Support Vector Machines for Face Detection

A. Fazekas, I. Buciu, C. Kotropoulos, and I. Pitas

Department of Informatics, Aristotle University of Thessaloniki
GR-54006 Thessaloniki Box 451, Greece
fattila@math.klte.hu,{nelu,costas,pitas}@zeus.csd.auth.gr

Abstract. In this paper we are going to construct a new kernel function, which is based on Walsh functions, for support vector machines. We prove some theoretical results about the VC-dimension of the support vector machines which are built in the space of the Walsh functions. This information is very important, because the learning capability of the support vector machines depends on the VC-dimension of the kernel function used. The paper also proposes the application of majority voting on the output of several support vector machines in order to select the most suitable learning machine for frontal face detection. The paper also reports the first experimental results related to application of support vector machines with Walsh kernel and majority voting technique to face detection.

1 Introduction

The Bayes likelihood ratio test yields the optimal classifier in the sense that it minimizes the probability of error [4]. However in order to construct the likelihood ratio, the conditional probability density function (pdf) for each class must be known. Although, there are several procedures for estimating a pdf from a finite number of observations [4], the problem of density estimation is ill-posed [10]. An alternative method to solve a two-class pattern recognition problem is to resort to example-based techniques, such as the support vector machines (SVM) [10].

SVM implement the following idea [10]: By mapping the input pattern vectors, which are the elements of the training set, onto a high-dimensional feature space through an a priori suitably chosen mapping, we expect that the elements of the training set will be linearly separable in the feature space. We construct the optimal separating hyperplane in the feature space (as is explained in Section 2) to get a binary decision whether the input vector belongs to a given class or not. For example, in the case of the application studied in the paper, face detection, the input vector comprises gray levels of pixels from a rectangular region of the digital image and the result of the binary decision is the answer whether this region is a face or not.

In general, the determination of the separating hyperplane is not easy, because the dimensionality of the feature space is high. However, in Hilbert spaces, one can estimate the inner product of two vectors in the feature space as a function of the inner product of the two vectors in the input space. These expressions for inner products are referred as kernel functions. Some kernel functions are well-known, for example the polynomial, the radial, the sigmoid, etc. [1, 2].

In this paper we will construct a new kernel function that is based on the 2-dimensional Walsh system. The 2-dimensional Walsh system is useful in pattern recognition [3]. Because the VC-dimension is the capacity factor of the SVM [11], its knowledge is very important in order to control its behavior. In this paper we prove several propositions about the VC-dimension of the class of the 2-dimensional Walsh functions. Finally, the paper reports experimental results in order to assess the properties of SVMs with Walsh kernels in face detection.

We will give an alternative approach instead of constructing a new kernel function to make a better face detector. We propose to rank an ensemble of SVMs trained on the same set by combining their outputs with majority voting in the decision making process. By doing so, we can define the most efficient SVM, i.e., the one whose outputs appear most frequently in the set of the outputs produced by the ensemble of SVMs. We apply this technique to frontal face detection and report a significant reduction of false acceptance rate.

The structure of the paper is the following: Section 2 and Section 3 are brief overviews of the principles of SVMs and the Walsh system, respectively. Section 4 explains the construction of the 2-dimensional Walsh kernel. Section 5 describes the theoretical results on the VC-dimension of the class of the Walsh functions. Finally, Section 6 presents the promising experimental results obtained with the proposed system of SVMs when applied to the face detection. The application of majority on the output of several support vector machines is explained in Section 7 and its experimental results are reported in Section 8.

The following notation is used throughout the paper. Bold face symbols will indicate vectors or matrices, and normal typeface will be used for vector and matrix elements as well as for scalars. We will denote the set of non-negative integers by $\mathbb{N}$, the set of positive integers by $\mathbb{N}^+$, the set of integers by $\mathbb{Z}$, and the set of real numbers by $\mathbb{R}$.

