Efficient Face Detection by a Cascaded Support Vector Machine Expansion

BY SAMI ROMDHANI[1], PHILIP TORR[2], BERNHARD SCHÖLKOPF[3], ANDREW BLAKE[4]

[1] *University of Basel, Bernoullistrasse, 16, 4056 Basel, Switzerland*
[2] *Oxford Brookes University, Department of Computing, Oxford OX33 1HX, UK*
[3] *MPI for Biological Cybernetics, Spemannstraße 38,72076 Tübingen, Germany*
[4] *Microsoft Research Ltd., 7 J J Thomson Ave, Cambridge CB3 0FB, UK*

We describe a fast system for the detection and localization of human faces in images using a nonlinear Support Vector Machine. We approximate the decision surface in terms of a reduced set of expansion vectors and propose a cascaded evaluation which has the property that the full support vectors expansion is only evaluated on the face-like parts of the image, while the largest part of typical images is classified using a single expansion vector (simpler and more efficient classifier). The cascaded evaluation offers a thirty-fold speed-up over an evaluation using the full set of reduced set vectors which itself already is thirty times faster than classification using all the support vectors.

Keywords: Face Detection, detection cascade, Cascaded Evaluation, Rare events detection, Support Vector Machines, Classification, Machine Learning, Reduced Set SVM.

1. Introduction

In this paper we consider the problem of face detection within a large collection of images, such as large photographic databases or images displayed on the internet. We consider the most general problem with no constraint on the position of the face. The input images are assumed to be monochrome; if a colour image is presented to the system, it is converted to monochrome, so that colour information alone cannot be used to reduce the search (leaving the exploration of colour cues to others).

This is a well established research problem and there have been a large number of different approaches to it. The most successful have included that of Osuna *et al.* [6] who applied support vectors (SVs) to the problem, that of Rowley *et al.* [11] who used a neural network, and that of Schneiderman & Kanade [12] who pursued a maximum likelihood approach based on histograms of feature outputs. The one common thing to all these methods is that they are all based on running a 20×20 pixel observation window across the image at all possible locations, scales and orientations. This involves a high degree of computation as (a) the observation window is a 400 dimensional vector that has to be classified in a non-linear classification task (b) there are hundreds of thousands of positions to search.

Within this paper we follow the support vector machine approach of Osuna *et al.* [6]. There are two novelties presented in this paper: (**i**) We use a *Reduced Set of Vectors* that yields a set of classifiers of increasing computational complexity. These classifiers have the property of being 'optimal' classifiers for a given complexity (This optimality is relative to the approximation criterion used, see §3). (**ii**) A *Cascaded Evaluation* is used such that the region of the image which are easily discriminated (non-face like) are rejected at an

early stage of the computation while more effort (and time) is spent on the more difficult parts of the images. This idea of cascaded evaluation and early rejection is also used by other and recent detection algorithms such as the Antiface detector [4] and the detector introduced by Viola & Jones [18]. The major difference between the detector presented in this paper and these detectors is the manner by which the cascade of classifiers is obtained, more specifically, the criterion optimized during training. The Antiface detector assumes that the negative examples (i.e. the non-faces) are modeled by a Boltzman distribution and that they are smooth. This assumption could increase the number of false positive in presence of a cluttered background. Here, we do not make this assumption: the negative example can be any image patch. The Viola & Jones detector, which was first presented shortly after the initial presentation of our method [10], uses a similar cascaded evaluation to ours. The main differences lie in the set of features selected for classification, in the classification function and in the training algorithm used to obtain them. They use Haar-like features selected by an AdaBoost algorithm. The choice of this set of features is based on performance reasons. Instead, to select and train our classifiers, we use a Support Vector Machine (SVM) known to yield classifiers with guaranteed generalization performances: given fixed but unknown probability distributions, an SVM minimizes an upper bound on the risk of misclassifying not only the examples in the training set (as all other learning techniques do) but also the *yet-to-be-seen* examples of the test set [17].

Nonlinear SVMs are known to lead to excellent classification accuracies in a wide range of tasks [13], including generic object detection [7, 9], full-body recognition [5], and face detection [6, 3]. They utilize a set of support vectors (SVs) to define a boundary between two classes, this boundary depending on a kernel function that defines a distance between two vectors. They are, however, usually slower classifiers than neural networks. The reason for this is that their run-time complexity is proportional to the number of SVs, i.e. to the number of training examples that the SVM algorithm utilises in the expansion of the decision function. Whilst it is possible to construct classification problems, even in high-dimensional spaces, where the decision surface can be described by two SVs only, it is normally the case that the set of SVs forms a substantial subset of the whole training set. This is the case for face detection where several hundred or thousand support vectors can be needed.

There has been a fair amount of research on methods for reducing the run-time complexity of SVMs [2, 14]. In the present article, we employ one of these methods and adapt it to the case where the reduced expansion is not evaluated at once, but rather in a cascaded way, such that in most cases a *very* small number of SVs are applied. A substantial speed up may be obtained not only for the face detection problem but for any classification application for which the number of instances of the two classes to be evaluated is highly asymmetric, as is the case for face detection.

The paper is organised as follows: In §2, the general theory of support vector machines is reviewed with emphasis on non-linear support vector machines. In §3 it is explained how to compute a set of reduced set vectors and how to deduce a suitable order for their evaluation. The training is explained in §4 and the face finding algorithm in §5. Results are given in §6 and conclusion plus avenues for future work suggested in §7.

2. Non-linear Support Vector Machines

Suppose that we have a labeled training set consisting of a series of 20×20 image patches $\mathbf{x}_i \in \mathcal{X}$ (arranged in a 400 dimensional vector) along with their class label $y_i \in \{\pm 1\}$.

