Advances in Component Based Face Detection

S. M. Bileschi

Center for Biological And Computational Learning
Massachusetts Institute of Technology
Cambridge, MA.

B. Heisele

Honda Research and Development

Boston, MA.

Abstract

We describe the design of a component based face detector for gray scale images. We show that including parts of the face into the negative training sets of the component classifiers leads to improved system performance. We also introduce a method of using pairwise position statistics between component locations to more accurately locate the parts of a face. Finally, we illustrate an application of this technology in the creation of an accurate eye detection system.

1. Introduction

The goal of the work presented here is to build an accurate face detection system for gray scale images. We limit our domain to faces which are not rotated in the image plane, and are rotated a maximum of 30 degrees left or right out of the plane. Our system will be developed as an extension to the component based face detection system described in [2].

Many early face detection systems eschewed component based architectures for a global approach. In [5], the distribution of faces is modelled with a mixture of gaussian curves. Faces are detected comparing novel patterns to the model distribution. A similar approach is taken in [4] and [6, 10], where a single SVM and a set of neural networks, respectively, are trained to build surfaces to separate faces from non-faces. In all of these systems, the feature vector is composed of image values sampled uniformly over the whole face pattern.

It makes sense intuitively to use a part based approach to face detection if one believes that small parts of the face are less sensitive to visual changes due to differences in lighting or pose. Part based systems can also be less sensitive to partial facial occlusion. Perhaps the most compelling reason to continue studying part based systems is the empirical evidence supporting their accuracy over global approaches [2].

All component based systems must at some point select which parts to use. Some systems, such as those described in [3, 1], use parts which seem naturally salient to humans, such as the eyes, nose, and mouth. Other systems have been designed to learn object parts automatically from the training images [13, 9, 2, 14]. The system described in [2] uses 14 features that were chosen automatically using a region growing algorithm in combination with a statistical error bound [11]. In [9], an interest operator was designed to collect image patches from the training set, which were then clustered to find salient object parts. Component based object detection systems in the literature have been built with as few as 2 or as many as 150 component parts. The system described in this paper uses exactly the same 14 components described in [2] for ease of direct comparison.

Once the part examples have been located within the input image, and perhaps labelled with a confidence in each detection, each component based object detection system will use still another classifier to judge whether or not the part detections are truly part of the target object, or they are simply doppelgangers stemming from similar patterns in non-object image sections. Some of these upper-level classifiers use the geometry of the detected parts to decide face examples from non-face examples. Others use only the confidence measure output by the individual part classifiers. The face detection system described in [7] uses a product of probabilities, indexed from histograms, to calculate confidence in some image patch stemming from the face class. In [2] in each test image only the best example of each part is used, and an SVM is utilized to decide whether the set of positions and confidences is likely to have come from a face. This SVM method of judging part detections, along with a few other top level classifiers for comparison, will be used in the system outlined in section 2.

2 Implementation Background

2.1 Global Classifiers for Vision

We use the term global image classifier to describe the opposite of a component-based image classifier. These machines do not search the input image for constituent object parts as a first step toward classification. A single SVM

0-7695-2010-3/03 $ 17.00 © 2003 IEEE

trained on images of faces and non faces is an example of a global face detector. The features input to a global classifier do not necessarily need to be pixel values; wavelet features, first derivatives of gray scale features, and other statistics could also be used.

To turn a classifier into an object detector, a common strategy is to use a windowing technique; where every image patch is independently fed into the classifier [10]. When the classifier output is larger than some threshold, the corresponding part of the image is labelled as being a member of the object class. It is possible to build a corresponding image, separate from the input image, where the value of this new image at some position (i, j) is equal to the value output from the classifier if the input to the classifier is the image patch taken from the input image, starting at position (i, j); refer to figure 1 for an illustration. This new image, which we will refer to as a *result image* will be precisely the size of the original input image, less the size of the classifier, and brighter where the classifier returned large values. Figure 2 illustrates the result image created when a classifier tuned to respond strongly to the bridge of the nose is run over both a face image and a non-face image. Note the strong response over the bridge of the nose.

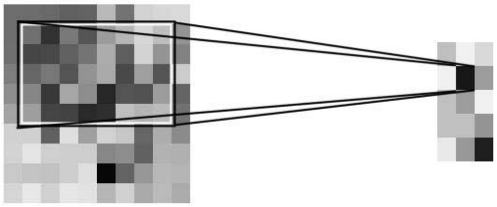

Figure 1: A 10×10 input image when fed into an 8×5 classifier yields a 3×6 result image. The window corresponding to result image position (2, 2) is illuminated to illustrate the correspondence.

