Support vector machines for 3D object recognition

(M. Pontil and A. Verri, "Support vector machines for 3D object recognition", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637-646, 1998 (on-line))

- **The problem**

  - Recognize 3D objects from appearance (i.e., no geometrical models).

- **The approach**

  - Linear SVM are used for 3D object recognition (COIL-100 database).

  - Images are regarded as points of a space of high dimensionality.

  - No features are extracted and recognition is performed without pose estimation.

- **Preprocessing**

  - Each image was converted to gray-scale (originals are RGB).

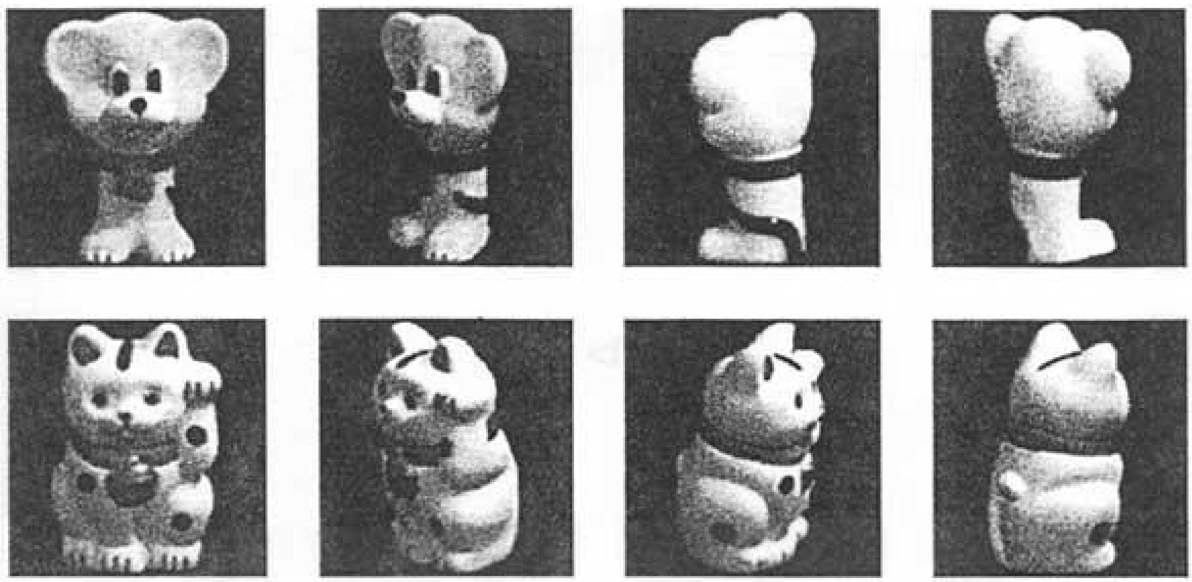
  - Spatial resolution was reduced to 32 x 32 by averaging over 4 x 4 patches.

  - Each image is thus represented as a vector with 32 x 32 = 1,024 values.

• Training

- One SVM was trained for each pair of objects (COIL-100 database).

- The images corresponding to some of the support vectors for a specific pair of objects are shown below.



- The typical number of support vectors found for each pair of objects was between 1/3 and 2/3 of the training images (72 images).

- The training stage takes about 15 minutes on a SPARC10 workstation.

• Testing

- Recognition was performed following the rules of a tennis tournament:

  * Each object is regarded as a *player*.

  * In each *match*, the system temporarily classifies an image using the SVM associated with the two players.

  * Suppose there are 2^K players, 2^{K-1} matches are played in the first round.

  * The 2^{K-1} winners are advanced to the next round.

  * The $k-1$ round is the final round which declares the winner (i.e., recognized object).

  * This procedures requires $1+2+\ldots+2^{K-1}=2^K-1$ classifications.

- The test stage is very fast (31 dot products need to be computed).

• Experiments and results

- The COIL-100 database was used in the experiments.

  * Contains 100 objects
  * 72 images/object (sampled every 5 degrees)

- Experiments were performed to test the following:

  * recognition accuracy
  * performance in the presence of noise
  * performance in the presence of bias in the registration
  * performance in the presence of occlusion

- In each experiment, a subset N from the 100 objects were considered (N was chosen randomly).

- Half images from each object (one every 10 degrees, i.e., 36 images) were used for training and the rest for testing.

