Facial Feature Detection with Optimal Pixel Reduction SVM

Minh Hoai Nguyen **Joan Perez** **Fernando De la Torre**
Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.
minhhoai@cmu.edu joan.jpi@gmail.com ftorre@cs.cmu.edu

Abstract

Automatic facial feature localization has been a long-standing challenge in the field of computer vision for several decades. This can be explained by the large variation a face in an image can have due to factors such as position, facial expression, pose, illumination, and background clutter. Support Vector Machines (SVMs) have been a popular statistical tool for facial feature detection. Traditional SVM approaches to facial feature detection typically extract features from images (e.g. multiband filter, SIFT features) and learn the SVM parameters. Independently *learning features and SVM parameters might result in a loss of information related to the classification process. This paper proposes an energy-based framework to* jointly *perform relevant feature weighting and SVM parameter learning. Preliminary experiments on standard face databases have shown significant improvement in speed with our approach.*

1. Introduction

Detection of facial features (*e.g.* eyes, nose) is a necessary step in a wide range of applications (*e.g.* face recognition, face tracking). Most successful approaches to facial feature detection frame the task as a classification or regression problem [11, 14, 17, 18, 22, 31]. Traditional approaches for classification/regression follow a two step process: (i) extracting features, (ii) building classifiers/regressors. Performing these two steps *independently* might result in a loss of information relevant to the classification/regression task.

Due to its importance, feature selection has been a central topic in a variety of fields including signal processing, computer vision, statistics, neural networks, pattern recognition, and machine learning. Traditionally, feature selection is performed independently of learning the classifier parameters [2–6, 12, 15, 20, 23, 24, 28, 30]. This paper extends previous work on feature selection and image classification by *jointly* learning optimal weighting of features (i.e. pixels) and SVM parameters.

Figure 1 illustrates the main point of the paper. Figure 1a displays a 17×29 rectangular patch around an eye. Figure 1b plots the ROC curve of a linear SVM using all available pixels inside the patch as features. Figure 1c displays a sparse set of 64 pixels chosen by our algorithm. These pixels and their weights are learned jointly with the SVM parameters. Using only 64 pixels (13% of the features), our SVM classifier produces a ROC curve (Fig. 1d) that is almost identical to the one shown in Figure 1b (using all pixels). Although the classification performance is not significantly better, using only 13% of the features lead to a dramatic increase in speed. Notably, most selected pixels are located around the edges of the eye, which is consistent with our intuition.

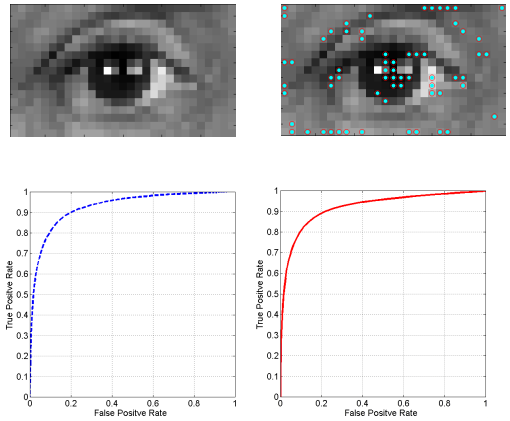

Figure 1. a) 17 × 29 rectangular patch used for eye detection. b)ROC curve of a linear SVM classifier using all pixels as features. c) 64 most discriminative pixels used by our SVM classifier that jointly optimizes pixel weighting and SVM parameters. d) ROC curve of the learned SVM classifier, using only 64 pixels.

The rest of the paper is organized as follows. Sec. 2 reviews previous work on SVMs and feature extraction. Sec. 3 derives a normalized error function to jointly learn a parameterized kernel and the SVM parameters. Methods for learning feature weights in the input space and kernel space are provided in Sec. 4 and 5 respectively. Sec. 6 describes experiments on two standard face databases.

978-1-4244-2154-1/08/$25.00

2. Previous work

This section reviews previous work on SVMs and feature construction for SVMs.

