Face Recognition based on Discrete Cosine Transform and Support Vector Machines

GABRIEL CHAVES AFONSO COUTINHO[1]
CRISTIANO LEITE DE CASTRO[2]

UFLA - Federal University of Lavras
DCC - Department of Computer Science
P.O. Box 3037 - Campus da UFLA 37200-000 - Lavras (MG)- Brazil
[1]gabrielcac@gmail.com
[2]ccastro@dcc.ufla.br

Abstract. In this paper we propose a new method for face recognition using a Support Vector Machine and the Discrete Cosine Transform. Experiments were conducted with two common benchmark face databases: the Yale and the AT&T.

Keywords: Facial Recognition, Support Vector Machines, Discrete Cosine Transform.

(Received February 23, 2010 / Accepted May 05, 2010)

1 Introduction

Computer-based face recognition has attracted the interest of many researchers in later years mainly due to its applicability in security systems, surveillance and criminal identification. Although advances have been reported and new solutions have been proposed, automatic face recognition is still considered a difficult computational problem [1], [20], [4]. Inherent issues of face set (raw data) such as the similarity of human faces and the unpredictable variations caused by 3D pose, facial expressions, lighting direction, and aging, are the greatest obstacles for designing a robust face recognition system.

According to [20], the face recognition problem can be divided into three stages: face detection, feature extraction, and recognition (classification). The first stage aims to detect (find the location) faces in input images. This is typically achieved by traditional techniques of image processing such as edge map, signature, hierarchical coarse-to-fine searches with template-based matching criteria, etc. [10], [6]. This stage was not considered in this paper, since the images used to evaluate the proposed methodology contain only the face of individuals.

Considering the feature extraction stage, computer-based face recognition approaches can be grouped into two categories: feature-based and holistic (or global) [20], [3]. Feature-based approaches basically rely on the shapes and geometrical relationships of individual facial features including eyes, mouth, nose and chin. Although the methods of this category are more robust to positional variations of the faces in the input image, e.g., rotation, scale, etc., they are highly dependent on the accuracy of facial feature detection techniques. For a detailed discussion of feature-based approaches, see [3] and [20].

The holistic approaches in turn, encode the images globally, extracting features in terms of components of the input signals (images). Once encoded, the face images are represented as points in high-dimension feature space. Many holistic methods proposed in the literature are based on Karhunen-Loeve transform (KLT) [16], [12], also known as principal component analysis (PCA). When applied to an input image, KLT produces an expansion in terms of a set of basis images or the so-called "eigenfaces". However, it has been observed that KLT does not achieve adequate robustness against variations in face orientation position and light-

ing [16]. To overcome this limitation, other methods such as Fisher's linear discriminant (FLD) [8] and its variants have been applied after KLT (on the space of feature vectors obtained by the KLT) to reduce the dimension and to obtain the most discriminating features. Compared with KLT being applied alone, the combined approach KLT + FLD ("fisherfaces") is more insensitive to large variations in lighting and facial expression. Nevertheless, it is worth notice that the computational requirements of this approach (KLT + FLD) are greatly related to the dimension of the original images and the number of training examples (individuals). Moreover, the KLT is not only more computationally intensive, but it must also be redefined every time the statistics of its input signals change, i.e., the "eigenfaces" (or "eigenvectors") should be recalculated every time a new face is added to the known face set [16]. A alternative holistic approach for face recognition is the discrete cosine transform (DCT) [13], [11]. Of the deterministic discrete transforms, the DCT is the one that best approaches the KLT. Moreover, DCT is independent of data set changes and can be implemented using fast and efficient algorithms [9].

Finally, the stage of recognition can be viewed as a pattern classification problem, where a classifier (or learning machine), learned from a known face set (training set), is used to assign a given unknown image, represented by a feature vector extracted in the previous stage, to one of the available classes (individuals). Among the learning machines proposed in the literature, support vector machines (SVMs) [7] have been successfully applied to many pattern classification real problems, including face regognition [15]. SVMs are based on Vapnik Chervonenkis' theory and the structural risk minimization principle [18], [19] which aims to obtain a classifier with high generalization performance through minimization of the training error and the complexity of the learning machine.