Recognition accuracy

- $N = 32$ in these experiments/ 32 random experiments.

- Perfect recognition accuracy was achieved in all the experiments.

- Using a training set including the most "difficult" objects (selected manually), the system misclassified a view of a packet of chewing gum for another very similar packet of chewing gum.

Performance in the presence of noise

* Zero mean random noise, uniformly distributed in the interval $[-n, n]$, was added to the gray value of each pixel.

* The analysis was carried out using the "difficult" training set only.

* Some noise was suppressed by the 4x4 averaging ...

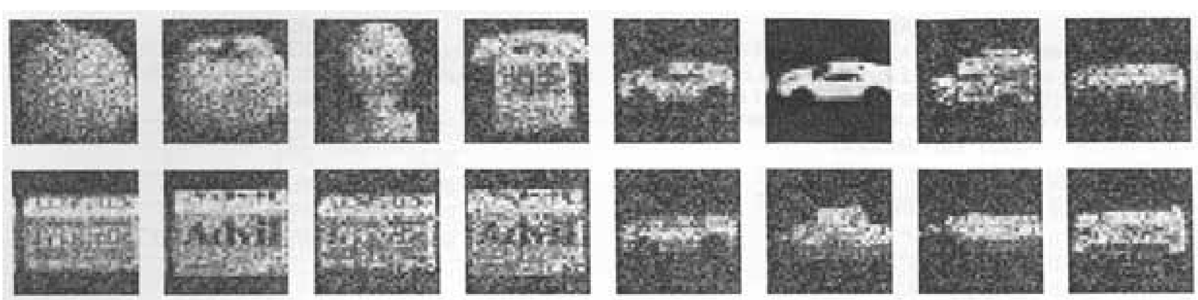


TABLE 2
ERROR RATES (E.R.) FOR COIL IMAGES CORRUPTED BY NOISE

Noise	e.r. (32 objs)	e.r. (30 objs)
±25	0.3%	0.0%
±50	0.8%	0.1%
±75	1.1%	0.2%
±100	1.6%	0.2%
±150	2.7%	0.7%
±200	6.2%	1.8%
±250	11.0%	5.8%

The noise is in gray levels (see text).

* Different spatial resolutions (8x8 to 128x128) were also tested using zero mean random noise, uniformly distributed in the interval $[-100, 100]$

* Recognition rates increase with spatial resolution.

TABLE 3
ERROR RATES (E.R.) FOR COIL IMAGES CORRUPTED BY NOISE UNIFORMLY DISTRIBUTED IN THE INTERVAL [−100, 100] AT DIFFERENT SPATIAL RESOLUTIONS

spat. res.	e.r. (32 objs)	e.r. (30 objs)
8×8	2.8%	0.5%
16×16	2.1%	0.3%
32×32	1.6%	0.2%
64×64	0.9%	0.1%
128×128	0.3%	0.0%

Performance in the presence of bias in the registration

- Each image in the most difficult test set was shifted by n pixels in the horizontal direction.

- Spatial registration seems to be very important.

TABLE 4
ERROR RATES (E.R.) FOR SHIFTED COIL IMAGES (SHIFTS ARE IN PIXEL UNITS)

shift	e.r. (32 objs)	e.r. (30 objs)
3	0.6%	0.1%
5	2.0%	0.8%
7	6.7%	4.8%
10	18.6%	12.5%

Performance in the presence of noise and bias in the registration

TABLE 5
ERROR RATES (E.R.) IN THE PRESENCE OF BOTH NOISE (IN GRAY LEVELS) AND SHIFTS (IN PIXEL UNITS)

shift	noise	e.r. (32 objs)	e.r. (30 objs)
3	±25	0.6%	0.1%
3	±50	0.8%	0.1%
3	±100	1.8%	0.2%
3	±150	3.0%	0.5%
5	±25	2.1%	0.6%
5	±50	2.7%	0.8%
5	±100	4.1%	1.3%
5	±150	7.3%	2.7%

Performance in the presence of occlusion

- Occlusion was introduced in two different ways:

  (1) by randomly selecting a subwindow in the test images and assigning a random value between 0 and 255 to the pixels inside the subwindow.

  (2) by randomly selecting n columns and m rows and assigning a random value to the corresponding pixels.