2 Support Vector Machines

SVMs are learning algorithms based on the statistical learning theory [11]. In this section we briefly describe the foundation of SVMs by [11]. Statistical learning from examples aims at selecting from a given set of functions $\{f_\alpha(\mathbf{x}) \mid \alpha \in \Lambda\}$, the one which predicts best the correct response (i.e. the response of a supervisor). This selection is based on the observation of l pairs that build the training set:

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l), \qquad \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{+1, -1\} \tag{1}$$

which contains input vectors $\mathbf{x}_i$ and the associated ground "truth" y_i given by an external supervisor.

Let the response of the learning machine $f_\alpha(\mathbf{x})$ belongs to a set of indicator functions $\{f_\alpha(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^m, \alpha \in \Lambda\}$ (see Definition 7 subsequently). If we define the loss-function:

$$L(y, f_\alpha(\mathbf{x})) = \begin{cases} 0, \text{if } y = f_\alpha(\mathbf{x}), \\ 1, \text{if } y \neq f_\alpha(\mathbf{x}) \end{cases} \tag{2}$$

that measures the error between the ground truth y to a given input $\mathbf{x}$ and the response $f_\alpha(\mathbf{x})$ provided by the learning machine, the expected value of the loss is given by:

$$R(\alpha) = \int L(y, f_\alpha(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy \tag{3}$$

where $p(\mathbf{x}, y)$ is the joint probability density function of random variables $\mathbf{x}$ and y. $R(\alpha)$ is called the expected risk. We would like to find the function $f_{\alpha_0}(\mathbf{x})$ which minimizes the risk functional $R(\alpha)$. The selection of the function is based on the training set of l random independent identically distributed observations (1). In order to minimize the risk functional $R(\alpha)$ the empirical risk minimization (ERM) induction principle is usually employed by replacing the expected risk functional $R(\alpha)$ by the empirical risk functional, which is measured on the training set:

$$R_{\mathrm{emp}}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} L(y_i, f_\alpha(\mathbf{x}_i)). \tag{4}$$

The idea underlying the ERM induction principle is to approximate the function $f_{\alpha_0}(\mathbf{x})$ which minimizes $R(\alpha)$ by the function $f_{\alpha_l}(\mathbf{x})$ which minimizes the empirical risk. This approach may be valid for training sets having large size (ideally infinite). It is known that for some η such that $0 \leq \eta \leq 1$, the expected risk is bounded for arbitrary $\alpha \in \Lambda$ with probability $1 - \eta$ [10]:

$$R(\alpha) \leq R_{\mathrm{emp}}(\alpha) + \sqrt{\frac{h(\log(\frac{2l}{h}) + 1) - \log(\frac{\eta}{4})}{l}}, \tag{5}$$

where h is a non-negative integer called the Vapnik-Chervonenkis (VC) dimension (see subsequent Definition 9), and is a measure of capacity of SVMs. In the following, we shall call the right hand side of inequality (5) the risk bound. The second term in the risk bound is called the VC-confidence. Inequality (5) reveals the necessity to minimize both the empirical risk and the VC-confidence. This is the aim of the so-called structural risk minimization (SRM) principle. The SVMs are learning machines that implement the SRM principle in their training.

In order to introduce the basic idea of SVMs, let us consider the construction of the optimal separating hyperplane. Suppose the training data (1) can be separated by a hyperplane, that is $\exists \mathbf{v} \in \mathbb{R}^m$:

$$(\mathbf{v}^{\mathrm{T}}\mathbf{x}_i) + b \geq 1, \qquad \text{if } y_i = +1 \tag{6}$$

$$(\mathbf{v}^{\mathrm{T}}\mathbf{x}_i) + b \leq -1, \qquad \text{if } y_i = -1, \tag{7}$$

where $\mathbf{v}$ is a normal to the hyperplane, $\frac{|b|}{||\mathbf{v}||}$ is the perpendicular distance from the hyperplane to the origin, and $||\mathbf{v}||$ is the Euclidean norm of $\mathbf{v}$. A compact notation for inequalities (6) and (7) is:

$$y_i \left((\mathbf{v}^{\mathrm{T}}\mathbf{x}_i) + b\right) \geq 1, \qquad i = 1, \ldots, l. \tag{8}$$