In this paper a face recognition approach based on discrete cosine transform (DCT) and support vector machines (SVMs) is investigated on two benchmark data bases: Yale and AT&T. An efficient algorithm for computing DCT is used on each image for obtaining feature vectors of 64 dimensions. Such vectors are used for teaching a SVM. After the learning stage, the proposed recognition system is then evaluated in terms of accuracy for independent test sets, also extracted from the preprocessed databases.

This paper is organized as follows: Section 2 reviews the theoretical concepts of DCT and SVMs learning algorithm. Section 3, presents our approach to the face recognition problem and describes how the experiments were performed. Section 4 presents the results obtained and the discussion. Finally, Section 5 is the conclusion.

2 Background

2.1 Discrete Cosine Transform

The discrete cosine transform (DCT) is a technique for converting a signal into elementary frequency components, much like the Fourier Transform presented in [10], it is used in the JPG (Join Photographic Experts Group) image compression.

Data compression is important both for biological facial recognition as for computer-based face recognition. According to [14], the human eye uses a compression of 100:1 for every signal received.

There are four kinds of DCT, named DCT-I, DCT-II, DCT-III and DCT-IV [11]. The DCT-II is the most common, usually referred to simply as DCT, and is the one used in this paper.

Let us consider an image as an array $u(n)$ of size N. To obtain an array $v(k)$ using the DCT method we have the following equation,

$$v(k) = \alpha(k) \sum_{n=0}^{N-1} u(n) \cos \frac{(2n+1)\pi k}{2N} \quad (1)$$
$$0 \leq k \leq N-1$$

where

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad (2)$$
$$\alpha(k) = \sqrt{\frac{2}{N}},$$
$$0 \leq k \leq N-1$$

Another way to understand it is to consider the sequence $u(n)$ as an array, and the DCT method as a transformation matrix applied to $u(n)$. In this case, the transformation matrix $C = \{c(k,n)\}$ is defined as,

$$c(k,n) = \frac{1}{\sqrt{N}}, \quad (3)$$
$$k = 0, 0 \leq n \leq N-1$$
$$c(k,n) = \sqrt{\frac{2}{N}} \cos \frac{(2n+1)\pi k}{2N},$$
$$1 \leq k \leq N-1, 0 \leq n \leq N-1$$

where k and n are indexes for row and column of the transformation matrix. Using equation 3 the DCT of $u(n)$ will be,

$$v = Cu \tag{4}$$

It is possible to obtain the original image $u(n)$ from $v(n)$ using the following equation,

$$u(n) = \sum_{k=0}^{N-1} \alpha(k)v(k)\cos\frac{(2n+1)\pi k}{2N}, \quad 0 \leq n \leq N-1 \tag{5}$$

Considering the DCT as a Transformation Matrix and the equation 3, it is possible to recover $u(n)$ from $v(n)$ through the equation,

$$u = C^{-1}v \tag{6}$$

The compression of data using DCT works by defining the size k of $v(k)$. The smaller the value of k, the greater is the data loss. By compressing an image using the DCT method, firstly, the high frequency characteristics of the image are lost, so after the decompression the image obtained is similar to the original image, with the loss of smaller details.

2.2 Support Vector Machines

Support Vector Machines (SVMs) were introduced by V. Vapnik and coworkers [2, 5] based on the structural risk minimization principle from statistical learning theory [19]. In their original formulation [2], SVMs were designed to estimate a linear function,

$$f(\mathbf{x}) = \text{sgn}\left(\langle \mathbf{w} \cdot \mathbf{x} \rangle + b\right) \tag{7}$$

of parameters $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, using only a training set drawn i.i.d. according to an unknown probability distribution $P(\mathbf{x}, y)$. This training set is a finite set of samples,

$$T = \{\mathbf{x}_1, y_1, \cdots, \mathbf{x}_n, y_n\} \tag{8}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.

