EE368 Final Project – Face Detection

By Ping Hsin Lee, Vivek Srinivasan, and Arvind Sundararajan

1. Introduction

Face detection has been a fascinating problem for image processing researchers during the last decade because of many important applications such as video face recognition at airports and security check-points, digital image archiving, etc. In this project, we attempt to detect faces in a digital image using various techniques such as skin color segmentation, morphological processing, template matching, Fisher linear discriminant (FLD), eigenface decomposition, and support vector machines (SVM). We determined that the more complex classifiers did not work as well as expected due to the lack of large databases for training. Reasonable results were obtained with color segmentation, template matching at multiple scales, and clustering of correlation peaks.

2. Color Segmentation and Morphological Processing

The first step in our face detection algorithm was to segment regions in the image based on skin color. It is important to decide which color space to operate on in order to effectively segment skin color of different shades and under a variety of lighting conditions. While the images are conventionally represented in the RGB domain for display, the color and intensity components are correlated. Therefore, using the RGB color space would require some pre-processing [1] in order to reduce the effects of unwanted variations in intensity. On the other hand, both the YCbCr and HSV color models separate intensity and color information of the image effectively. Various studies[2] have shown that the skin color regions are highly correlated in their hue and saturation channels (HSV model) or chrominance channel (YCbCr model). The information of different shades is usually present only in the intensity channel (Y or V), and it captures variations in the lighting conditions. Therefore, CbCr or HS channels are used in skin color segmentation.

In this project, we decided to implement the MAP rule to achieve skin color segmentation. This technique requires determination of probability maps based on the ground truth data. The YCbCr representation is convenient for this approach because the color levels are discrete in this domain.

2.1 MAP Decoder for Skin Segmentation

A variation of the *maximum a posteriori* (MAP) probability decoder[3] is used to segregate non-skin pixels (NS) from the skin pixels (S). We define the decoder as follows:

$$D(I(x,y)) = 1 \text{ if } P(I(x,y) \mid S)P(S) > P(I(x,y) \mid NS)P(NS) \quad \ldots\ldots\ldots\ldots (1)$$

$$= 0 \text{ other wise,}$$

where

P(S) is the probability of skin pixels in a typical image
P(NS) is the probability of non-skin pixels in a typical image
P(I(x,y) | S) is the *a priori* probability distribution of pixels in skin region
P(I(x,y) | NS) is the *a priori* probability distribution of the pixels in non-skin region
I(x,y) = [Cr(x,y) Cb(x,y)], a vector of the two chrominance components

Using the ground truth data from the training images 1 through 4, probability maps of the chrominance components for skin and non-skin pixels are determined. In figure 1, the probability maps of P(I(x,y) | S) and P(I(x,y) | NS) are plotted. The MAP rule exploits the peaks of the two distributions and the distance between them. An additional advantage of using the MAP decoder is that the decision boundary between skin and non-skin regions does not need be linear (Fig 2).

We modified decision rule to include a threshold to reduce the error of misclassifying skin regions as non-skin regions.

$$D(I(x,y)) = 1 \quad \text{if } P(I(x,y) \mid S)P(S) > T*P(I(x,y) \mid NS)P(NS) \qquad (2)$$

The threshold T is determined heuristically for the best performance in skin segmentation.