Let d_+ (d_-) be the Euclidean distance from the separating hyperplane to the closest positive (negative) example. Define the margin of the separating hyperplane to be $d_+ + d_-$. For the linearly separable case, SVM simply seeks for the separating hyperplane with the largest margin. The optimal hyperplane minimizes

$$\frac{1}{2}\|\mathbf{v}\|^2 \quad \text{subject to the inequalities (8).} \tag{9}$$

After training a support vector machine, one simply determines the side of the decision boundary where a given test pattern $\mathbf{x}$ lies on and assigns the corresponding class label, i.e., $\theta(\mathbf{v}^{\mathrm{T}}\mathbf{x} + b)$.

Let us investigate the generalization to the case where the decision function is not a linear function of the input vector [11]. Now suppose we have first mapped the data to some other Euclidean space $\mathcal{H}$, using a mapping Φ:

$$\Phi : X \to \mathcal{H}. \tag{10}$$

Then the training algorithm would only depend on the data through inner products in $\mathcal{H}$, i.e., on functions of the form $\Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$. If there were a kernel function K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$, we would only need to use $K(\mathbf{x}_i, \mathbf{x}_j)$ in the training algorithm without necessarily explicitly knowing $\Phi(\mathbf{x})$.

How can we use this machine? After all, we need $\mathbf{v}$ that will be in $\mathcal{H}$ as well. The solution of the optimization problem (9) yields a coefficient vector $\mathbf{v}$ that is expressed as a linear combination of a subset of training vectors, whose associated Lagrange multipliers are non-zero. These training vectors are called support vectors. In our case we will denote the images of the support vectors $\mathbf{s}_i$, $\Phi(\mathbf{s}_i)$. In the test phase, an SVM computes the sign of:

$$f(\mathbf{x}) = \theta\left(\sum_{i=1}^{N_S} \lambda_i y_i \Phi(\mathbf{s}_i)^{\mathrm{T}}\Phi(\mathbf{x}) + b\right) \tag{11}$$

where λ_i are the Lagrange multipliers that are associated to $\Phi(\mathbf{x}_i)$, and N_s is the number of the support vectors. Again, we can avoid computing $\Phi(\mathbf{s}_i)^{\mathrm{T}}\Phi(\mathbf{x})$ explicitly and use $K(\mathbf{s}_i, \mathbf{x}) = \Phi(\mathbf{s}_i)^{\mathrm{T}}\Phi(\mathbf{x})$.

3 The Walsh System

In the literature the term “Walsh functions” refers to one of three orthonormal systems: the Walsh-Paley system, the original Walsh system, or the Walsh-Kaczmarz system [9]. These systems contain the same functions and differ only in enumeration. We will investigate the Walsh-Paley system which will be referred as the Walsh system henceforth. For more details the interested reader may consult [9]. In the following, $n, m \in \mathbb{N}$.

Definition 1. *Let $r(x)$ be the function defined on $[0,1)$ by:*

$$r(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}), \\ -1, & x \in [\frac{1}{2}, 1) \end{cases}$$

extended to $\mathbb{R}$ periodically with period 1. The Rademacher system $R = \{r_n(x)\}$ is defined by:

$$r_n(x) = r(2^n x), \qquad x \in \mathbb{R}, n \in \mathbb{N}.$$

Definition 2. *Given $n \in \mathbb{N}$, it is possible to write n uniquely as:*

$$n = \sum_{k=0}^{\infty} n[k] 2^k,$$

where either $n[k] = 0$ or 1 for $k \in \mathbb{N}$. This expression will be called the binary expansion of n and the numbers $n[k]$ will be called the binary coefficients of n.

Definition 3. *Let x be an arbitrary element of the interval $[0,1)$. If x has the form $\frac{p}{2^n}$ for some p $(0 \leq p < 2^n)$, x will be a dyadic rational in the interval $[0,1)$.*

Definition 4. *Any $x \in [0,1)$ can be written in the form:*

$$x = \sum_{k=0}^{\infty} x[k] 2^{-(k+1)},$$

where each $x[k]$ is equal to either 0 or 1. We will call it the dyadic expansion of x. When x is a dyadic rational there are two expressions of this form, one which terminates in 0's and one which terminates in 1's. In this paper, we will confine ourselves to the dyadic expansion of x the one that terminates in 0's.