The SVMs learning aims to find the hyperplane which gives the largest separating margin between the two classes. For a linearly separable training set, the margin ρ is defined as euclidean distance between the separating hyperplane and the closest training examples. The hyperplane is considered in its canonical form, meaning that its parameters $(\mathbf{w}, b)$ are normalized such that the training examples closest to the hyperplane satisfy $|f(\mathbf{x})| = 1$ and, consequently, the margin is given by $1/\|\mathbf{w}\|$ [7]. Thus, for the linearly separable case, the learning problem can be stated as follows: find $\mathbf{w}$ and b that maximize the margin while ensuring that all the training samples are correctly classified,

$$\begin{aligned} min_{(\mathbf{w},b)} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ s.t. \quad & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b\right) \geq 1, \ \ \forall\, i \in T \end{aligned} \tag{9}$$

For the non-linearly separable case, slack variables ε_i are introduced to allow for some classification errors (soft-margin hyperplane) [5]. If a training example is located inside the margins or the wrong side of the hyperplane, its corresponding ε_i is greater than 0. The $\sum_{i=1}^{n}\varepsilon_i$ corresponds to an upper bound of the number of training errors. Thus, the optimal hyperplane is obtained by solving the following constrained (primal) optimization problem,

$$\begin{aligned} min_{(\mathbf{w},b,\varepsilon_i)} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\varepsilon_i \\ s.t. \quad & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b\right) \geq 1 - \varepsilon_i, \forall\, i \in T \\ & \varepsilon_i \geq 0, \forall\, i \in T \end{aligned} \tag{10}$$

where the constant $C > 0$, controls the trade-off between the margin size and the misclassified examples. Instead of solving the primal problem directly, one considers the following dual formulation,

$$\begin{aligned} max_{(\alpha)} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ s.t. \quad & 0 \leq \alpha_i \leq C, \ \forall\, i \in T \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{11}$$

Solving this dual problem the Lagrange multipliers α_i are obtained whose sizes are limited by the *box constraints* ($\alpha_i \leq C$); the parameter b can be obtained from some training example (support vector) with non-zero corresponding α_i. This leads to the following decision function,

$$f(\mathbf{x}_j) = \text{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + b\right) \tag{12}$$

Notice that the SVM formulation presented so far is limited to linear decision surfaces in input space, which are definitely not appropriate for many classification tasks. The extension to more complex decision surfaces is conceptually quite simple and, is done

by mapping the data into a higher dimensional feature space F, where the problem becomes linear. More precisely, a non-linear SVM first maps the input vectors by $\Phi : \mathbf{x} \rightarrow \mathbf{\Phi}(\mathbf{x})$, and then estimates a separating hyperplane in F,

$$f(\mathbf{x}) = \text{sgn}\left(\langle \mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{w} \rangle + b)\right) \tag{13}$$

It can be observed, in (11) and (12), that the input vectors are only involved through their inner product $\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$. Thus, to map the data is not necessary to consider the non-linear function Φ in explicit form. The inner products can only be calculated in the feature space F. In this context, a *kernel* is defined as a way to directly compute this product [7]. A *kernel* is a function K, such that for all pair $\mathbf{x}, \mathbf{x}'$ in input space,

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{x}') \rangle \tag{14}$$

Therefore, a non-linear SVM is obtained by only replacing the inner product $\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ in equations (11) and (12) by the *kernel* function $K(\mathbf{x}_i, \mathbf{x}_j)$ that corresponds to that inner product in the feature space F. Some *Kernel* functions commonly used in SVM learning are linear and RBF functions given, respectively, by the following expressions [7],

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}' \tag{15}$$

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-(\mathbf{x} - \mathbf{x}')^2}{2r^2}\right) \tag{16}$$

3 Methodology

Our face recognition approach uses the DCT holistic method to transform the whole image into a 8×8 feature matrix. The size of this matrix was suggested by [11].

