Over-Complete Wavelet Approximation of a Support Vector Machine for Efficient Classification

Matthias Rätsch[1], Sami Romdhani[1], Gerd Teschke[2], and Thomas Vetter[1]

[1] University of Basel, Computer Science Department,
Bernoullistrasse 16, CH - 4056 Basel, Switzerland
[2] University of Bremen, ZETEM, PF 33 04 40, D-28334 Bremen, Germany
{matthias.raetsch, sami.romdhani, thomas.vetter}@unibas.ch
teschke@math.uni-bremen.de

Abstract. In this paper, we present a novel algorithm for reducing the runtime computational complexity of a Support Vector Machine classifier. This is achieved by approximating the Support Vector Machine decision function by an over-complete Haar wavelet transformation. This provides a set of classifiers of increasing complexity that can be used in a cascaded fashion yielding excellent runtime performance. This over-complete transformation finds the optimal approximation of the Support Vectors by a set of rectangles with constant gray-level values (enabling an Integral Image based evaluation). A major feature of our training algorithm is that it is fast, simple and does not require complicated tuning by an expert in contrast to the Viola & Jones classifier. The paradigm of our method is that, instead of trying to estimate a classifier that is jointly accurate and fast (such as the Viola & Jones detector), we first build a classifier that is proven to have optimal generalization capabilities; the focus then becomes runtime efficiency while maintaining the classifier's optimal accuracy. We apply our algorithm to the problem of face detection in images but it can also be used for other image based classifications. We show that our algorithm provides, for a comparable accuracy, a 15 fold speed-up over the Reduced Support Vector Machine and a 530 fold speed-up over the Support Vector Machine, enabling face detection at 25 fps on a standard PC.

1. Introduction

Image based classification tasks are time consuming. For instance, detecting a specific object in an image, such as a face, is computationally expensive, as all the pixels of the image are potential object centres. Hence all the pixels must be classified.

Therefore, recently, more efficient methods have emerged based on a cascaded evaluation of hierarchical filters: image patches easy to discriminate are classified by a simple and fast filter, while patches that resemble the object of interest are classified by more involved and slower filters. In the area of face detection [11], cascaded based classification algorithms were introduced by Keren *et al.*[7], by Romdhani *et al.* [10] and by Viola and Jones [16]. The detector from Keren *et al.* [7] assumes that the negative examples (i.e. the non-faces) are modeled by a Boltzmann distribution and that they are smooth. This assumption could increase the number of false positive in presence of a cluttered background. Romdhani *et al.* [10] use a Cascaded Reduced Set Vectors (RSV) expansion of a Support Vector Machine (SVM) [15]. The speed bottleneck of [10] is that at least one convolution of a 20×20 filter has to be carried out on the full image,

W. Kropatsch, R. Sablatnig, and A. Hanbury (Eds.): DAGM 2005, LNCS 3663, pp. 351–360, 2005.

resulting in a computationally expensive evaluation of the kernel with an image patch. Kienzle *et al.* [8] present an improvement of this method, where the first (and only the first) RSV is approximated by a separable filter. Viola & Jones [16] use Haar-like oriented edge filters having a block like structure enabling a very fast evaluation by use of an Integral Image. These filters are weak, in the sense that their discrimination power is low. They are selected, among a finite set, by the Ada-boost algorithm that yields the ones with the best discrimination. A drawback of their approach is that it is not clear that the cascade achieves optimal generalization performances. Practically, the training proceeds by trial and error, and often, the number of filters per stage must be manually selected so that the false positive rate decreases smoothly. Another drawback of the method is that the set of available filters is limited and manually selected. Additionally, the training of the classifier is very slow, as every filter (and there are about 10^5 of them) is evaluated on the whole set of training examples and this is done every time a filter is added to a stage of the cascade.

