Shift Invariant Support Vector Machines Face Recognition System

J. Ruiz-Pinales, J. J. Acosta-Reyes, A. Salazar-Garibay, and R. Jaime-Rivas

Abstract**—**In this paper, we present a new method for incorporating global shift invariance in support vector machines. Unlike other approaches which incorporate a feature extraction stage, we first scale the image and then classify it by using the modified support vector machines classifier. Shift invariance is achieved by replacing dot products between patterns used by the SVM classifier with the maximum cross-correlation value between them. Unlike the normal approach, in which the patterns are treated as vectors, in our approach the patterns are treated as matrices (or images). Cross-correlation is computed by using computationally efficient techniques such as the fast Fourier transform. The method has been tested on the ORL face database. The tests indicate that this method can improve the recognition rate of an SVM classifier.

Keywords**—**Face recognition, support vector machines, shift invariance, image registration.

I. Introduction

FACE recognition is a difficult task with a wide range of applications such as identity authentication, access control, and surveillance. Over the last few years, research in this area has increased notably. This is due in part to the availability of faster computational resources and the development of new powerful pattern recognition techniques. However, even though current face recognition systems have reached a high degree of sophistication, their success is still limited to applications of controlled conditions. This contrasts with the human visual system which is able to recognize faces under unconstrained conditions.

In general, face recognition systems can be classified as: analytic and holistic [1]. In analytic systems, the facial features such as eyes, nose, mouth and chin are detected and a set of geometrical features such as areas, distances and angles are computed from them. These geometrical features are used to search for a candidate from a face database. These systems are very robust to translation changes but their performance depends to a great extend on the accuracy of facial feature detection. In holistic systems, the face is recognized as a whole without detecting the different facial features. However, these systems depend on techniques that transform the image to a low dimensional space with better discriminatory power. For instance, the eigenface approach [2] is based on the Karhunen-Loève transform (KLT) for the representation of faces. In this case, the image is projected into a point in the eigenface space and a distance metric is used to recognize the input face. Analytic and holistic systems may be combined in order to exploit their advantages [1]. For instance, in the first step an analytic method can be used to locate a set of feature points on a face. Then, rotation of the face can be estimated by using geometrical measurements and a head model. The positions of the feature points can then be adjusted so that their corresponding positions in the frontal view are approximated. Next, these feature points are compared with those of the faces in a database in order to leave only similar faces for the next step. In the second step, an approach based on correlation with templates of the eyes, nose, and mouth is used for recognition. This hybrid approach achieves a good recognition rate under different perspective variations [3]. A far more complete description of the different systems can be found in [4].

Support Vector Machines (SVM) [5] is one of the most powerful techniques proposed for classification and regression. This technique finds the optimal separating hyper-plane which minimizes the risk of misclassification. SVMs have been successfully applied to face recognition [6]. For instance, the image can be first transformed to another space by using PCA and the resulting vector is used for recognition by using a support vector machine classifier. This approach has given good results on a benchmark database [6]**.** Up to now, some of the best approaches incorporate knowledge about the expected variations of the patterns. For instance, new training samples (virtual examples) can be artificially generated by the transformation of some samples from the training set [7].

In this work, we have used SVMs for classification in a holistic face recognition system. However, instead of using self-organizing maps or principal component analysis as in previous approaches, we have chosen a very different approach consisting only on scaling and the application of a technique based on the alignment of two images for rendering the SVM classifier shift invariant.

II. Support Vector Machines

One of the goals of SVMs for pattern recognition is to find the optimal separating hyper-plane that minimizes the risk of misclassification [5]. Unlike other classifiers, SVMs control their generalization ability by minimizing their error rate on the training set and their capacity [8].

Manuscript received August 30, 2006.

The authors are with Faculty of Mechanical, Electrical, end Electronics Engineering, University of Guanajuato, Salamanca, Gto. Mexico (Corresponding author: J. Ruiz-Pinales, phone: +524646480911, email:pinales@salamanca.ugto.mx).

Given pattern $\mathbf{x} \in \Re^N$, support vector machines find hyper-planes of the form:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

The optimal separating hyper-plane is the hyper-plane farthest away from the two classes (with maximal margin and lowest capacity) and maximizes the risk of misclassification. More explicitly, the optimal hyper-plane is a one that minimizes $\|\mathbf{w}\|$ and whose margin is $2/\|\mathbf{w}\|$ (see Fig. 1).

The cost function minimized by SVMs is given by:

$$\min J = \tfrac{1}{2}\|\mathbf{w}\|^2$$

subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1,\ \forall i .$$

These constraints ensure that all the patterns of each class lie at a distance greater than or equal to $1/\|\mathbf{w}\|$ from the hyper-plane, see Fig. 1.

