# Face Class Modeling Using Mixture of SVMs

Julien Meynet[1], Vlad Popovici, and Jean-Philippe Thiran

Signal Processing Institute, Swiss Federal Institute of Technology Lausanne,
CH-1015 Lausanne, Switzerland
http://itswww.epfl.ch

**Abstract.** We [1] present a method for face detection which uses a new SVM structure trained in an expert manner in the eigenface space. This robust method has been introduced as a post processing step in a real-time face detection system. The principle is to train several parallel SVMs on subsets of some initial training set and then train a second layer SVM on the margins of the first layer of SVMs. This approach presents a number of advantages over the classical SVM: firstly the training time is considerably reduced and secondly the classification performance is improved, we will present some comparisions with the single SVM approach for the case of human face class modeling.

## 1 Introduction

Human face detection is one of the most important tasks of the face analysis and can be viewed as a pre-processing step for face recognition systems. It is always important to find a precise localization of faces in order to be able to later recognize them. The difficulty resides in the fact that the face object is highly deformable and its aspect is also influenced by the environmental conditions. On the other hand, the class of objects which do not belong to the face class is large and can not be modeled. Thus finding a model for the face class is a challenging task. In the last years, many methods have been proposed, we give a brief overview of the most significant of them.

A fast face detection alorithm has been proposed by Viola and Jones[1] , it uses simple rectangular Haar-Like features boosted in a cascade structure. We have used this fast approach as a pre-processing step in order to obtain a fast and robust face detection system.

Then, one of the most representative approaches for the class of neural networks–based face detectors is the work reported by Rowley et. al. in [2]. Their system has two major components: a face detector made of a scanning window at each scale and position, and a final decision module whose role is to arbitrate multiple detections.

Sung and Poggio have developed a clustering and distri-bution-based system for face detection [3]. There are two main components in their system: a model

[1] The authors thank the Swiss National Science Foundation for supporting this work through the National Center of Competence in Research on "Interactive Multimodal Information Management (IM2)".

of the face/non–face patterns distribution and a decision making module. The two class distributions are each approximated by six Gaussian clusters.

A naive Bayes classifier based on local appearance and position of the face pattern at different resolutions is described by Schneiderman and Kanade in [4]. The face samples are decomposed in four rectangular subregions which are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns.

Osuna et. al. developed a face detector based on SVM that worked directly on the intensity patterns [5]. A brief description of the SVM is given in this paper also. The large scale tests they performed showed a slightly lower error rate than the system of Sung and Poggio, while running approximately 30 times faster.

In [6], Popovici and Thiran proposed to model the face class using a SVM trained in eigenfaces space. They showed that even a very low dimensional space (compared with the original input space) suffices to capture the relevant information when used in conjunction with a powerful classifier, like a non linear SVM. We propose here an extension of these ideas that employs a mixture of SVMs (MSVM in the following) for better capturing the face class variability. We use the analysis from [6] for choosing the input space of our classifier, but we will also extend the feature vector by adding a new term that accounts for the information lost through the PCA process. The idea of using mixture of experts (in our case SVMs) is not new, but we will use a slightly different approach: the final decision is taken by a SVM that is trained using the margins output by the first layer of SVMs. In training this final SVM we penalize more the false negative type of errors (missed faces) to favor the detection of faces. Other ways of combining the experts can be used: for example, in [7] the EM algorithm was used to train the experts. Later [8] replaced neural network experts by SVMs but still trained each expert on the whole dataset. The use of parallel SVMs trained on subsets of large scale problem has been studied in 2002 in [9]. However, the second layer remained a neural network.

We will introduce the MSVM and we will justify its use both from a theoretical perspective and a more practical one. In section 2 we will briefly review the SVM theory and then we will describe the MSVM approach. The MSVM will be trained on face and non face examples pre-processed by PCA, as described on section 2.3. Finally, in sections 3 and 4 we present some experiments and comparisons with classical SVM and we draw some conclusions.

## 2 Mixtures of SVMs

### 2.1 An overview of Classical SVM

Let us begin with a brief overview of the classical SVM algorithm. More information about SVM can be found in [10], [11].

Let $\{(\mathbf{x}_i, y_i)|i = 1, \ldots, l\} \subset \mathbb{R}^n \times \{-1, +1\}$ be a set of examples. From a practical point of view, the problem to be solved is to find that hyperplane that

correctly separates the data while maximizing the sum of distances to the closest positive and negative points (i.e. *the margin*). The hyperplane is given by[2]:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{1}$$

and the decision function is

$$f(\mathbf{x}) = \operatorname{sgn}(h_{\mathbf{w},b}(\mathbf{x})) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \tag{2}$$

In the case of linearly separable data, maximizing the margins means to maximize $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, to minimize $\|\mathbf{w}\|^2$, subject to $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1$. Suppose now that the two classes overlap in feature space. One way to find the optimal plane is to relax the above constraints by introducing the *slack variables* $\xi_i$ and solving the following problem (using 2-norm for the slack variables):

$$\min_{\xi, \mathbf{w}, b} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 \tag{3}$$

$$\text{subject to} \qquad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \tag{4}$$

where $C$ controls the weight of the classification errors ($C = \infty$ in the separable case).