In this paper, we present a novel efficient classification algorithm based on following features:

1. Use of an SVM classifier that is known to have optimal generalization capabilities.
2. To achieve high run-time efficiency we use a reduced set of Support Vector (RVM in [10]).
3. The high run-time efficiency is also obtained by a coarse-to-fine approximation of the classifier enabling a cascaded evaluation. For non-symmetric data (i.e. only few positives to many negatives) we achieve an early rejection of easy to discriminate vectors. The granularity of the accuracy of the approximation is set by the following parameters, which are automatically selected at detection time based on the image patch to be classified: (i) the number of Reduced Set Vector (RSV) used and (ii) the accuracy of the wavelet representation of these RSV's. This constitutes the major novelty of this paper. The trade-off between accuracy and speed is very continuous.
4. As the RSV's are approximated by a Haar wavelet transform, the Integral Image method is used for their evaluation, similarly to [16].
5. We use the over-complete wavelet theory to obtain the global optimum approximation of RSV's. As shown in Section 2.1.3.3, the over-complete wavelet theory provides an upper bound on the distance between the decision function of the RVM and of the proposed W-RSV. The proposed learning stage is fast, straightforward, automatic and does not require the manual selection of ad-hoc parameters, as opposed to the Viola and Jones method [16]. For example, the training time, on the data set mentioned in Section 3, was two hours which is a vast improvement over the Viola detector.

The novelty to [9] is 3. (ii) and 5.: The Simulated Annealing optimization using morphological filters is replaced by the over-complete wavelet transformation. The problems with the Simulated Annealing method is that it did not provide the global optimum of the RVM approximation in all cases and it was difficult to adjust the approximation accuracy. It should be noted that, in this work, we apply an over-complete wavelet transformation of the Reduced Support Vector Machine itself, and not of the input space as a pre-processing like [6,1].

Section 2 details our novel training algorithm that constructs a Wavelet Approximated Reduced Set Vectors expansion having a block-like structure. It is shown in Section 3 that the new expansion yields a comparable accuracy to the SVM while providing a significant speed-up.

2 Over-Complete Wavelet Approximated Support Vector Machine

Support Vector Machines (SVM), used as classifiers, are now well-known for their good generalisation capabilities. Their decision function has the following form: $y(\mathbf{x}) = \sum_k \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_k) + b$, where $k(\cdot)$ is the kernel used. In order to improve the runtime performance [13] proposed to approximate the SVM by a Reduced Support Vector Machine (RVM), used with a cascaded evaluation in [10]. The RVM aims to approximate the Support Vectors, $\mathbf{x}_k$ by a *smaller* set of Reduces Set Vectors (RSV's), $\mathbf{z}_k$. During evaluation, most of the time is spent in kernel evaluations. In the case of the Gaussian kernel, $k(\mathbf{x}, \mathbf{z}_k) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{z}_k\|^2}{2\,\sigma^2}\right)$, chosen here, the computational load is spent in evaluating the norm of the difference between a patch and a RSV. This norm can be expanded as follows: $\|\mathbf{x} - \mathbf{z}_k\|^2 = \mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{z}_k + \mathbf{z}_k'\mathbf{z}_k$. As $\mathbf{z}_k$ is independent of the input image, it can be pre-computed. The sum of squares of the pixels of a patch of the input image, $\mathbf{x}'\mathbf{x}$ is efficiently computed using the Integral Image ([4,16]) of the squared pixel values of the input image. As a result, the computational load of this expression is determined by the term $2\mathbf{x}'\mathbf{z}_k$.

The novelty of this paper is the approximation the RSV's, $\mathbf{z}_k$, by a set of Wavelet Approximated Reduced Set Vectors (W-RSV), $\mathbf{u}_k$ that have a block-like structure, as seen in Figure 1. Then the term $2\mathbf{x}'\mathbf{u}_k$ can be evaluated very efficiently by use of the Integral Image. If $\mathbf{u}_k$ is an image patch with rectangles of constant (and optionally different) grey levels then the dot product is evaluated in constant time by the addition of four pixels of the Integral Image of the input image per rectangle and one multiplication per grey level value.