2.2 Simplified Component-based Classifier

Illustrated in figure 3 in block diagram format is a schematic depicting a simple part based face detection system. In this abstraction, which is similar to the more complicated system described in section 3, a result image is created for each component. They are then used as the input to some higher classifier which will detect faces based on the part results.

For each sub-window of the original image, the x and y position of maximization in each result image is recorded (relative to the top of the sub-window), along with the values at that position. This process yields a set of triplets of the form $((x_0, y_0, v_0), (x_1, y_1, v_1), ..., (x_{n-1}, y_{n-1}, v_{n-1}))$ where n is the number of facial components used by the system. This ordered set of triplets will be referred to as a *constellation* and can be thought of as the set of points within the sub-window where the face parts fit best. This constellation is then input to the higher level classifier, which decides between constellations stemming from faces and constellations from non-faces. The output of this upper-level classifier is recorded in the final result image. The top level classifier can be of any number of types (SVMs and Baysian approaches are commonly used) and is constrained only in that it must be a function mapping valid constellations to real values.

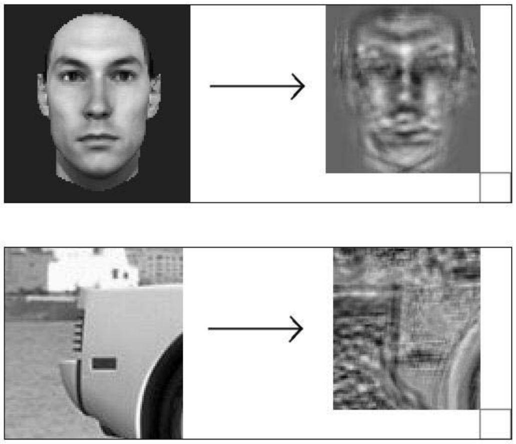

Figure 2: An 18×16 global classifier trained on images of the bridge of the nose is run over two input images, one of a face and one of a non-face.

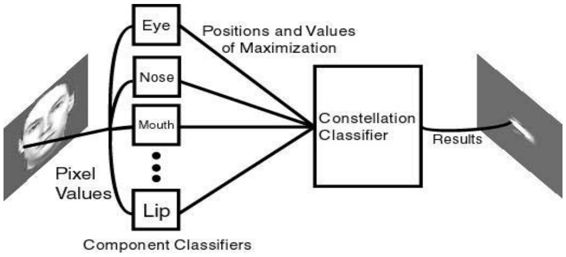

Figure 3: A schematic for a component based classifier.

3 Detailed Implementation

Figure 4 shows a detailed block structure diagram of our face detection system. Each sub-section will be described separately in order of data processing. The biasing, or model step between the creation of the component result images and the construction of the constellations is optional, and will be described at the end of this section .

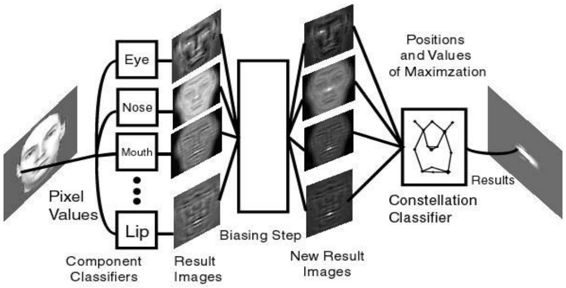

Figure 4: A block diagram schematic of the major components of our face detection system.

3.1 Component Classifiers

Our face detector uses the 14 parts illustrated in figure 5. All of the parts, when situated over a frontal face, lie completely within the frame of the face and include no hairline, jawline, or ear structure. These parts were chosen in particular to match the component classifiers used in [2], which were in turn selected automatically using a statistical error bound.

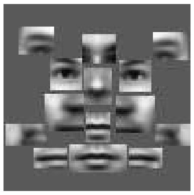

Figure 5: The 14 components used in our component based face detection system arranged in a geometrically salient and vaguely disturbing pattern.

3.2 Training Data

The training data, which was used to train the component classifiers as well as the top-level classifier, is a set of images divided into positive and negative examples of faces. The negative training data consists of 13,654 gray scale images. Each of these images is a 58 × 58 crop from a larger set of images known not to contain any faces. Many, but not all of these images are difficult examples of non faces, selected by using a simple face detector to bootstrap examples out of larger images.

