Subspace Based Object Recognition Using Support Vector Machines

O. G. Sezer, A. Ercil and M. Keskinoz

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
ogsezer@su.sabanciuniv.edu, aytulercil@sabanciuniv.edu, keskinoz@sabanciuniv.edu

ABSTRACT

In this paper, we propose an object recognition technique using higher order statistics without the combinatorial explosion of time and memory complexity. The proposed technique is a fusion of two popular algorithms in the literature, Independent Component Analysis (ICA) and Support Vector Machines (SVM). We propose to use ICA to reduce the redundancy in the images and obtain some feature vectors for every image which has lower dimensions and then make use of SVM to classify these feature vectors coming from the ICA step. Experimental results are shown for Coil-20 and an internally created database of 2D manufacturing objects. Comparative analysis of independent component analysis and principal component analysis (PCA) is also given for each experiment.

1- INTRODUCTION

Object recognition, which is an easy task for a human observer, has long been the focus of much research in Computer Vision, forming an essential component of many machine-based object recognition systems. A large number of different approaches have been proposed in the literature. An extensive survey of shape matching in computer vision can be found in [1, 2, 3]. There are two main approaches: 1) feature based, which involve the use of spatial arrangements of extracted features such as edge elements or junctions, 2) brightness-based, which make more direct use of pixel brightness.

Brightness-based (or appearance-based) methods make direct use of the gray values within the visible portion of the object, instead of focusing on the shape of the occluding contour or other extracted features. Subspace methods such as ICA and PCA have been applied to face recognition [4] and robot vision systems successfully. Also in the literature there are many works on comparison between ICA and PCA, because ICA uses higher order statistics it is expected that it outperforms PCA which uses second order statistics. One of these works is Sahambi et al.'s [5], which compares performance of ICA and PCA on object recognition tasks. They have applied independent and principal component analysis to the Coil-20 database by training with different sampling degrees (25^0 and 50^0) and have used Euclidean distance to find the difference between test object coefficients of the independent and principal components and the means of coefficients of objects that are used in the training stage. Recognition rates for this method are found around 70-80% for the Coil-20 database.

Another approach for object recognition was proposed by Pontil et al. [6] using Support Vector Machines (SVM). Since SVM is known for its strength in classification of high dimensional data, Pontil et. al. did not use any feature extraction technique to reduce the dimensionality. They scaled the originally 128x128 images into 32x32 images for increasing the speed of calculations and performed recognition on images regarded as points on a high dimensional space without estimating pose. However, since algorithm takes 32x32 images as data points it has high computational load.

Our proposal is to combine the advantages of ICA for modelling higher order dependencies in an image including nonlinear relationships among the pixel intensity values and SVM for constructing an optimal separating hyperplane. We also include object recognition results of PCA for enabling performance comparison with ICA.

Organization of the paper is as follows: Background on principal and independent component analysis are given in sections 2 and 3 respectively, the idea behind SVM is explained briefly in section 4, the approach used in the paper is outlined in section 5 and the experimental results are given in section 6 with discussion and finally conclusions are presented in section 7.

2- PRINCIPAL COMPONENT ANALYSIS

PCA technique, also known as Karhunen-Loeve transform, chooses a dimensionality reducing linear projection that maximizes the scatter of all projected samples. If the total scatter is defined by S_T;

$$S_T = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T \tag{1}$$

where $x_k \in R^n$ are sample images, N is the total number of sample images and μ is the mean image of all sample images then after applying a linear transform we will obtain transformed features $y_k \in R^m$ in the reduced dimensional subspace;

$$y_k = W^T x_k \tag{2}$$

where $W^T \in R^{nxm}$ is a matrix with orthonormal columns. In PCA , the projection matrix W_{opt} will be chosen to maximize the determinant of the total scatter of the transformed features.

$$W_{opt} = \arg\max_W \left| W^T S_T W \right| \tag{3}$$

where $W^T S_T W$ is the scatter matrix of transformed features. [7]

3- INDEPENDENT COMPONENT ANALYSIS

ICA is a method that can perform blind source separation. Since both the source signals and how these signals are mixed are unknown, the separation is named as blind. ICA algorithm will find such a linear coordinate system that resulting signals will be statistically independent. ICA not only makes signals uncorrelated like PCA does but also reduces higher order dependencies between the signals.

Compared with the classical methods, ICA is a powerful method for finding the factors that are mutually independent with the non-gaussian distributions. In ICA model, linear or nonlinear mixtures of the hidden factors or independent components constitute the observed data. Basic linear mixture model of ICA can be expressed mathematically as [8]:

$$\boldsymbol{x}=\boldsymbol{A}\boldsymbol{s} \tag{4}$$

where $\boldsymbol{x}$ is the observation vector containing the observed data $\boldsymbol{x_j}$*'s*, $\boldsymbol{s}$ is source vector and $\boldsymbol{A}$ is the mixing matrix. The aim is to estimate unknown $\boldsymbol{A}$ and $\boldsymbol{s}$ from observation vector $\mathbf{x}$. Our only assumption is non-gaussianity and statistically independence of the sources [8].