2.1 Learning Haar-Like Reduced Set Vectors Using OCWT

In contrast to other approaches ([6,1]), we do not use a wavelet transformation of the input images as a pre-processing at runtime. The novelty is that we apply the over-complete wavelet transformation at the learning stage. Our approach proposes a wavelet transformation of the Reduced Support Vector Machine itself as a means to speedup the runtime performance.

2.1.1 Wavelet-Shrinkage for Haar-Like Structured Reduced Set Vectors

In order to make full usage of the concept of Integral Images, it would be desirable to approximate the computed RSV's, $\mathbf{z}$, by block-wise structured images that are not too far off while keeping the number of rectangular regions with constant grey value much smaller than in $\mathbf{z}$. Mathematically speaking, we are searching for an approximation of a given image $\mathbf{z}$ by a piecewise block structured image $\mathbf{u}$ which is as sparse as possible. This optimization problem can be casted in the following variational form

$$\min_{\hat{\mathbf{u}}} \left\{ \|\mathbf{z} - \hat{\mathbf{u}}\|^2_{L_2} + 2\alpha|\hat{\mathbf{u}}|_{B^1_1(L_1)} \right\}, \tag{1}$$

where $B^1_1(L_1)$ denotes a particluar Besov semi–norm; for an overview we refer the reader to [14,12] and for a detailed discussion of the problem to [2]. The Besov (semi) norm of a given function can be expressed by means of its wavelet coefficients and, moreover, in two dimensions the Besov penalty is nothing else than a ℓ_1 constraint on the wavelet coefficients (promoting sparsity as required).

The minimization of (1) is easily obtained: Let $\{\psi_\lambda\}_{\lambda\in\Lambda}$ be the wavelet basis, where Λ is the double index over all grid points and all scalings. Then we may express $\mathbf{z}$

and $\hat{\mathbf{u}}$ as follows: $\mathbf{z} = \sum_{\lambda \in \Lambda} z_\lambda \psi_\lambda$, $\hat{\mathbf{u}} = \sum_{\lambda \in \Lambda} \hat{u}_\lambda \psi_\lambda$, where $z_\lambda = \langle \mathbf{z}, \psi_\lambda \rangle$ and $\hat{u}_\lambda = \langle \hat{\mathbf{u}}, \psi_\lambda \rangle$ (here $\langle \cdot, \cdot \rangle$ stands for the inner product in the underlying Hilbert space). We may completely represent (1) by means of the associated wavelet coefficients,

$$\mathbf{u} = \arg \min_{\hat{\mathbf{u}}} \sum_{\lambda \in \Lambda} \left\{ (z_\lambda - \hat{u}_\lambda)^2 + 2\alpha |\hat{u}_\lambda| \right\} . \tag{2}$$

Since the wavelet basis is linearly independent, we can minimize summand–wise and obtain the following explicit expression for the optimum u_λ, see, e.g. [5],

$$u_\lambda = S_\alpha(z_\lambda) = \operatorname{sgn}(z_\lambda) \max\{|z_\lambda| - \alpha, 0\} , \tag{3}$$

where S_α is the soft–shrinkage operation with threshold α. Consequently, the optimum $\mathbf{u}$ is simply obtained by soft–shrinking the wavelet coefficients of $\mathbf{z}$, i.e. $\mathbf{u} = \sum_{\lambda \in \Lambda} S_\alpha(z_\lambda) \psi_\lambda$.

2.1.2 Over-Complete Wavelet Transformation

Typically, a wavelet representation of an image is computed by fast discrete wavelet schemes. However, non–redundant representations and filtering very often creates artifacts in terms of undesirable oscillations or non–optimally represented details, which manifest themselves as ringing and edge blurring. For our purpose, it is essential to pick a representation that optimally meets the local image structure and is not restricted to a fixed grid (see Figure 1). The most promising method for adequately solving this kind of problem has its origin in translation invariance (the method of cycle spinning, see, e.g. [3]), i.e. representing the image by all possible shifted versions of the underlying (Haar) wavelet basis. But contrary to the idea of introducing redundancy by averaging over all possible representations of $\mathbf{z}$, we aim to pick only that one which is optimally suited for our given image.

