14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

FACE VERIFICATION WITH A MLP AND BCH OUTPUT CODES[1]

Marcos Faundez-Zanuy

Escola Universitària Politècnica de Mataró, Univesistat Politècnica de Catalunya
Avda. Puig i Cadafalch 101-111, 08303 MATARO (BARCELONA), SPAIN
Phone: +(34) 93 757 44 04 Fax: +(34) 93 757 05 24 e-mail: faundez@eupmt.es

ABSTRACT

This Paper studies several classifiers based on Multi-layer Perceptrons (MLP) for face verification. We use the Discrete Cosine Transform (DCT) instead of the eigenfaces method for feature extraction. Experimental results using a Nearest Neighbour classifier show a minimum Detection Cost Function (DCF) of 1.76% when using DCT, and 7.14% when using eigenfaces. We also study several MLP architectures, and we get better accuracies when using Bose-Chaudhuri-Hocquenghem (BCH) codes. In this case, we reduce the minimum DCF to 0.97% when using DCT feature extraction.

1. INTRODUCTION

Face recognition is probably the most natural way to perform a biometric authentication between human beings. However, the technology still presents some drawbacks, which have been described in the literature, such as Vulnerability [1], Privacy [2], and others [3].

In this paper we use a DCT approach in combination with a neural net classifier and data fusion [4]. Experimental results have been evaluated using the DET plots [5] and reveal a significant improvement over previous techniques [6].

1.1 The eigenface approach

Turk and Pentland [7], proposed an eigenface system which projects face images onto a feature space that spans the significant variations among known face images using the Karhunen-Loéve Transform. It is an orthogonal lineal transform of the signal that concentrates the maximum information of the signal with the minimum number of parameters using the minimum square error (MSE). The significant features are known as eigenfaces, because they are the eigenvectors (principal components) of the set of images. The projection operation characterizes an individual face by a weighted sum of the eigenface features, and so to recognize a particular face it is only necessary to compare these weights to those of known individuals.

Recognition is performed by finding the training face that minimizes the face distance with respect to the input test face. In other terms, the identification of the test image is done locating the database entry, whose weights are closest (in Euclidean distance) to the weights of the face.

1.2 The DCT approach

The Discrete Cosine Transform (DCT) is closely related to the Discrete Fourier Transform (DFT). It is a separable, linear transformation; that is, the two-dimensional transform is equivalent to a one-dimensional DCT performed along a single dimension followed by a one-dimensional DCT in the other one.

The application of the DCT to an image (real data), produces a real result. The DCT tends to concentrate information, making it useful for image compression applications, dimensionality reduction, etc.

An important advantage of the DCT is that basis functions are not data dependent (it will generalize better than KLT).

2. EXPERIMENTAL RESULTS

This section evaluates the results obtained using the dct and compares them with the classical eigenface method. On the other hand, several classifiers are applied.

2.1 Database

The database used is the ORL (Olivetti Research Laboratory) faces database [8]. This database contains a set of face images taken between April 1992 and April 1994 at ORL. The database was used in the context of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department.

There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

In [9] the minimum size of the test data set, *n*, which guarantees statistical significance in a pattern recognition task, is derived. The goal in the abovementioned work is

[1] This work has been supported by FEDER and the Spanish grant MEC TIC2003-08382-C05-02.

to estimate n so that it is guaranteed, with a risk α of being wrong, that the error rate p does not exceed that estimated from the test set, $\hat{P}$, by an amount larger than $\varepsilon(N,\alpha)$, that is, $\Pr\left\{P > \hat{P} + \varepsilon(N,\alpha)\right\} < \alpha$. Letting $\varepsilon(N,\alpha) = \beta P$ and supposing recognition errors as bernoulli trials (i.i.d. Errors), we can derive the following relation after some approximations: $N \approx \frac{-\ln\alpha}{\beta^2 P}$. For typical values of α and β (α =0.05 and β =0.2), the following simplified criterion is obtained: $N \approx \frac{100}{P}$

If the samples in the test data set are not independent (due to correlation factors that may include variations in recording conditions, in the type of sensors, etc.), then n must be further increased. The reader is referred to [8] for a detailed analysis of this case, where some guidelines for computing the correlation factors are also given. In [6] we compared several algorithms results using ORL and FERET databases and we found that ORL is statistically significant for verification applications studies. On the other hand, the reduced set of people lets to perform fast simulations when compared against FERET.