The positive training data consists of 1,323 100 × 100 images of textured 3-D head models provided through the work in [12]. A few of these are illustrated in figure 6. The images are of 21 different heads viewed at 7 angles of rotation between head-on and 30 degrees to the right. At each position each head is viewed with 9 different illuminations. In the images of the 3-D model, the facial part of interest is about 58 pixels square.

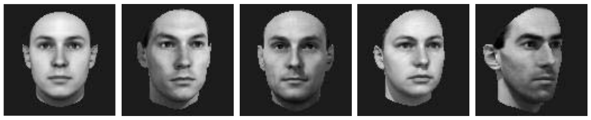

Figure 6: Five examples from the positive training set.

In order to create the component training set, it was necessary to crop all 14 target parts out of each training image. This process was made much easier with the correspondence between images available from the artificial head data. Along with the images of the heads are included the pixel positions of 25 sentinel points on the head. Figure 7 illustrates the positions of some of these points on a typical head model. Each of our components is defined as a sentinel point and extensions up, down, left and right. For instance that the first classifier is an 18 by 16 rectangle around a point centered at the bridge of the nose. Figure 8 shows a few examples of extracted training data for this classifier. When we extract the components we also extract components from the left-right mirror images of the training data. This corpus of 2,646 images for every component comprises the positive training data of the component classifiers.

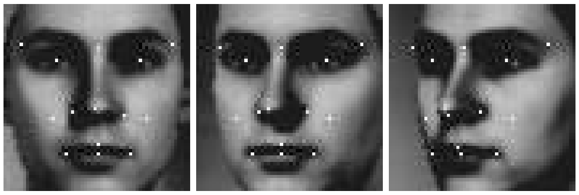

Figure 7: The 58 × 58 region around the face in three training images, with the 14 utilized sentinel points highlighted.

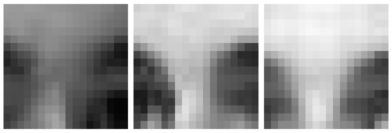

Figure 8: Selected examples of the positive training set for component 0, the bridge of the nose component

Two different negative component training sets were built; the first one was built from the negative examples of faces. For each component classifier, a random rectangle, the size determined by the classifier, was extracted from each of the 13,654 negative training images. This will be referred to as the non-facial negative training set, examples of which can be seen in figure 9.

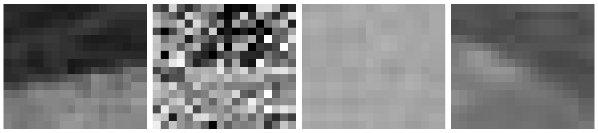

Figure 9: Selected examples of the non-face negative training set for component 0, the bridge of the nose component.

The second negative training set was created using extractions from the *positive* face data. Care was taken so that the extractions did not overlap the canonical positions by more than 50% of the area of the classifier. From each of the 1,323 training images 4 such rectangles were cropped out, and 4 again from the mirror image. This body of 10,584 images per classifier will be referred to as the facial negative training set, examples of which are shown in figure 10.

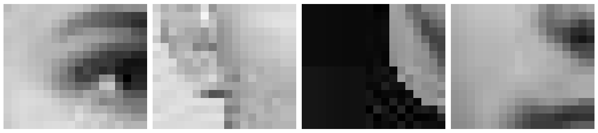

Figure 10: Selected examples of the facial negative training set for component 0, the bridge of the nose component

For our feature space we use the gray scale values of the pixels in each training image. Gray scale pixel values have been shown to be a good feature space for frontal face detection in comparison to derivative or wavelet type features [1]. Two separate arrays of component classifiers were trained, one using the non-facial negatives, and the other using the facial negatives.

3.3 Judgement of Constellations

Once the constellation has been calculated for every 58 × 58 window in the input image, a higher level classifier is employed to judge the constellations.

Our first constellation judging algorithm uses histogram based classifiers. In this approach, we collected data from the artificial head models to produce a model of $P(x_n, y_n|n)$ for each component n. Figure 11 illustrates this position histogram for the bridge of the nose classifier, the left cheek classifier, and the mouth classifier. If we assume that the position of facial components are independent random variables, we can calculate the probability of a constellation stemming from a face by simply multiplying all the probabilities indexed from the histograms.