This problem is solved by means of Lagrange multipliers method. Let $\alpha_i \geq 0$ be the Lagrange multipliers solving the problem above, then the separating hyperplane, as a function of $\alpha_i$, is given by

$$h_{\alpha_i,b}(\mathbf{x}) = \sum_{i, \alpha_i > 0} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b \tag{5}$$

Note that usually only a small proportion of $\alpha_i$ are non-zero. The training vectors $\mathbf{x}_i$ corresponding to $\alpha_i > 0$ are called *support vectors* and are the only training vectors influencing the separating boundary.

In practice however, a linear separating plane is seldom sufficient. To generalize the linear case one can project the input space into a higher–dimensional space in the hope of a better training–class separation. In the case of SVM this is achieved by using the so–called "kernel trick". Basically, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. As the data vectors are involved only in this inner products, the optimization process can be carried out in the feature space directly. Some of the most used kernel functions are:

$$\text{the polynomial kernel} \qquad K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \tag{6}$$

$$\text{the RBF kernel} \qquad K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \tag{7}$$

### 2.2 Mixture of SVMs (MSVM)

SVM techniques are well known since a few years for many reasons, among them their generalization capabilities. However, as explained in the previous

[2] We use $\langle \cdot, \cdot \rangle$ to denote the inner product operator

subsection, training a SVM usually requires solving a quadratic optimization problem, which means it also varies quadratically with the number of training examples. We know by experience that because of the large variability of both face and non face classes, building a face detection system requires a large amount of examples. So in order to make easier the training of the SVM (in term of training time) we use a parallel structure of SVMs similar to the one introduced in [9]. A first part of the dataset is splitted and clustered and each cluster is used to train each SVM of the first layer. And then the remaining example are used to train a second layer SVM, based on the margins of the first layer SVMs. Basically, the input space for the $2^{\text{nd}}$ layer SVM is the space of margins generated by the $1^{\text{st}}$ layer SVMs. We can represent the output of such a mixture of $M+1$ experts as follows:

$$h_{\alpha_i,b}(\mathbf{x}) = \sum_{i,\alpha_i>0} y_i \alpha_i K(\mathbf{m}_i(x_i), \mathbf{m}(x)) + b \tag{8}$$

where $\mathbf{m}(\mathbf{x})$ is the vector of margins output by the $M$ SVMs in the first layer given the input $\mathbf{x}$.

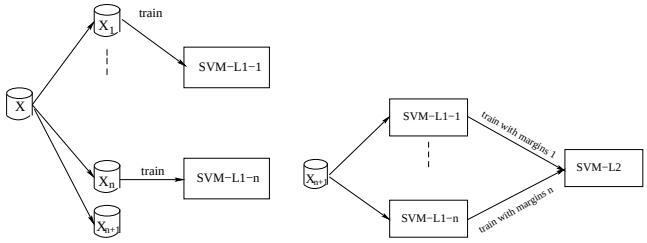
Assuming that we want to train $M$ SVMs in the first layer, we will need $M+1$ training sets (an additional one is used to train the second layer SVM) - see figures 1. We use two different approaches for generating the $M+1$ subsets. One consists of a random partitioning of the original training set. The second one is more elaborated: we first randomly draw a sample that will be used for training the second layer and then we use a clustering algorithm, like k-Means[12] , for building the $M$ subsets needed for training the first layer SVM.

In both cases we train each SVM-L1-i using a cross-validation process to select the best parameters then we use the $M+1$-th dataset for training the second layer SVM (SVM-L2): we let each of SVM-L1-i to classify the examples from this dataset and we take the margins output by the SVM-L1-i as input for SVM-L2. The margin can be seen as a measure of confidence in classifying an example, so, in some sense, the second layer SVM learns a non linear function that depends on the input vector and which assembles the confidences of each individual expert.

From a practical point of view, we have decomposed a problem of $O(N^2)$ complexity in $M+1$ problems of $O(\lceil \frac{N}{M+1} \rceil^2)$ complexity. As $N >> M$ this decomposition is clearly advantageous, and has the potential of being implemented in parallel, reducing even more the training time. Another issue that should be mentionned here is related to the robustness of the final classifier. In the case of a single SVM, if the training set contains outliers or some examples heavily affected by noise, its performance can be degraded. However, the chances of suffering from such examples are less important in the case of MSVM.