2.2 Conditions of the experiments

Our results have been obtained with the ORL database [8] in the following situation: 40 persons, faces 1 to 5 for training, and faces 6 to 10 for testing.

We obtain one model from each training image. During testing each input image is compared against all the models inside the database (40x5=200 in our case) and the closest model to the input image (using Mean Square Error criterion) indicates the recognized person.

Biometric recognition systems can be operated in two ways:

a) Identification: In this approach no identity is claimed from the person. The automatic system must determine who is trying to access.

b) Verification: In this approach the goal of the system is to determine whether the person is who he/she claims to be. This implies that the user must provide an identity and the system just accepts or rejects the users according to a successful or unsuccessful verification. Sometimes this operation mode is named authentication or detection.

For identification, if we have a population of N different people, and a labelled test set, we just need to count the number of identities correctly assigned.

Verification systems can be evaluated using the False Acceptance Rate (FAR, those situations where an impostor is accepted) and the False Rejection Rate (FRR, those situations where a user is incorrectly rejected), also known in detection theory as False Alarm and Miss, respectively. There is trade-off between both errors, which has to be usually established by adjusting a decision threshold. The performance can be plotted in a ROC (Receiver Operator Characteristic) or in a DET (Detection error trade-off) plot [5].

We have used the minimum value of the Detection Cost Function (DCF) for comparison purposes. This parameter is defined as [5]:

$$DCF = C_{miss} \times P_{miss} \times P_{true} + C_{fa} \times P_{fa} \times P_{false} \qquad \textbf{(1)}$$

Where c_{miss} is the cost of a miss (rejection), c_{fa} is the cost of a false alarm (acceptance), p_{true} is the a priori probability of the target, and $p_{false} = 1 - p_{true}$. We have used $c_{miss} = c_{fa} = 1$.

In our experiments, we are making for each user, all other users' samples as impostor test samples, so we finally have, that n=40×5 (client)+40×39×5 (impostors)=8000. So, with 95% confidence, our experiments guarantee statistical significance in experiments with an empirical error rate, $\hat{P}$, down to 1.25%, which is certainly suitable for our experiments.

2.3 Dimensionality reduction using the DCT

The first experiment consisted of the evaluation of the identification rates as function of the vector dimension. Thus, 200 tests (40 persons x 5 test images per person, being each image size 92x112 pixels) were performed for each vector dimension (92 different vector dimensions) and the corresponding identification rates were obtained. Experimental results revealed better performance for N'=100 coefficients.

The classifier consists of a nearest neighbor (NN) classifier using the Mean Square Error (MSE) or the Mean Absolute Difference (MAD) defined as:

$$MSE(\vec{x},\vec{y}) = \sum_{i=1}^{(N')^2} (x_i - y_i)^2 \qquad \textbf{(2)}$$

$$MAD(\vec{x},\vec{y}) = \sum_{i=1}^{(N')^2} |x_i - y_i| \qquad \textbf{(3)}$$

Where N' is the dimensionality of the vectors that represent faces.

Table 1. Comparison between different systems

FEATURE EXTRACTION	VECTOR DIMENS.	CLASSIFIER	DCF (%)
EIGENFACES	200	NN (MAD)	7.14
EIGENFACES	100	NN (MAD)	7.67
DCT	100	NN (MAD)	6.28
DCT	100	NN (MSE)	5.84
DCT	100	MLP	1.76
DCT	100	RBF	1.62
DCT+EIGENFA	100	RBF+NN (MAD)	1.35
DCT	100	RBF+NN (MAD)	1.5

In addition, we have also tested two different neural network architectures, acting as classifiers: Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) [9-11].

Table 1 compares the results obtained with different feature extraction, classifiers, and some combinations. We provide results for verification (minimum value of the Detection Cost Function).Table 1 reveals that the DCT approach outperforms face verification using eigenfaces. In addition, the neural net classifier outperforms the nearest neighbour classifier, probably because it is a discriminative learning.

2.4 Neural net classifier trained in a discriminative mode

In our experiments, a neural net has been trained as discriminative classifier in the following fashion: when the input data belongs to a genuine person, the output (target of the NNET) is fixed to 1. When the input is an impostor person, the output is fixed to –1. Figure 1 shows the obtained intra/inter-distance histogram result for a face recognition system using a Multi-Layer Perceptron (MLP) and ORL database with the conditions given in previous section. A fitted Gaussian is also plotted in each histogram.