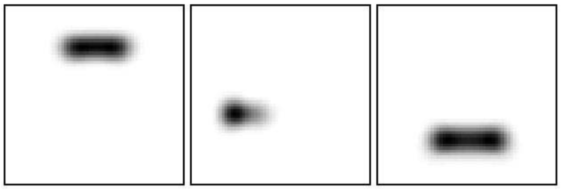

Figure 11: Position histograms for the bridge of the nose, left cheek, and mouth components. Darker pixels indicate areas of likely location for these components

3.4 Biasing Step

One common error of the system outlined in [2] is that the classifiers don't always maximize at the correct location. Using only the position of the maximum stimulation per component unfortunately ignores any local maximum over the correct position of the component.

Using classifiers that have maximized in the correct position, we can guess more likely positions for classifiers that were wrong using geometric clues. Since we don't know which classifiers were correct a priori, we propose the following algorithm to improve the accuracy of the constellations.

Once the constellation has been determined, for every classifier i, and for every other classifier $j \neq i$, we multiply every position in the result image of j by a value representative of how likely j is to maximize at that location, given the location of i. These representative values are drawn from a histogram of pairwise position statistics (see figure 12), modulated by a strength parameter. Paraphrased, given the position of classifier i, we change the result of classifier j to more closely model the expected position of classifier j. This is illustrated in figure 14.The strength parameter is implemented by linearly normalizing the values in the pairwise position images to $[(\alpha)^{\frac{1}{n-1}}, 1]$, where n is the number of component classifiers, and alpha is in $[0, 1]$. This way, the most any value in any result image can be reduced is by a factor of α, which happens only if the $(n-1)$ other classifiers had a value of 0 in their pairwise position histogram at this particular point.

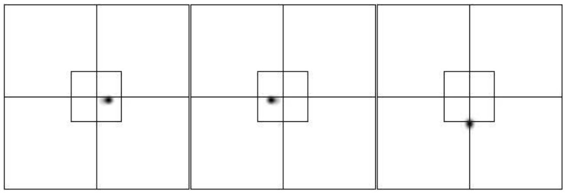

Figure 12: Pairwise position images indicate the expected position of the right eye, the left eye, and the mouth in comparison to the position of the bridge of the nose (×).

The biasing step only works by assuming that some subset of the component classifiers maximized in the correct positions, and the constructive interference from their biasing will serve to correct the errant components. Unfortunately, since the individual classifiers are weak, the global maximum of some classifier i is very often not at the correct position. Indeed, for images that are not much like our training images, it is often the 3^{rd} or 4^{th} ranked local maximum that is at the correct position. As a generalization of the biasing step outlined above, consider biasing from more than one local maximum per component. In brief, we record the N strongest local maxima whose corresponding windows of support in the original window do not overlap at all. We then bias as before from *each* of these points and refer to the technique as N-level biasing.

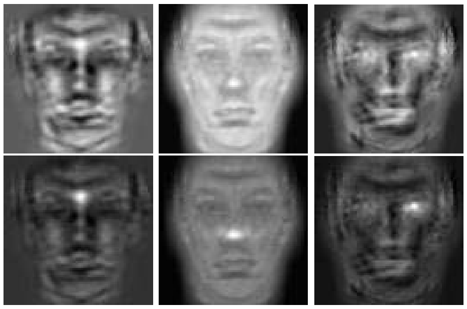

Figure 13: *Top:* Result images for the bridge of the nose, the nose, and the right eye. *Bottom:* The same result images after biasing.

4 Results

The positive test data were drawn from the CMU PIE database available at [8]. In order to save time computationally, the heads were cropped out by hand before testing. After removing from the data all heads at rotations out of the plane more than 30 degrees, we were left with a positive test set of 1,834 images, examples of which are illustrated in figure 15.

The negative test data were extracted from a set of non face images, different from the set used to generate the nonface training data. In order to make the test set difficult, we selected bootstrapped examples using a simple face classifier[1]. In total, 8,848 images comprised the negative test set. For each image in the test data, we recorded only the strongest response over all scales and positions, and used this to build an ROC curve.

The ROC curve in figure 16 is the curve gleaned from running our system using the component classifiers trained

[1] This classifier was different from the one used to generate the bootstrapped training data.