Support Vector classifiers implicitly map the data $\mathbf{x}_i$ into a dot product space F via a (usually nonlinear) map $\Phi : \mathcal{X} \rightarrow F, \ \mathbf{x} \mapsto \Phi(\mathbf{x})$. Often, F is referred to as the *feature space*. Although F can be high-dimensional, it is usually not necessary to explicitly work in that space [1]. There exists a class of kernels $k(\mathbf{x}, \mathbf{x}')$ which can be shown to compute the dot products in associated feature spaces, i.e. $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$. The SVM training algorithm computes a hyperplane which separates the data in F into two classes by the largest margin. Once this geometrical problem is cast in terms of dot products, the kernel trick is used and thus all computations in F are reduced to the evaluation of the kernel. It can be shown that the resulting training problem consists of optimizing the following cost function with respect to the scalars α_i (for some positive value of the parameter C determining the trade-off between margin maximisation and training error minimisation)

$$\max_{\alpha} \ \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{2.1}$$

$$\text{subject to } \ 0 \leq \alpha_i \leq C, \ \ i = 1, \ldots, \ell, \ \ \sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{2.2}$$

It can also be shown that the resulting classification function $f(\mathbf{x})$ has the following expansion:

$$f(\mathbf{x}) = \operatorname{sgn} \left(\sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b \right). \tag{2.3}$$

It turns out that a large number of the α_i are null. Those training examples $\mathbf{x}_i$ with $\alpha_i > 0$ are called *Support Vectors*.

Kernels commonly used include polynomials $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$, which can be shown to map into a feature space spanned by all order d products of input features, and the Gaussian Radial Basis Function kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp \left(\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2 \, \sigma^2} \right). \tag{2.4}$$

Performance-wise, they have been found to do similarly well; in the present paper, we focus on the latter of the two. This means that support vectors act as templates for faces and non-faces.

3. Reduced Set Vectors

Assume we are given a vector $\Psi \in F$, that is in the linear span of the $\Phi(\mathbf{x}_i)$:

$$\Psi = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i), \tag{3.1}$$

with $\alpha_i \in \mathbb{R}, \mathbf{x}_i \in \mathcal{X}$. Ψ is in fact the normal vector of a Support Vector Machine and to reduce the complexity of using it to classify an input pattern $\mathbf{x}$ (using equation 2.3), one can approximate it by a *reduced set* expansion [2, 15]

$$\Psi' = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i), \tag{3.2}$$

with $N_z \ll N_x$, $\beta_i \in \mathbb{R}$, and *reduced set vectors* $\mathbf{z}_i \in \mathcal{X}$. To this end, one can minimise

$$\|\Psi - \Psi'\|^2 = \sum_{i,j=1}^{N_x} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i,j=1}^{N_z} \beta_i \beta_j k(\mathbf{z}_i, \mathbf{z}_j) - 2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_z} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{z}_j). \tag{3.3}$$

The key point of that method is that although Φ is not given explicitly, (3.3) can be computed (and minimised) in terms of the kernel. This minimisation is carried out over both the $\mathbf{z}_i$ and β_i.

The cascaded approach used here requires an extension of the reduced set method, to compute a whole sequence of reduced set approximations

$$\Psi'_j = \sum_{i=1}^{j} \beta_{j,i} \Phi(\mathbf{z}_i), \tag{3.4}$$

for $j = 1, \ldots, N_z$. The reduced set vectors $\mathbf{z}_i$ and the coefficients $\beta_{j,i}$ are computed by iterative optimisation [14]. For the first vector, we need to approximate $\Psi = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$ by $\Psi' = \beta \Phi(\mathbf{z})$. Minimising the distance $\|\Psi - \Psi'\|^2$ between Ψ and Ψ', with respect to $\mathbf{z}, \beta$, to give the first reduced set vector $\mathbf{z}_1$ and its coefficient $\beta_{1,1}$, using the method set out in the appendix.

Recall that the aim of the reduced set algorithm is to approximate a vector Ψ as in equation (3.1) by an expansion of the type of equation (3.2) with $N_z > 1$. The required higher order reduced set vectors $\mathbf{z}_i,\ i > 1$ and their coefficients β_i, are obtained in recursive fashion by defining a residual vector

$$\Psi_j = \Psi - \sum_{i=1}^{j-1} \beta_{j-1,i} \Phi(\mathbf{z}_i), \tag{3.5}$$

where Ψ is the original feature-space vector defined in (3.1). Then the procedure for obtaining the first reduced set vector $\mathbf{z}_1$ is repeated, now with Ψ_j in place of Ψ to obtain $\mathbf{z}_j$. However, the optimal β from this step is not used, instead optimal $\beta_{j,i},\ i = 1, \ldots, j$ are jointly computed, using proposition 7.1 in the appendix.

Thus, each Ψ'_j can be plugged into the SVM decision function to give rise to a classifier $f_j(\mathbf{x})$:

$$f_j(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \beta_{j,i} k(\mathbf{x}, \mathbf{z}_j) + b \right), \quad j = 1, \ldots, N_z. \tag{3.6}$$

These classifiers are such that, for a given complexity (i.e. number of reduced set vectors) they provide the optimal greedy approximation of the full SVM decision boundary: The first one is the one which, using the objective function (3.3), is closest to the full SVM (equation (2.3)) constrained to using only one reduced set vector. This optimisation is different from the SVM training algorithm as reduced set vectors are not constrained to be within the training set (whereas the support vectors are.) For this reason the first reduced set classifier performs better than a classifier using any of the support vectors. These arguments are valid for series of reduced set classifiers.

Figure 1 demonstrates, on a two-class (white and black dots) and 2 dimensional example, the effects on the classification boundary of cascaded reduced set vector machines for different m (examples with m = 1, 2, 3, 4, 9, and 13 are shown). Observe that with $m = 1$, the classification boundary is a simple circle and as m increases the boundary gets more complex and is able to discriminate in a more flexible way the two classes. Note that there is a law of diminishing returns, the first few reduced set vectors yielding the greatest increase in discrimination.

