FACE DETECTION IN CLUTTERED BACKGROUND USING SUPPORT VECTOR MACHINES

D.V.Karthik A.N.Rajagopalan

Department of Electrical Engineering, I.I.T-Madras
Chennai-600036
{ee99253,raju}@ee.iitm.ernet.in

ABSTRACT

Face detection in a cluttered background is a challenging problem in image processing. Current methods are plagued by the problem of a large number of false positives. Support Vector Machines (SVMs) are based on the solid foundation of statistical learning theory and have proven themselves by addressing a variety of pattern recognition problems. In conventional methods of face detection information about the background in an image is not used in identifying the faces. Doing so may help in reducing the number of false positives. We propose a new method of addressing the problem of face detection in a cluttered image by combining eigenfeature extraction, background learning and SVMs.

1. INTRODUCTION

This paper first gives a brief introduction to the face detection problem and methods employed in detecting faces in cluttered images. We then give a somewhat detailed but by no means exhaustive introduction to the method of support vector machines. This is followed by a description of our proposed method, Results and conclusions form the remaining part of the paper.

1.1. Face Detection

In the literature, several works have appeared that address the face detection and the face recognition problem separately [1, 2, 3,4]. There are also papers that combine recognition with detection such as the principal component analysis (PCA)-based approach of [4]. The PCA, which is one of the very successful and well-known face recognition methods, is based on the Karhunen-Loeve (KL) expansion [3]. Sirovich and Kirby [3] were the first to study the problem of KL representation of faces. They showed that if the eigenvectors corresponding to a set of training face images are obtained, any image in that database can be optimally reconstructed using a weighted combination of these eigenvectors. Turk and Pentland [4] later used these eigenvectors (or eigenfaces as they are called) for face recognition. Methods such as the eigen face recognition (EFR) technique work quite well provided the input test pattern is a face i.e., the face image has already been cropped and plucked out of a scene. The more general and difficult problem of recognizing faces in a cluttered background has also received some attention in [1, 4]. The authors in [1, 4] propose the use of distance from face space (DFFS) and distance in face space (DIFS) to detect and eliminate non-faces against arbitrary background patterns. In the absence of background information, the method is not robust and cannot discriminate against arbitrary background patterns. The traditional EFR technique either ends up missing faces or throws up many false alarms, depending on the threshold value. This is due to the inability of the method to pick out complex decision regions separating the faces and backgrounds.

In this paper, we use the method proposed in [4] to first segregate prominent background samples and then use the PCA feature vectors of these samples along with those of faces in an SVM framework for robust face detection.

1.2. Support Vector Machine

Support Vector Machine is a pattern recognition method arising out of statistical learning theory. It owes its foundation to the work of Vapnik in the seventies. There has been a great rise in interest and activity in SVM over the past few years, largely due its very impressive performance over other pattern classification methods [6,7].

The SVM algorithm works by finding the hyperplane that separates two given classes in a 'N' dimensional space. The restriction placed is to find the 'optimal' hyperplane which does so with the maximum margin.

.

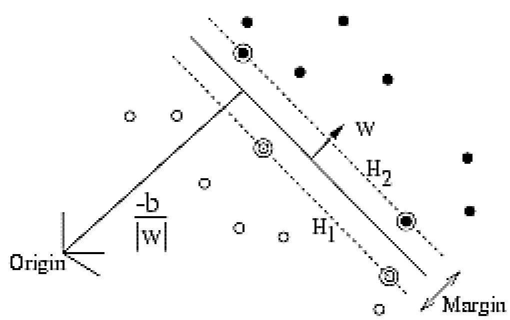

Figure 1. *Linear Separable Case. Support Vectors are circled.*

This problem can be solved as a standard quadratic programming problem to obtain a unique hyperplane in linearly separable cases such that

$$\mathbf{X_i.w} + b > +1 \text{ for } y_i = +1$$
$$\mathbf{X_i.w} + b < -1 \text{ for } y_i = -1$$

These can be combined as follows

$$y_i \,.(\mathbf{X}_i.\mathbf{w} + b) - 1 > 0 \;\forall\; i$$

Here X_i, y_i are the features and the class of the i^{th} data point respectively. **w** is the direction vector of the separating hyperplane and $b/\,|w|$ is its distance from the origin. The maximum margin is achieved by minimizing $\|w\|^2$.