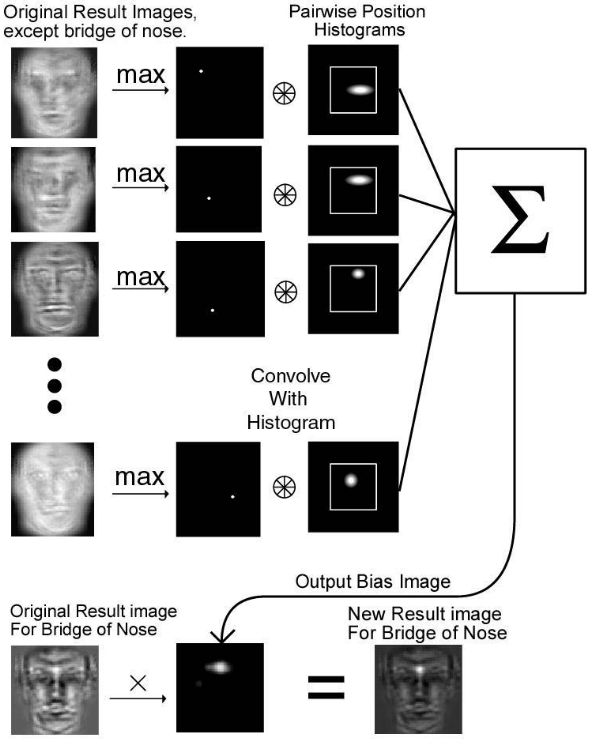

Figure 14: Illustration of the biasing step for the bridge of the nose component.

Figure 15: 5 example images from our positive test set, extracted from the CMU PIE database. The full size images are all between 200 and 300 pixels wide and roughly square

with facial-negative training set. The images were tested for faces at every scale from 60 × 60 to 110 × 110 in 11 geometric increments. Biasing was performed using 5 local maxima per component. The dashed line below, for comparison, is the result from a linear SVM trained on the full 58 × 58 facial extractions.

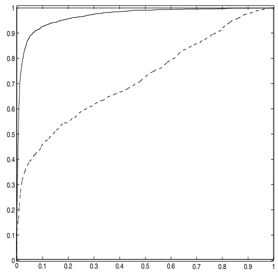

Figure 16: ROC curve illustrating comparative performance between a 58 × 58 linear kernel SVM (dashed line) and the full 14 component system with 5 level biasing (solid line).

In figure 17 we again see the same solid curve. The dashed line is now the exact same system as above, with the component classifiers replaced with component classifiers trained on non-facial negatives. The two systems are about on par in this performance measure.

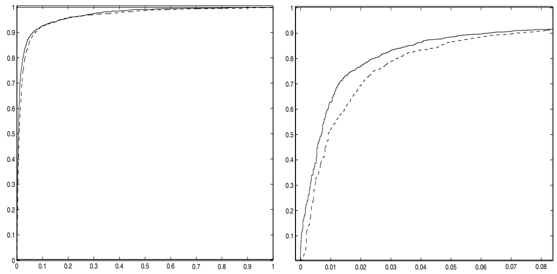

Figure 17: *Left:* ROC curve illustrating comparative performance between the 14 component system using facial negatives in the training set (solid line) and using non-facial negatives in the training set (dashed line). *Right:* Rescaled view of the graph on the left

In figure 18 are ROC curves for three systems which differ only in the biasing step. The solid curve near the bottom is from a system which is using no biasing at all. Performance is increased greatly by using a 5-level biasing routine on the result images. The system which generated the dotted curve uses first a 5-level biasing step and second a 1-level biasing step before the constellations are created. The reduction in performance is perhaps due to forcing the negative examples into constellations which look like they came from faces.

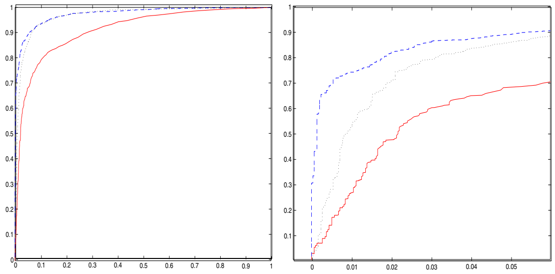

Figure 18: *Left:* ROC curve illustrating comparative performance between three systems which differ only in the biasing step. These systems all included the facial-negative component classifiers, and the histogram based constellation classifier. 5 level biasing *dashed line*, 5 level biasing followed by 1 level biasing *dotted line*, and no biasing *solid line* were all tested. *Right:* Zoomed in view.