In order to minimize the cost function, the problem is transformed into another form by using Lagrange multipliers. So the new cost function to be minimized is:

$$\min L = \tfrac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

with constraints on the Lagrange multipliers $\alpha_i \geq 0$, $\forall i$. Conveniently, the problem can be transformed into its dual form which only contains parameters α_i and dot products between the patterns. The dual problem is thus written as:

$$\max_{\alpha_i} W = -\tfrac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{\ell} \alpha_i$$

with the constraints

$$0 \leq \alpha_i \leq C,\ \forall i$$

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0$$

where C is a constant which allows the penalization of misclassifications when the patterns are not linearly separable. For large values of C, the classifier seeks to separate perfectly the patterns. For the non separable case, a relatively small value of C allows the classifier to tolerate misclassifications.

For a given training set $\{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\}$, the parameter $\mathbf{w}$ of the optimal hyper-plane is given by:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

and b can be found by means of

$$y_i \mathbf{w} \cdot \mathbf{x}_i + b y_i = 1$$

for any $\mathbf{x}_i$ such that $\alpha_i > 0$.

The support vectors are those patterns $\mathbf{x}_i$ for which $\alpha_i > 0$. The number of support vectors is usually small compared with the number of training patterns. The complexity of a SVM classifier is given by the number of support vectors. The resulting decision function takes the form:

$$I(\mathbf{x}) = \text{sign}(\sum_{\mathbf{x}_i \in SV} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b)$$

where SV is a subset of the training vector samples $\mathbf{x}_i$ (also called support vectors), α_i are the coefficients and $y_i \in \{-1, 1\}$ are the class labels.

For the case of spaces that are not linearly separable, the basic idea of SVMs is to map the patterns to a high dimensional space (feature space), via a dot product and a nonlinear kernel function, where patterns may be linearly separable. Thus, in the general case, they find decision functions of the form:

$$I(\mathbf{x}) = \text{sign}(\sum_{\mathbf{x}_i \in SV} \alpha_i y_i K(\mathbf{x}_i \cdot \mathbf{x}) + b)$$

where $K(\cdot,\cdot)$ is the kernel function. The usual choices for the kernel are: the linear kernel, the polynomial kernel, the Gaussian kernel and the sigmoidal kernel. From these, the Gaussian kernel is the most popular. The expression for the linear kernel is given by:

$$K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x} \cdot \mathbf{x}_i$$

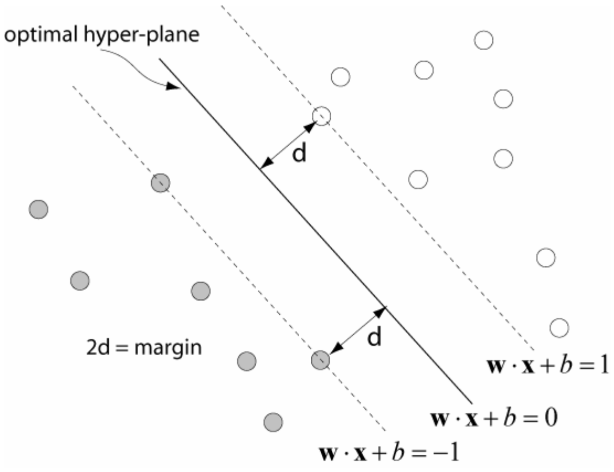
optimal hyper-plane
d
d
2d = margin
$\mathbf{w} \cdot \mathbf{x} + b = 1$
$\mathbf{w} \cdot \mathbf{x} + b = 0$
$\mathbf{w} \cdot \mathbf{x} + b = -1$

Fig. 1 Geometric interpretation of the optimal hyper-plane

whereas for the polynomial kernel is:

$$K(\mathbf{x}_i, \mathbf{x}) = (\gamma \mathbf{x} \cdot \mathbf{x}_i + r)^d,\ \gamma > 0$$

and for the Gaussian kernel is:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2),\ \gamma > 0 .$$

Some kernels may have extra parameters (e.g. γ in the Gaussian kernel) that need to be chosen in order to obtain the best possible classifier from the training data.