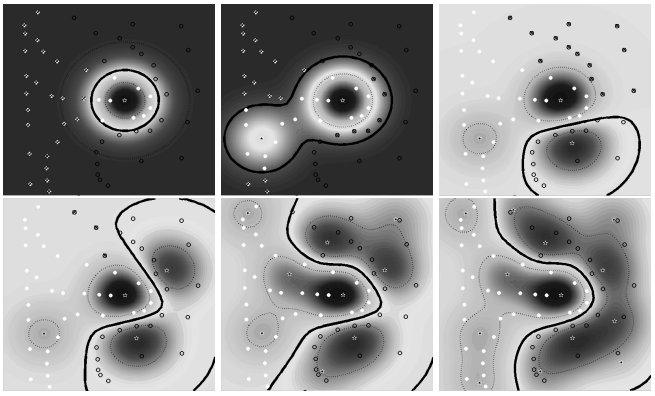


Figure 1. *The result of the cascaded application of reduced set vectors (stars) to a classification problem, showing the result of using 1,2,3,4,9 and 13 reduced set vectors. Darker regions indicate strong support for the classification.*

Note that it is not optimal to simply re-use the offset b stemming from the original SV machine for all f_j as it is done in equation (3.6). Reduced set approximations of decision functions can be improved by recomputing a threshold (a.k.a. offset), b_j, for the different classifiers, based on the training set or some validation set [14], hence the classification functions is modified to:

$$f_j(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{j} \beta_i k(\mathbf{x}, \mathbf{z}_i) + b_j\right). \tag{3.7}$$

4. Training

Initially the SVM was trained on 3600 frontal face and 25000 non-face examples using Platt's Sequential Minimal Optimisation [8]. The kernel used was Gaussian (equation (2.4)) with a standard deviation σ of 3.5. The trade-off parameter C between margin maximization and training error minimization was set to 1. The non-face patches were taken randomly from a set of 1000 images containing no faces. The SVM selected 1742 support vectors.

A retraining using the misclassifications of a previous training has been shown in [11] to produce a greatly improved classifier. Hence, to improve the performance of the classifier a second bout of training was initiated: To decrease the number of false positives the face detector was applied on a new set of 100 images which did not contain any faces. This generated 110000 false positive patches which were then added to the training. To decrease

the number of false negatives, virtual faces were generated and added to the training set. These virtual faces were computed by modifying the contrast or by adding an illumination plane to the faces of the original training set. This alleviates the need of computing a pre-processing at detection time and increase the run-time performance of our algorithm. The SVM was then retrained using this new training set which yielded 8291 support vectors. These were subsequently decreased to 100 reduced set vectors. The first 10 reduced set vector are depicted in figure 2.

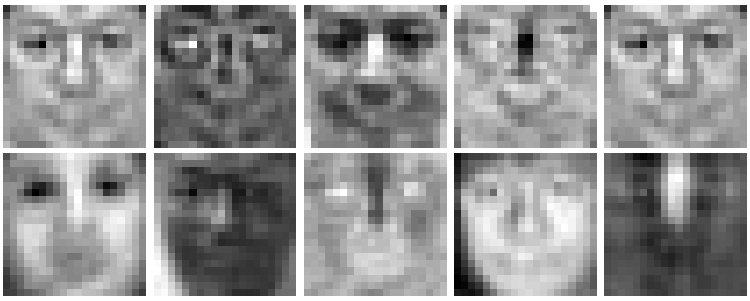


Figure 2. *First 10 reduced set vectors. Note that all vectors can be interpreted as either faces (e.g. the first one) or anti-faces (or inverse face, e.g. the second one). An anti-face is a reduced set vector with negative* $\beta_{m,i}$.

5. Face Detection by Cascaded Evaluation

At detection time, *each* pixel of an input image is the potential center of a face. To detect faces at different scales an image pyramid is constructed. If *w* and *h* are the width and the height of the input image and L and s the number of sub-sampling levels and the sub-sampling rate, respectively, the total number of patches to be evaluated is $N_p = \sum_{l=1}^{L} whs^{2(l-1)}$. Evaluating the full SVM or even the whole set of reduced vectors on all patches would be extremely slow. A large portion of the patches can be easily classified using only a few reduced set vectors. Hence we propose the following *Cascaded Reduced Set Machine* (CRSM) algorithm using a cascaded evaluation, to be applied to each overlapping patch $\mathbf{x}$ of an input image.

1. Set the hierarchy level to $j = 1$.
2. Evaluate $y_j = \mathrm{sgn}\left(\sum_{i=1}^{j} \beta_{j,i} K_i + b_j\right)$ where $K_i = k(\mathbf{x}, \mathbf{z}_i)$.
3. - If $y_j < 0$, $\mathbf{x}$ is classified as a non-face and the algorithm stops.
   - If $y_j \geq 0$, j is incremented. If $j < N_z$ the algorithm continues at step 2. If $j = N_z$, it continues at step 4.
4. When reaching this step $y_j \geq 0$ and $j = N_z$, the full SVM is applied on the patch $\mathbf{x}$, using equation (2.3). If the evaluation is positive the patch is classified as a face.

The main feature of this approach is that on average, very few kernels K_i have to be evaluated at any given image location — i.e., for most patches, the algorithm above stops at a level $j \ll N_z$. This speeds up the algorithm relative to the full reduced set (by more than

an order of magnitude in the face classification experiments reported below). A substantial speed up may be obtained not only for the face detection problem but for any classification application for which the number of instances of the two classes to be evaluated is highly asymmetric, as is the case for face detection (where most of the patches do not depict a face). Note that in the case of Gaussian kernels, the application of one reduced set vector amounts to a simple template matching operation.

Setting offsets. The offsets b_m are fixed to obtain a desired point on the Receiver Operator Curve (ROC) for the overall cascade. Suppose an overall false negative rate κ is required, then, given a 'decay rate' α, we express κ as a geometric series by setting false negative rates κ_m for the mth level in the hierarchy to $\kappa_j = \alpha\kappa_{j-1}$ where $\kappa_1 = \kappa(1-\alpha)$. This is because $\kappa(1-\alpha)(1+\alpha+\alpha^2+\ldots) = \kappa$. Now each b_m is fixed to achieve the desired κ_m over a validation set. The free parameter α can now be set to maximize the overall true positive rate over the validation set.