Mixing matrix will be estimated by using assumptions in the model. Sources will be calculated with the following formulation.

$$\boldsymbol{s}=\boldsymbol{W}\boldsymbol{x} \tag{5}$$

where $\boldsymbol{W}$ is pseudo-inverse of the mixing matrix, $\boldsymbol{A}$. In our application these sources represent the coefficients of independent components.

4- SUPPORT VECTOR MACHINES

Support vector machines is a generalization method that is increasingly becoming popular in pattern recognition. SVM tries to find an optimum hyperplane that separates a given set of points belonging to two-class data.

In this section, first the simple case of linearly separable data is explained then support vectors concept and the general case for non-separable data is given in details.

4.1 Linearly Separable Data

In linearly separable data case we are given a set S of $\boldsymbol{x_i}$ element of R^n , i=1,…,N each belonging to one of the classes represented by y_i={-1,1}. The goal is to separate the set of data according to given labels by an hyperplane, hence leaving all data points belonging to one class in the same side of the hyperplane.

A set of data $\boldsymbol{x_i}$'s are linearly separable if there exists a w that satisfies

$$y_i(\boldsymbol{w}.\boldsymbol{x_i}+\boldsymbol{b}) \geq 1 \tag{6}$$

for i=1,…N, where ($\boldsymbol{w}$,$\boldsymbol{b}$) defines a hyperplane with the following equation;

$$\boldsymbol{w}.\boldsymbol{x}+\boldsymbol{b}=0 \tag{7}$$

and it is named as the separating hyperplane and the product in equation (6) defines that the data point and its label are in the same side of the hyperplane. There are, of course, infinitely many possible hyperplanes that can separate the two classes. Let's now define a distance measure from the hyperplane to a data point $\boldsymbol{x_i}$ as $\boldsymbol{d_i}$:

$$d_i=\frac{wx+b}{\|w\|} \tag{8}$$

If you combine equation (6) and (7) we will get the following equation,

$$y_i d_i \geq \frac{1}{\|\mathbf{w}\|} \tag{9}$$

Here we observe that $1/\|\boldsymbol{w}\|$ is a lower bound on the distance between ($\boldsymbol{w}$,$\boldsymbol{b}$) hyperplane and the data point $\boldsymbol{x_i}$. Hence if we can find an $\boldsymbol{x_i}$ in the set of data points that satisfies this lower bound with equality that means we have found the closest point to hyperplane. Also note that in order to obtain the optimum hyperplane, we should maximize the distance between hyperplane and the closest data point and this margin in fact corresponds to $1/\|\mathbf{w}\|$ value. Moreover, maximizing the margin corresponds to minimizing the norm of $\mathbf{w}$ which is $\|\mathbf{w}\|$.

After obtaining $\boldsymbol{w}$ and $\boldsymbol{b}$ by maximizing the margin, the problem of classification reduces to looking at the sign of

$$\boldsymbol{w}.\boldsymbol{x}+\boldsymbol{b} \tag{10}$$

where $\boldsymbol{x}$ is the incoming data. Hence, our decision function will be,

$$\begin{aligned} f(\boldsymbol{x}) &= \text{sign}(\boldsymbol{w}.\boldsymbol{x}+\boldsymbol{b}) \\ &= \text{sign}(<\boldsymbol{w},\boldsymbol{x}>+\boldsymbol{b}) \end{aligned} \tag{11}$$

4.2 Linearly Non-separable data

It is well known that in real life we generally do not have linearly separable data due to noise and nonlinearity of the data classes. So in order to apply SVM for this kind of data, we relieve the constraints that are presented in linearly separable case by introducing a slack variable ξ_i.

Our problem now is to find a hyperplane that separates two-class data by leaving maximum possible fraction of data belonging to same class in the same side of hyperlane. Solving the appropriate equation gives us that optimum separating hyperplane. [6, 9].

Using duality as before, we can represent $\boldsymbol{w}$ as linear combinations of $y_i\boldsymbol{x_i}$' s hence the decision function turns into:

$$\begin{aligned} f(\boldsymbol{x}) &= \text{sign}(<\boldsymbol{w},\boldsymbol{x}>+\boldsymbol{b}) \\ &= \text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i <\boldsymbol{x_i},\boldsymbol{x}>+\boldsymbol{b}\right) \end{aligned} \tag{12}$$

Observe that we have a dot product in our decision function which can be replaced by an appropriate nonlinear kernel to move the data into a higher dimensional space.[6, 9].