The problem is best approached in a Lagrangian formulation. In this formulation the problem can be defined as

$$L_P \equiv \frac{1}{2}\|w\|^2 - \frac{1}{2}\sum_{i=1}^{t}\alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l}\alpha_i$$

We must now minimize L_p with respect to **w**; b, and simultaneously require that the derivatives of L_p with respect to all the α_i vanish, all subject to the constraints $\alpha_i>0$. This quadratic programming problem can be solved in a standard manner in its dual form.

It will be found that other than a few critical α_i the rest go to 0. The α_i which do not go to 0 correspond to those data points which lie closest to the separating hyper plane and in effect define it. These points are called the support vectors.

In the cases where the data set is not linearly separable the SVM algorithm raises the points in consideration into a higher dimensional space finds a maximum margin hyperplane in that higher dimension and maps this back to the lower dimension.

It can be expected that the raising of the dimensions to a higher number will greatly increase computational complexity. SVM overcomes this problem very intelligently by defining a kernel function

$$K(x_i, x_j) = \Phi(x_i)\cdot\Phi(x_j) \qquad (1)$$

where $\Phi(x_i)$ is a mapping from a lower to a higher dimensional space.

Using this definition, the kernel is used in place of the inner product in the Lagrangian. The net result is that we will be solving the problem (of mapping to a high dimensional space, and doing a linear separation in it, followed by a mapping back to lower dimensional space, all of which is computationally expensive) completely in a lower dimensional space, which is tractable. In the process, we learn a nonlinear decision surface in the lower dimensional space.

In almost all cases, the explicit mapping $\Phi(X_i)$ is never done. The mapping is done implicitly by defining the Kernel function for a given problem. The sufficient condition for a function to be a valid kernel function is that it must satisfy the Mercers criterion [5]. A $\Phi(X_i)$ can exist satisfying (1) for a given $K(X_i, X_j)$ iff, for any $g(x)$ such that

$$\int g(x)^2 dx < \infty \ ,$$

$$\int K(x,y)g(x)g(y)dxdy \geq 0 \qquad (2)$$

(2) is always satisfied.

The choice of the Kernel for a problem is one of the critical parameters in SVM learning and the best choice of Kernel function for a given problem is open to research.

2. THE PROPOSED SCHEME

The eigen face approach has the advantage of intrinsically deriving the important features in the images under consideration. A representation of a given image with these feature vectors greatly reduces the dimensionality of the image.

Background information, specific to an image, can be of great use especially in classifying correctly the frames likely to give false positives. This is the reason for using background learning in our approach.

We conjecture that a combination of "background" information and the use of this information in conjunction with "face" data to train a learning machine to detect faces in a sub-image will lead to improved performance in face detection in cluttered images. The output will be a 'true' for the case when the sub-image is a face and 'false' for it being a background.

SVMs have, over the past few years, achieved remarkable success in pattern recognition in a large variety of problems. [6,7]. By intrinsically following structural risk minimization they make excellent learning machines. They are capable of capturing various kinds of decision regions with ease while arriving at a global solution for the same.

2.1. Extraction of Background Images

PCA is used on the face data set to get eigenfaces. Given a test image with face embedded in a background each of the sub-images in the test image is projected onto the eigenface space using a running window. The reconstruction error (or DFFS) is a measure of it being a face or not. By choosing a high enough threshold the prominent background images can be extracted with confidence.

The number of such background images is usually very large and this is reduced by running the K-means clustering algorithm on the background images to get a representative set of background samples. These are nothing but the means of the clusters in the data. 'K' is usually in the range of hundreds.

2.2. Face Detection

We now have a set of "face" as well as "background" images. We reapply PCA but the new training set consists of both faces as well as the background points. It must be noted that the significant eigenvectors derived from PCA will have characteristics of both faces and background. A large number of either will bias the eigenvectors to the dominant features of that group. This may lead to decreased "separability" in the space spanned by the top eigenvectors resulting in inadequate training of the SVM.

Hence, clustering of the background is an important step. It may be noted that the number of training points alone may not affect the SVM as only the support vectors determine the decision surface.