In this example, the number of genuine training samples is 40×5, while the number of impostors is 40×40×5–40×5=39×40×5. It is interesting to observe that There is a preponderance of the negative responses. This is because of the most part of the training vectors are inhibitory. Thus, the MLP tends to learn that "all is inhibitory". Although we bounded the MLP to learn +1 or –1, all the values are shifted to negative ones (the mean of the genuine values is close to 0 and far of 1). In general, in patter recognition applications, the number of samples for impostors is always higher than the number of genuine persons, because each person can be considered as impostor for all the other ones.

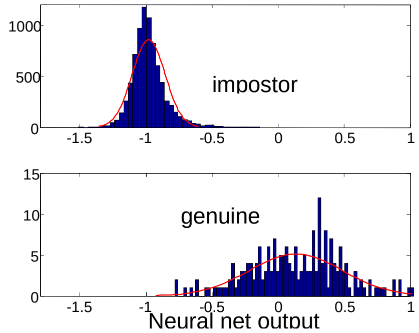

Figure 1 Inter and intra distance histograms for one-per-class approach.

One of the problems that occur during neural network training is called *overfitting*: the error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. The adopted solution to the overfitting problem has been the use of regularization. The regularization involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set. So, this technique helps take the mystery out of how to pick the number of neurons in a network and consistently leads to good networks that are not overtrained. In addition, there is another important topic: the random initialization. We have done 100 random initializations in each experiment, and we provide the minimum error, mean error and standard deviation.

2.5 Error correction codes

Error-control coding techniques [13] detect and possibly correct errors that occur when messages are transmitted in a digital communication system. To accomplish this, the encoder transmits not only the information symbols, but also one or more redundant symbols. The decoder uses the redundant symbols to detect and possibly correct whatever errors occurred during transmission.

Block coding is a special case of error-control coding. Block coding techniques map a fixed number of message symbols to a fixed number of code symbols. A block coder treats each block of data independently and is a memoryless device. The information to be encoded consists of a sequence of message symbols and the code that is produced consists of a sequence of codewords. Each block of k message symbols is encoded into a codeword that consists of n symbols; in this context, k is called the message length, n is called the codeword length, and the code is called an $[n, k]$ code.

A message for an $[n, k]$ BCH (Bose-Chaudhuri-Hocquenghem) code must be a k-column binary Galois array. The code that corresponds to that message is an n-column binary Galois array. Each row of these Galois arrays represents one word.

BCH codes use special values of n and k:

- n, the codeword length, is an integer of the form $2^m–1$ for some integer $m > 2$.
- k, the message length, is a positive integer less than n.

However, only some positive integers less than n are valid choices for k. This code can correct all combinations of t or fewer errors, and the minimum distance between codes is $d_{\min} \geq 2t+1$. Table 2 shows some examples of suitable values for BCH codes.

n	7	5			31				
k	4	11	7	5	26	21	16	11	6
t	1	1	2	3	1	2	3	5	7

Table 2. Examples of values for BCH codes.

2.6 Multi-class learning problems via error-correction output codes.

Multi-class learning problems involve finding a definition for an unknown function $f(\vec{x})$ whose range is a discrete set containing $k > 2$ values (i.e. k classes), and $\vec{x}$ is the set of measurements that we want to classify. The definition is acquired by studying large collections of training examples of the form $\{\vec{x}_i, f(\vec{x}_i)\}$.

We must solve the problem of learning a k-ary classification function $f : \Re^n \to \{1,\cdots,k\}$ from examples of the form $\{\vec{x}_i, f(\vec{x}_i)\}$. The standard neural network approach to this problem is to construct a 3-layer feed-forward network with k output units, where each output unit designates one of the k classes. During training, the output units are clamped to 0.0, except for the unit corresponding to the desired class, which is clamped at 1.0. During classification, a new $\vec{x}$ value is assigned to the class whose output unit has the highest activation. This approach is called [14-15] the ***one-per-class*** approach, since one binary output function is learnt for each class.

An alternative method, proposed in [14-15] and called ***error-correcting output coding*** (ECOC), gives superior performance. In this approach, each class i is assigned an m-bit binary string, c_i, called a codeword. The strings are chosen (by BCH coding methods) so that the Hamming distance between each pair of strings is guaranteed to be at least d_{min}. During training on example $\vec{x}$, the m output units of a 3-layer network are clamped to the appropriate binary string $c_{f(\vec{x})}$. During classification, the new example $\vec{x}$ is assigned to the class i whose codeword c_i is closest (in Hamming distance) to the m-element vector of output activations. The advantage of this approach is that it can recover from any $t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$ errors in learning the individual output units. Error-correcting codes act as ideal distributed representations.