2.1. Support Vector Machines

Given a set of training data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Re^{d\times 1}$ (see notation [1]) with corresponding labels $y_1, \ldots, y_n \in \{-1, 1\}$, SVMs seek a separating hyperplane with maximum margin [26]:

$$\begin{aligned} \underset{\overline{\mathbf{w}},\overline{b},M}{\text{maximize}} \quad & M \\ \text{s.t.} \quad & y_i(\overline{\mathbf{w}}^T\varphi(\mathbf{x}_i) + \overline{b}) \geq M \ \ \forall i \\ & ||\overline{\mathbf{w}}||_2 = 1. \end{aligned} \tag{1}$$

Here, M is the margin, $\overline{\mathbf{w}}$ is the normal vector of the hyperplane, and $\varphi(\cdot)$ represents the mapping from the input space to the feature space.

Let $\mathbf{w} = \overline{\mathbf{w}}/M$, $b = \overline{b}/M$, then Eq. 1 is equivalent to:

$$\begin{aligned} \underset{\mathbf{w},b}{\text{maximize}} \quad & \frac{1}{||\mathbf{w}||_2} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \geq 1 \ \ \forall i. \end{aligned} \tag{2}$$

The above is equivalent to:

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \geq 1 \ \ \forall i. \end{aligned} \tag{3}$$

Using a soft-margin instead of a hard-margin, we obtain the primal problem for SVMs:

$$\begin{aligned} \underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i \ \ \forall i \\ & \xi_i \geq 0 \ \ \forall i. \end{aligned} \tag{4}$$

Here, $\{\xi_i\}_1^n$ are slack variables which allow for penalized constraint violations. C is the parameter controlling the trade-off between a large normalized margin and less constrained violations.

2.2. Feature construction in SVM

This section discusses previous work on selecting features for SVMs.

[1] Bold uppercase letters denote matrices (e.g. $\mathbf{X}$), bold lowercase letters denote column vectors (e.g. $\mathbf{x}$). $\mathbf{x}_i$ represents the i^{th} column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the row j^{th} and column i^{th} of the matrix $\mathbf{X}$. x_{ij} also denotes the scalar in the row j^{th} of column vector $\mathbf{x}_i$. Non-bold letters represent scalar variables. $||\mathbf{x}||_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$ designates Euclidean norm of $\mathbf{x}$. $diag(\cdot)$ is the operator that extracts the diagonal of a square matrix or constructs a diagonal matrix from a vector.

One popular technique for selecting features is RELIEF [19]. RELIEF assigns the weight to a particular feature based on the differences between the feature values of nearest neighbor pairs. Cao *et al* [5] further develop this method by learning feature weights in kernel space. This method is often done as a data processing step, independent of classifier construction. De la Torre and Vinyals [12] learn a subpace-parameterized Taylor series kernel expansion that effectively weights irrelevant pixels for classification with SVMs. Recently, there have also been several papers that learn kernel matrices for classification [10, 16, 21]. A popular approach is to define a parameterized family of kernel matrices and optimize the parameters to align with an ideal kernel. Another popular approach is to determine a desired property and learn a kernel which exhibits that property. In these approaches, the kernel is learned independently of the SVMs parameters. This is the key difference between our proposed method and previous work.

To solve the problem of jointly learning the SVM parameters and kernel, Chapelle *et al* [8] and Weston *et al* [30] propose a method for choosing SVM's parameters including the kernel parameters by minimizing the Leave-One-Out Cross Validation (LOOCV) error. However, since the LOOCV error cannot be expressed analytically, they instead propose to minimize some differentiable functions that are upper bounds of the LOOCV error. Mangasarian & Wild [23] introduce a modification to the objective function of the SVMs, and performs feature selection by repeatedly sweeping through all features to decide weather select or deselect a feature depending on which will decrease the value of the objective function.

One way to select a subset of good features is to prune away unnecessary ones. Hermes and Buhmann [15] start by constructing a SVM classifier using all available features and recursively remove the feature that has the least impact on the decision function if removed. Similarly, Avidan [3] uses a greedy sequential forward selection method to find a subset of features and support vectors that approximate the SVM solution obtained using all available features.

