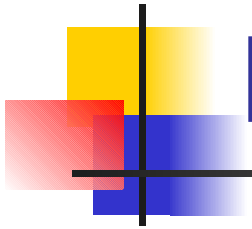
Training Support Vector Machines: an Application to Face Detection

Edgar Osuna, Robert Freund, Federico Girosi

Presented by: Amrudin Agovic



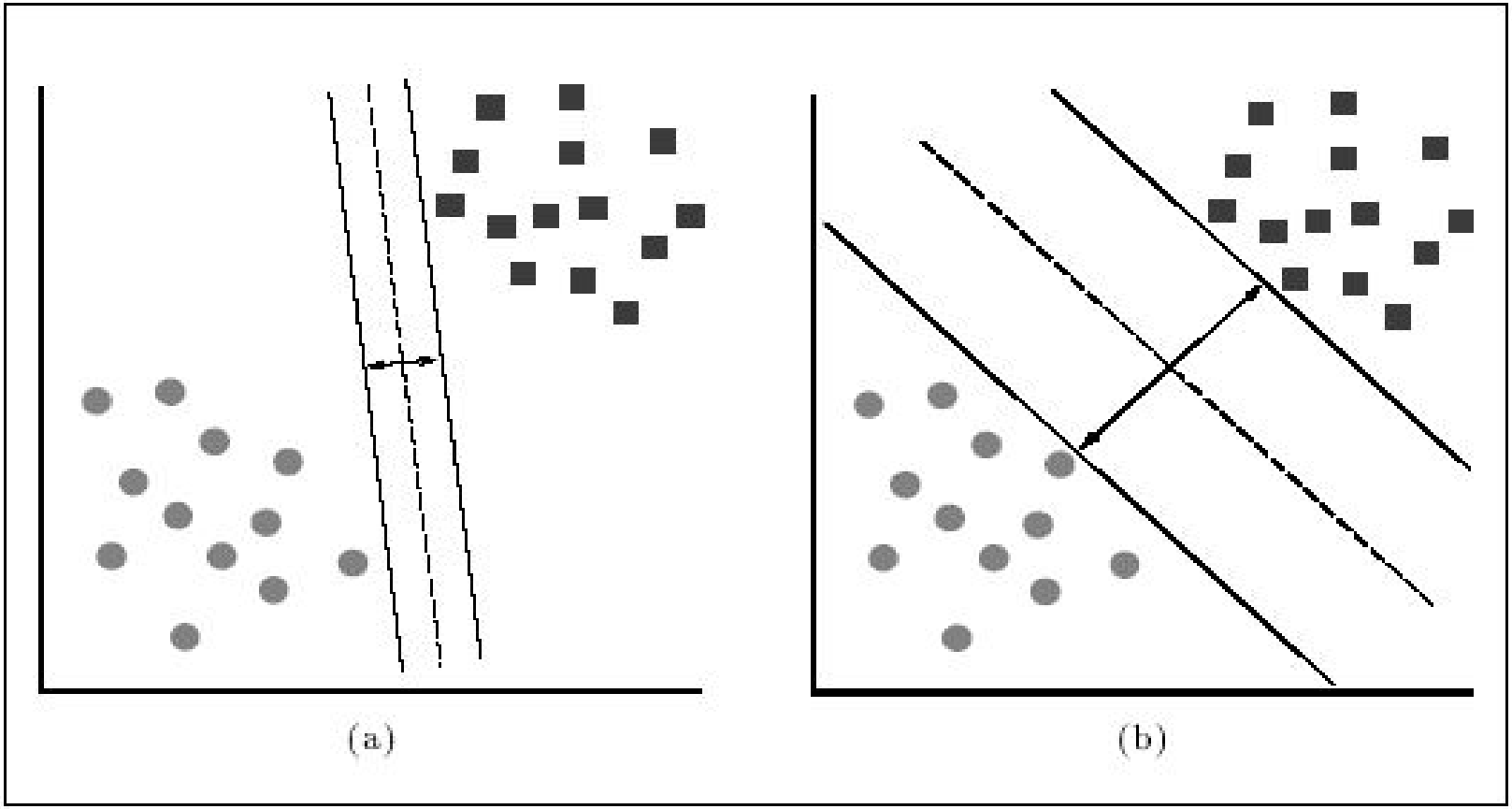
Motivation

In this paper the authors propose a decomposition method to train support vectors using large data sets

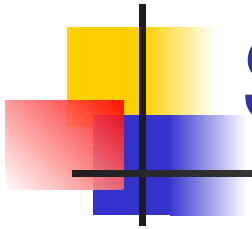
Obtaining the support vectors involves solving a QP problem. If the dataset is huge it would take a very long time.