### 2.3 Construction of the Eigenfaces

As we use a large number of examples, we use Principal Component Analysis(PCA) to decrease the dimensionality of the image space. We first recall the definition of PCA and then we will discuss some possible improvements.




(a) First layer (b) Second layer

**Fig. 1.** *Training of the SVMs of the first ans second layer.*

**Principal Component Analysis (PCA) and Eigenfaces** Let $\mathbf{x}_1, \ldots, \mathbf{x}_l \in \mathbb{R}^n$ be a set of $n-$dimensional vectors and consider the following linear model for representing them

$$\mathbf{x} = W_{(k)}\mathbf{z} + \mu \tag{9}$$

where $W_{(k)}$ is a $n \times k$ matrix, $\mathbf{z} \in \mathbb{R}^k$ and $\mu \in \mathbb{R}^n$. For a given $k < n$, the PCA can be defined ([13]) as the transformation $W_{(k)}$ whose column vectors $\mathbf{w}_j$, called *principal axes*, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors $\mathbf{w}_j$ are given by the dominant $k$ eigenvectors of the sample covariance matrix[3] $S = \frac{1}{l}\sum_l (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)'$ such that $S\mathbf{w}_j = \lambda \mathbf{w}_j$ and where $\mu$ is the sample mean. The vector $\mathbf{z}_i = W'_{(k)}(\mathbf{x}_i - \mu)$ is the $k-$dimensional representation of the observed vector $\mathbf{x}_i$. The projection defined by PCA is optimal in the sense that amongst the $k-$dimensional subspaces, the one defined by the columns of $W_{(k)}$ minimizes the reconstruction error $\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ where $\hat{\mathbf{x}}_i = W_{(k)}\mathbf{z}_i + \mu$.

Now let us view an image as a vector in $\mathbb{R}^n$ space by considering its pixels in lexicographic order. Then the PCA method can be applied to images as well, and in the case of face images the principal directions are called *eigenfaces* [14],[15]. Some details about the estimation of the eigenfaces space dimensionality such as classification in eigenfaces space using SVMs are shown in [6].

**Distance From Feature Space (DFFS)** Traditionally, the distance between a given image and the class of faces has been decomposed in two orthogonal components: the *distance in feature space* (corresponding to the projection onto the lower dimensional space) and the *distance from feature space (DFFS)* (accounting for the reconstruction error).

$$DFFS = \sqrt{\|\mathbf{x} - \mu\|^2 - \|\mathbf{z}\|^2} \tag{10}$$

Given this and considering that the DFFS still contains some useful information for classification, we can improve the discrimination power by adding the value

[3] We denote with a prime symbol the transpose of a matrix or a vector.

of the DFFS to the projection vector. Thus considering that we keep 85% of total variance with the $k$ first eigenvectors, we use the following vectors to perform the classification.

$$X = (x_1, \ldots, x_k, x_{k+1}), \tag{11}$$

where $x_1, \ldots, x_k$ represent the projection onto the $k-$ dimensional eigenfaces space and $x_{k+1}$ the DFFS.

## 3 Experiments and results

Our experiments have been based on images from the BANCA [16] and the XM2VTS[17] databases for the faces whereas the non faces examples were chosen by bootstrapping on randomly selected images. In order to test the accuracy and the validity of the method, we have used a dataset as follows: A training set made of 8256 faces and 14000 non face examples, all images with the fixed size $20 \times 15$ pixels. The validation set had 7822 faces and 900000 non faces of the same size. We first tried to find a coherent PCA decomposition before training the SVMs. The PCA reduces the dimensionality of the input space but also the eigenfaces proved to be more robust features in real-world applications than the raw pixel values.

We first estimated the dimensionality of the eigenfaces space that we need to keep 85% of total variation. For this we have estimated the number of examples from which the eigenfaces space has a stable dimensionality for keeping 85% of total variation. So we performed the PCA decomposition on a randomly selected part of the training set and from the 300-dimensional input space we kept only 17 eigenfaces. As explained earlier, the vector used for the classification task is made by adding the DFFS value to the projection onto the eigenfaces space.

Then, the face training set has been splitted into 2 subsets. The first part, containing 5000 examples, has been splitted into 5 subsets, either by clustering or by random sampling. We trained the SVM-L1-i on these 5 subsets, each combined with 2000 negative examples and the remaining subset (3000 faces and 4000 non faces) was passed through all the trained SVM-L1-i.The output margins were used to train the second layer SVM. Table 1 shows the classification results on the validation set for each SVM.

Using the random sampling for generating the training sets for the first layer has the advantage of reducing the importance of outliers or unusual examples, but leads to SVMs that need more support vectors for good performances. On the other hand, using k-Means clustering leads to SVMs that perform like experts on their own domain, but whose common expertise should cover the full domain.