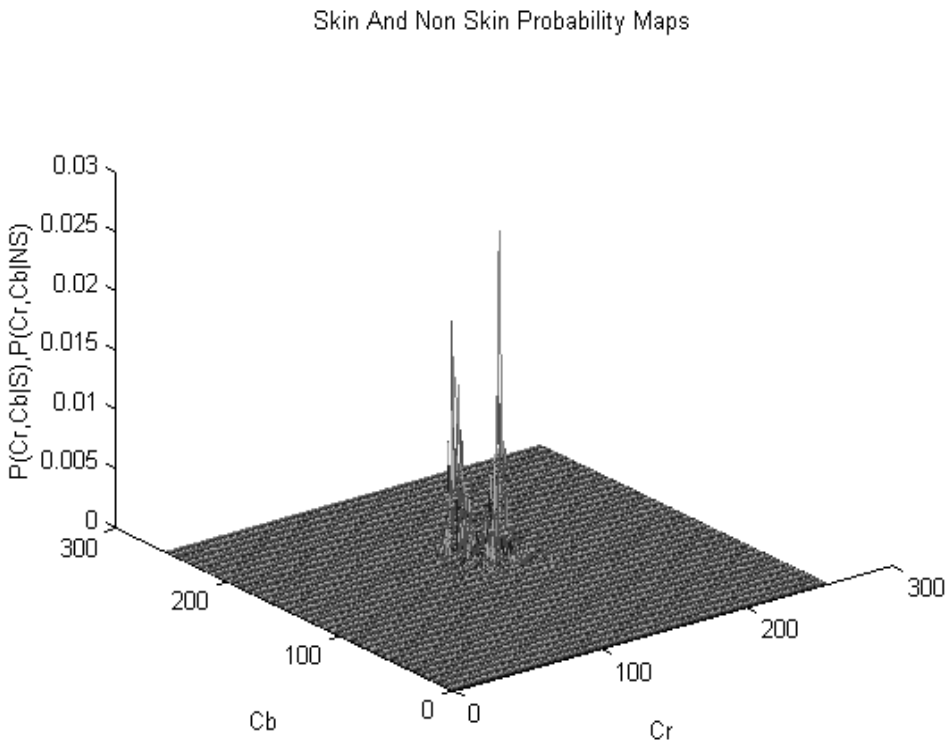


Fig. 1 - Probability maps of skin and non-skin regions
The two peaks correspond to skin and non-skin (larger peak) pixels.

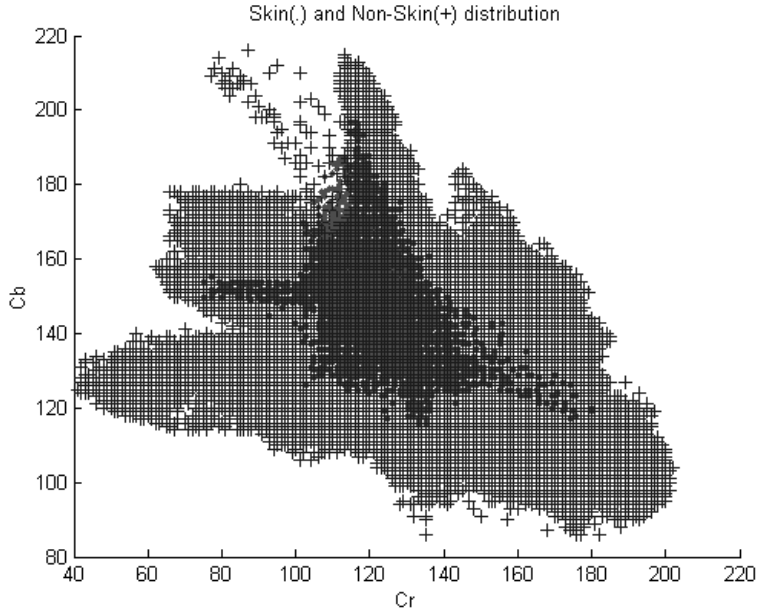


Fig. 2 - Distribution of skin and non-skin pixels
Cr and Cb have been translated from [-128:127] to [0:256].
Decision boundary between skin and non-skin pixels is non-linear.

Fig. 3 - Training_5.jpg from the training data set

Fig. 4 - Image after skin segmentation

2.2 Results of Skin Segmentation

Once the probability maps are obtained, the decision rule (2) is applied to each pixel of the image to classify it either as a skin or non-skin pixel. In order to speed up the process, the original image is subsampled by a factor of 4 before the skin segmentation routine is applied. Figures 3 and 4 show the results of skin segmentation. After skin segmentation, the image mask is up-sampled back to the original image size for further processing.

As we can see in figure 4, most of the face regions pass through the skin segmentation routine along with some background noise. We now filter out the background noise by rejecting connected region (50x50 square pixels) that are smaller than a typical face region. After the routine, features such as the eyes appear as holes in the face objects. We will make use of these features in the next step of our algorithm, which is template matching. In order to recover these features, a closing operation is performed using a structure element of size 5x5, followed by hole removal. The result of these operations can be seen in figure 5.

Fig. 5 – The mask after rejection of small blobs, closing and hole removal

2.3. Hand and Arm Rejection

After blob rejection and closing, several non-face regions remained in the image. These regions included arms, hands, and legs. These areas would present problems in the template matching because they cause high correlation peaks. In order to reject these regions, a method based on standard deviations was devised. The general idea was to make reasonable assumptions about the shape, size, and location of possible faces in the image, and reject all blobs that did not fit these assumptions.