6. Results

Within this section the new cascaded evaluation algorithm, CRSM, is tested for speed and accuracy.

(*a*) *Speed Improvement.*

At detection time, due to the cascaded evaluation of the patches, very few reduced set vectors are applied. Figure 6 shows the number of reduced set vectors evaluated per patches for different methods (SVM, RSM and CRSM), when the algorithm is applied to the photograph in figure 4. The Full SVM and the RSM evaluate all their support or reduced set vectors on all the patches, while the CRSM uses on average only 2.8 reduced set vectors per patch. Figure 4 show the patches of an input image which remain after 1, 10, 20 and 30 cascaded reduced set evaluations on an image with one face. Figure 5 shows the results on an image with multiple faces.

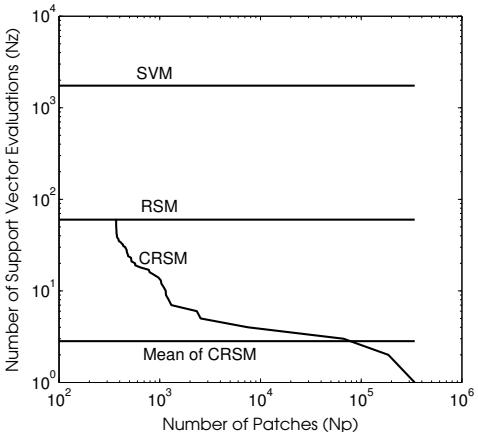


Figure 3. *Number of reduced set vectors used per patch for the full SVM (8291 support vectors), RSM and CRSM (both at 100 reduced set vector).*

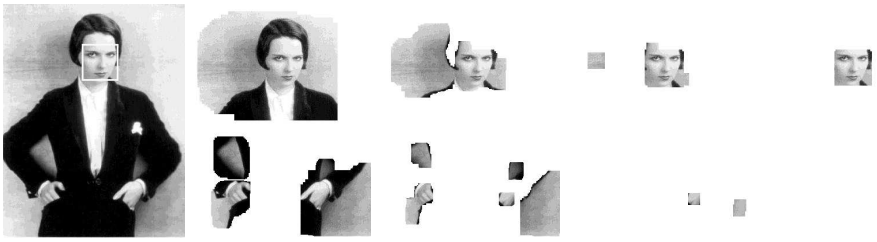


Figure 4. **From left to right:** *input image, followed by portions of the image which contain un-rejected patches after the cascaded evaluation of 1* (13.3% patches remaining)*, 10* (2.6%)*, 20* (0.01%) *and 30* (0.002%) *support vectors. Note that in these images, a pixel is displayed if it is part of any remaining un-rejecteded patch at any scale or position. This explains the apparent discrepancy between the above percentages and the visual impression.*

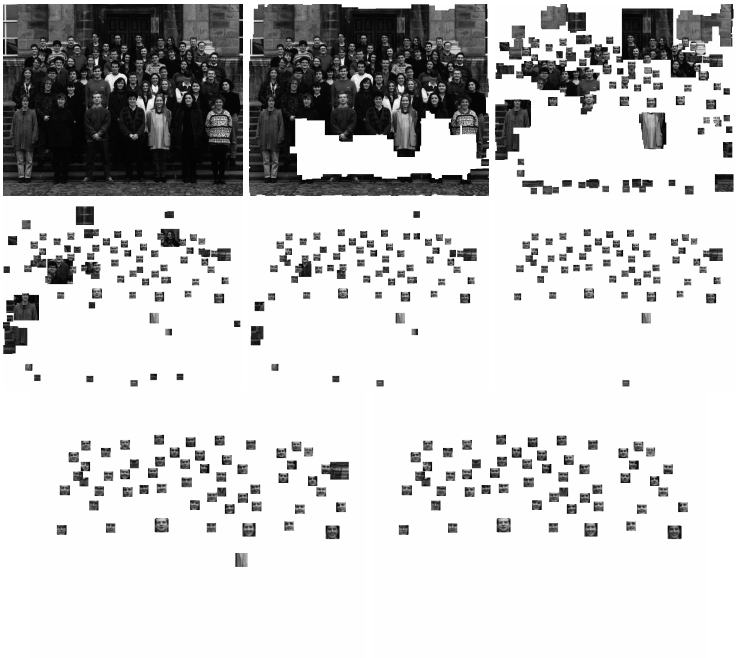


Figure 5. *Input image, followed by patches which remain after the evaluation of 1* (19.8% patches remaining)*, 10* (0.74%)*, 20* (0.06%) *and 30* (0.01%) *. . . 70* (0.007%) *support vectors. Note the comment in the caption of Figure 4.*

Figure 6 shows the number of reduced set vectors used to classify each patch of an image. The darkness of the pixels of the right image are proportional to the number of reduced set vectors used to classify the corresponding spot in the left image (note that the intensities are displayed at the center of the corresponding patches only). The uniform parts of the input image are easily rejected using a single reduced set vector, whereas the cluttered background requires more reduced set vectors. Note that very few patches needed all the reduced set vectors (only the patches containing the faces used all the reduced set vectors).



Figure 6. *Left: Input image. Right: The darkness of the pixels of this image is proportional to the number of reduced set vectors used to classify their associated patches. Light grey corresponds to the use of a single reduced set vector, black to the use of all the vectors.*

(*b*) *Accuracy.*

Figure 7 shows a comparison of the accuracy of the different methods. These Receiver Operator Curves (ROC) were computed on a test set containing 800 faces and 5000 non-faces. The accuracy of the CRSM (100 reduced set vectors) is comparable to the accuracy of the full SVM (8291 support vectors) and the RSM (100 reduced set vectors) which perform equally well.