The Gaussian kernel can also be formulated in terms of dot products as:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x}\|^2 - \gamma \|\mathbf{x}_i\|^2 + 2\gamma \mathbf{x} \cdot \mathbf{x}_i)$$

For the case of unity vector patterns, the Gaussian kernel becomes:

$$K(\mathbf{x}_i,\mathbf{x}) = \exp(-2\gamma(1-\mathbf{x}\cdot\mathbf{x}_i)) .$$

Admissible kernel functions must satisfy the Mercer condition which is the condition for the convergence of SVMs to a global optimal solution. That is, there must exist a function $\varphi(\cdot)$ whose range is in an inner product space such that the kernel can be written as:

$$K(\mathbf{x}_i,\mathbf{x}) = \varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}) .$$

Thus, the resulting non-linear algorithm is equivalent to a linear algorithm operating in the range space of φ.

The SVM classifier is two-class classifier. Thus, for dealing with multi-class problems (with k classes), it is possible to construct $k(k-1)/2$ classifiers, each trained with data of two different classes (one-against-one strategy). Then the decision is made by using a voting strategy. An alternative is to use the one-against-the-rest strategy where one SVM is constructed for each class. The performance of these two methods is very similar. However, in terms of training time the one-against-one approach is the best [9].

III. New Method

A. Invariance in Support Vector Machines

Robust SVM classifiers require the incorporation of a priori knowledge about the expected variations of the patterns. There are at most three methods for incorporating invariances in SVMs [7]: modified kernel functions, artificially transformed examples from the training set and a combination of those two.

In the first case, invariances are incorporated by modifying the kernel functions. One method to achieve this is to transform the patterns by means of $B\mathbf{x}$ such that the SVM be invariant to local transformations of the patterns. One expression for B has been proposed in [10]. The basic idea is to minimize the magnitude of the tangent vector of the decision function $I(\mathbf{x})$ with respect to the parameter of the transformation at each pattern. For instance, the tangent vector around a given pattern $\mathbf{x}$ can be calculated as:

$$\frac{\partial}{\partial t} I(\mathcal{L}_t\mathbf{x})\bigg|_{t=0}$$

where t is the parameter of the transformation indicated by $\mathcal{L}_t$.

So the goal is to minimize

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\frac{\partial}{\partial t} I(\mathcal{L}_t\mathbf{x})\bigg|_{t=0}\right)^2$$

Since the magnitude of the tangent vector is cero for patterns which are not support vectors, B depends only on the support vectors. One expression for B is given by

$$B = C^{-2}$$

where

$$C = \frac{1}{\ell}\sum_{i=1}^{\ell}\left(\frac{\partial}{\partial t} I(\mathcal{L}_t\mathbf{x})\bigg|_{t=0}\right)\left(\frac{\partial}{\partial t} I(\mathcal{L}_t\mathbf{x})\bigg|_{t=0}\right)^T .$$

In the second case, invariance is introduced by generating transformed versions of the training patterns (virtual examples) with the hope that the SVM be able to learn the invariances. Training of the SVM on the artificially enlarged data set is considerably slower mainly due to the increase in the number of patterns. However, the advantage of this method is that it can be applied to any learning machine. An alternative is to first train an SVM on the original dataset in order to find the support vectors. Then, the support vectors are transformed in order to generate the virtual examples. Finally, another SVM is trained by using only the support vectors and their virtual examples.

An alternative to the virtual support vector (VSV) approach is to perform the transformations of the patterns inside the kernel function itself, the jittering support vector (JSV) approach. For example, any two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ are jittered around by transforming pattern $\mathbf{x}_i$ until a close match with the other pattern is found. The match between a transformed pattern $\mathbf{x}_q$ and the pattern $\mathbf{x}_j$ can be computed in the feature space by means of the Euclidean norm:

$$\sqrt{K(x_q,x_q)-2K(x_q,x_j)+K(x_j,x_j)} .$$

The VSV approach scales quadratically with the number of virtual examples whereas the JSV approach scales at least linearly with the number of jitters.

B. Image Registration

The estimation of the displacement vector of a shifted version of an image is a common operation in many applications of computer vision and image processing [11]. This operation is often referred as registration. Image registration is thus the task of finding the optimal spatial and intensity transformations so that two images are matched.

Cross-correlation is considered as the basic approach to image registration. It is usually used in pattern recognition for template matching because it can be regarded as a similarity measure of two images.

Let I_1 and I_2 be two images. The cross-correlation of these two images is given by

$$C(u,v) = \frac{\sum_x\sum_y I_1(x,y)I_2(x-u,y-v)}{\left[\sum_x\sum_y I_2^2(x-u,y-v)\right]^{\frac{1}{2}}}$$

If the images match perfectly, except for an intensity scale factor, cross-correlation will present a peak at some position. Sometimes it is preferable to compute the correlation coefficient which can be regarded as a linear indicator (in the range $[-1,1]$) of the degree of similarity.

$$\rho(u,v)=\frac{\sum_x\sum_y I_1(x,y)I_2(x-u,y-v)}{\left[\sum_x\sum_y (I_1(x,y)-\mu_1)^2\sum_x\sum_y (I_2(x-u,y-v)-\mu_2)^2\right]^{\frac{1}{2}}}$$

One of the reasons for the widespread use of correlation is that it can be computed efficiently (especially for large images) in the frequency domain by using the Fast Fourier Transform (FFT).