Definition 5. *The Walsh system $W = \{w_n(x)\}$ is product of Rademacher functions in the following way. If $n \in \mathbb{N}$ has binary coefficients $\{n[k], k \in \mathbb{N}\}$ then:*

$$w_n(x) = \prod_{k=0}^{\infty} r_k^{n[k]}(x).$$

It is easy to see that this product is always finite, $w_0(x) = 1$ and $w_{2^n}(x) = r_n(x)$ $\forall x \in [0,1)$. It is worth noticing that each Walsh function is piecewise constant with finitely many jump discontinuities on $[0,1)$, and takes only the values of either $+1$ or -1.

Definition 6. *The 2-D Walsh system $W^{(2)} = \{w_{(n,m)}(x,y)\}$ is defined in the following way:*

$$w_{(n,m)}(x,y) = w_n(x) \cdot w_m(y).$$

4 The Construction of the Walsh Kernel

It is well-known, that the Walsh system is a complete orthonormal system on $[0, 1)$ and the Walsh system is a Schauder basis in L^p for $1 < p < \infty$ [9]. This linear space is a Hilbert space, where the inner product, denoted by $\langle \cdot, \cdot \rangle$ is simply the integral of the product of two functions $< p(x), q(x) >= \int_0^1 p(\xi)q(\xi)d\xi$ [9]. Walsh functions have already been used in image analysis [3].

In this section we define a new kernel function for support vector machines. The construction is based on Vapnik's idea [11]. Let us suppose that we would like to analyze an one-dimensional signal in terms of Walsh functions. Let us map the input variable x into the N-dimensional feature space, as follows:

$$\mathbf{\Phi}_{(N)}(x) = \left(\frac{1}{\sqrt{2^0}} w_o(x), \frac{1}{\sqrt{2^1}} w_1(x), \ldots, \frac{1}{\sqrt{2^{N-1}}} w_{N-1}(x) \right)^{\mathrm{T}}. \tag{12}$$

The inner product of two vectors in this space has the form:

$$K_N(x, x') = \mathbf{\Phi}_{(N)}(x)^{\mathrm{T}} \mathbf{\Phi}_{(N)}(x') = \sum_{k=0}^{N-1} \frac{1}{2^k} w_k(x) w_k(x'). \tag{13}$$

It is convenient to assume that $N = 2^n$, $n \in \mathbb{N}$. The relations stem from the theoretical characteristics of the Walsh functions [9]. In the case of digital images, the input variable x is the gray level of an arbitrary image pixel. Let us suppose x is normalized to $[0, 1)$. If we were to linearly combine the elements of (12), then the weights of the linear combination would be chosen so that linear separability was maintained in an ideal point-wise SVM. The required inner product computations are then replaced by bit-wise operations that can be executed fast on either dedicated hardware or software.

To construct the SVM for the 2-dimensional vector space $X = \{\mathbf{x} \mid \mathbf{x} = (x, y)^{\mathrm{T}}\}$, it is sufficient to use the generating kernel that is a product of one-dimensional kernels:

$$K^2_{(N,M)}(\mathbf{x}, \mathbf{x}') = K_N(x, x') \cdot K_M(y, y'). \tag{14}$$

The calculation of the mapping (12) is not a difficult task, because we can use the idea of fast Walsh transformation [5].

5 The VC-dimension of the Class of the 2-D Walsh Functions

The Vapnik-Chervonenkis dimension has a very important role in the statistical learning. The VC-dimension of the support vector machines characterizes the learning capacity of the machine. With control of the VC-dimension one can avoid overfitting of the support vector machines and one can minimize the expected value of the error [11]. So the knowledge of the VC-dimension of the class of the functions employed in the learning algorithm is very important.

At first we quote some definitions from [11], which are important to understand the theoretical results of this section.