As mentioned in [9], when the DCT is used to transform an image into a 8×8 matrix, even with the loss of data, the most important characteristics used for face recognition, such as the hair silhouette, eyes, nose and mouth position are kept. This characteristics are all low frequency components, much more reliable for face recognition.

After the image compression with the DCT, the 8×8 matrix must be converted into an one dimensional array of 64 positions. Following sugestion of [9], we scan the DCT coefficient matrix in a zig-zag manner starting from the upper-left corner and subsequently convert it to a one-dimensional (1-D) vector, as illustrated in Figure 1. Once each image is converted into an array of 64 characteristics, the array should be normalized to be used as input to a SVM classifier.

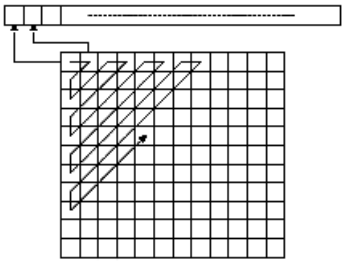

Figure 1: Scheme of scanning a 2 dimensional matrix into a one dimensional array

In order to compare our face recognition approach with other results already published in literature, experiments were conducted with two common benchmark face databases: the Yale and the AT&T. The following sections describe the characteristics of these databases and the methodology used to obtain the learning (training) and classification (test) subsets for each experiment.

3.1 Yale database

The Yale database contains 165 images in GIF format (Graphic Interchange Format). These images are all in gray scale, and have dimensions 320×243. They represent a total of 15 individuals in 11 different situations. An example of the 11 images of a individual is shown in Figure 2.

Figure 2: Images of the Yale database: centered light, with glasses, happy, light at left, no glasses, normal, light at right, sad, leepy, surprised and blinking an eye.

As there are 11 pictures of each individual in the database, 11 different experiments were conducted through the following methodology: in each experiment, the first image of each individual was separated to compound the test subset, and the other 10 images were

used as training examples. This training set (containing 150 examples) was then used to train a Multi-Class SVM. Once trained, the SVM classifier was evaluated using the test set containing 15 faces. The Yale database can be obtained at http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

3.2 AT&T database

The AT&T database contains 400 images of 40 individuals. The images are all in gray scale and have dimensions 92×112. An example of 10 images of a individual is shown in Figure 3.

Figure 3: Example of AT&T database images.

In these database, there are 10 pictures of each individual. Thus, 10 experiments were conducted with different training e test subsets obtained from the same methodology adopted with Yale database. The AT&T database, previously known as the ORL database can be obtained at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

3.3 The Multi-Class SVM Configuration

To achieve better results with SVM classifier, it was necessary to tune the following parameters of the SVM learning algorithm,

- Type of kernel function: linear or RBF;
- The regularization parameter C, which controls the trade-off between the margin size and the misclassified examples in training set (see expression 11);

Since the linear kernel function obtained good results in initial empirical tests, it was chosen for all experiments. The optimal range $(0.01 - 0.04)$ for parameter C was selected using a grid search procedure, as described in [17]. In this range, the best results for all experiments were obtained with $C = 0.01$. These results are presented in the following Section.

4 Results and Discussion

Tables 1 and 2 show the results achieved with the Yale and AT&T databases, respectively. In each experiment, the accuracy (percentage of individuals correctly classified) was calculated from a different subset of the test. Moreover, in the last row of each table is provided the global accuracy (average) and the standard deviation over all test subsets.

As can be observed in Table 1, our approach for face recognition (DCT + SVMs) achieved 72.8% of global accuracy over all test subsets of Yale database. However, it is important to notice that for the experiments 4 and 7, the results were not good. This can be explained due to the characteristics presented by the pictures that compound the 4 and 7 test subsets. These images were obtained when the light source was at the side of the individual, and not in front of it, as in the other pictures. Great changes in light source position can affect the efficiency of face recognition system, as explained in [20].