- The system seems to tolerate small amounts of noise.

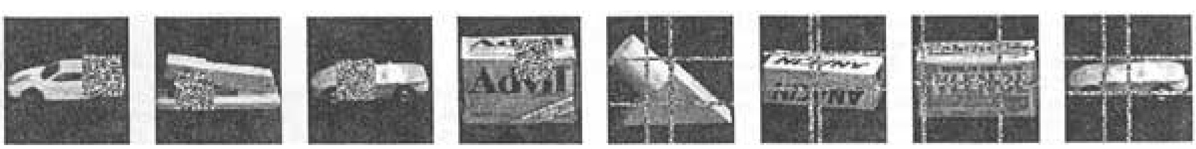


TABLE 6
ERROR RATES (E.R.) FOR COIL IMAGES OCCLUDED BY A RANDOMLY PLACED K × K WINDOW OF UNIFORMLY DISTRIBUTED RANDOM NOISE

k	e.r. (32 objs)	e.r. (30 objs)
4	0.7%	0.4%
6	2.0%	1.2%
8	5.7%	4.3%
10	12.7%	10.8%

TABLE 7
ERROR RATES (E.R.) FOR COIL IMAGES IN WHICH N COLUMNS AND M ROWS (RANDOMLY SELECTED) WERE REPLACED BY UNIFORMLY DISTRIBUTED RANDOM NOISE

n	m	e.r. (32 objs)	e.r. (30 objs)
1	1	2.1%	1.3%
1	2	3.2%	1.9%
2	1	4.5%	2.8%
2	2	6.1%	3.2%

Example-based object detection in images by components

(A. Mojan, C. Papageorgiou and T. Poggio, "Example-based object detection in images by components", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349-361, 2001 (on-line))

• The problem

- Build a general example-based (i.e., appearance-based) framework for component-based object detection.

- A component-based object detection system searches for an object by looking for identifying its components rather than the whole object.

- The proposed system is demonstrated on the problem of detecting people in cluttered scenes.

• Applications and challenges

- Detection of people in images has many applications including
  * Surveillance systems
  * Driver assistance systems
  * Image indexing

- More challenging that detecting other objects due to several reasons:
  * People are highly articulated objects.
  * Difficult to build a single model that captures the shape variation.
  * People dress in a variety of colors and garment types.

• The approach

- Four exampled-based detectors (implemented as SVM) are used to detect the following four components of the human body: head, legs, left arm, and right arm.

- The input to each detector are features based on the Haar wavelet transform.

- The spatial configuration of the detected parts is validated.

- An example-based classifier (implemented as a SVM) combines the results of the component detectors to classify a pattern as either a "person" or a "non-person".

• Why using components?

- It allows to combine the visual information present in an image with the geometric information concerning the human body.

- Often, it is difficult to detect the human body as a whole due to variations in lighting and orientation.

- Can provide tolerance to partial occlusions.

- Hierarchical classification schemes have been shown to perform better than single classifiers.

• Overview of proposed system

- Given an input image, a 128 x 64 window is shifted across and down the image, starting from the upper left corner.

- To allow detecting various sizes of people, the image is processed at several sizes ranging from 0.2 to 1.5 times its original size.

- Each input window is classified as a "person" or "non-person" as follows:

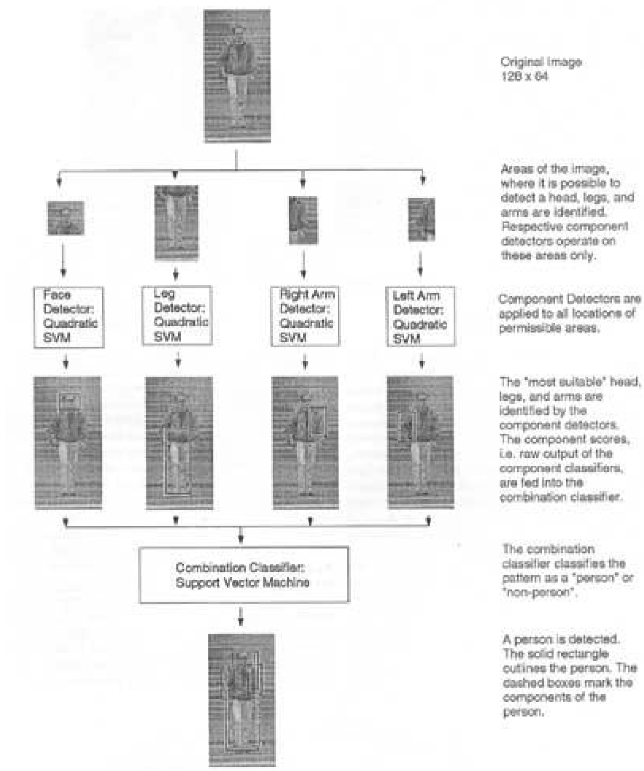
  * Apply the component detectors within the window.