In [14-15] some improvements using this strategy were obtained when dealing with some classification problems, such as vowel, letter, soybean, etc., classification. In this paper, we apply this same approach for biometric face recognition.

If we observe the output codes (targets) learnt by the neural network when the input pattern $\vec{x} \in k$ user, we can see that just the output number k is activated, and the number of outputs is equal to the number of users.

If we observe the output codes (targets) where each user has his own code, and these codes are selected from the BCH [n, k] codes, in fact, it yields up to 2^k output codes. However, we just need 40, because this is the number of users. It is interesting to observe that BCH [n, k] provides a more balanced amount of ones and zeros, while in one-per-class approach almost all the outputs will be inhibitory.

3. EXPERIMENTAL RESULTS

We use a Multi-layer perceptron with 100 inputs, and h hidden neurons, both of them with *tansig* nonlinear transfer function.

This function is symmetrical around the origin. Thus, we modify the output codes replacing each "0" by "–1". In addition, we normalize the input vectors $\vec{x}$ for zero mean and maximum modulus equal to 1.

Figure 1 shows the histograms of the neural net outputs for genuine scores (top) and impostors (bottom), using one-per-output approach. A fitted Gaussian is also plotted for each distribution. Figure 2 shows the same information when using BCH (31, 6) output codes during training (targets). It corresponds to MSE (see equation 9) computation between expected values (changing "0" per "–1") and obtained outputs. Obviously equations 9 and 10 yield a resulting *distance*, which is always greater or equal to zero. For this motivation, and for comparison purposes, we have plotted (1 – *distance*). For this reason, the maximum value in previous figure is equal to 1. A good error-correcting output code for a k-class problem should satisfy two properties [14]:

- Row separation: each codeword should be well-separated in Hamming distance from each of the other codewords.
- Column separation: each bit-position function f_i should be uncorrelated from the functions to be learnt for the other bit positions $f_j, j \neq i$.

Error- correcting codes only succeed if the errors made in the individual bit positions are relatively uncorrelated, so that the number of simultaneous errors in many bit positions is small. For this purpose, we have used the algorithm proposed in [16] for random ECOC generation. On the other hand, ECOC approaches can be interpreted as a combination of pattern classifiers [16].

For 40 outputs (classifiers) and 40 users we get a minimum row (class) Hamming distance H_c = 24 bit and minimum column (classifier) Hamming distance H_L = 24 bit, after 500 random iterations. On the other hand, BCH (31,6) provides H_c = 15 bit and H_L = 16 bit, and BCH (15, 7) implies H_c = 10 bit and H_L = 16 bit.

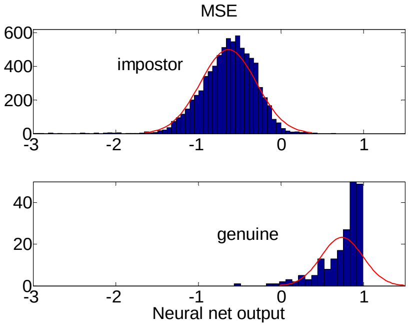

Figure 2. MLP histograms for BCH approach, using MSE.

We will summarize the Multi-Layer Perceptron number of neurons in each layer using the following nomenclature: *inputs*× *hidden*× *output*. In our experiments, the number of inputs is fixed to 100, and the other parameters can vary according to the selected strategy.

Table 3 shows the results of a single MLP with 40 outputs (one-per-class), BCH, and ECOC strategies.

CLASSIFIER	STRATEGY	MIN (DCF) (%)		
MLP		MEAN	σ	MIN
100×40×40	1-PER-CLASS	2.37	0.19	1.7
100×40×14	BCH MAD	3.91	0.41	3.05
100×40×14	BCH MSE	3.07	0.36	2.35
100×40×31	BCH MAD	1.47	0.096	1.17
100×40×31	BCH MSE	1.24	0.094	0.97
100×40×40	ECOC MAD	3.03	0.29	2.43
100×40×40	ECOC MSE	1.7	0.34	1.03

Table 3. Minimum Detection Cost function for several strategies.