Definition 7. *An arbitrary* $\{+1,-1\}$*-valued function with domain* $\mathbb{R}^2$ *is called an 2-D indicator function.*

Definition 8. *Let* f *be an arbitrary 2-D indicator function. The sets* $\{(x,y) \mid f(x,y) = +1,\ x,y \in \mathbb{R}\}$ *and* $\{(x,y) \mid f(x,y) = -1,\ x,y \in \mathbb{R}\}$ *are the separated classes of the domain by using* f*.*

Definition 9. *The VC-dimension of a set of 2-D indicator functions is equal to the largest number* h *of points of the domain of the functions that can be separated into two different classes in all the* 2^h *possible ways using a function from this set. If for any* m *there exists a set of* m *points that can be shattered by the functions of the set, then the VC-dimension is equal to infinity.*

In the rest of the paper we assume that the domain of 2-D Walsh functions and of the 2-D indicator functions is the set $[0,1)^2$.

Theorem 1. *The VC-dimension of the class of all the 2-D Walsh functions is equal to* ∞*.*

The relations $N = 2^n$, $M = 2^m$ and $n, m \geq 2$ are assumed in the following.

Theorem 2. *The VC-dimension of the set* $W_{(N,M)} = \{w_{(k_1,k_2)}(x,y) \mid k_1 = 0,\ldots,N-1, k_2 = 0,\ldots,M-1\}$ *is equal to* $\log_2(N \cdot M)$*.*

Definition 10. *Let* $G = \{x[n] \mid n \in \mathbb{N}, x[n] = 1$ *or* $x[n] = 0\}$*. Set* $I_0(x) = G$ *for all* $x \in G$*. For each* $x \in G$ *and* $n \in \mathbb{N}^+$ *define:*

$$I_n(x) = \{y \in G \mid y[i] = x[i], 0 \leq i < n\}.$$

We call the sets $I_n(x)$ *the dyadic intervals of order* n *in* $[0,1)$*.*

Definition 11. *By a dyadic step function of order* n *we mean a finite linear combination of characteristic functions of dyadic intervals of order* n *in* $[0,1)$*.*

Definition 12. *By a 2-D dyadic step function of order* (n,m) *we mean a product of two dyadic step functions of orders* n *and* m*, respectively.*

We use the following notation:

$$\theta(x) = \begin{cases} 1, \text{if } x >= 0, \\ -1, \text{if } x < 0, \end{cases}$$

where $x \in \mathbb{R}$.

Definition 13. *Let* $f_\alpha(x,y)$ *be* $\mathbb{R}$*-valued functions. We call the set of indicator functions* $\theta(f_\alpha(x,y) - t)$*, where* $t \in \Big(\inf_{(x,y)} f_\alpha(x,y), \sup_{(x,y)} f_\alpha(x,y)\Big)$*, the set of indicators for functions* $f_\alpha(x,y)$*.*

Theorem 3. *The VC-dimension of the set* $lin(W_{(N,M)}) = \{f(x,y) \mid f(x,y) = \alpha_{(0,0)} w_{(0,0)}(x,y) + \ldots + \alpha_{(0,M-1)} w_{(0,M-1)}(x,y) + \ldots + \alpha_{(N-1,M-1)} w_{(N-1,M-1)},$ $\alpha_{(i,j)} \in \mathbb{R}\}$ *is equal to* $N \cdot M$*.*

Corollary 1. *The VC-dimension of the kernel function $K_{(N,M)}$ is equal to $N \cdot M$.*

The theoretical developments presented in this section lead to the construction of an SVM that employs the values admitted by the Walsh basis functions as elements of the training vectors and resorts to coefficients of the linear combination that solve a two-class pattern recognition. The latter coefficients are **not** the ones given by the analysis equation of the Walsh transformation.

6 Experimental Results about Walsh SVMs

The purpose of the designed experiments is to assess the performance of Walsh SVMs in face detection. A training data set of 112 images, 57 images containing a face and another 55 images with non-face patterns, is built. The images containing face patterns have been derived from the face database of IBERMATICA where several sources of degradation are modeled. For a description of this database the interesting reader may refer to [8].