  * Each candidate body part region is processed by applying the Haar wavelet transform.

  * A vector containing the wavelet coefficients is then classified by the component detectors (quadratic SVM).

  * The detector with the highest output (*component score*) determines the classification of the body part.

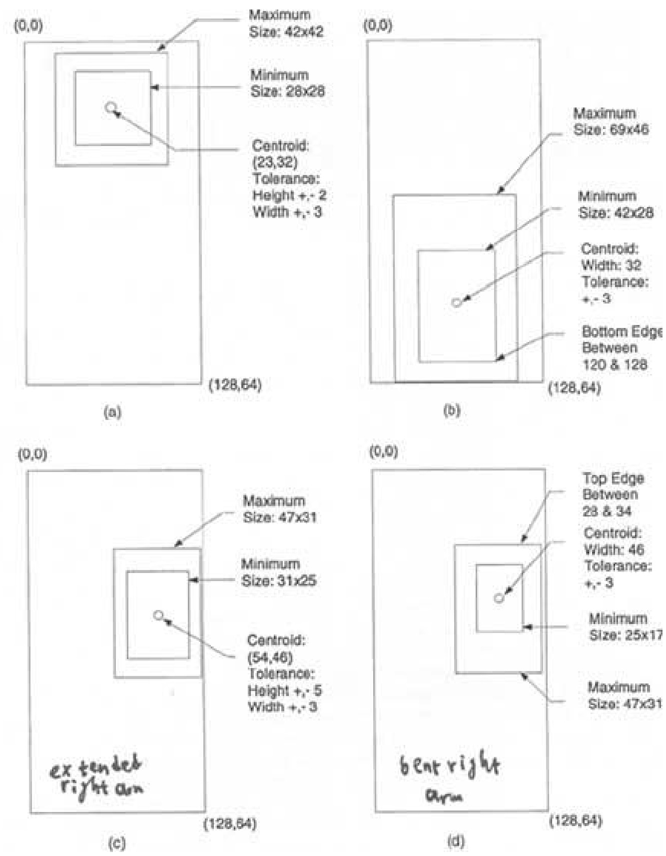
  * The highest component score for each body part is fed to the combination classifier (linear SVM) which determines if the input window is a person.



- **First stage: identifying components of people**

  - The component detectors are applied only to specific areas of the window (i.e., approximate configuration of body parts is known) and only at particular scales (i.e., relative proportions must match).

  - These areas were determined from the training set based on geometric constraints for each component within a 128 x 64 window (training images have been aligned such as people are in the center of the image).


(0,0)
Maximum Size: 42x42
Minimum Size: 28x28
Centroid: (23,32) Tolerance: Height +,- 2 Width +,- 3
(128,64)
(a)
(0,0)
Maximum Size: 69x46
Minimum Size: 42x28
Centroid: Width: 32 Tolerance: +,- 3
Bottom Edge Between 120 & 128
(128,64)
(b)
(0,0)
Maximum Size: 47x31
Minimum Size: 31x25
Centroid: (54,46) Tolerance: Height +,- 5 Width +,- 3
extended right arm
(128,64)
(c)
(0,0)
Top Edge Between 28 & 34
Centroid: Width: 46 Tolerance: +,- 3
Minimum Size: 25x17
Maximum Size: 47x31
bent right arm
(128,64)
(d)

- A 32x32 window is used for the head and a 48x32 window for the lower body and the arms.

- The information in each window is represented by a set of Haar wavelet coefficients (582 coefficients for the head window and 954 coefficients for the lower body and the arms windows - see paper and ref. [16] for more details):

    * They consider two scales only (8x8 and 16x16).