In order to give a rough sketch of this technique, assume that we are given an RSV $\mathbf{z}$ with $2^M \times 2^M$ pixel. Following the cycle–spinning approach, see again [3], we have to compute $2^{2(M+1-j_0)}$ different representations of $\mathbf{z}$ with respect to the $2^{2(M+1-j_0)}$ translates, s of the underlying wavelet basis. The scale j_0 denotes the coarsest resolution level of $\mathbf{z}$. The family $\{\mathbf{z}^s\}_s$ generated this way serves now as our reservoir of possible wavelet representations of one single $\mathbf{z}$. The best shift s^* is that one for which we have a minimal discrepancy to the SVM hyper-plane per operations for the kernel-evaluation. We evaluate all possible local shifts (in our case $s = 64$), hence the global optimum shift is guaranteed (see Section 2.1.4.4).

2.1.3 Hyper-Plane Approximation by Wavelet Shrinkage

Once we approximate the Support Vectors of the SVM by the W-RSV's, the question arises whether the hyper-plane approximation $\Psi''_{N_z} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{u}_i)$ is close to $\Psi_{N_x} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, where $\Phi : \mathcal{X} \to F$, $\mathbf{x} \mapsto \Phi(\mathbf{x})$ is the map into the best discriminating hyper-space F. (The dot product in F is computed using a kernel function: $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ [15].) Indeed, we can demonstrate that the discrepancy between the W-RVM and the RVM is upper-bounded by the L_2 distance (which is minimised by (3) with a ℓ_1 constraint on the coefficients $\hat{u}_\lambda$) of the sparse approximation $\mathbf{u}_i$ of $\mathbf{z}_i$ (The derivation is omitted here for space reasons)(Note that the fact that the RVM ($\Psi'_{N_z} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i)$) minimises $\|\Psi'_{N_z} - \Psi_{N_x}\|$ is shown in [13].):

$$\|\Psi''_{N_z} - \Psi'_{N_z}\| \leq \sigma^{-1} \sum_{i=1}^{N_z} |\beta_i| \, \|\mathbf{z}_i - \mathbf{u}_i\|. \tag{4}$$

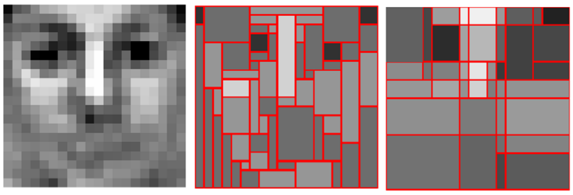

Fig. 1. Examples for Haar-like approximations of an RSV (*left*) using morphological filter (H-RSV [9], *middle*) and using an over-complete wavelet transformation (W-RSV, *right*). The OCWT representation meets optimally the local image structure. The ratio of the decreasing of the hyper-plane distance to the used operations (see Section 2.1.1.4) is more efficient for the W-RSV (0.73) than for the H-RSV (0.51).

2.1.4 Algorithm for Generation of the W-RSV's

First, the RSV's, $\mathbf{z}_i$ are computed by minimising $\|\Psi'_{N_z} - \Psi_{N_x}\|^2$ (see [10]). Then, the W-RSV's, $\mathbf{u}_i^l$, of the RSV's, $\mathbf{z}_i$ $(i = 1, \ldots, N_z)$ are computed using local best shift approximations at the level $l = 1$. The approximation at one level automatically selects the best shift and the number of wavelet basis used for this shift for all the $\mathbf{u}_i^l, i = 1, \ldots, N_z$. Once one level is computed, the residual of the previous level is approximated using the same procedure. The usage of the different approximation levels enables a smooth trade-off between accuracy and speed (see Section 2.2).