All images in this database are recorded in 256 grey levels and they are of dimensions 320×240. The procedure for collecting face patterns is as follows. From each image a bounding rectangle of dimensions 128×128 pixels has been manually determined that includes the actual face. This area has been subsampled four times. At each subsampling, non-overlapping regions of 2×2 pixels are replaced by their average. Accordingly, training patterns of dimensions 8×8 are built. The ground truth, that is, the class label $+1$ has been appended to each pattern. Similarly, 55 non-face patterns have been collected from images in the same way, and labeled by -1.

We have trained the three different SVMs indicated in Table 1. For all experiments the `SVMLight` toolbox developed by T. Joachims was used [7]. The trained SVMs have been applied to 442 test images (249 faces and 193 non-faces) from the IBERMATICA database that have not been included in the training set. The resolution of each test image has been reduced four times yielding a final image of dimensions 8×8. The test images are classified as either non-face or face ones.

Table 1 summarizes the results of the test. The first row in Table 1 depicts the cpu time needed for test experiments using each kernel functions for a Pentium III at 543 MHz. As can be seen, the support vector machine that is based on the Walsh function requires more time than the other SVMs. This is attributed to the high dimension of the feature space. The numbers of errors shown in the second row are the misclassification errors, that is either the number of real faces classified as non-faces (number of false rejections) or the number of non-face instances classified as faces (number of false acceptances). From this point of view the Walsh kernel outperforms the other two competitors. Finally, the number of support vectors in the case of Walsh kernel is larger than that of the linear and the polynomial kernels.

Table 1. Experimental Results

	Linear	Walsh	Polynomial with degree 2
Time (sec)	1.78	7.62	5.32
Number of the False Acceptances	18	**0**	2
Number of the False Rejections	11	**0**	1
Number of SVs	24	74	14

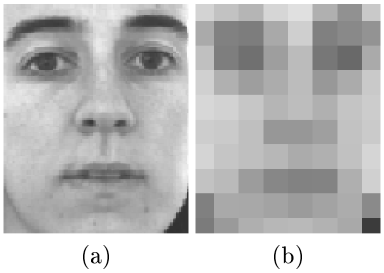


Fig. 1. Correctly detected face in the (a) original, and in the (b) subsampled resolution.

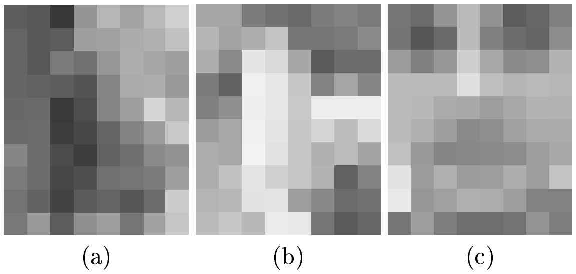


Fig. 2. False accepted cases (a)-(b), and false rejected case (c) in the subsampled resolution in the case of linear and polynomial SVM.

7 Majority Vote Applied in the case of the Outputs of Several SVMs

Let us consider five SVMs were based on the several kernels as follows: (1) Linear kernel; (2) Polynomial kernel with degree 2; (3) Gaussian Radial Basis Function (GRBF) having σ equal to 10; (4) Sigmoid with $k = 0.5$ and $\theta = 0.2$; (5) Exponential Radial Basis Function (ERBF) having σ equal to 10. The penalty was set to 500. These kernels have the analytical form listed in Table 2, where $||\cdot||_p$ denotes the vector p-norm, $p = 1, 2$.