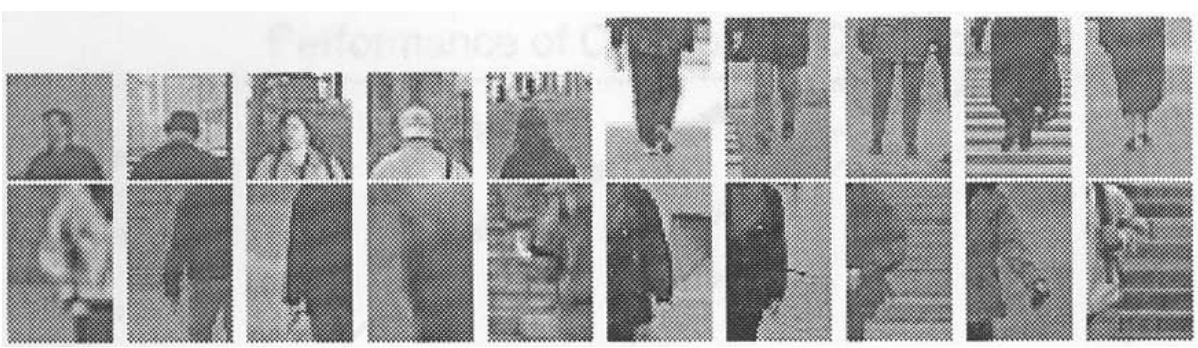
    * They run the Haar transform (non-standard basis) over each color channel separately.

    - For each scale, they keep the largest wavelet coefficient among the three color channels.

- The wavelet coefficients are fed to component detectors which are implemented as quadratic SVM.

$$K(x, x_k) = (x.\, x_k + 1)^2$$

- The component SVM are trained on positive and negative examples.



• Second stage: combining component classifications

- The highest response of each component detector (*component score*) is fed to the combination classifier.

- The component score is a rough measurement of how "well" a test point fits into its designated class (i.e., proportional to the distance of the test point from the SVM hyperplane).

- The combination classifier is implemented as a linear SVM:

$$K(x, x_k) = (x.\, x_k + 1)$$

• Data sets

- The data set contains images of people taken with different cameras, under different lighting conditions, and in different seasons.

- There are images of people who are

  * rotated in depth
  * walking
  * stationary (frontal and rear views)

- The positive examples of the lower body include images of

  * women in skirts
  * people wearing full length overcoats
  * people dressed in pants

• Training data

- The positive examples for the arms included arms at various positions in relation to the body.

- The negative examples were taken from scenes that do not contain people.

- Number of positive/negative examples used to train the component detectors:

    * head detector: 856 positive, 9,315 negative
    * lower body: 866 positive, 9,260 negative
    * left arm: 835 positive, 9,260 negative
    * right arm: 838 positive, 9,260 negative

- Number of positive/negative examples used to train the combination classifier:

    * 889 positive, 3,106 negative

• Test set

- The proposed system was run on a database containing 123 images of people to determine the detection rate.

- The system was also run on a database containing 50 images that do not contain people to determine the false-alarm rate (796,904 windows).

• Comparisons

- The proposed method was compared with two other methods:

- A method similar to the proposed but with the combination of component scores being done through voting (*voting-based combination*)

    * Classifies an input as a person only if all components have been detected in the proper configuration.

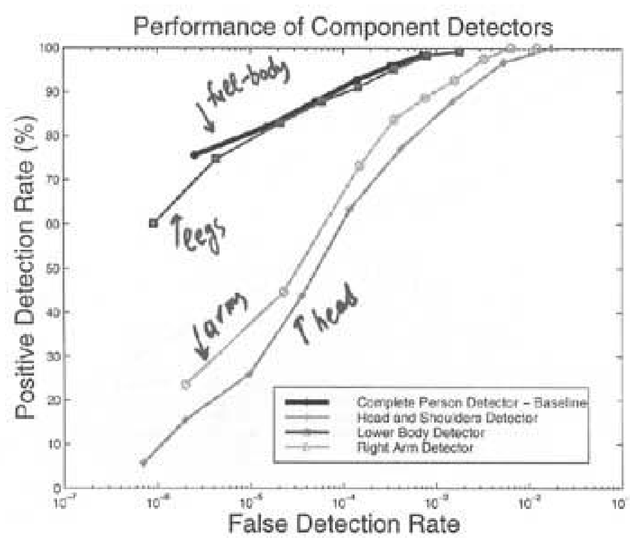
- A full-body person detector (based on their previous work).