To further constraint the SVMs' parameters, some authors propose modifying the objective function of SVMs by including regularization terms or constraints on the parameter $\mathbf{w}$ of SVMs. For example, Chan *et al* [6] include two additional constraints on the L_1 and L_2 norms of $\mathbf{w}$ in the formulation of SVMs to achieve a sparse weight vector $\mathbf{w}$. Stoeckel & Fung [27] add a constraint on $\mathbf{w}$ to have the weight for each pixel depend not only on the pixel itself but also on its neighbors. Dundar *et al* [13] add a regularization term on $\mathbf{w}$ in the objective function to encourage the decision function to produce similar results for neighboring pixels.

3. SVMs and parameterized kernels

Suppose the mapping from the input space to the feature space can be parameterized by a parameter $\mathbf{p}$, *i.e.* $\varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i, \mathbf{p})$. We would like to find a parameter vector $\mathbf{p}$ and a separating hyperplane that have the largest margin. However, different values of $\mathbf{p}$ correspond to different feature spaces, and since the margins in two different feature spaces can not be directly compared, it is necessary to consider *normalized margins*. Let us consider the normalized margin as the ratio of the margin over the square root of sum of squared distances (in the feature space) between same-class data instances. In other words, the normalized margin is defined as:

$$\frac{M}{\sqrt{\sum_{i,j} \frac{1+y_i y_j}{2} ||\varphi(\mathbf{x}_i, \mathbf{p}) - \varphi(\mathbf{x}_j, \mathbf{p})||_2^2}} \tag{5}$$

Observe that normalized margin defined above is invariant to scale and translation in the feature space.

The problem of finding the parameter $\mathbf{p}$ for the mapping and the parameters of the separating hyperplane that provides the largest normalized margin can be stated as:

$$\begin{aligned} \underset{\overline{\mathbf{w}},\overline{b},M,\mathbf{p}}{\text{maximize}} \quad & \frac{M}{\sqrt{\sum_{i,j} \frac{1+y_i y_j}{2} ||\varphi(\mathbf{x}_i, \mathbf{p}) - \varphi(\mathbf{x}_j, \mathbf{p})||_2^2}} \\ \text{s.t.} \quad & y_i(\overline{\mathbf{w}}^T \varphi(\mathbf{x}_i, \mathbf{p}) + \overline{b}) \geq M \ \ \forall i \\ & ||\overline{\mathbf{w}}||_2 = 1. \end{aligned} \tag{6}$$

Recall that if $\mathbf{p}$ is fixed, finding the hyperplane with maximum normalized margin is equivalent to finding the hyperplane that maximizes the normal margin M.

Let $\mathbf{w} = \overline{\mathbf{w}}/M$, $b = \overline{b}/M$, and let $\phi(\mathbf{p})$ denote $\sum_{i,j} \frac{1+y_i y_j}{2} ||\varphi(\mathbf{x}_i, \mathbf{p}) - \varphi(\mathbf{x}_j, \mathbf{p})||_2^2$, Eq. 6 is equivalent to:

$$\begin{aligned} \underset{\mathbf{w},b,M,\mathbf{p}}{\text{maximize}} \quad & \frac{1}{\sqrt{\phi(\mathbf{p})}||\mathbf{w}||_2} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \varphi(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 \ \ \forall i. \end{aligned} \tag{7}$$

The above is equivalent to:

$$\begin{aligned} \underset{\mathbf{w},b,\mathbf{p}}{\text{minimize}} \quad & \frac{1}{2}\phi(\mathbf{p})||\mathbf{w}||_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \varphi(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 \ \ \forall i. \end{aligned} \tag{8}$$

Using soft-margin instead of hard-margin, we get:

$$\begin{aligned} \underset{\mathbf{w},b,\mathbf{p},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\phi(\mathbf{p})||\mathbf{w}||_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \varphi(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 - \xi_i \ \ \forall i \\ & \xi_i \geq 0 \ \ \forall i. \end{aligned} \tag{9}$$

Here, $\{\xi_i\}_1^n$ are slack variables which allow for penalized constraint violation. C is the parameter controlling the trade-off between having large normalized margin and having less constraint violation.