So we are replacing $<\boldsymbol{x_i},\boldsymbol{x}>$ with $<Q(\boldsymbol{x_i}),Q(\boldsymbol{x})>$, where Q(.) is the kernel function. There are many kernels in literature for different applications but in this paper, performances of three basic kernels were compared:

- Linear: $K(x_i,x_j)=x_i^T x_j$.
- Polynomial: $K(x_i,x_j)=(\gamma x_i^T x_j+r)^d$, $\gamma>0$.
- Radial basis function (RBF) : $K(x_i,x_j)=\exp(-\gamma\|xi-xj\|^2)$, $\gamma>0$

Finally, the decision function becomes,

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b) \qquad (12)$$

5- METHODOLOGY

Object recognition method used in the paper can be summarized as shown in Figure 1. In the offline stage, independent components are obtained and the SVM classifier is built. In the online stage, information produced in the offline stage is used to extract features from the object image and then with SVM classifier parameters, multi-class decision is performed to predict the identity of the object.

In the offline stage, ICA algorithm is trained using images of every object in the training set, In the first stage of ICA, dimensionality is reduced using principal components analysis such that at least 90% of the total energy in the images are preserved. Thus, reducing the dimensions enabled ICA algorithm to work faster.

The SVM part in the offline stage is aiming to obtain the generalizations for test data from the training data. Here training data are the coefficients of independent components (or basis vectors) obtained by multiplying the de-mixing matrix (*W*) coming from the ICA stage with the object images from the database. Finally SVM classifier is trained with object images in the training set with optimum SVM parameters.

In the online stage, first the image of the incoming object is fed to the system, then the ICA feature vectors (coefficients of independent components (ICs)) of the given image are extracted and finally the SVM classifier decides on the identity of the object.

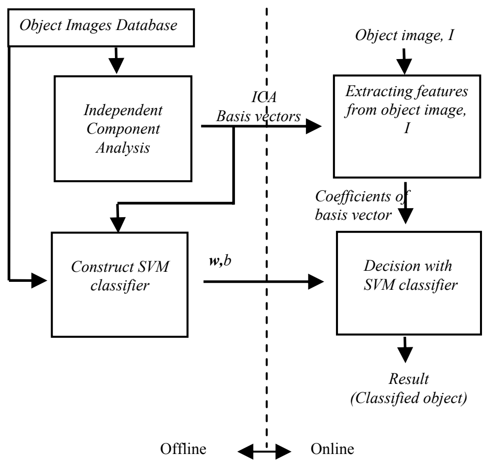

Figure 1 Block diagram of object recognition system in this paper

6- EXPERIMENTAL RESULTS

We have used two different databases to test the proposed method. First one is COIL-20 database which is widely used in 3D object recognition researches [10] (refer to Figure 2). This database consists of images of 20 different objects; each one is rotated with 5 degree angle interval in vertical axis. Hence for every object there are 72 images which sum up to 1440 images for the whole database.

Figure 2. Front-view of objects in Coil-20 database

The other database that we have used to test the proposed method is an in-house database of 2D objects created as a part of an industrial project with Festo, Germany (refer to Figure 3). Different from COIL-20 this database contains images of 15 industrial objects rotated in inwards axis with 10 degrees angle. Thus, for each object we have 36 images and a total of 540 images. In this paper two different experimental set-up of FESTO database is used. First experiment (Exp. #1) has a white background as shown in Figure 3 and second experiment (Exp. #2) has a black background. In this paper we have used LibSVM software for building SVM classifiers [11].

Figure 3. Objects in FESTO database

6.1 Results for COIL-20 Database

For the sake of completeness in comparisons, we have provided performance of different methods in object recognition for COIL-20 database. We have divided the database into two set as training and test sets for performance analysis. Two different training sets are formed for two different sampling angles (10 and 30 degrees) of the object images. In the PCA method we have used the highest 20 eigenvalues and corresponding eigenvectors. Thus 90% of the total energy in the images is preserved. We have used *tanh(x)* nonlinearity with fixed-point arithmetic for ICA. In the Figure 4, independent component filters obtained by ICA are given.

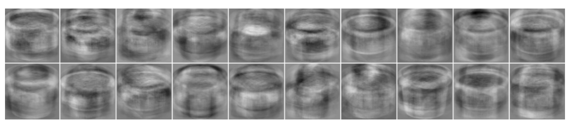

Figure 4. Independent component filters of COIL-20 database

Table 1 gives the classification results for two different sampling intervals. Here, a nearest mean classifier is used for classification of the features obtained by ICA and PCA. We observed that ICA has better object recognition performance than PCA in general.

