition*, 43(1):318 – 330, 2010. ISSN 0031-3203.

[47] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, volume 1, pages 586–591, 1993.

[48] Gerhard X. Ritter and Gonzalo Urcid. A lattice matrix method for hyperspectral image unmixing. *Information Sciences*, 181(10):1787–1803, May 2010. ISSN 0020-0255.

[49] G.X. Ritter, P. Sussner, and J.L. Diaz de Leon. Morphological associative memories. *Neural Networks, IEEE Transactions on*, 9(2):281 –293, mar. 1998. ISSN 1045-9227.

[50] G.X. Ritter, G Urcid, and Schmalz M.S. Autonomous single-pass endmember approximation using lattice auto-associative memories. *Neurocomputing*, 72(10-12):2101–2110, 2009.

[51] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[52] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13:1443–1471, July 2001. ISSN 0899-7667.

[53] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, MAY 4 2009. ISSN 0262-8856.

[54] Zhan-Li Sun, Tsan-Ming Choi, Kin-Fan Au, and Yong Yu. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, 46(1):411–419, December 2008. ISSN 0167-9236.

[55] Hui-Xin Tian and Zhi-Zhong Mao. An ensemble elm based on modified adaboost.rt algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Transactions On Automation Science And Engineering*, 7 (1):73–80, January 2010.

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

[57] MA Veganzones and M Graña. Endmember extraction methods: A short review. In *Knowledge-Based Intelligent Information and Engineering Systems, pt 3*, volume 5179 of *Lecture Notes In Computer Science*, 2008.

[58] Ivan Villaverde, Borja Fernandez-Gauna, and Ekaitz Zulueta. Lattice independent component analysis for mobile robot localization. In E Corchado, MG Romay, and AM Savio, editors, *Hybrid Artificial Intelligence Systems, pt 2*, volume 6077 of *Lecture Notes in Artificial Intelligence*, pages 335–342. Springer-Verlag, 2010. ISBN 978-3-642-13802-7.

[59] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518, 2001.

[60] W.S. Yambor. *Analysis of PCA-based and Fisher Discriminant-Based Image Recognition Algorithms.* Technical report cs-00-103, Computer Science Department, Colorado State University, July 2000.

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