4. Learning feature weights

Consider a mapping that assigns different weights to different features $\varphi(\mathbf{x}_i, \mathbf{p}) = diag(\mathbf{p})^{1/2}\mathbf{x}_i$, where $\mathbf{p} = [p_1 ... p_d]^T$ are the feature weights, and $p_i \geq 0 \ \forall i$. We have:

$$\phi(\mathbf{p}) = \sum_{k=1}^{d} p_k \sum_{i,j} \frac{1+y_i y_j}{2}(x_{ik} - x_{jk})^2 \tag{10}$$

Since $\phi(\mathbf{p})$ is homogeneous in $\mathbf{p}$, we can always scale $\mathbf{w}$ and $\mathbf{p}$ appropriately to get $\phi(\mathbf{p}) = 1$. Therefore Eq. (9) is equivalent to:

$$\begin{aligned} \underset{\mathbf{w},b,\mathbf{p},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T diag(\mathbf{p})^{1/2}\mathbf{x}_i + b) \geq 1 - \xi_i \ \ \forall i \\ & \sum_{k=1}^{d} p_k \sum_{i,j} \frac{1+y_i y_j}{2}(x_{ik} - x_{jk})^2 = 1 \\ & \xi_i \geq 0 \ \ \forall i, p_k \geq 0 \ \ \forall k. \end{aligned} \tag{11}$$

Let $\mathbf{v} = diag(\mathbf{p})^{\frac{1}{2}}\mathbf{w}$ and consider the function $g : \Re \times \Re^+ \rightarrow \Re$ defined by:

$$g(x, y) = \begin{cases} \frac{x^2}{y} & \text{if } y > 0 \\ 0 & \text{if } y = 0, x = 0 \\ \infty & \text{if } y = 0, x \neq 0. \end{cases} \tag{12}$$

Eq. 11 is equivalent to:

$$\begin{aligned} \underset{\mathbf{v},b,\mathbf{p},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\sum_i g(v_i, p_i) + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{v}^T \mathbf{x}_i + b) \geq 1 - \xi_i \ \ \forall i \\ & \sum_{k=1}^{d} p_k \sum_{i,j} \frac{1+y_i y_j}{2}(x_{ik} - x_{jk})^2 = 1 \\ & \xi_i \geq 0 \ \ \forall i, p_k \geq 0 \ \ \forall k. \end{aligned} \tag{13}$$

Since $g(\cdot, \cdot)$ is convex, the above optimization problem is also convex.

5. Feature weighting in feature space

Let $\mathbf{X} \in \Re^{d \times n}$ be the training data set and $\mathbf{X}' \in \Re^{d \times m}$ be the testing data set. Let $\varphi(\mathbf{X})$ denote $[\varphi(\mathbf{x}_1) \ldots \varphi(\mathbf{x}_n)]$. The training kernel is $\mathbf{K}_{train} = \varphi(\mathbf{X})^T\varphi(\mathbf{X})$, and the testing kernel is $\mathbf{K}_{test} = \varphi(\mathbf{X}')^T\varphi(\mathbf{X})$. Suppose $\mathbf{K}_{train} = \mathbf{U}\mathbf{S}\mathbf{U}^T$ is non-singular. Let $\mathbf{B} = \mathbf{S}^{-\frac{1}{2}}\mathbf{U}^T$, then $\mathbf{B}^T\mathbf{B} = \mathbf{K}_{train}$. Consider the mapping $\widetilde{\varphi} : \Re^d \rightarrow \Re^n, \widetilde{\varphi}(\mathbf{x}) = \mathbf{B}\varphi(\mathbf{X})^T\varphi(\mathbf{x})$. Based on these conditions, the correspond-

ing train and test kernels are:

$$\widetilde{\mathbf{K}}_{train} = \widetilde{\varphi}(\mathbf{X})^T\widetilde{\varphi}(\mathbf{X}) = \varphi(\mathbf{X})^T\varphi(\mathbf{X})\mathbf{B}^T\mathbf{B}\varphi(\mathbf{X})^T\varphi(\mathbf{X}) = \mathbf{K}_{train}. \quad (14)$$

$$\widetilde{\mathbf{K}}_{test} = \widetilde{\varphi}(\mathbf{X}')^T\widetilde{\varphi}(\mathbf{X}) = \varphi(\mathbf{X}')^T\varphi(\mathbf{X})\mathbf{B}^T\mathbf{B}\varphi(\mathbf{X})^T\varphi(\mathbf{X}) = \mathbf{K}_{test}. \quad (15)$$