The projection of the faces and the background on the significant eigenvectors yields the PCA weights for each of the images in the training set. This, along with the class to which they belong, is used to train a SVM. The SVM we used had a polynomial kernel

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$

with the exponent set to 2. This was empirically found to be the most promising kernel. Other functions tried out were linear kernel and polynomial with exponent of 3 and 4. The SVM on training with the PCA weights comes up with a decision surface to separate the background class from the face class.

The test image in question is now given a second run through and the projection of each of the sub-images on the eigenvectors is given to the trained SVM, which automatically assigns a class to each of them.

We summarize the steps in the algorithm below:

1. Apply PCA on the face data set to get eigenfaces.
2. Run through the image and identify prominent background sub-images based on reconstruction error, setting the error threshold very high.
3. Cluster the generated background sub-images to get a smaller representative set for the background.
4. Perform PCA again but this time taking both the background and face images into the training set so as to obtain a modified eigenspace.
5. Obtain weights of each of the background and face images with respect to the eigenvectors of the modified eigenspace.
6. Train the SVM using the above weights and class label.
7. Run through the test image again and use the trained SVM to classify each sub-image as a face or background.

3. RESULTS

A generic setting of 40 eigen faces, 200 clustered background images, 10 eigenvectors and a SVM of polynomial degree 2 as the kernel was used for all the images. The test was done on a data set of 6 images against a cluttered background. The method exactly and uniquely identified the face sub-window in 2 of the cases. It gave one false alarm in 2 cases and two false alarms in the others. However, faces were correctly detected in all the cases. Specific adjustment of parameters for each image results in improved performance for that image. Two images with correct detection are shown here. On the other hand, the pure PCA based approach to face detection [4] gave many false alarms in all the above cases.

4. CONCLUSIONS

Background learning of a given image combined with prior knowledge from a face data bank was used to train a SVM to detect faces in a cluttered scene and there was 100% recognition of face locations with a very small number of false alarms in all the cases tested, even with a generic setting for the parameters. Hence, the method does hold promise.

 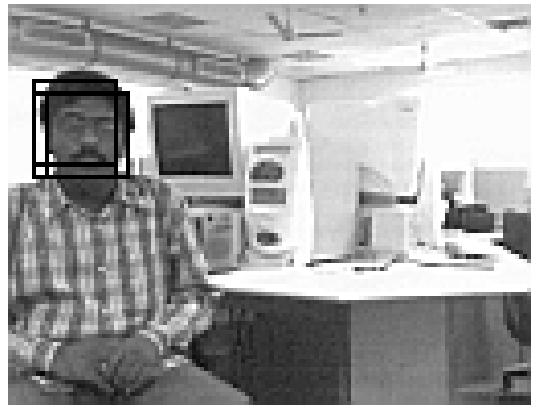

Figure 3, 4. *Examples of the algorithm at work.*

The findings are only preliminary and we are hopeful that with some more modifications the false alarms can be brought down to almost zero We propose to test the scheme on many more images.

5. REFERENCES

[1] B. Moghaddam and A. Pentland. “Probabilistic visual learning for object representation”. *IEEE Trans. Pattern Analysis and Machine Intell.*, 19:696–710, 1997.

[2] A. N. Rajagopalan, K. S. Kumar, J. Karlekar, R. Manivasakan, M. M. Patil, U. B. Desai, P. G. Poonacha, and S. Chaudhuri. “Locating human faces in a cluttered scene”. *Graphical Models in Image Processing*, 62:323–342, 2000.

[3] L.Sirovich and M. Kirby. “Low dimensional procedure for the characterization of human faces”. *J. Opt. Soc. Am. A*, 4:519–524, 1987.

[4] M. Turk and A. Pentland. ”Eigenfaces for recognition.”, *J. Cognitive Neurosciences*, 3:71–86, 1991.

[5] C.J.C.Burges "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol. 2,no. 2, pages 121-167,1998.

[6] S. Mukherjee, E. Osuna, and F. Girosi. “Nonlinear prediction of chaotic time series using a support vector machine”. *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 511-519, 1997.

[7] K.-R. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. "Predicting time series with support vector machines”, *Proceedings, International Conference on Artificial Neural Networks* Springer Lecture Notes in Computer Science, 1997.