5 Application: Eye Detection

As an illustration of the classifier training and biasing techniques described above, it was decided to apply the technology to the domain of eye detection. The goal was to construct an algorithm such that when input an image of a face the system would pinpoint the location of the center of the eyes.

5.1 Architecture

The eye-detection system works identically to the face detection system. Two of the 14 components are centered over the eyes. By simply outputting the positions of these components we are able to locate the eyes.

In order to bench-mark the system, it was necessary to construct another, more simple eye detection scheme to compare against. Two such benchmarking systems were built. The first system ran two classifiers, one for the left and right eye. It then extracted a list of the ten best local maxima across the scale space. These lists were then checked pairwise for good matches using the pairwise position statistics drawn from our artificial training data. The best pair was chosen based on how it fit the geometric constraint, and the strength of the detection in the eye classifier.

The second benchmark system started by searching the image for the position of the face. This was done by searching the image with a 19 × 19 polynomial face classifier trained on real images of frontal faces, as described in [1]. Once the best example of the face was found, a window around the expected position of the eyes was searched for the best example of the eyes. This pair of positions was reported by the system as the correct position of the eyes.

5.2 Performance

It was decided to use a subset of the labelled CMU PIE database [8], removing all heads turned more than 30 degrees out of the plane, leaving a total of 476 images. After correcting a very small number of mislabelled images, we benchmarked the system by recording the difference between the output and the human-defined ground truth.

In figure 19 are listed the mean Euclidean distances from the ground truth position. We see that the 14 component classifier is on average twice as close to the expected position of the eye as the classifier which searches first for the face. Both of these classifiers outperform the system which only searches for the eyes and chooses examples based on the geometrical constraint.

	Left	Right
Convolution and Constraint System	57.8	70.0
19 × 19 Face Detecting System	27.0	27.6
14 Component System with 5 Level Biasing	11.6	16.9

Figure 19: Table of the mean Euclidean distance from the ground truth position.

Although it might seem obvious, it is worth mentioning that the eye finder architecture is made more robust by searching for objects we normally find near the eye. As we add or remove component classifiers for the nose, mouth, etc. we can strike a balance between the desired accuracy and the required speed of the system.

6 Conclusions

While working with the component based face detection system in [2] we found that often component classifiers would maximize in the incorrect locations. By training component classifiers using negative examples drawn from the rest of the face, we were able to lessen the occurrence of such mistakes, and thereby make the system more robust. It is noteworthy that we built a face detector trained only on face images which outperforms a comparable system trained with non-face data.

Often when finding the best examples of the components in an image of a face, several of the components would classify in the correct positions while others would maximize elsewhere. This led to the idea of pairwise biasing, where classifiers would report their position to each other in order to find a set of positions which more closely match the geometrical relationships we expect from a face. It was shown that using the pairwise position statistics to bias the result images before calculating the constellations led to much improved face detection.

Finally we outlined the implementation of a robust eye detection scheme which used all 14 component classifiers in an attempt to both locate the face in an image, and pinpoint the center of the eyes. It was shown that by using the remainder of the face in a component based manner we were able to more accurately locate the center of the eye.

References

[1] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. A.I. memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.

[2] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 657–662, Hawaii, 2001.

[3] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. International Conference on Computer Vision*, pages 637–644, Cambridge, MA, 1995.

[4] E. Osuna. *Support Vector Machines: Training and Applications*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1998.

[5] Tomaso Poggio and Kah Kay Sung. Finding human faces with a gaussian mixture distribution-based face model. In *ACCV*, pages 437–446, 1995.

[6] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[7] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 746–751, 2000.

[8] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database of human faces. Computer Science Technical Report 01-02, CMU, 2001.

[9] S. Ullman and E. Sali. Object classification using a fragment-based representation. In *Biologically Motivated Computer Vision (eds. S.-W. Lee, H. Bulthoff ad T. Poggio)*, pages 73–87 (Springer, New York), 2000.

[10] R. Vaillant, C. Monrocq, and Y. LeCun. An original approach for the localisation of objects in images. In *International Conference on Artificial Neural Networks*, pages 26–30, 1993.

[11] V. Vapnik. *The nature of statistical learning*. Springer Verlag, 1995.

[12] T. Vetter. Synthesis of novel views from a single face. *International Journal of Computer Vision*, 28(2):103–116, 1998.

[13] Paul Viola. Complex feature recognition: A bayesian approach for learning to recognize objects. Technical Report AIM-1591, MIT, 11 1996.

[14] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision - to appear*, 2002.