For brevity, we index each SVM by k, $k = 1, \ldots, 5$. To distinguish between training and test patterns, the latter ones are denoted by $\mathbf{z}_i$. Let $\mathcal{Z}$ be the test

Table 2. Kernel functions used in SVMs.

k	SVM type	Kernel function $K(\mathbf{x},\mathbf{y})$
1	Linear	$\mathbf{x}^T\mathbf{y}$
2	Polynomial with degree q	$(\mathbf{x}^T\mathbf{y}+1)^q$
3	GRBF	$\exp(-\frac{\vert\vert\mathbf{x}-\mathbf{y}\vert\vert_1^2}{2\sigma^2})$
4	Sigmoid	$\tanh(k\cdot\mathbf{x}^T\mathbf{y}-\theta)$
5	ERBF	$\exp(-\frac{\vert\vert\mathbf{x}-\mathbf{y}\vert\vert_1}{2\sigma^2})$

set. We define the two values of the histogram of labels assigned to all $\mathbf{z}_i \in \mathcal{Z}$ as:

$$\begin{aligned} h_1(\mathbf{z}_i) &= |\{k \mid f_k(\mathbf{z}_i) = 1, k = 1, \ldots, 5\}|, \\ h_{-1}(\mathbf{z}_i) &= |\{k \mid f_k(\mathbf{z}_i) = -1, k = 1, \ldots, 5\}|. \end{aligned} \tag{15}$$

The we may combine the decision taken separately by the SVMs indexed by $k = 1, \ldots, 5$ in the following manner:

$$g(\mathbf{z}_i) = \begin{cases} 1, \text{if } h_1(\mathbf{z}_i) > h_{-1}(\mathbf{z}_i), \\ -1, \text{otherwise.} \end{cases} \tag{16}$$

Let us now define the quantities:

$$\begin{aligned} F_k &= |\{\mathbf{z}_i \mid f_k(\mathbf{z}_i) = 1, \mathbf{z}_i \in \mathcal{Z}\}|, \\ G_k &= |\{\mathbf{z}_i \mid g(\mathbf{z}_i) = 1, \text{ and } f_k(\mathbf{z}_i) = 1, \mathbf{z}_i \in \mathcal{Z}\}|. \end{aligned} \tag{17}$$

To determine the best SVM, we simply choose:

$$k' = \arg\max_k \frac{G_k}{F_k}. \tag{18}$$

If more than one indices are determined by (18) we count additionally the number of consistent classifications $|\{\mathbf{z}_i \mid f_{k'}(\mathbf{z}_i) = g(\mathbf{z}_i), \mathbf{z}_i \in \mathcal{Z}\}|$, and we resolve the tie by selecting the index that maximizes that number.

8 Experimental Results about Majority Voting Technique

We have trained the five different SVMs indicated in Table 2. The trained SVMs have been applied to six test imaged from the IBERMATICA database that were not included in the training set. Each test image corresponds to a different person. The resolution of images has been decreased four time using a pyramidal algorithm, the final size of the image being of the dimension 15 × 20. Scanning row by row the reduced image by a 10×8 rectangular window, test patterns are classified as non-face ones or face pattern. When a face pattern is found by the

Table 3. Ratio G_k/F_k achieved by the various SVMs.

SVM type	Test Images numbers					
	1	2	3	4	5	6
Linear	0.83	0.20	0.57	0.66	**1.00**	0.74
Polynomial	0.52	0.28	0.57	0.44	**1.00**	0.71
GRBF	0.67	0.25	0.44	0.44	0.80	0.83
Sigmoid	0.64	0.14	0.15	0.11	0.22	0.13
ERBF	**1.00**	**0.50**	**0.80**	**0.80**	0.80	**1.00**

machine, a rectangle is drawn for locating the face. We have tabulated the ratio G_k/F_k in Table 3.

From this table it can be seen that ERBF is found to maximize the ratio in (18) for the five test images. On the contrary the machine built using the sigmoid kernel attains the worst performance with respect to (18). Two quantities measurements have been used for performing the performance of each SVM, namely, the false acceptance rate (FAR) and the false rejection rate (FRR) during the test phase. We have measured FAR and FRR for each individual SVM before and after applying the majority vote procedure. FRR is always zero, meaning that each machine is able to detect faces. Instead FAR varies. The values of FAR attained by each SVM individually and after applying majority vote are shown in Table 4.