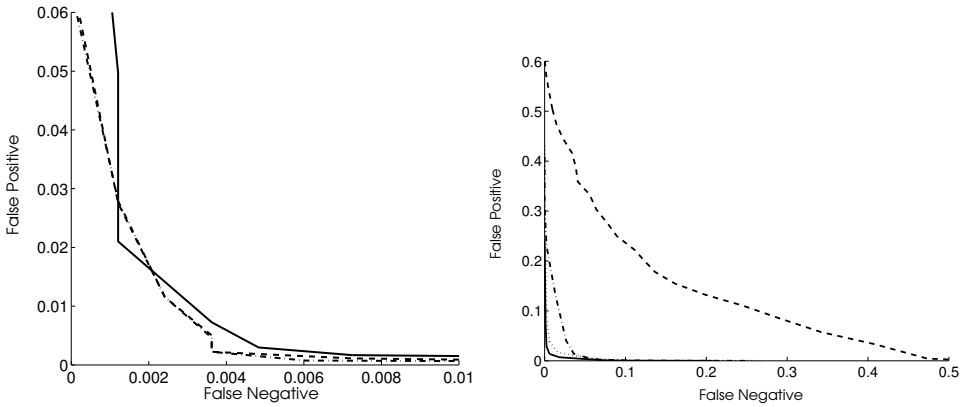


Figure 7. **Left:** *ROC for the SVM using 8291 support vectors (dotted line), the RSM using 100 reduced set vectors (dashed line) and CRSM using also 100 reduced set vectors (solid line). Note that the SVM and RSM curves are so close as to be indistinguishable.* **Right:** *ROC for an CRSM using 1 (dashed line), 2 (dash-dot line), 3 (dotted line) and 4 (solid line) reduced set vectors.*

To compare our system with others, we used the Rowley *et al.* [11] test set (which also includes the Sung & Poggio [16] and the Osuna *et al.* [6] test images). This set consists of 130 images containing 507 faces. We used a sub-sampling ratio of $s = 0.7$ and the input images were sub-sampled as long as their width and height was larger than 20 (i.e. the number of levels in the sub-sampling pyramid is $\min\left(\text{floor}\left(\frac{\log(20/w)}{\log 0.7}\right), \text{floor}\left(\frac{\log(20/h)}{\log 0.7}\right)\right)$

where w and h are, respectively, the width and the height of the input image). We obtained a detection rate of 80.7% with a false detection rate of 0.001% (i.e. one false positive in 100,000 image patch). These numbers are slightly worse than Rowley's, Sung's and Osuna's results, although they are hard to compare due to the fact that they pre-process the patches before feeding them into their classifier (histogram equalisation, background pixel removal and illumination gradient compensation). Our main objective was speed, hence no pre-processing was made. Secondly, we used a different training set as their training set was partly proprietary. Speed figures are also hard to compare, but from the information given, we conjecture that the Osuna *et al.* RS system is comparable in speed to our RS system, which in turn is 30 times slower than our cascaded evaluation system: The classification of one patch on a 500MHz Pentium takes 32μs for the cascaded evaluation, 1.2ms for the reduced set evaluation and 26ms for the full SVM.



Figure 8. **Top left***: The darkness of the pixels of the left image are proportional to the number of reduced set vectors used to classify their associated patches of the middle image. Light grey corresponds to the use of a single reduced set vector, black to the use of all the vectors.* **Top middle:** *153×263 middle image contains 76108 patches and was detected in* 2.58*s.* **Top right:** *A 601×444 image containing 518801 patches detected in 27.9s.* **Bottom Left:** *1280×1024 contains 2562592 patches and was detected in* 80.1*s.* **Bottom right:** *A 320×240 image containing 147289 patches detected in 10.4s (Note the false positives).*

(*c*) *Speed - Accuracy trade off*

The value of the threshold used for each reduced vector set classifier has a major impact on the speed vs accuracy trade off. As shown in §5, the parameter κ sets the false negative rate of the overall detector. Varying this parameters affects the thresholds of the reduced vector set classifiers and hence the computational load of the detector. If we accept a high false negative rate, the computational load (i.e. the number of reduced set vectors evaluation) will decrease. Figure 9 shows the false negative rate as a function of the computational load obtained by varying the parameter κ. These results were computed on a evaluation set containing 800 faces and 5000 non-faces. It shows that for a false negative rate of 2%, 4.45 kernel evaluations are required; for a false negative rate of 5%, 3.25 kernel

evaluations are required and for a false negative rate of 10%, 2.82 kernel evaluations are required.

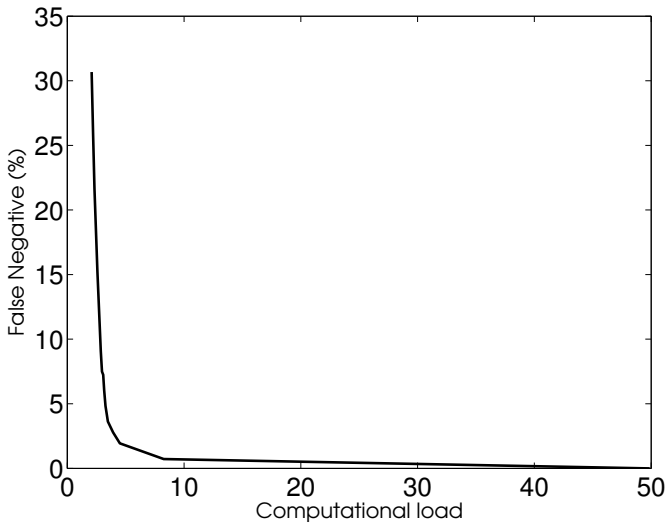


Figure 9. *Computational load, i.e. average number of reduced set vector evaluations, as a function of the false negatives, i.e. percentage of faces falsely classified as non-faces.*

7. Conclusion and Future Work

Pattern detection systems usually have to scan large images. Therefore, the greatest challenge in engineering systems for real-world applications is that of reducing computational complexity. Within this paper we have demonstrated computational savings in classification by the use of a cascaded reduced support vector evaluation. There are several avenues for future research. (a) We have explored the use of the Gaussian kernel, however it may be possible to tailor the kernel to something much more suited to facial detection. (b) It may be that the criteria for choosing the reduced set of support vectors can be improved. At present the reduced set of support vectors is chosen to minimize (3.3), which affects classification error only indirectly. However, it might be advantageous to choose a reduced set that minimizes classification error directly. (c) It would be interesting to adapt the thresholds based on contextual information: for instance, if a face is detected in the image, this places strong priors on the scale and orientation of any other faces we expect to see. This could further speed up the detection. Finally, although the method has been implemented for the task of face detection, it could be readily applied to a wide class of other detection and classifications problems.