    * uses the Haar wavelet too

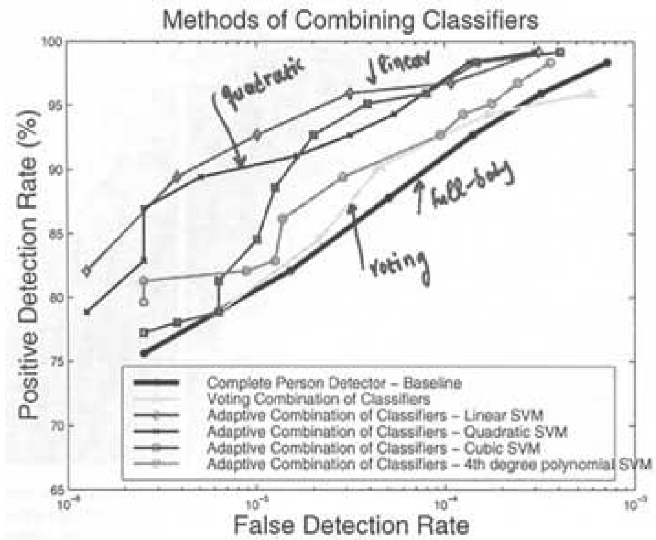
    * was trained using 869 positive examples and 9,225 negative examples

• Experiments and results

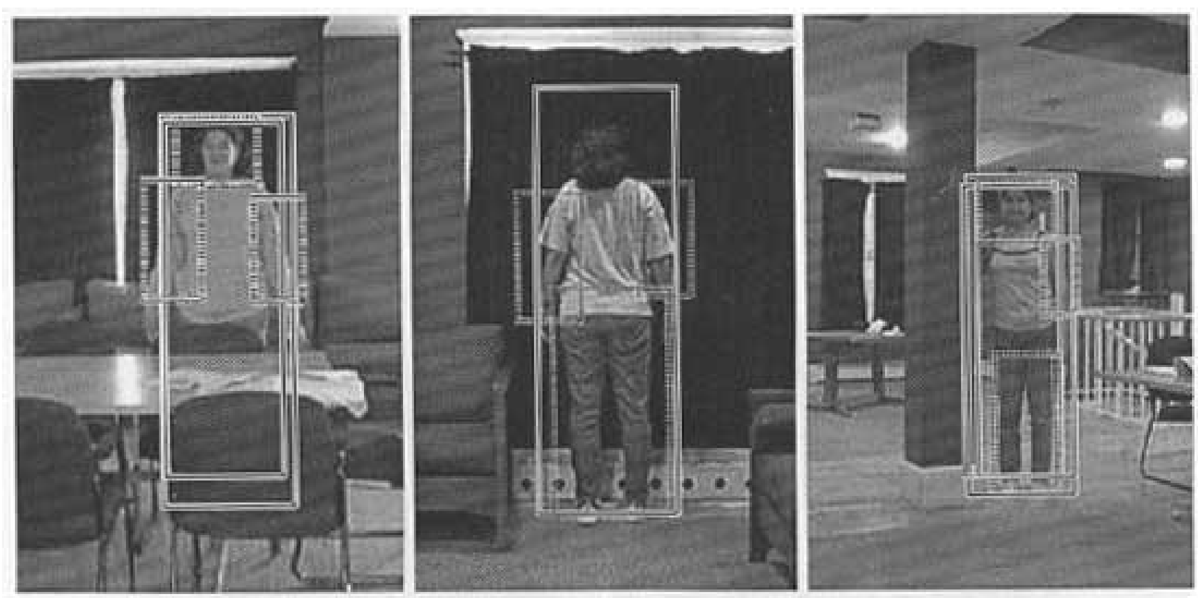
<u>Compare component detectors with full-body detector</u>