One of the Fourier methods to align two images is phase correlation. For the ideal case when the two images differ only by a displacement (d_x, d_y) as in

$$I_2(x,y)=I_1(x-d_x,y-d_y)\,.$$

Their Fourier transforms are related by

$$F_2(u,v)=F_1(u,v)e^{-j(ud_x+vd_y)}$$

and

$$\frac{F_1(u,v)F_2^*(u,v)}{\left|F_1(u,v)\right|\left|F_2(u,v)\right|}=e^{j(ud_x+vd)_y} \qquad (1)$$

which corresponds to an impulse at position (d_x, d_y) in the spatial domain. That is, the two images have the same magnitude spectrum but their phase spectrum varies as a function of the displacement. Image registration is therefore reduced to finding the peak of the cross-power spectrum phase (1).

The use of phase information for correlation is sometimes referred as whitening of the images. For instance, cross-correlation is robust to white noise in the images.

C. Modified Support Vector Machine

Being the maximum value of the cross-correlation matrix the dot product of the two images when they are aligned, it makes sense to compute in this way the dot product needed by SVMs. The intrinsic advantage of this method is that in this way we can consider at once all the shifted versions of the same images, avoiding the computational cost of augmenting the training set with shifted versions of each image. Furthermore, we can make use of computationally efficient techniques such as the fast Fourier transform.

Fig. 2 shows how we use the FFT to compute the maximum cross-correlation value of two images $\mathbf{x}_i$ and $\mathbf{x}_j$. First, we transform the two images to the Fourier domain. Then, we compute the product

$$\mathscr{F}(\mathbf{x}_i)\mathscr{F}(\mathbf{x}_j)^*\,.$$

The correlation matrix is obtained by using

$$M=\mathscr{F}^{-1}\left\{\mathscr{F}(\mathbf{x}_i)\mathscr{F}(\mathbf{x}_j)^*\right\}.$$

At the last step, we find the maximum value of M.

Using this approach, it is possible to model other types of transformations of the image such as scaling, and rotation though it might become computationally too expensive. For instance, invariance to rotation and shift can be obtained by using the following algorithm.

1. Transform the two images to the Fourier domain.
2. Compute the product $$\mathscr{F}(\mathbf{x}_i)\mathscr{F}(\mathbf{x}_j)^*\,.$$
3. Compute the correlation matrix by using $$M=\mathscr{F}^{-1}\left\{\mathscr{F}(\mathbf{x}_i)\mathscr{F}(\mathbf{x}_j)^*\right\}.$$
4. Find the maximum value of M.
5. Use $\mathscr{F}(\mathbf{x}_i)$ to interpolate for a rotation angle $\Delta\theta$ and use it to compute a new maximum correlation value.
6. Repeat several times and retain the highest cross-correlation value.

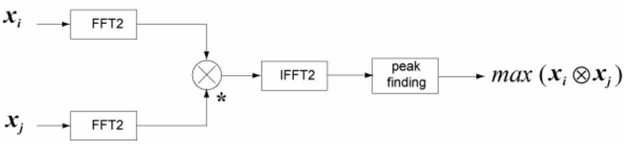

Fig. 2 Computation of the maximum cross-correlation value between two input patterns (images)

For the case of a Gaussian kernel the new expression becomes:

$$K(\mathbf{x}_i,\mathbf{x})=\exp(-\gamma\left\|\mathbf{x}\right\|^2-\gamma\left\|\mathbf{x}_i\right\|^2+2\gamma\max(\mathbf{x}\otimes\mathbf{x}_i))\,.$$

An alternative for reducing the computational cost of this approach is to train the SVM as usually and then to use the new dot product for the test patterns. One possible drawback is that the resulting classifier might have a higher number of support vectors. However, the excellent tolerance of SVMs to local variations makes viable another alternative, the computation of the correlation at a limited number of positions.

IV. Recognition Experiments

In order to evaluate our approach, we have performed some experiments on the publicly available Cambridge ORL database which contains 10 grayscale images of 40 people in an upright position (see Fig. 3). The different images were taken at different times with a uniform dark background. Variations present in the images include: smiling/non-smiling, glasses/no-glasses, open-eyes/closed-eyes, slight scale and illumination changes. All grayscale images are of size 92×112 pixels.