Table 1. Object Recognition Performance of ICA and PCA methods (%)

COIL-20	Sampling Interval	
	10^0	30^0
PCA	74.03	75.17
ICA	88.33	88.00

On the other hand, using different kernels in SVM for classification of the PCA and ICA features of the object images, we have achieved perfect recognition performance by polynomial and Rbf kernels (refer to Table 2). Performance values given in Table 2 show the maximum values acquired by grid-search of the SVM variables (cost, degree of the polynomial or gamma value in rbf).

Table 2. Object Recognition Performance of ICA and PCA methods fused with SVM (%)

COIL	Linear		Polynomial		Rbf	
	10^0	30^0	10^0	30^0	10^0	30^0
PCA+ SVM	99.4	95.5	100	97.2	100	97.4
ICA + SVM	99.9	97.3	100	97.5	100	97.5

6.2 Results for FESTO Database

Similar experimental procedures that we have performed to COIL-20 database are repeated for Exp. #1 and Exp. #2 datasets of FESTO database. Different then COIL-20 in this case, since objects in FESTO database are rotated with 10 degrees angle, we have just used 30 degrees as our sampling angle of the object images in the training phase.

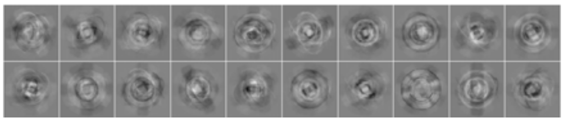

Figure 5. Independent component filters of FESTO database

Independent component filters shown in Figure 5 are obtained when ICA is applied to FESTO database.

Table 3. Object Recognition Performance of ICA and PCA methods (%)

FESTO	Exp. #1	Exp. #2
PCA	57.22	47.50
ICA	98.33	100.00

Table 3 demonstrates that object recognition performance of ICA method is superior to PCA in both Exp. #1 and Exp. #2 dataset of FESTO. We achieved perfect recognition result for Exp. #2 by ICA even without fusion of SVM. Again classification is done by a nearest mean classifier in this case too.

Table 4. Object Recognition Performance of ICA and PCA methods fused with SVM (%)

FESTO	Linear		Polynomial		Rbf	
Exp.	#1	#2	#1	#2	#1	#2
PCA+ SVM	83.9	75	93.3	91.9	93.1	93.1
ICA + SVM	96.4	95.6	97.2	97.2	98.1	100

When fused with SVM, we observe that classification performance improves significantly for PCA. On the other hand, fusion of ICA and SVM reduces the recognition rate for all kernels except Rbf (refer to Table 4).

7- CONCLUSION

In this paper the effect of the fusion of SVM with ICA and PCA methods is investigated for object recognition task. We observed that classification performances of ICA and PCA are generally improved by the fusion of SVM. Moreover, using feature extraction methods to reduce the dimensionality of the object images before SVM enabled faster computations in offline stage during building the classifiers. It is also important to note the difference between classification performance of ICA and PCA for FESTO database. Here, we can say that those features of the object images that we obtained by ICA are more invariant to rotations in inward axis described in FESTO database than PCA features. This means ICA features of the images of a single object cluster better around their mean value and the classification results of ICA features with Rbf kernel support this conclusion.

8- REFERENCES

[1] M. Hagedoorn, Pattern Matching using Similarity Measures, PhD Thesis, University of Utrecht, 2000

[2] S. Loncaric, A Survey of Shape Analysis Techniques, Pattern Recognition, Vol. 31, No. 8, pp. 983-1001, 1998

[3] R.C. Veltkamp, M. Hagedoorn, State of the Art in Shape Matching, Technical Report, UU-CS-1999-27, Utrecht, 1999.

[4] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski, Independent component representations for face recognition, Proc. SPIE Symp. Electron. Imaging: Science Technology; Conf. Human Vision Electronic Imaging III, CA, 1998, pp. 528–539.

[5] H. S. Sahambi, K. Khorasani, A Neural-Network Appearance-Based 3-D Object Recognition Using Independent Component Analysis, IEEE Trans. on Neural Networks, vol. 14, no. 1, January 2003, pp. 138-149.

[6] M. Pontil, A. Verri, Support Vector Machines for 3D Object Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 6, June 1998, pp. 637-646.

[7] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. CognitiveNeuroscience*, vol. 3, no. 1, 1991

[8] A. Hyvarinen, Survey on independent component analysis, Neural Computing Surveys, 2,(99) 94-128.

[9] N. Cristianini, J.Shawe-Taylor, An Introduction to Support Vector Machines ans Other Kernel-based Learning Methods, Cambridge Univ. Press 2000.

[10] S. A. Nene, S. K. Nayar and H. Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96, February 1996.

[11] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical Guide to Support Vector Classification, Department of Computer Science and Information Engineering National Taiwan University.