Thus we have defined a feature mapping $\widetilde{\varphi}$ that induces the same training and testing kernels. Now, we can learn the feature weights as if the training data was $\mathbf{B}\mathbf{K}_{train}$ and the testing data was $\mathbf{B}\mathbf{K}_{test}^T$.

If $\mathbf{K}_{train}$ is singular or if we want to reduce the number of dimensions of the feature space, we can take $\mathbf{B}$ as $\mathbf{B} = \mathbf{S}_k^{-\frac{1}{2}}\mathbf{U}_k^T$. Here $\mathbf{U}_k$ contains the first k columns of $\mathbf{U}$ (corresponding to the largest eigenvalues of $\mathbf{K}_{train}$) and $\mathbf{S}_k$ is the sub-matrix of $\mathbf{S}$ containing the first k columns and k rows. In this case, $\widetilde{\mathbf{K}}_{train}$ might not exactly match $\mathbf{K}_{train}$, but it is the best *rank-k* approximation.

6. Experiments

This section compares the performance of weighted SVMs and normal SVMs on two standard face databases.

6.1. Pose classification

We performed experiments on the CMU Face Images Data Set from the UCI machine learning repository [1]. The database contains 30×32 pixel facial images of 20 people under different expressions and poses. Some examples of faces from the database are given in Fig. 2. The classification task was to distinguish between two different poses: looking up and looking to the camera. Because the number of data instances in this database is small (only 312 faces), the experimental results were taken as the accuracy of 10-fold cross validation. We constructed four different SVM classifiers, namely linear SVM, linear weighted SVM, Gaussian SVM, and Gaussian weighted SVM. For all classifiers, we repeated the experiments for different values of the C parameter (and γ for Gaussian SVMs) and reported the best results. Table 1 shows the best results from all methods. Notably, weighted SVMs achieve similar classification accuracy while using a much smaller number of pixels and support vectors. Fig. 3 displays the pixels selected by applying our weighted SVM method.

6.2. Eye detection

Following the approach of Everingham and Zisserman [14], we performed eye detection experiments on the gray-scale FERET database [25]. This database contains facial images of various subjects under different expressions and poses. All images have a 256×384 pixel resolution and limited lighting variation. Some images are associated with

Figure 2. Examples of faces from the CMU Face Database

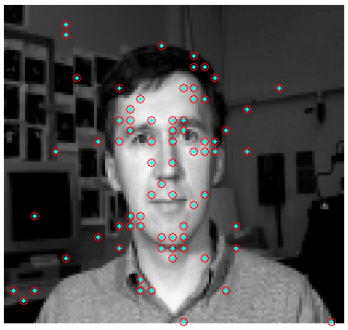

Figure 3. Pixels chosen by our weighted SVM. Most chosen pixels lie around the face region, which is the informative region about the pose. Several pixels outside the facial region are also chosen. This is due to noise and insufficient training data.

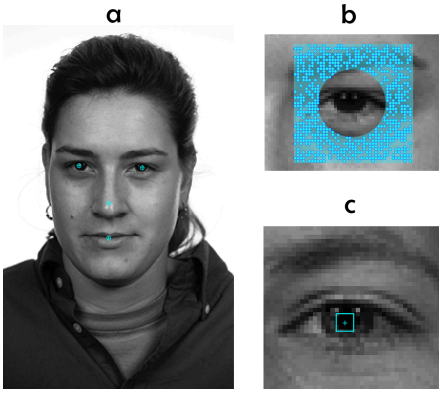

Figure 4. (a) Example of four landmarks used in the FERET database. (b) Centers of negative training patches were sampled randomly inside the cyan region. (c) Region of correct classification, positively classified pixels were considered correct if they are located inside the square.

a set of four hand labeled landmarks (Fig. 4a). Among the images with labeled landmarks, we extracted all the 2963 available frontal faces for experiments. These images were further divided into disjointed training and testing sets (60% and 40% respectively).