The approximation $\mathbf{u}_i^l$ of the RSV, $\mathbf{z}_i$, at the level l, is obtained by minimising the distance δ_i^l to the SVM hyper-plane with respect to β_i^l and $\mathbf{u}_i^l$:

$$\delta_i^l = \|\Psi_{i-1}^l - \beta_i^l \Phi(\mathbf{u}_i^l)\|^2, \text{ where } \Psi_{i-1}^l = \Psi_{N_z}^{l-1} - \sum_{k=1}^{i-1} \beta_k^l \Phi(\mathbf{u}_k^l), \tag{5}$$

where Ψ_{i-1}^l is the residual vector (in the feature space F) between the SVM and the classifier obtained using N_z RSV's for the levels 1 to $l-1$, and using $i-1$ RSV's for the level l. For the first level, this residual is the SVM itself: $\Psi_{N_z}^0 = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$. The formal algorithm that provides the set of β_i^l and $\mathbf{u}_i^l$ for $i = 1, \ldots, N_z$ and $l = 1, \ldots, N_l$ follows:

1. Set $\Psi_{N_z}^0 = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$ and $\forall_{i=1,\ldots,N_z} : \mathbf{r}_i^1 = \mathbf{z}_i$, where $\mathbf{z}_i$ are the Reduced Set Vectors.
2. Start at the first approximation level $l = 1$.
3. Start with the first Reduced Set Vector $i = 1$.
4. Evaluate $\forall_s : \tilde{\mathbf{u}}^s = (W^s)^{-1} S_\alpha \left(W^s \mathbf{r}_i^l\right)$ where W^s is the wavelet decomposition and $(W^s)^{-1}$ the reconstruction with a shifted wavelet basis by the two dimensional shift $s \in \left\{1, 2, \ldots, 2^J\right\} \times \left\{1, 2, \ldots, 2^J\right\}$. For a 20×20 patch size a shift $J = 3$ is sufficient. S_α is the Shrinkage function with the sparsity parameter α (see Section 2.1.1.1 and 2.2).
5. Evaluate $\forall_s : \Delta_\delta^s = \delta_{i-1}^l - \delta_i^l$ where $\delta_0^l = \delta_{N_z}^{l-1}$ and the number of operations $\Delta_\omega^s = 4 * \# [\tilde{\mathbf{u}}^s] + v(\tilde{\mathbf{u}}^s)$ where $\# [\tilde{\mathbf{u}}^s]$ is the number of piecewise constant rectangles and $v(\tilde{\mathbf{u}}^s)$ the number of grey values of $\tilde{\mathbf{u}}^s$.
6. Select the best shift s^*, for which the ratio $\frac{\Delta_\delta^s}{\Delta_\omega^s}$ is maximum.

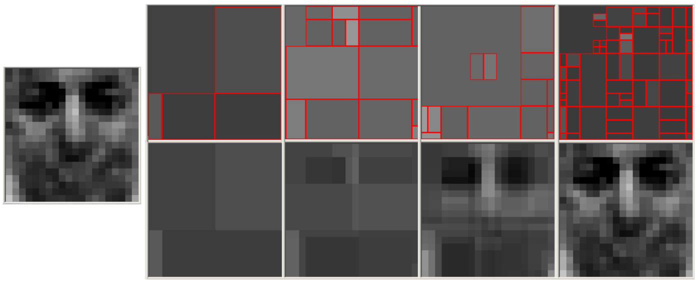

Fig. 2. Example of the approximation of a RSV (*left*), $\mathbf{z}_i$, by its W-RSV $\mathbf{u}_i^l$ at different approximation levels (*top row, left to right*: $l = 1, 2, 10, 19$). The *bottom row, (left to right)* shows the sum of the W-RSV's over the approximation levels: $\sum_{l=1}^{n} \mathbf{u}_i^l$ with $n = 1, 2, 10, 19$.