Complex problems such as face detection require large training sets to achieve a high accuracy in classification. So having a method that can deal with large training sets efficiently would be desirable.

SVMs Revisited



Linearly separable data



SVMs Revisted

The idea is to find a separating hyperplane with the maximum margin to get the best generalization.

The dual problem problem allows us to solve the same problem in a high dimensional space by using kernels.

SVMs Revisted

The dual problem can be expressed as follows (note: this is using a soft margin)

$$\begin{array}{lll} \text{Minimize} & W(\mathbf{\Lambda}) & = -\mathbf{\Lambda}^T \mathbf{1} + \frac{1}{2}\mathbf{\Lambda}^T D \mathbf{\Lambda} \\ \mathbf{\Lambda} & & \\ \text{subject to} & & \\ & \mathbf{\Lambda}^T \mathbf{y} & = 0 \\ & \mathbf{\Lambda} - C\mathbf{1} & \leq \mathbf{0} \\ & -\mathbf{\Lambda} & \leq \mathbf{0} \end{array}$$

where D is a symmetric, semi-positive definite, $\ell \times \ell$ matrix with elements $D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. , $\mathbf{y} = (y_1, \cdots, y_\ell)$, $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_\ell)$ is the vector of non-negative Lagrange multipliers

C - the cost of constraints violation (for relaxed margin)

SVMs Revisted

By looking at the objective function

$$W(\mathbf{\Lambda}) \quad = -\mathbf{\Lambda}^T \mathbf{1} + \tfrac{1}{2}\mathbf{\Lambda}^T D \mathbf{\Lambda}$$

we can see that only non-zero lagrangian multipliers contribute to the value.

Data vectors whose corresponding multiplier is non-zero are called support vectors

Vectors whose multiplier is bigger than zero and less than C are called margin vectors. This is because they lie on one of the canonical hyperplanes. (assumed to exist in this method)

SVMs Revisted

Classification is done using :

$$f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{\ell} \lambda_i y_i \mathbf{x}^T \mathbf{x}_i + b)$$

Where

$$g(\mathbf{x}_i) = \sum_{j=1}^{\ell} \lambda_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + b$$

is called the discriminant function

The proposed method

The method assumes that most data vectors in the data set will not be support vectors.

Let D be our data set, such that |D|=l

We split D into two sets B and N, such that |B| <= l and B is big enough to contain all of the support vectors. Initialize $\boldsymbol{\Lambda}_N = 0$

The QP problem defined by the variables in B is solved

While there exists some j in N, such that $g(\mathbf{x}_j)y_j < 1$ replace $\lambda_i = 0, i \in B$ with $\lambda_j = 0$ and solve the new subproblem.

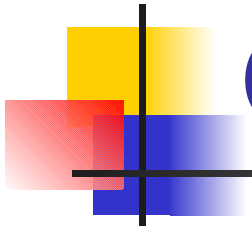
Correctness

Using the Karush-Kuhn-Tucker conditions one can show that the following holds:

1.) If $0 < \lambda_i < C$ hen $\mu = b$

2.) If $\lambda_i = C$ then $y_i g(\mathbf{x}_i) \leq 1$

3.) If $\lambda_i = 0$ then $y_i g(\mathbf{x}_i) \geq 1$



Correctness

One needs to show that after pivoting the objective function is always improved. In other words one needs to prove the following proposition:

Proposition 2.1 *Given an optimal solution of a subproblem defined on* B*, the operation of replacing* $\lambda_i = 0$*,* $i \in B$*, with* $\lambda_j = 0$*,* $j \in N$*, satisfying* $y_j g(\mathbf{x}_j) < 1$ *generates a new subproblem that when optimized, yields a strict improvement of the objective function* $W(\mathbf{\Lambda})$*.*

Correctness

Assume that there exists an λ_p such that

$$0 < \lambda_p < C$$

Also assume that $y_p = y_j$

Then there exists some epsilon>0 such that

$$\lambda_p - \delta > 0, \text{ for } \delta \in (0, \epsilon)$$

Also note: $g(\mathbf{x}_p) = y_p$ since corresponding data vector is a margin vector