It is interesting to see that the MSVM has better generalization capabilities than a single SVM trained on the initial dataset. This result shows that as explained in section 2, MSVM does not only give improvements in term of training time but also in term of classification performances. We can also notice the importance of the SVM-L2: The TER (Total error rate) has been improved from a single SVM but it is really more interesting for face detection as it improves the true positive rate (even if the false positive rate is degraded). Just recall

| Classifier | Faces(%) | | NonFaces(%) | |
|---|---|---|---|---|
| | r.s. | k-m | r.s. | k-m |
| SVM-L1-1 | 86.23 | 76.47 | 99.00 | 98.86 |
| SVM-L1-2 | 84.91 | 82.32 | 99.00 | 97.68 |
| SVM-L1-3 | 85.13 | 81.23 | 99.02 | 98.77 |
| SVM-L1-4 | 84.64 | 77.12 | 99.13 | 99.12 |
| SVM-L1-4 | 85.66 | 74.29 | 99.12 | 99.12 |
| SVM-L2 | 93.60 | 95.37 | 98.14 | 96.43 |

**Table 1.** Performances on the validation set for both random sampling (r.s.) and k-Means clustering (k-m).

| Classifier | Faces(%) | Non Faces(%) | Total N$^o$ SV |
|---|---|---|---|
| MSVM,r.s | 93.60 | 98.14 | 1673 |
| MSVM,$k$-m | 95.37 | 96.43 | 1420 |
| Single SVM | 92.8 | 99.52 | 2504 |

**Table 2.** Comparison between MSVM with random sampling (MSVM,r.s), MSVM with $k$-Means clustering (MSVM,$k$-m) and a single SVM trained on the complete training set.

that in the face detection world, we often want to detect a maximum number of faces even if some non face examples are misclassified. Another advantage of this method compadred to the single SVM trained on the complete dataset is that the total number of support vectors (last column in table2) is radically inferior in the case of MSVM. This emphasizes the gain of time and computation complexity given by the MSVM.

## 4 Conclusions

In this paper we presented a method for face class modeling using mixtures of SVMs. This approach presents an extension to the SVM technique which allows a better use of particularly large datasets. We have used this mixture of experts approach in the context of face detection using a PCA decomposition and then adding the DFFS to the features in order to decrease the information loss through the PCA process. We have proposed here a mixture of SVMs made of several SVMs in a first layer trained on independent subsets of the initial dataset and a second layer trained on the margins predicted by the first layer SVMs given another independent subset. It has been shown that this structure allowed a significant improvement from the single SVM trained on the complete database. On the first hand, the training time is largely reduced because of the parallel structure and the splitting of the original subset, and on the other hand, the discrimination capabilities are improved because of the possible presence of noise and outliers in the dataset. In order to have a structure more adapted to the datasets, we are now working on more specialized experts, for example by using a clustering in eigenfaces space based on a more appropriated metrics.

## References


1. Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
2. Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Human face detection in visual scenes," in *Advances in Neural Information Processing Systems*, David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, Eds. 1996, vol. 8, pp. 875–881, The MIT Press.
3. Kah-Kay Sung and Tomaso Poggio, "Example-based learning for view-based human face detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.
4. H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationship for object recognition," in *Proceedings of Computer Vision and Pattern Recognition*, 1998, pp. 45–51.
5. Edgar Osuna, Robert Freund, and Federico Girosi, "Training support vector machines: an application to face detection," in *Proceedings of Computer Vision and Pattern Recognition*, 1997.
6. V. Popovici and J.-P. Thiran, "Face Detection using SVM Trained in Eigenfaces space," in *Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003, pp. 925–928.
7. Robert A. Jacobs Michael I. Jordan Steven J. Nowlan and Geo rey E. Hinton, "Adaptive mixtures of local experts," in *Neural Computation 3(1), 1991*, 1991, pp. 79–87.
8. J. T. Kwok, "Support vector mixture for classification and regression problems," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 1998, pp. 255–258.
9. R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of svms for very large scale problems," 2002.
10. Vladimir Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
11. Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.
12. Darken, C., and Moody, J., "Fast adaptive k-means clustering: Some empirical results," 1990.
13. H. Hotteling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, , no. 24, pp. 417–441, 498–520, 1933.
14. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, pp. 519–524, 1987.
15. Matthew Turk and Alex Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
16. S. Bengio, F. Bimbot, J. Mariéthoz, V. Popovici, F. Porée, E. Bailly-Baillière, G. Matas, and B. Ruiz, "Experimental protocol on the BANCA database," IDIAP-RR 05, IDIAP, 2002.
17. K Messer and J Matas and J Kittler and J Luettin and G Maitre, "XM2VTSDB: The Extended M2VTS Database," in *Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999.