7. Set $\mathbf{u}_i^l = \tilde{\mathbf{u}}^{s^*}$ and save the rectangle structure for each approximation level of $\mathbf{u}_i^l$ separately. Then, the residual is updated: $\mathbf{r}_i^{l+1} = \mathbf{r}_i^l - \mathbf{u}_i^l$.
8. If $i \leq N_z$, increment i and proceed to step 4. If $i > N_z$ and $l \leq N_l$, increment l and proceed to step 3; else, stop.

Using this algorithm, we obtain for each RSV, $\mathbf{z}_i$, N_l levels of W-RSV's, $\mathbf{u}_i^l$ (see Figure 2 *top row*). The approximation level $l + 1$ of the W-RSV is not computed by a finer approximation of the original RSV, $\mathbf{z}_i$ (e.g. by increasing sparsity parameter α). Instead the algorithm achieves the approximation $\mathbf{u}_i^{l+1}$ from the residual $\mathbf{r}_i^{l+1} = \mathbf{z}_i - \sum_{h=1}^{l} \mathbf{u}_i^h$. Thus $\sum_{l=1}^{N_l} \mathbf{u}_i^l$ converge to $\mathbf{z}_i$ if $N_l \to \infty$ (see Figure 2 right column). We call it a local best shift method because the shift s^* is generally different for each approximation level. It is also noticed, that the rectangle structure of $\mathbf{u}_i^l$ is evaluated and stored during the training and applied at the classification process for each l separately because $\# \left[\sum_{h=1}^{l} \mathbf{u}_i^h\right] > \sum_{h=1}^{l} \# \left[\mathbf{u}_i^h\right]$. As seen in Figure 2 (*bottom row*) we obtain more rectangles, because the rectangles overlay by adding the approximations levels.

2.2 Detection Process

The classification function of the input patch $\mathbf{x}$ of the W-RVM, denoted by $y_i^l(\mathbf{x})$, using l levels and i RSV's at the level l is as follows:

$$y_i^l(\mathbf{x}) = \operatorname{sgn}\left(\sum_{h=1}^{l-1} \sum_{j=1}^{N_z^h} \beta_{h,j}^{l,i} k(\mathbf{x}, \mathbf{u}_j^h) + \sum_{j=1}^{i} \beta_{l,j}^{l,i} k(\mathbf{x}, \mathbf{u}_j^l) + b_i^l\right), \tag{6}$$

where, N_z^h, for $h = 1, \ldots, l-1$, denotes the number of RSV's used for the approximation of level h (see hereafter how to set N_z^h), b_i^l are the thresholds obtained automatically from an R.O.C. for a given accuracy. These thresholds are set to yield a given False Rejection Rate (FRR) so that the accuracy of the W-RVM is the same as the one of the full SVM (see [10] for details). The trade-off between FRR and FAR is the only parameter of our algorithm to be set by the user.

To achieve high run-time efficiency, we use a cascade of coarse-to-fine approximations of the SVM classifier. The aim is too reject as early as possible image parts that do not present the object of interest. This is performed by the following algorithm:

1. Start at the first approximation level $l = 1$.
2. Start with the first W-RSV, $\mathbf{u}_1^l$ at the level l.
3. Evaluate $y_i^l(\mathbf{x})$ for the input patch $\mathbf{x}$ using (6).
4. If $y_i^l < 0$ then the patch is classified as not being the object of interest. The evaluation stops.
5. If $i < N_z^l$, i is incremented and the algorithm proceeds to step 3; else if $l < N_l$, l is incremented and the algorithm proceeds to step 2; otherwise the full SVM is used to classify the patch.

When computing a RSV approximation of an SVM, it is not clear how many RSV's N_z should be computed (see [10]). This number of vectors may vary depending on the level of the approximation. This is why in Equation (6) the number of vectors used for the level h is denoted by N_z^h. The rationale of this dependency is that, at some point in the evaluation algorithm, it might be more efficient to increment l (and reset i), rather than to increment i. The best value of N_z^l is computed in an offline process using a validation dataset: N_z^l is set to the smallest i for which $\frac{\Delta_\omega(y_{i+1}^l)}{r(y_{i+1}^l)} > \frac{\Delta_\omega(y_1^{l+1})}{r(y_1^{l+1})}$, where $r(y_i^l)$ is the number of rejections of the negative examples obtained with i RSV's for the level l, and $\Delta_\omega(y_{i+1}^l)$ is the number of operations required to evaluate y_{i+1}^l (see Section 2.1.4).