Now consider: $\overline{\mathbf{\Lambda}} = \mathbf{\Lambda} + \delta e_j - \delta e_p$

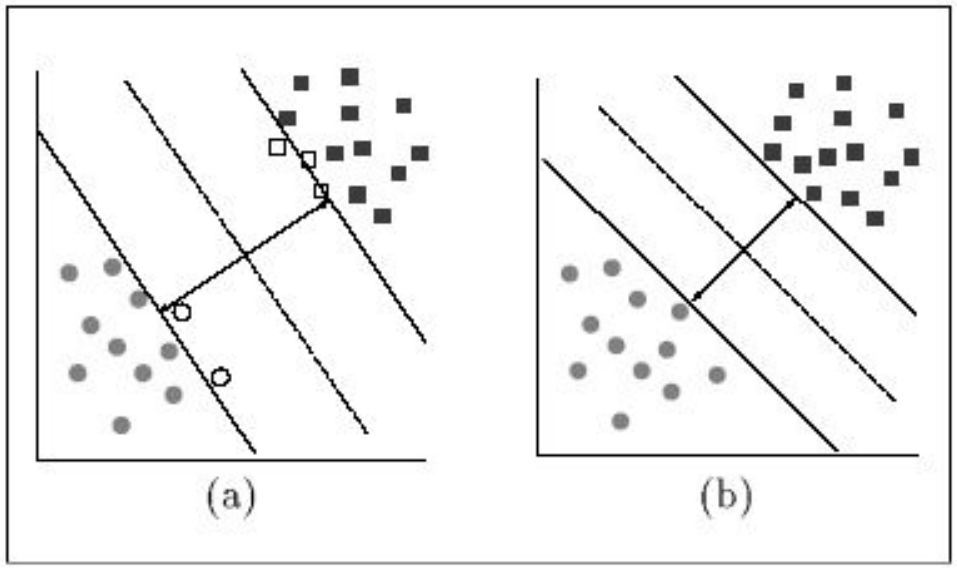
Correctness

$$
\begin{aligned}
W(\overline{\boldsymbol{\Lambda}}) &= -\overline{\boldsymbol{\Lambda}}^T 1 + \frac{1}{2}\overline{\boldsymbol{\Lambda}}^T D\overline{\boldsymbol{\Lambda}} \\
&= -\boldsymbol{\Lambda}^T 1 + \frac{1}{2}\boldsymbol{\Lambda}^T D\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T D(\delta e_j - \delta e_p) + \\
&\quad + \frac{1}{2}(\delta e_j - \delta e_p)^T D(\delta e_j - \delta e_p) \\
&= W(\boldsymbol{\Lambda}) + \delta\left[(g(\mathbf{x}_j) - b)y_j - 1 + by_p\right] + \\
&\quad + \frac{\delta^2}{2}[K(\mathbf{x}_j, \mathbf{x}_j) + K(\mathbf{x}_p, \mathbf{x}_p) - 2y_p y_j K(\mathbf{x}_p, \mathbf{x}_j)] \\
&= W(\boldsymbol{\Lambda}) + \delta\left[g(\mathbf{x}_j)y_j - 1\right] + \frac{\delta^2}{2}\left[K(\mathbf{x}_j, \mathbf{x}_j)\right. \\
&\quad \left. + K(\mathbf{x}_p, \mathbf{x}_p) - 2y_p y_j K(\mathbf{x}_p, \mathbf{x}_j)\right]
\end{aligned}
$$

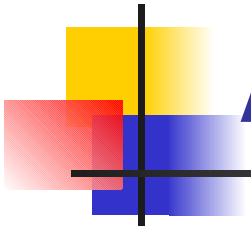
Since $g(\mathbf{x}_j)y_j < 1$ for δ sufficiently small we have $W(\overline{\boldsymbol{\Lambda}}) < W(\boldsymbol{\Lambda})$.

Correctness

Given that objective function is improved at each iteration, the algorithm will not cycle. Instead the method will converge



By including points that violate optimality conditions, the decision surface is redefined.

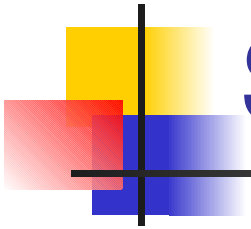


Application: Face Detection

Images are preprocessed using: Masking (gets rid of background patterns), Light Correction (reduction of light and heavy shadows) and Histogram equalization

Images are divided into two classes: face and non-face. Images in the training set are 19x19.