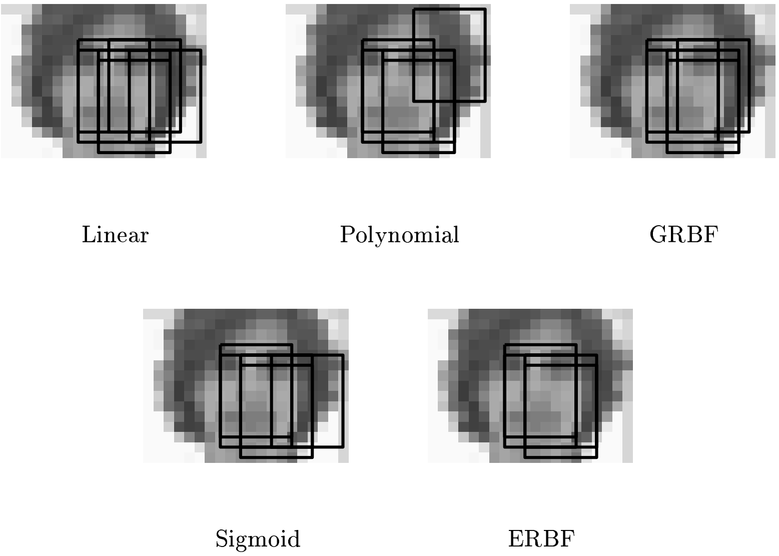
Table 4. False acceptance rates (in %) achieved by the various SVMs individually and after applying the majority vote.

SVM type	Test Images numbers					
	1	2	3	4	5	6
Linear	3.9	10.5	6.5	5.2	2.6	6.5
Polynomial	6.5	6.5	6.5	9.2	2.6	6.5
GRBF	5.2	7.8	9.2	9.2	3.9	5.2
Sigmoid	7.8	17.1	31.5	44.7	21.0	47.3
ERBF	2.6	2.6	3.9	3.9	3.9	3.9
Combining	2.6	1.3	2.6	2.6	2.6	3.9

One can see that the application of majority voting reduces the number of false acceptances in all cases and particularly when $F_k \neq G_k$.

Figure 3 and Figure 4 depict face localization determined by the five SVM types along with the face localization using majority vote.

It is seen that majority vote helps to discard many of the candidate face regions returned by single SVMs (Fig.3) yielding the best face localization (Fig.4).



Linear Polynomial GRBF

Sigmoid ERBF

Fig. 3. Face localization in a test image for individual SVMs.

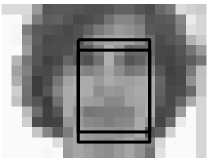


Fig. 4. More accurate face localization after using majority vote procedure.

Acknowledgements

This work was supported by the European Union Research Training Network "Multi-modal Human-Computer Interaction (HPRN-CT-2000-00111)".

References

1. C.J.C. Burges. A tutorial on support vector machines for pattern recognition, *Knowledge Discovery and Data Mining*, **2**, 1998.
2. L. Devroye, L. Györfi and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York, 1996.

3. A. Fazekas and A. Hajdu. Recognizing typeset documents using Walsh transformation, *Journal of Computing and Information Technology*, Vol. **9**, pp. 101–112, 2001.
4. K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, 1990.
5. R.C. Gonzalez and R.E. Woods. *Digital Image Processing.* Addison-Wesley, New York, 1993.
6. S. Gunn. Support vector machines for classification and regression, *ISIS Technical Report ISIS-1-98*, Image Speech Intelligent System Research Group, University of Southapton, 1998.
7. T. Joachims. Making large-scale SVM learning practical, in *Advances in Kernel Methods - Support Vector Learning*, pp. 169–184, MIT Press, Cambridge, MA, 1998.
8. C. Kotropoulos, A. Tefas, and I. Pitas. Morphological elastic graph matching applied to frontal face authentication under well-controlled and real conditions, *Pattern Recognition* **33**, 31–43, 2000.
9. F. Schipp, W.R. Wade, P. Simon and J. Pál. *Walsh Series: An Introduction to Dyadic Harmonic Analysis.* Adam Hilger, Budapest, 1990.
10. V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.
11. V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.