By a similar evaluation the last used approximation level, N_l can be achieved. For this $N_l = l$ it is more efficient to classify the last few remaining patches by the SVM, instead of incrementing l. How many levels this are depends also on the sparsity parameter α of the OCWT. The smaller is α, the closer $\mathbf{u}_i^l$ is from $\mathbf{z}_i$ and the less approximation levels are required. However, the number of levels does not play a decisive role as the higher N_l, the sooner the evaluation process selects the next level, i.e. the less N_z^l. Therefore our proposed approach is not very sensitive to the parameter for setting the approximation accuracy (e.g. α), opposite to former methods using only one approximation level.

3 Experimental Results

We applied our novel over-complete wavelet approximated SVM to the task of face detection. The training set includes 3500, 20×20, face patches and 20000 non-face patches and, the validation set, 1000 face patches, and 100,000 non-face patches. The SVM computed on the training set yielded about 8000 Support Vectors that we approximated by $N_z = 90$ W-RSV's at $N_l = 5$ approximation levels by the method detailed in the previous section.

The first graph on Figure 3 plots the residual distance of the RVM (dashed line) and of the W-RVM (plain line) to the SVM (in terms of the distance $\Psi_{N_x} - \Psi''_{N_z}$) as a function of the number of vectors used, N_z. It can be seen that for a given accuracy more Wavelet Approximated Set Vectors are needed to approximate the SVM than for the RVM. However, as shown on the second plot, for a given computational load, the W-RVM rejects much more non-face patches than the RVM. This explains the improved

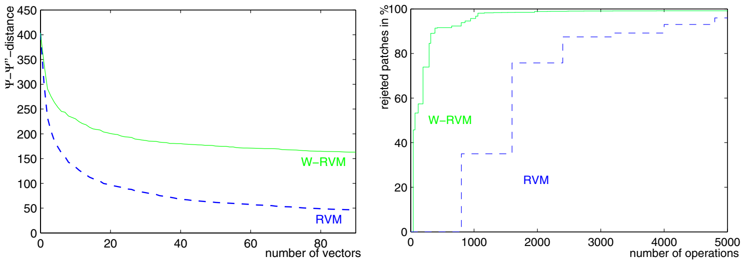

Fig. 3. *Left:* $\Psi_{N_x} - \Psi''_{N_z}$ distance as function of the number of vectors for the RVM (*dashed line*), and the W-RVM (*solid line*). *Right:* Percentage of rejected non-face patches as a function of the number of operations required.

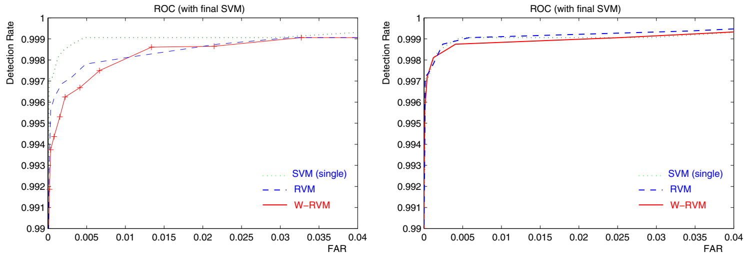

Fig. 4. R.O.C.'s for the SVM, the RVM and the W-RVM (*left*) without and (*right*) with the final SVM classification for the remaining patches. The FAR is related to non-face patches.

run-time performances of the W-RVM. Additionally, it can be seen that the curve is more smooth for the W-RVM, hence a better trade-off between accuracy and speed can be obtained by the W-RVM.