An SVM with a 2^{nd} degree polynomial kernel function and un upper bound of C=200 is used to obtain the decision surface

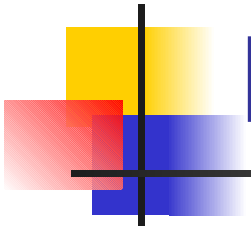


Speeding up SVM training…

In addition to the iterative method described in the paper the authors used a simplification method to speed up the computations.

The simplification method was proposed by C. Burges in 96. It provides a way to reduce the number of support vectors by replacing them with a smaller set of points, which are not necessarily data points.

The method has been shown to speed up the training for digit recognition up to ten times with essentially no loss in accuracy.



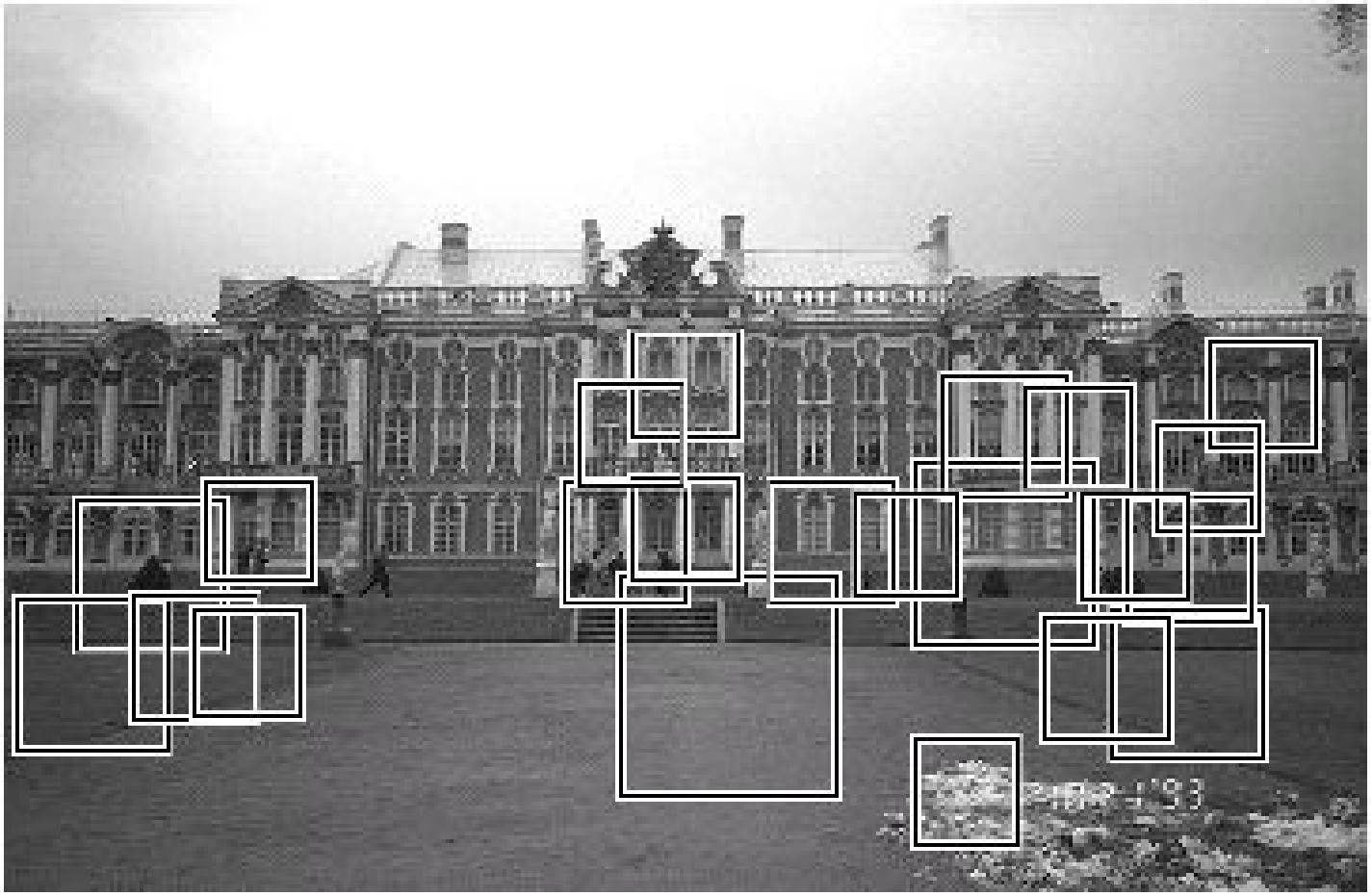
Building up a non-face class

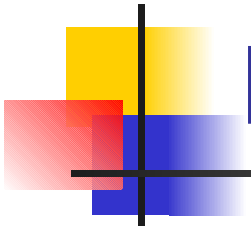
Given that the class of non-face images is a lot more complex it is difficult to find a set of images that will represent it accurately.

The authors applied their initial SVM classifier to non-face images such as trees, buildings, and landscapes. The misclassifications that were obtained were stored as non-face images. The SVM classifier was recomputed.

The authors claim that the chosen non-face images are a good source of false positives because they contain many different textured patterns.

Misclassifications





Detecting Faces (frontal views)

Once a classifier is obtained it is used a follows:

- The input image is rescaled several times
- 19x19 window patterns are considered from the scaled image
- The window is preprocessed using masking, light correction and histogram equalization.
- Then the window is classified using the SVM