Each blob was first characterized by the maximum ratio of standard deviations in perpendicular directions ($\sigma_{max}/\sigma_{min}$). This ratio is the maximum over all possible orientations. For example, $\sigma_{max}/\sigma_{min}$ could be obtained by rotating a blob by all angles about its centroid and finding the maximum ratio σ_y/σ_x over all angles of rotation. $\sigma_{max}/\sigma_{min}$ is the same for any shape regardless of its orientation, and indicates how long a shape is. For instance, a circle has $\sigma_{max}/\sigma_{min} = 1$, while this same quantity is larger than 1 for an ellipse. In order to reject arms, only blobs with relatively small $\sigma_{max}/\sigma_{min}$ were retained. The choice of threshold here was difficult because, although a face has a standard deviation ratio just above one, several faces grouped together as a blob can have a standard deviation ratio that is much higher. Using a combination of size, location, and the aforementioned $\sigma_{max}/\sigma_{min}$ to characterize the blobs, we ensured that the chance of losing a face in this process was minimized. Finally, small blobs corresponding to hands that appeared relatively low in the image were rejected.

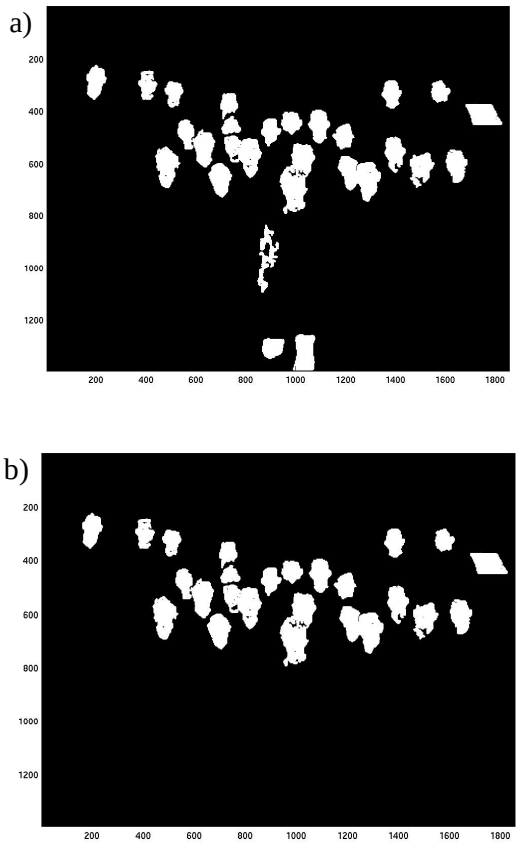


Fig. 6 - Example of blob rejection based on $\sigma_{max}/\sigma_{min}$
a) Mask before $\sigma_{max}/\sigma_{min}$ based rejection
b) Mask after $\sigma_{max}/\sigma_{min}$ based rejection and location-dependent thresholding

3. Template Matching

After color segmentation, blob rejection, closing, and hole filling, template matching was used to isolate faces from non-faces. The template matching was performed in the luminance domain, using the FFT. Because the template size was small compared to the image size, using circular convolution instead of linear convolution introduced negligible error.

The initial template was an average of all face regions in the image. From the reference data, the centroid of each face region was computed, and a rectangular region of pixels about the centroid was extracted for each face region. These regions were then averaged and mean subtracted. Regions in the rectangular template deemed to be background were manually set to zero after mean subtraction so that they would not affect the correlation result. This template performed reasonably well, and did not require multiresolution template matching, since faces of all scales appeared in the template. A disadvantage of this template was that features were not sharply defined. As can be seen below, the first template was not "face -like" in appearance.

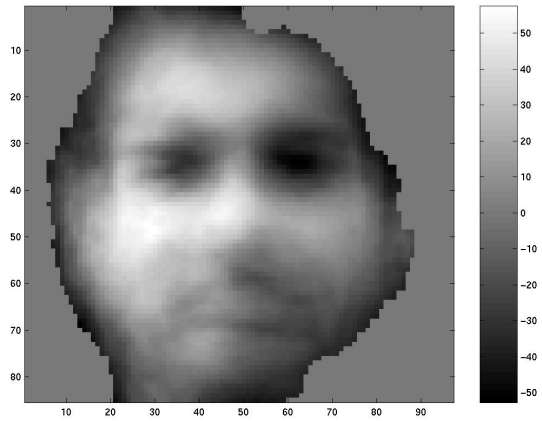


Fig. 7 - Face template based on extracting face regions from the image test set without resizing. (Not used in final version)

The final method of template creation was to resample each face region to the same size before averaging. This caused the features in the different faces to align better when the averaging operation was performed. As can be seen below, this template looks more "face-like" in appearance. For this method, because every face used in taking the average was resampled to the same scale, multiresolution template matching was used.

A symmetric template (Figure 8a) was obtained by including the mirror image of every face region in the averaging process. The symmetric template yielded sharp correlation peaks near the center of most faces, and performed better than an otherwise identical non-symmetric template. This may seem somewhat surprising since the symmetric template does not incorporate luminance differences due to lighting from one side. However, the symmetric template, while losing information about the lighting, performs better in isolating symmetric regions. Furthermore, the lighting was not very dramatic for the test images. (Note: the symmetric template background is *not* zero. Reasonable results were obtained using this template, so the background was not set to zero.)