For all of our experiments, we first split at random the database into two subsets of 200 images (5 of each individual). Then, we use the first half for training and the second for testing. Next, we train in the second half and test in the first. This procedure is repeated several times and we compute the average error rate as the mean of the average error rates of all runs.

Other researchers have reported that smaller images can yield an additional improvement [12]. Thus, we first scale each image and then feed it to the input of the modified SVM classifier. In order that each value of the input pattern be in the interval $[0,1]$, we divide each image by its highest intensity value. The implementation of the SVM classifier was based on the LIBSVM library described in [13] and we used only Gaussian kernels. The computation of the cross-correlation value was based on the FFTW library described in [14].

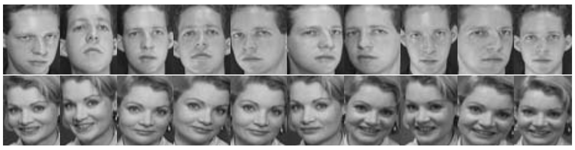

Fig. 3 Samples from the database for two of the classes

TABLE I
AVERAGE ERROR RATE AS A FUNCTION OF IMAGE SIZE

Size	12×14	23×28	31×37	46×56	92×112
%	2.58	**1.5**	2.13	2.29	2.75

V. CONCLUSION

We have successfully tested a method for achieving shift invariance in a holistic face recognition system using support vector machines. Table I shows the average error rate on the ORL database for different image sizes. The lowest average error rate was obtained for images of size 23×28. This result is clearly better than the average error rate of 3.25% achieved by the unmodified SVM classifier for the same resolution. As we had expected, the improvement in the recognition rate was small because of the small displacements present in the images. Thus, we expect that better improvements be possible on other databases in which the face may be at any location of the image. This might be advantageous because the errors made by face recognition systems are usually due to failures in face detection. Thus, in some sense the system also performs face detection. The modeling of other transformations such as scale changes and rotations may as well improve the result. The computational demand of the method may be reduced significantly by using image registration methods based on projections. Another alternative is to compute the correlation at a limited number of positions. Other methods achieve local invariance whereas our method achieves global shift invariance. However, our method can easily achieve local shift invariance by simply computing the cross-correlation matrix for small displacements. In general, it is possible to replace the dot product of an SVM classifier with a similarity measure. For instance, we can transform each pattern by using the probabilities given by several HMMs each trained with examples of one class [15]. Then, an SVM can be used for recognition in the new space.

REFERENCES

[1] K.M. Lam and H. Yan, “An Analytic-to-holistic Approach for Face Recognition Based on a Single Frontal View”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 673–686, 1998.

[2] M. Turk, A. Pentland, “Eigenfaces for Recognition”, *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[3] M.S. Kamel, H.C. Shen, A.K.C. Wong, T.M. Hong and R.I. Campeanu, “Face Recognition using Perspective Invariant Features”, *Pattern. Recognition Letters*, vol. 15, no. 9, pp. 877-883, 1994.

[4] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, “Face Recognition: A Literature Survey”, *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003.

[5] V.N. Vapnik, *The Nature of Statistical Learning Theory*. 2nd edn. Springer-Verlag, New York, 2000.

[6] G. Guo, S. Z. Li and C. Kapluk, “Face recognition by support vector machines”, *Image and Vision Computing*, vol. 19, pp. 631-638, 2001.

[7] D. DeCoste and B. Schölkopf, “Training Invariant Support Vector Machines”, *Machine Learning*, vol. 46, pp. 161-190, 2002.

[8] C. Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[9] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multi-class support vector machines”, *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[10] B. Schölkopf, C. Burges and V. Vapnik, “Incorporating Invariances in Support Vector Learning Machines”, In Lecture Notes in Computer Science, vol. 1112, pp. 47-52, 1996.

[11] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325-376, 1992.

[12] R. Fernandez and E. Viennet, “Face identification using support vector machines”, in *Proceedings of the European Symposium on Artificial Neural Networks* (ESANN99), 1999, pp.195-200.

[13] C. C. Chang and C. J. Lin, “LIBSVM: a library for support vector machines”, 2001. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] M. Frigo and S. G. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[15] T. Jaakkola, M. Diekhans and D. Haussler, “Using the Fisher Kernel Method to Detect Remote Protein Homologies”, In Proceedings of the Seventh international Conference on intelligent Systems For Molecular Biology, 1999, pp. 149-158.