Figure 4 shows the R.O.C.'s, computed on the validation set, of the SVM, the RVM and the W-RVM. It can be seen that the accuracies of the three classifiers are similar without (left plot) and almost equal with the final SVM classification for the remaining patches (right plot), see step 5. of the evaluation algorithm. Table 1 compares the accuracy and the average time required to evaluate the patches of the validation set. The speed-up over the former approach [9] is about a factor 2.5 ($3.85\mu s$). The novel W-RVM algorithms provides a significant speed-up (530-fold over the SVM and more than 15-fold over the RVM), for no substantial loss of accuracy.

We also proved the performance and detection accuracy under real life conditions in the "Institut für Techno- und Wirtschaftsmathematik" (ITWM) in Kaiserslautern. To demonstrate the fast and accurate detection algorithm, we implemented an application using a small webcam. Accurate face detection one obtained at 25 fps (on a Intel Pentium M Centrino 1600 CPU, at a resolution of 320x240, stepsize 1 pixel, 5 scales).

Table 1. Comparison of accuracy and speed improvement of the W-RVM to the RVM and SVM

method	FRR	FAR	time per patch
SVM	1.4%	0.002%	787.34μs
RVM	1.5%	0.001%	22.51μs
W-RVM	1.4%	0.002%	1.48μs

4 Conclusion

In this paper, we presented a novel efficient method for SVM classifications on image based vectors. We used an over-complete wavelet transformation of the Reduced Set Vectors. It was demonstrated on the task of face detection.

As opposed to the RVM, the sparseness of operations required for classification is not only controlled by the number of Reduced Set Vectors but also by the number of wavelets basis functions used to approximate a Reduced Set Vector. Hence, negative examples can be rejected with much fewer number of operations, making the run-time algorithm very efficient. Moreover, as the Haar wavelets are used, the SVM kernel may be evaluated extremely efficiently using Integral Images. The main advantage of this algorithm compared to other algorithm based on boosting, such as the Viola & Jones detector [16], is the fact that the training is much faster and does not require manual intervention.

References

1. G. Zikos C. Garcia and G. Tziritas. Face detection in color images using wavelet packet analysis. *IEEE Int. Conf. on Multimedia Computing and Systems*, 1999.
2. A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear Approximation and the Space $BV(\mathbb{R}^2)$. *American Journal of Mathematics*, (121):587–628, 1999.
3. R.R. Coifman and D. Donoho. Translation–invariant de–noising. in *Wavelets and Statistics, A. Antoniadis and G. Oppenheim,* eds., Springer–Verlag, New York, pages 125–150, 1995.
4. F. Crow. Summed-area tables for texture mapping. *In Proc. of SIGGRAPH*, 18(3):207 – 212, 1984.
5. I. Daubechies and G. Teschke. Variational image restoration by means of wavelets: simultaneous decomposition, deblurring and denoising. *Applied and Computational Harmonic Analysis*, 2005.
6. D.A. Karras. Improved defect detection in textile visual inspection using wavelet analysis and support vector machines. *International Journal on Graphics, Vision and Image Processing*, 2005.
7. 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.
8. W. Kienzle, G. H. Bakir, M. O. Franz, and B. Schölkopf. Efficient approximations for support vector machines in object detection. *Proc. DAGM'04*, pages 54 – 61, 2005.
9. M. Rätsch, S. Romdhani, and T. Vetter. Efficient face detection by a cascaded support vector machine using haar-like features. *Proc. DAGM'04: 26th Pattern Recognition Symposium*, pages 62 – 70, 2005.
10. S. Romdhani, P. Torr, B. Schölkopf, and A. 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.-J. Schmeisser and H. Triebel. *Topics in Fourier Analysis and Function Spaces*. John Wiley and Sons, New York, 1987.
13. 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 TNN*, 10(5):1000 – 1017, 1999.
14. H. Triebel. *Interpolation Theory, Function Spaces, Differential Operators*. Verlag der Wissenschaften, Berlin, 1978.
15. V. Vapnik. *Statistical Learning Theory*. Wiley, N.Y., 1998.
16. 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.