3.1 Improvements offering several trials to verify.

One way to improve the face verification application is to offer several trials, in a similar fashion than the ATM machines, which offer three trials for entering the password. In our case, we have evaluated the system with 5 trials per person. This is equivalent to a data fusion on the decision level (parallel combination) [4]. Figure 3 shows the difference. It plots the FAR and FRR magnitudes for different thresholds, using a combination of RBF+NN (MAD). The output probabilities of each classifier have been previously normalized.

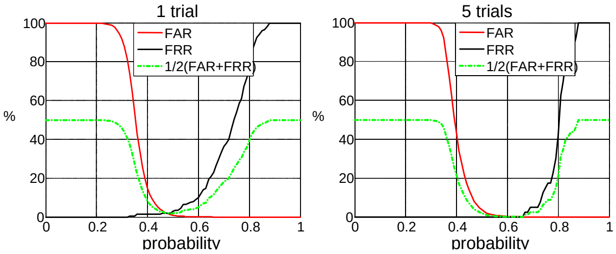

Figure 3. FAR and FRR offering 1 and 5 trials.

Those users with a probability higher than the threshold can pass, while the others are rejected. With five trials (a user can enter if he achieves at least one trial with higher probability than the threshold), there is a slight degradation of FAR, and a great improvement on FRR. Using a proper threshold setup, it is possible to trade-off both magnitudes and to get FAR=FRR=0%. In addition, the threshold setup is less critical, because there is a wider range of values that provide good results. However, we have not worked out more exhaustive experimentation of "several trials to verify" strategy, due to the limited amount of testing samples.

4. CONCLUSIONS

In this paper we have proposed the use of DCT for feature extraction in combination with a discriminative MLP classifier. Several strategies have been studied, and BCH output correction codes outperform classical results, providing a minimum DCF equal to 0.97%, while classical eigenfaces approach and NN classifier provides 7.14%.

5. REFERENCES

[1]Faundez-Zanuy, M., "On the vulnerability of biometric security systems". IEEE Aerospace and Electronic Systems Magazine. Vol.19 nº 6, pp.3-8, June 2004.

[2]Faundez-Zanuy, M., "Privacy issues on biometric systems". IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 2, pp13-15. February 2005

[3]Faundez-Zanuy, M., "Biometric recognition: why not massively adopted yet?". IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 8, pp.25-28, August 2005.

[4]Faundez-Zanuy M., "Data fusion in biometrics" IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 1, pp.34-38, January 2005.

[5]Martin A., Doddington G., Kamm T., Ordowski M., and Przybocki M., "The DET curve in assessment of detection performance", V. 4, pp.1895-1898, Eurospeech 1997

[6]Roure J., Faundez-Zanuy, M. "face recognition with small and large size databases" IEEE ICCST'2005, Pp.153-156. October 2005

[7]Turk M. & Pentland A., "Eigenfaces for Recognition" Journal Cognitive Neuroscience, Vol. 3, nº 1 pp 71-86, Massachusetts Institute of Thecnology.1991.

[8]Samaria F., & Harter A. "Parameterization of a stochastic model for human face identification". 2nd IEEE Workshop on Applications of Computer Vision. December 1994, Sarasota (Florida).

[9]Guyon, I., Makhoul, J., Schwartz, R., and Vapnik, V.: 'What size test set gives good error rate estimates?', IEEE Trans. Pattern Anal. Mach. Intell., 1998, 20, (1), pp. 52–64

[10]Schalkoff R.,"Pattern recognition statistical, structural and neural approaches" Ed. John Wiley & sons Inc. 1992

[11]Bishop C.M. "Neural networks for pattern recognition" Ed. Clarendon press. 1995

[12]Haykin S., "Neural nets. A comprehensive foundation", 2on edition. Ed. Prentice Hall 1999

[13]Faundez-Zanuy, M. "Biometric verification of humans by means of hand geometry". 39º IEEE International Carnahan Conference on Security Technology, October 2005.

[14]Wicker, Stephen B., Error Control Systems for Digital Communication and Storage, Upper Saddle River, N.J., Prentice Hall, 1995.

[15]Dietterich T. G., Bakiri G., "Error-correcting output codes: A general method for improving multiclass inductive learning programs". Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91). Anaheim, CA: AAAI Press.

[16]Dietterich T., "Do Hidden Units Implement Error-Correcting Codes?" Technical report 1991

[17]Kuncheva, L. I. "Combining pattern classifiers". Ed. John Wiley & Sons 2004.