Thanks to Henry Rowley for assisting us and for providing test images. Thanks to Mike Tipping, Kentaro Toyama for useful conversations.

Appendix A.

The first and subsequent reduced set vectors are obtained by optimising equation (3.3). In this appendix, we explain how to compute them.

Computing the first reduced set vector.

First observe that rather than minimizing

$$\|\Psi - \Psi'\|^2 = \sum_{i,j=1}^{N_x} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \beta^2 k(\mathbf{z}, \mathbf{z}) - 2\sum_{i=1}^{N_x} \alpha_i \beta k(\mathbf{x}_i, \mathbf{z}), \tag{A 1}$$

we minimize the distance between Ψ and the orthogonal projection of Ψ onto span $(\Phi(\mathbf{z}))$,

$$\left\| \frac{(\Psi \cdot \Phi(\mathbf{z}))}{(\Phi(\mathbf{z}) \cdot \Phi(\mathbf{z}))} \Phi(\mathbf{z}) - \Psi \right\|^2 = \|\Psi\|^2 - \frac{(\Psi \cdot \Phi(\mathbf{z}))^2}{(\Phi(\mathbf{z}) \cdot \Phi(\mathbf{z}))}. \tag{A 2}$$

i.e. maximize

$$\frac{(\Psi \cdot \Phi(\mathbf{z}))^2}{(\Phi(\mathbf{z}) \cdot \Phi(\mathbf{z}))}, \tag{A 3}$$

which can be expressed in terms of the kernel. The maximization of (A 3) over $\mathbf{z}$ is preferable to the one of (A 1) over $\mathbf{z}$ and β, since it comprises a lower-dimensional problem, and since $\mathbf{z}$ and β have different scaling behaviour. Once the maximum is found, it is extended to the minimum of $\|\Psi - \Psi'\|^2$ by setting (cf. (A 2)) $\beta = (\Psi \cdot \Phi(\mathbf{z}))/(\Phi(\mathbf{z}) \cdot \Phi(\mathbf{z}))$. The function (A 3) can either be minimized using standard techniques (as [2]), or, for particular choices of kernels, using fixed-point iteration methods, as shown presently.

For kernels which satisfy $k(\mathbf{z}, \mathbf{z}) = 1$ for all $\mathbf{z} \in \mathcal{X}$ (e.g. Gaussian kernels), (A 3) reduces to

$$(\Psi \cdot \Phi(\mathbf{z}))^2. \tag{A 4}$$

For the extremum, we have

$$0 = \nabla_{\mathbf{z}} (\Psi \cdot \Phi(\mathbf{z}))^2 = 2(\Psi \cdot \Phi(\mathbf{z})) \nabla_{\mathbf{z}} (\Psi \cdot \Phi(\mathbf{z})). \tag{A 5}$$

A sufficient condition for this to hold is that the second factor equal zero. To evaluate the gradient in terms of k, we substitute (3.1) to get the sufficient condition

$$0 = \sum_{i=1}^{N_x} \alpha_i \nabla_{\mathbf{z}} k(\mathbf{x}_i, \mathbf{z}). \tag{A 6}$$

For $k(\mathbf{x}_i, \mathbf{z}) = k(\|\mathbf{x}_i - \mathbf{z}\|^2)$ (e.g. Gaussian, or $(\|\mathbf{x}_i - \mathbf{z}\|^2 + 1)^c$ for $c = -1, -1/2$), we obtain

$$0 = \sum_{i=1}^{N_x} \alpha_i k'(\|\mathbf{x}_i - \mathbf{z}\|^2)(\mathbf{x}_i - \mathbf{z}), \tag{A 7}$$

leading to $\mathbf{z} = \frac{\sum_{i=1}^{N_x} \alpha_i k'(\|\mathbf{x}_i - \mathbf{z}\|^2) \mathbf{x}_i}{\sum_{i=1}^{N_x} \alpha_i k'(\|\mathbf{x}_i - \mathbf{z}\|^2)}$. For the Gaussian kernel $k(\mathbf{x}_i, \mathbf{z}) = \exp(-\|\mathbf{x}_i - \mathbf{z}\|^2/(2\sigma^2))$ we thus arrive at $\mathbf{z} = \frac{\sum_{i=1}^{N_x} \alpha_i \exp(-\|\mathbf{x}_i - \mathbf{z}\|^2/(2\sigma^2)) \mathbf{x}_i}{\sum_{i=1}^{N_x} \alpha_i \exp(-\|\mathbf{x}_i - \mathbf{z}\|^2/(2\sigma^2))}$, and devise an iteration

$$\mathbf{z}_{n+1} = \frac{\sum_{i=1}^{N_x} \alpha_i \exp(-\|\mathbf{x}_i - \mathbf{z}_n\|^2/(2\sigma^2)) \mathbf{x}_i}{\sum_{i=1}^{N_x} \alpha_i \exp(-\|\mathbf{x}_i - \mathbf{z}_n\|^2/(2\sigma^2))}. \tag{A 8}$$

The denominator equals $(\Psi \cdot \Phi(\mathbf{z}_n))$ and thus is nonzero in a neighbourhood of the extremum of (A 4), unless the extremum itself is zero. The latter only occurs if the projection

of Ψ on the linear span of $\Phi(\mathcal{X})$ is zero, in which case it is pointless to try to approximate Ψ. Numerical instabilities related to $(\Psi \cdot \Phi(\mathbf{z}))$ being small can thus be approached by restarting the iteration with different starting values.

Without further detail, we note that (A 8) can be interpreted as a type of clustering which takes into account both positive and negative data [14].