For training, we first performed Procrustes analysis [9] to align the landmarks w.r.t. the mean shape, removing rotation, translation, and scale variations. Positive training examples were obtained by sub-sampling 17×29 patches inside 27×47 rectangular regions around the left iris landmark of every training image. Similarly, negative exam-

Table 1. Comparison of weighted SVMs and normal SVMs on the UCI CMU Face Images Data Set. The weighted SVMs (both linear and Gaussian) achieve similar accuracy rates while using much fewer features and support vectors.

10-fold CV acc				#features used				#SVs			
Linear		Gaussian Kernel		Linear		Gaussian Kernel		Linear		Gaussian Kernel	
Normal	Weight	Normal	Weight	Normal	Weight	Normal	Weight	Normal	Weight	Normal	Weight
95.5	95.5	97.48	98.06	960	67	312	74	102	85	186	73

ples were created by extracting rectangular patches around random points in the iris neighborhood. The neighborhood was defined as in Fig. 4b. Each patch was normalized by subtracting the mean intensity and dividing by the standard deviation.

For each training image, the OpenCV Viola-Jones face detector [29] was used to produce a square centered on the face. A linear regression predictor was implemented to approximate the iris landmark from the position and scale of the face detector's output [14].

We performed experiments with two different SVM classifiers, namely normal SVM and weighted SVM. For weighted SVM, we first applied the method described in Sec. 4 to learn the optimal pixel weights. Pixels with insignificant weights ($< 10^{-5}$) were discarded, and a SVM classifier was constructed based on the remaining pixels, taking their weights into account. Fig. 1c shows the locations of 64 pixels (out of 493) chosen by our weighted SVM (cyan dots).

For each testing image, we used the previously learned linear regression to produce the first approximation for the iris' position. A searching window was placed around this initial guess. With a sliding window approach, the pixel with the highest SVM decision value was chosen as the final result for the localization of the iris.

The performance of different algorithms was evaluated in two different ways. Figure 5 plots the localization error threshold (x-axis) and the proportion of successful localizations within the threshold (y-axis). The Euclidean distance from the ground truth landmark to the predicted iris location was normalized by the inter-ocular distance (distance between the two iris landmarks) to account for different scales. Compared with normal SVM, weighted SVM achieves similar performance results while using a much smaller number of pixels.

To analyze the trade-off between true detections and false alarms, we classified all pixels inside the searching window and produced ROC curves (Fig. 6) by varying the threshold of the SVM classifier. The positively classified pixels were considered correct if they fell inside a square neighborhood around the true landmark. The size of this neighborhood was proportional to the inter-ocular distance of the subject (illustrated in Fig. 4c). As can be observed, the ROC curve produced by our weighted SVM is similar to the one produced by standard SVM. However, weighted

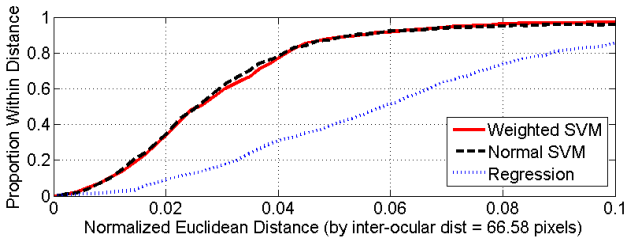

Figure 5. Distance threshold versus the proportion of iris localization within the threshold. The distance is taken as the Euclidean distance from the ground truth landmark to the predicted iris location normalizing by the inter-ocular distance. Weighted SVM performs as well as the other method while using much less pixels. The *Regression* curve is the result of using initial guess produced by the linear regression predictor.