Table 1: Results obtained with Yale database

Experiment n°	Accuracy
1	80%
2	73.3%
3	93.3%
4	20%
5	93.3%
6	100%
7	0%
8	86.7%
9	86.7%
10	86.7%
11	80%
Global Acc.	72.8%
STD Acc.	32.2%

Although the global accuracy achieved with AT&T database has not been satisfactory, as can be seen in Table 2, the results obtained in each experiment were more stable (see standard deviation) than the Yale's ones. We believe that the discrepancy of global accuracies (between the databases) was due to the AT&T database images are not geographically normalized. We also speculate that, the use of algorithms to correct these geometric differences before application of DCT, as presented in [1], could improve the results obtained with this database.

5 Conclusions

One face recognition approach was presented and tested. Its main characteristics are fast and efficient

Table 2: Results obtained with AT&T database

Experiment n°	Accuracy
1	42.5%
2	35%
3	30%
4	35%
5	35%
6	35%
7	35%
8	27.5%
9	30%
10	35%
Global Acc.	34%
STD Acc.	4.1%

extraction of feature vectors using the DCT holistic method, and improved accuracy in the image classification stage due to the choice of a learning machine (SVMs) designed to ensure high generalization performance.

Although the experimental results obtained with both databases, Yale and AT&T, have not been entirely satisfactory, they point out that our approach is promising. Furthermore, they show that the incorporation of digital image processing techniques and algorithms, such as normalization of illumination direction and correction of geographic differences can improve the global accuracy and robustness of the proposed method.

Concerning SVMs learning algorithm, we also believe that a detailed investigation with other types of kernel functions such as polynomial, sigmoid and RBF could help to improve the results.

References

[1] Abate, A. F., Nappi, M., Riccio, D., and Sabatino, G. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28:1885–1906, 2007.

[2] Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, New York, NY, USA, 1992. ACM.

[3] Brunelli, R. and Poggio, T. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[4] Chellappa, R., Wilson, C. L., and Sirohey, S. Human and machine recognition of faces: a survey. *Proceedings of the IEEE*, 83(5):705–741, 1995.

[5] Cortes, C. and Vapnik, V. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

[6] Craw, I., Tock, D., and Bennett, A. Finding face features. In *European Conference on Computer Vision*, pages 92–96, 1992.

[7] Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.

[8] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, 2000.

[9] Er, M. J., Chen, W., and Wu, S. High-speed face recognition based on discrete cosine transform and rbf neural networks. *IEEE Trans Neural Netw*, 16(3):679–91, 2005.

[10] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[11] Hafed, Z. M. and Levine, M. D. Face recognition using discrete cosine transform. *International Journal of Computer Vision*, 43:167–188, 2001.

[12] Kirby, M. and Sirovich, L. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, 1990.

[13] Pan, Z., Rust, A., and Bolouri, H. Image redundancy reduction for neural network classification using discrete cosine transforms. In *in Proceedings of the International Joint Conference on Neural Networks*, pages 149–154, 2000.

[14] Sekuler, R. and Bake, R. *Perception*. McGraw-Hill Humanities, 2005.

[15] Shen, L., Bai, L., and Ji, Z. A svm face recognition method based on optimized gabor features. In *VISUAL'07: Proceedings of the 9th international conference on Advances in visual information systems*, pages 165–174. Springer-Verlag, 2007.

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

[17] Van Gestel, T., Suykens, J. A. K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., and Vandewalle, J. Benchmarking least squares support vector machine classifiers. *Mach. Learn.*, 54(1):5–32, 2004.

[18] Vapnik, V. N. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[19] Vapnik, V. N. An overview of statistical learning theory. *IEEE Trans. on Neural Networks*, 10(5):988–999, 1999.

[20] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. Face recognition: A literature survey. *ACM Computing Surveys*, 35:399–458, 2003.