Computing higher order reduced set vectors

Higher order, reduced set vectors $\mathbf{z}_m,\ m > 1$ are required and each z_m is computed from Ψ_m (defined above) as follows. equation (A 8) is applied to Ψ_m (in place of Ψ) by expressing Ψ_m in its representation in terms of mapped input images:

$$\Psi_m = \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}_i) - \sum_{i=1}^{m-1} \beta_i \Phi(\mathbf{z}_i), 1 \tag{A 9}$$

i.e. we need to set $N_x = \ell + m - 1$,

$$(\alpha_1, \ldots, \alpha_{N_x}) = (\alpha_1, \ldots, \alpha_\ell, -\beta_1, \ldots, -\beta_{m-1}), \tag{A 10}$$

and

$$(\mathbf{x}_1, \ldots, \mathbf{x}_{N_x}) = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \mathbf{z}_1, \ldots, \mathbf{z}_{m-1}). \tag{A 11}$$

At each step, we compute the optimal coefficients $\beta = (\beta_1, \ldots, \beta_m)$ using Proposition 7.1 (note that if the discrepancy Ψ_{m+1} has not yet reached zero, then K^z will be invertible). The iteration is stopped after N_z steps, either specified in advance, or by monitoring when $\|\Psi_{m+1}\|$ (i.e. $\|\Psi - \sum_{i=1}^{m} \beta_i \Phi(\mathbf{z}_i)\|$) falls below a specified threshold.

Proposition 7.1 ([14]). *The optimal coefficients $\beta = (\beta_1, \ldots, \beta_m)$ for approximating $\Psi = \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}_i)$ by $\sum_{i=1}^{m} \beta_i \Phi(\mathbf{z}_i)$ (for linearly independent $\Phi(\mathbf{z}_1), \ldots, \Phi(\mathbf{z}_m)$) in the 2-norm are given by*

$$\beta = (K^z)^{-1} K^{zx} \alpha. \tag{A 12}$$

Here, the element at row i and column j of K^z and K^{zx}, denoted by K^z_{ij} and K^{zx}_{ij}, respectively, are equal to:

$$K^z_{ij} := (\Phi(\mathbf{z}_i) \cdot \Phi(\mathbf{z}_j)), \quad K^{zx}_{ij} := (\Phi(\mathbf{z}_i) \cdot \Phi(\mathbf{x}_j)). \tag{A 13}$$

The solution vector takes the form of equation (3.2).

References

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of the 5th ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

[2] C. J. C. Burges. Simplified support vector decision rules. In *13th Intl. Conf. on Machine Learning*, pages 71–77, 1996.

[3] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. AI Memo 1687, Massachusetts Institute of Technology, 2000.

[4] D. Keren, M. Osadchy, and C. Gotsman. Antifaces: a novel, fast method for image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:747–761, July 2001.

[5] Chikahito Nakajima, Massimiliano Pontil, Bernd Heisele, and Tomaso Poggio. Full-body person recognition system. *Pattern recognition*, 36(9):1997–2006, 2003.

[6] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *CVPR*, pages 130–136, 1997.

[7] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[8] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

[9] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[10] Sami Romdhani, Philip Torr, Bernhard Schlkopf, and Andrew Blake. Computationally efficient face detection. In *Proceedings of the 8th International Conference on Computer Vision*, July 2001.

[11] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 20:23–38, 1998.

[12] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to face and cars. In *CVPR*, pages 746–751, 2000.

[13] B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods — Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

[14] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000 – 1017, 1999.

[15] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[16] K-K. Sung and T. Poggio. Example-based learning for view-based human face detection. In *Proceedings from Image Understanding Workshop*, Monterey, CA, November 1994.

[17] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

List of Figures

Fig. 1. The result of the cascaded application of reduced set vectors (stars) to a classification problem, showing the result of using 1,2,3,4,9 and 13 reduced set vectors. Darker regions indicate strong support for the classification.

Fig. 2. First 10 reduced set vectors. Note that all vectors can be interpreted as either faces (e.g. the first one) or anti-faces (or inverse face, e.g. the second one). An anti-face is a reduced set vector with negative $\beta_{m,i}$.

Fig. 3. Number of reduced set vectors used per patch for the full SVM (8291 support vectors), RSM and CRSM (both at 100 reduced set vector).

Fig. 4. From left to right: input image, followed by portions of the image which contain un-rejected patches after the cascaded evaluation of 1 *(13.3% patches remaining)*, 10 *(2.6%)*, 20 *(0.01%)* and 30 *(0.002%)* support vectors. Note that in these images, a pixel is displayed if it is part of any remaining un-rejected patch at any scale or position This explains the apparent discrepancy between the above percentages and the visual impression.

Fig. 5. Input image, followed by patches which remain after the evaluation of 1 *(19.8% patches remaining)*, 10 *(0.74%)*, 20 *(0.06%)* and 30 *(0.01%)* . . . 70 *(0.007%)*support vectors. Note the comment in the caption of Figure 4.

Fig. 6. Left: Input image. Right: The darkness of the pixels of this image is proportional to the number of reduced set vectors used to classify their associated patches. Light grey corresponds to the use of a single reduced set vector, black to the use of all the vectors.

Fig. 7. Left: ROC for the SVM using 8291 support vectors (dotted line), the RSM using 100 reduced set vectors (dashed line) and CRSM using also 100 reduced set vectors (solid line). Note that the SVM and RSM curves are so close that they are not distinguishable. **Right:** ROC for an CRSM using 1 (dashed line), 2 (dash-dot line), 3 (dotted line) and 4 (solid line) reduced set vectors.

Fig. 8. Top left: The darkness of the pixels of the left image are proportional to the number of reduced set vectors used to classify their associated patches of the middle image. Light grey corresponds to the use of a single reduced set vector, black to the use of all the vectors. **Top middle:** 153×263 middle image contains 76108 patches and was detected in 2.58s. **Top right:** A 601×444 image containing 518801 patches detected in 27.9s. **Bottom Left:** 1280×1024 contains 2562592 patches and was detected in 80.1s. **Bottom right:** A 320×240 image containing 147289 patches detected in 10.4s (Note the false positives).