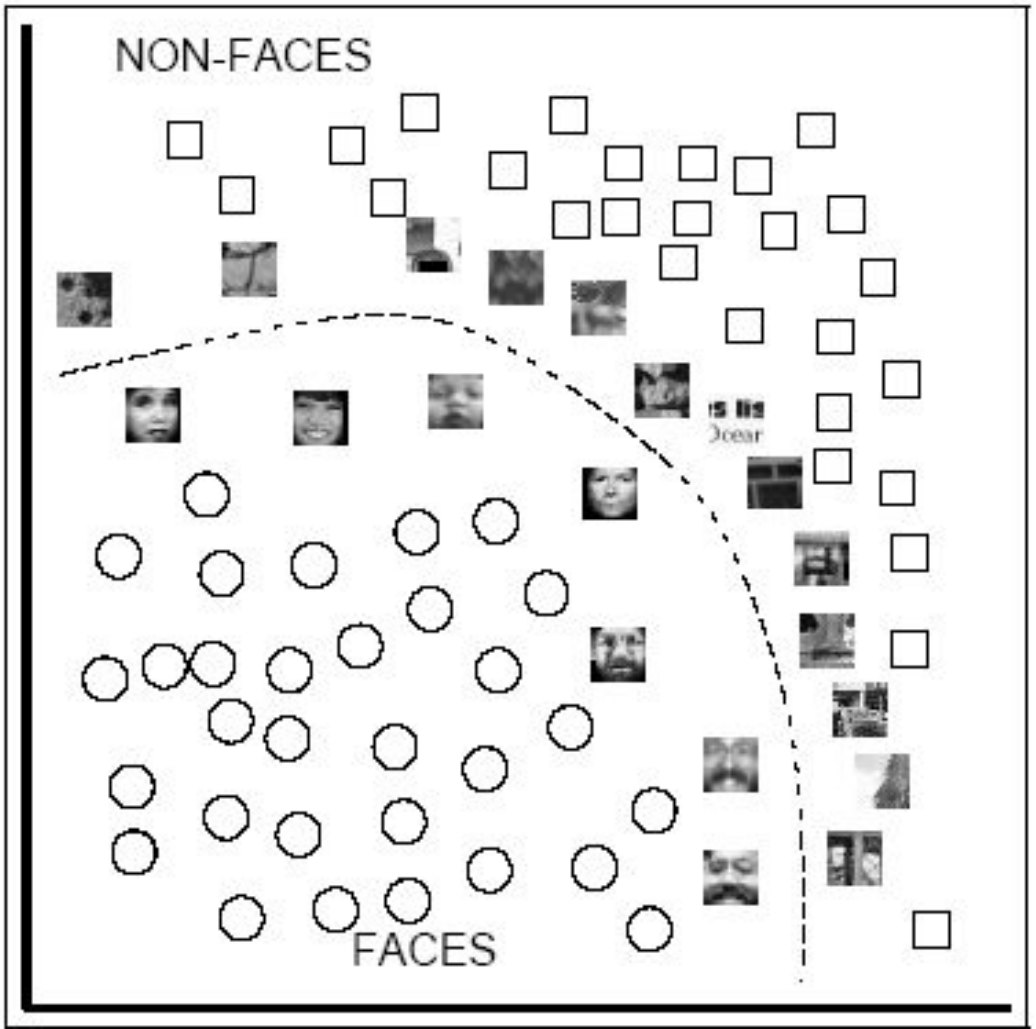
Detected Faces

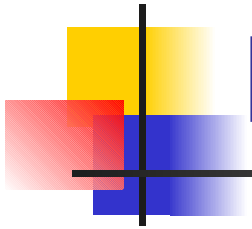
Frontal views and slightly oriented images

Detected Faces

Note that occluded faces are not recognized at all.
This is because the training set did not include any occluded faces.

How some of the support vectors look like



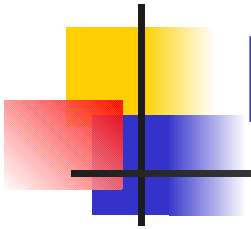


Results

The system was run on the same data set that was used by Sung and Poggio

Sung and Poggio used a very similar approach. For classification they trained a Multi Layer Perceptron

	Test Set A		Test Set B	
	Detect Rate	False Alarms	Detect Rate	False Alarms
SVM	97.1 %	4	74.2%	20
Sung *et al.*	94.6 %	2	74.2%	11