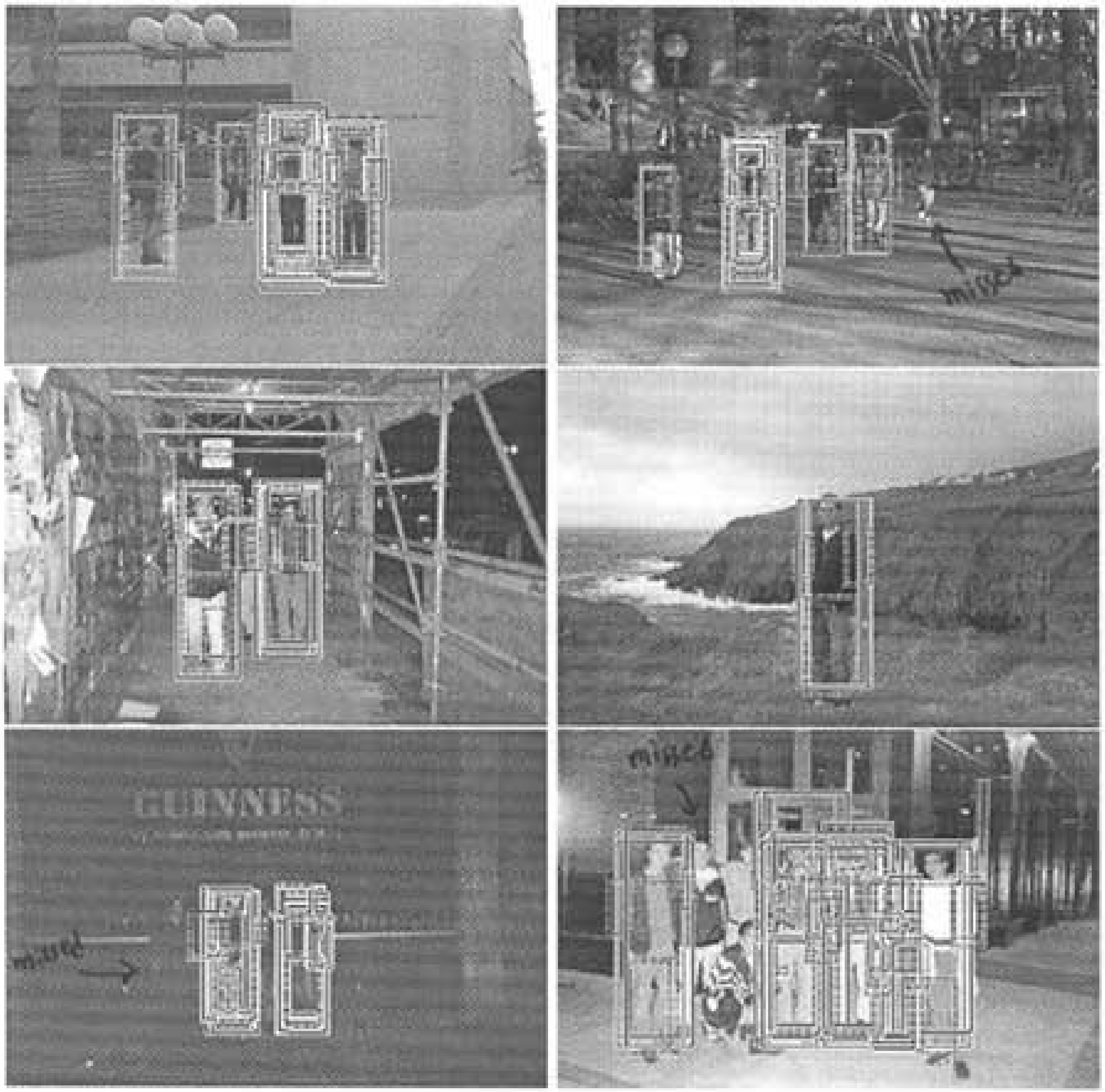


<u>Compare various methods</u>



<u>Performance under occlusion and clutter</u>





- **Extensions**

  - Learn the geometric constraints to be placed on the components of an object from examples (some preliminary results are presented in section 3.3).

  - Build more sophisticated systems where the important components of an object are learned too.

Gender Classification with SVM

(B. Moghaddam and M. Yang, "Gender Classification with SVM", *IEEE Conference on Face and Gesture Recognition*, pp. 306-311, 2000 (on-line)).

• The problem

- Visual gender classification from face images.

• The approach

- Use SVM to learn and classify gender from a large set of images.

- Low resolution, hairless face image are used.

• The dataset

- 1755 images (1044 males and 711 females) from the FERET database were used in the experiments.

- The face images were normalized (i.e., feature alignment, hair removal through masking) and their resolution was reduced to 21 x 12.