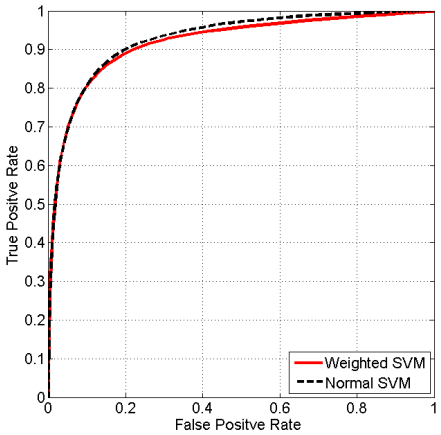

Figure 6. ROC curves of three different methods. Weighted SVM performs as well as normal SVM while using a much smaller number of pixels.

SVM used only 13% of available pixels.

In our experiments, SVM classifiers were built using LibSVM [7]. The C parameter of SVMs and other parameters were tuned using cross validation.

7. Conclusion

In this paper, we have presented a method for jointly performing feature extraction and building SVM classifiers. Learning feature weights and parameters of SVM classifiers is formulated as a convex optimization problem. The method has been applied to solve two important computer

vision problems: pose classification and facial feature detection. Experiments on standard face databases produce SVM classifiers that employ sparse sets of features while retaining classification performance.

Acknowledgements This work was supported by the U. S. Naval Research Laboratory under Contract No. N00173-07-C-2040. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U. S. Naval Research Laboratory.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] S. Avidan. Subset selection for efficient svm tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[3] S. Avidan. Joint feature-basis subset selection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[4] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proceedings of International Conference on Machine Learning*, 1998.

[5] B. Cao, D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Feature selection in a kernel space. In *Proceedings of International Conference on Machine Learning*, 2007.

[6] A. B. Chan, N. Vasconcelos, and G. R. G. Lanckriet. Direct convex relaxtions of sparse SVM. In *Proceedings of International Conference on Machine Learning*, 2007.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[8] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

[9] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, 2001.

[10] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, 2001.

[11] D. Cristinacce and T. F. Cootes. Facial feature detection and tracking with automatic template selection. In *Automatic Face and Gesture Recognition*, 2006.

[12] F. De la Torre and O. Vinyals. Learning kernel expansions for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[13] M. Dundar, J. Theiler, and S. Perkins. Incorporating spatial contiguity into the design of a support vector machine classifier. In *IEEE International Conference on Geoscience and Remote Sensing Symposium*, 2006.

[14] M. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, 2006.

[15] L. Hermes and J. Buhmann. Feature selection for support vector machines. In *Proceedings of International Conference on Pattern Recognition*, 2000.

[16] S. C. H. Hoi, M. R. Lyu, and E. Y. Chang. Learning the unified kernel machines for classification. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2006.

[17] H. Jee, K. Lee, and S. Pan. Eye and face detection using svm. In *Sensor Networks and Information Processing Conference*, 2004.

[18] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia. A hybrid classifier for precise and robust eye detection. In *International Conference on Pattern Recognition*, 2006.

[19] K. Kira and L. Rendell. The feature selection problem: traditional methods and new algorithm. In *Proceedings of AAAI Conference on Artificial Intelligence*, 1992.

[20] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Proceedings of European Conference on Computer Vision*, 1994.

[21] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidenite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[22] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *Automatic Face and Gesture Recognition*, 2004.

[23] O. L. Mangasarian and E. W. Wild. Feature selection for nonlinear kernel support vector machines. In *Workshop on Optimization-based Data Mining Techniques with Applications, IEEE International Conference on Data Mining*, 2007.

[24] B. Moghaddam, Y. Weiss, and S. Avidan. Fast pixel/part selection with sparse eigenvectors. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[25] P. Philips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation methodology for face-recognition. *Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[26] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond*. MIT Press, Cambridge, MA, 2002.

[27] J. Stoeckel and G. Fung. SVM feature selection for classification of spect images of alzheimer's disease using spatial information. In *IEEE International Conference on Data Mining*, 2005.

[28] Y. Sun and J. Li. Iterative RELIEF for feature weighting. In *Proceedings of International Conference on Machine Learning*, 2006.

[29] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[30] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems*, 2001.

[31] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.