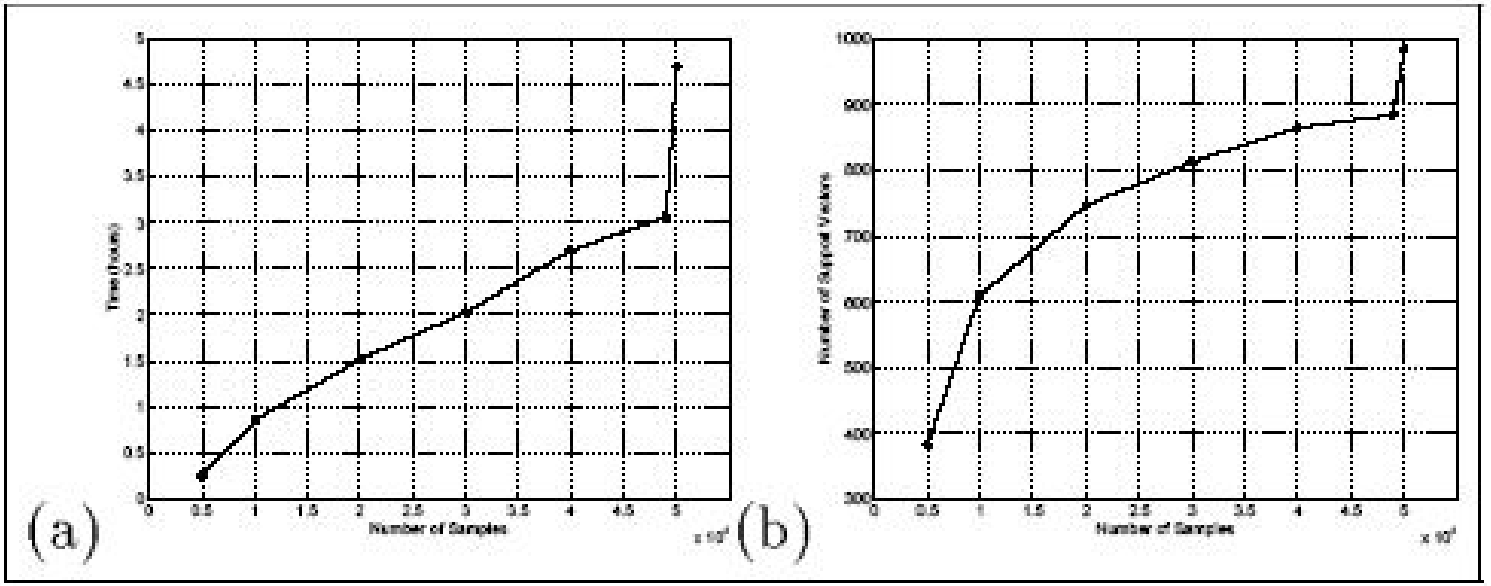
Results

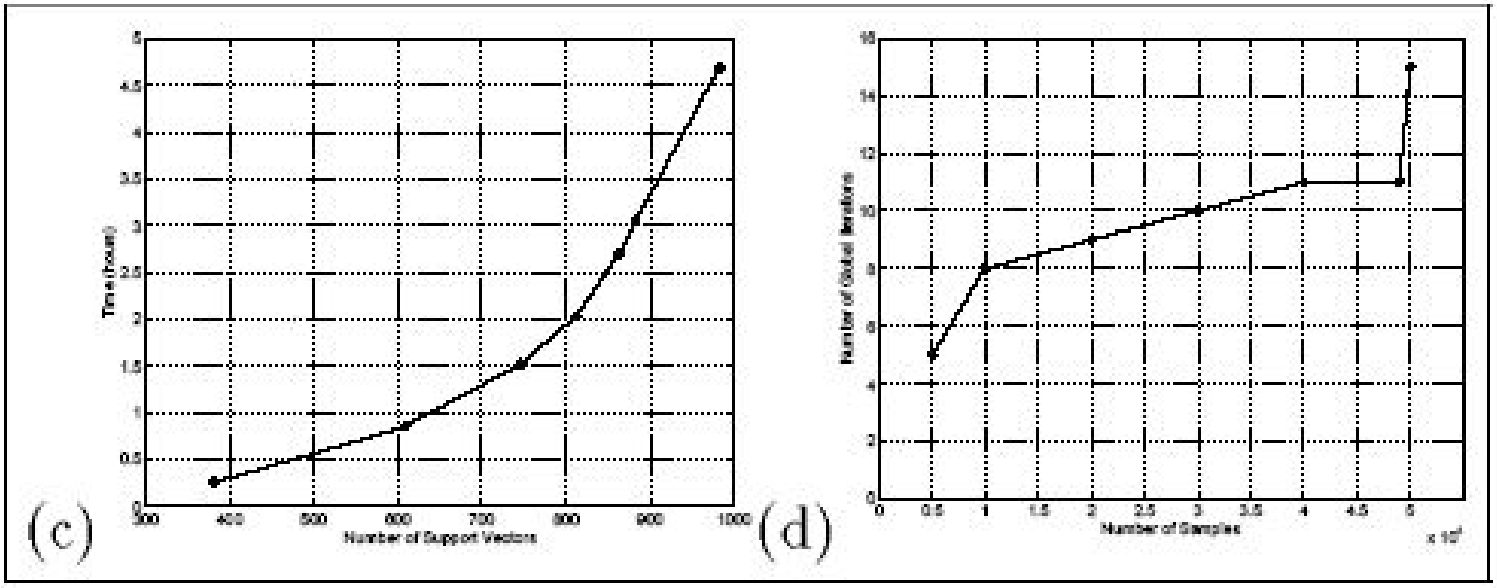
The SVM system turned out to be 30 times faster then the MLP system.

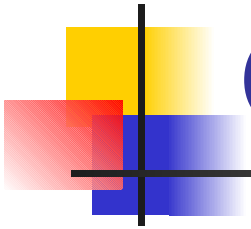
The memory requirements are quadratic. For 50000 data points, a working set of 1200 was used. The memory requirement was 25 MB.

However if we go up to 2800 support vectors, the memory requirement becomes 128MB

Performance for data sets of different size



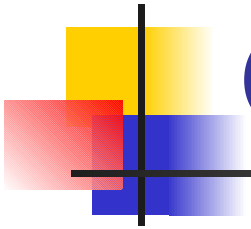




Conclusions

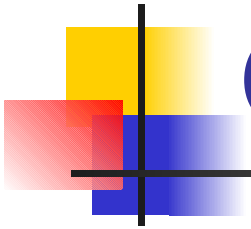
Weaknesses:

- The authors do not address the issue of determining the size of the initial working set. How does one know upfront how many support vectors will be needed?
- The proofs show that the method works, but no arguments are given on the convergence rate. Even if the system is much smaller than the original problem, if the convergence is very slow, the overall improvement might not be so big. Especially if at each step only one data point is exchanged.



Conclusions

It is not clear how much of the speed up in the face detection is due to the decomposition algorithm and how much is due to simplification method, since both were used.



Conclusions

Strengths:

- The method seems to perform very well on high quality images (97 % detection rate). Given the speed up it is definitely an improvement over the MLP method.
- The decomposition method works provably and it can handle large data sets. That makes it applicable to a lot of problems