Fig. 9. Computational load, i.e. average number of reduced set vector evaluations, as a function of the false negatives, i.e. percentage of faces falsely classified as non-faces.

Affiliation of Authors

Sami Romdhani
University of Basel
Bernoullistrasse, 16
4056 Basel, Switzerland
sami.romdhani@unibas.ch

P. H. S. Torr
Oxford Brookes University,
Department of Computing,
Oxford OX33 1HX, UK
philtorr@brookes.ac.uk

Bernhard Schölkopf
MPI for Biological Cybernetics
Spemannstraße 38
72076 Tübingen, Germany
bernhard.schoelkopf@tuebingen.mpg.de

Andrew Blake
Microsoft Research Ltd., 7 J J Thomson Ave,
Cambridge CB3 0FB, UK
ablake@microsoft.com

8. Response to referee 1

This is a list of responses to the referee. There is one item for each item of the referee, in the same order.

1. Page 1. 'furthermore we allow the images to be monochrome or colour' is replaced by 'The input images are assumed to be monochrome; if a colour image is presented to the system, it is converted to monochrome', to disambiguate the statement.
2. Page 2. A forward reference to Section 3 is given for the set of 'optimal' classifiers.
3. Page 7. The caption of Figure 2 has been clarified by adding: 'An anti-face is a reduced set vector with negative $\beta_{m,i}$.'
4. Comments of the reviewer: '... classification problems even in high-dimensional spaces, where the decision surface can be described by two SV's only...'. I thought that if n is the dimension of the space then the number of support vectors is n+1?

   The reviewer is wrong here. This is an example R^N: Assume x1=(1,0,...,0), y1=1, x2=(-1,0,...,0), y2=-1, x3=(2,*,...,*), y3=1, x4=(-2,*,...,*), y4=-1, where arbitrary numbers are filled in for *. Then only x1 and x2 are SVs. The problem is separable (x1=0 is a separating hyperplane) and the optimal separation has margin at most 2 (since x1 and x2 have distance 2 and are in different classes) and this is attained by x1=0. We know that if an optimal separating hyperplane exists, it is unique. Hence x1=0 is the optimal one. Finally: it is clear that none of the other points is an SV for this hyperplane since SVs since SVs for this optimal hyperplane have coordinate (+1,*,...) or (-1,*,...). Note that we can add as many points x3, x4 as we wish. As we did not want to further complicate the paper, we did not include this example in the paper.
5. Page 3. We emphasised the fact that the map of the data does not involved y_i.
6. Page 3. 'separates the data in F by a large margin' is replaced by 'separates the data in F into two classes by the largest margin'.
7. Page 4. 'RBF' is explicitly written.
8. Page 4. The psi in equation 5 (and not 4 as the referee suggested) is clarified and its evaluation also.
9. Page 5. It is specified that the minimisation in Equation (7) is carried out over both zi and beta.
10. The connection between Psi, Psi' and f(x) is more precise: 'This algorithm provides a set of classifiers fj(x)' is replaced by: 'each Ψ'_j gives rise to a classifiers $f_j(x)$'.
11. Page 6. Figure 1 is explained in the text more fully.
12. The relationship between b and b_j is clarified.
13. Page 7. Yes the number of 110000 false positive is correct. The reason why this number is huge is because the initial set of non-face example on which the first SVM is trained is not representative enough. Therefore a re-training is performed on a new training set including these false positive to increase the performance of the classifier.

14. Page 8. The new algorithm is given the name Sequential Reduced Set Machine (SRSM), that is then related to when reporting the results on page 9.
15. There was an inconsistency in the algorithm with the indices j and m that is now resolved.
16. Page 9. 'The intensity values...' is replaced by 'The darkness...'
17. Page 10 (now page 12). a false detection rate of 0.001% means one false positive for every 100,000 image patch. This is now explicitly stated in the paper.
18. Page 13. The caption of the figure is updated.
19. Page 14. We now refer back to the text where the material of the appendix is used.
20. Page 15 (now 16). The notation Kzij is explained.
21. Page 16 (now 17). The numbers at the end of each reference are back references to the section where the references where cited. They are not shown anymore.

9. Response to referee 2

This is a list of responses to the referee. There is one item for each item of the referee, in the same order.

1. Page 2. 'Osuna at el' is changed to 'Osuna *et al.*'.
2. Page 3. $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$
3. Page 4. Equation (2) ends with a dot.
4. Page 4. 'Performance-wise... vector quantisation'. The support vectors are the training vectors that are close to the boundary surface. So the SVM learning effectively computes a subset of the training vectors, therefore it can be related to vector quantisation. Additionally, an input vector is compared with the support vectors and the classification decision is taken based on the distance with the support vectors (the distance function is given by the kernel function), therefore the support vectors can be viewed as template of faces and non-faces. We did not find necessary to include these explanations in the paper.
5. Page 4. Reference corrected.
6. Page 14 (now 15). It would not be correct to set $k(.,.) = k'(...)$, as the prime in Equation (18) is the usual derivation symbol of a function of one variable and k' is not another function as the reviewer assumed.
7. References to the paper suggested by the reviewer have been added on Page 3.

10. Modifications w.r.t. Version 2, submitted on the 5 April 2004

1. Page 1: 'well established researched problem' is replaced by 'well established research problem'.
2. We chose to replace the naming of our method: sequential evaluation is replaced by cascaded evaluation. We feel that the word 'cascade' reflects better the proposed method. Hence, the abbreviation of our method, introduced in Section 5, is now CRSM, for Cascaded Reduced Set Expansion. The caption of Figure 3 is changed accordingly.
3. Page 6: Steps 3 and 4 of the algorithm have been reformulated and made clearer.
4. Page 7 and following: R. O. C. is replaced by ROC.
5. Page 7: ν is replaced by κ, as ν is used to denote another variable in the context of SVM.
6. Caption of Figure 7: 'the curves are so close that they are not distinguishable' is replaced by 'the curves are so close as to be indistinguishable'.
7. All references to equations are now bracketed.