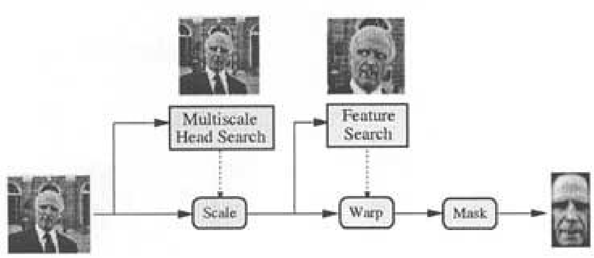


Figure 2. Face alignment system

• Training/Test sets

- They created 5 train/test sets randomly.

  * The number of training patterns was 1496 (793 males, 713 females).

  * The number of test patterns was 259 (133 males and 126 females).

- The average error was estimated for each classifier tested.

• Other methods used for comparison

- The proposed method based on SVM was compared with the following methods:

(1) Radial basis functions

$$g(x) = \sum_{i}^{K} w_i G(x; \mu aubi, \sigma_i) + b$$

(2) LDA

(3) Linear classifier (Gaussian densities, same covariance Σ, equal priors)

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_{\mathbf{i}})^t \Sigma^{-1}(\mathbf{x} - \mu_{\mathbf{i}})$$

which can be written as $g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + w_{i0}$

(4) Quadratic classifier (Gaussian densities, different covariance matrices Σ_1 and Σ_2, equal priors)

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_{\mathbf{i}})^t \Sigma_i^{-1}(\mathbf{x} - \mu_{\mathbf{i}}) - \frac{1}{2}\ln|\Sigma_i|$$

which can be written as $g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i \mathbf{x} + w_{i0}$

(5) Nearest neighbor classifier

• **Results**

Table 1. Experimental results with thumbnails.

Classifier	Error Rate		
	Overall	Male	Female
SVM with Gaussian RBF kernel	**3.38%**	**2.05%**	**4.79%**
SVM with cubic polynomial kernel	**4.88%**	**4.21%**	**5.59%**
Large ensemble-RBF	5.54%	4.59%	6.55%
Classical RBF	7.79%	6.89%	8.75%
Quadratic classifier	10.63%	9.44%	11.88%
Fisher linear discriminant	13.03%	12.31%	13.78%
Nearest neighbor	27.16%	26.53%	28.04%
Linear classifier	58.95%	58.47%	59.45%

- The number of support vectors found by SVM was about 20% of the training data.

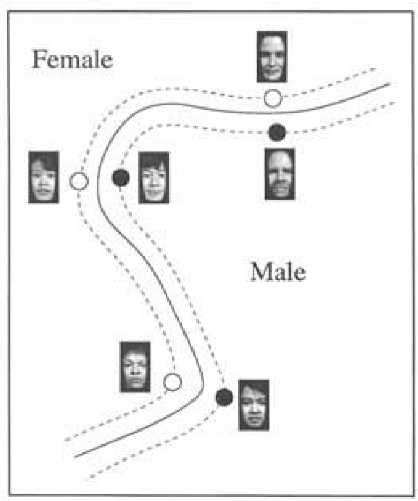


Figure 5. Support faces at the boundary

• Comparisons with human performance

- 30 subjects (22 males and 8 females) participated in an experiment with high resolution images.

- 10 subjects (6 males and 4 females) participated in an experiment with low resolution images.

- All subjects were asked to classify the gender of 254 faces.

Table 2. Human error rates

Gender of human subject	Error Rate	
	High resolution	Low resolution
Male	7.02%	30.87%
Female	5.22%	30.31%
Combined	**6.54%**	**30.65%**

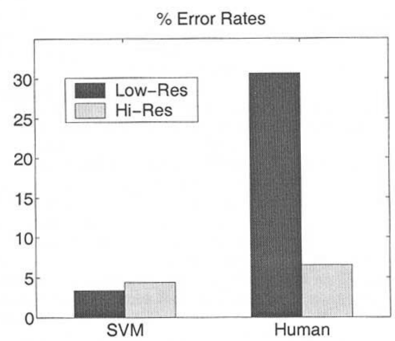


Figure 6. SVM vs. Human performance

- Faces misclassified by SVM were almost always misclassified by humans as well (the converse was not true).