In addition to the symmetric template, a downsampled, non-symmetric, partial template (Figure 8b) was used to capture information about smaller and partially obscured faces in the image. The template size was chosen to match faces near the back of the test images. Because larger faces obscure the mouth and chin of many smaller faces in the test set, the smaller template was chosen to match only the eyes and forehead. Since the smaller template was *not* symmetric, its lighting matched the lighting in the test image.

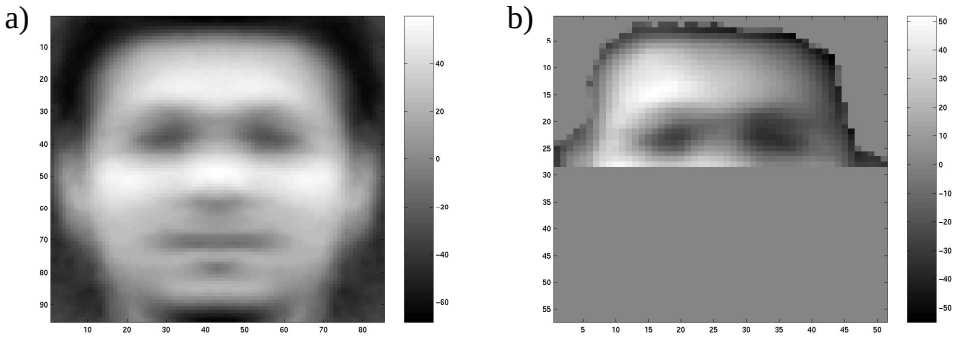


Fig. 8 - a) Symmetric template obtained by extracting face regions from the image and resampling. b) Non-symmetric partial template based on a similar technique. Both templates were used in the final version.

Next, correlation results obtained from each of the two templates were separately thresholded and combined. Peaks appeared at or near the centroid of most candidate face regions (Figure 9a). In addition, some smaller peaks appeared at boundaries between faces and at face edges. These occurred due to off-center regions that matched one of the two templates. Many of these peaks were eliminated by thresholding on pixel size.

3.1 Grouping Correlation Peaks (Clustering)

In order to group peaks, information from the color based segmentation mask was used. If the centers of two peaks were sufficiently close together and a line between the centers did not pass through any non-skin regions, then the two peaks were grouped together as belonging to the same face. Various heuristic techniques based on morphology of the skin regions were also used in the grouping process. The final face locations were the cluster centroids.

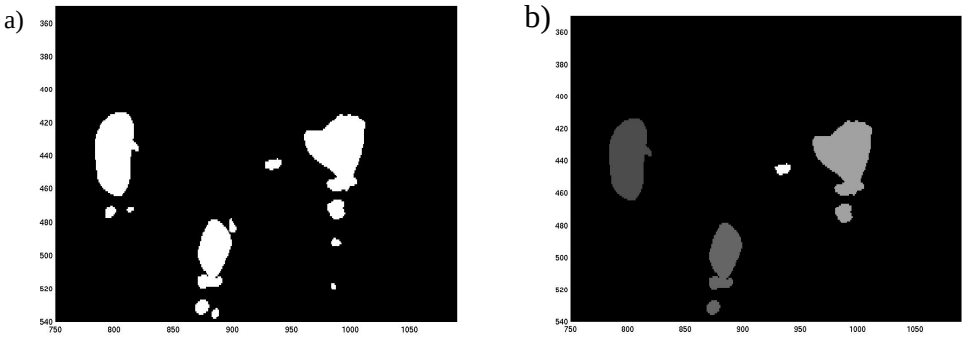
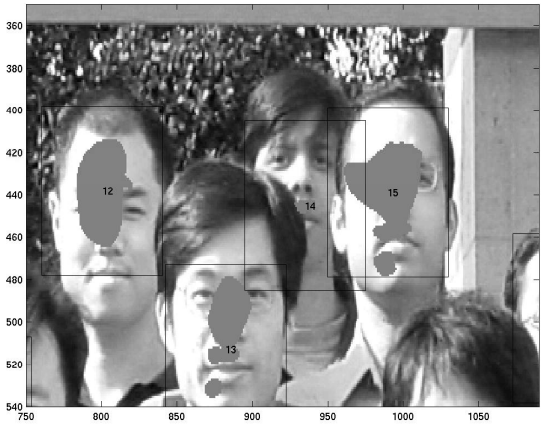
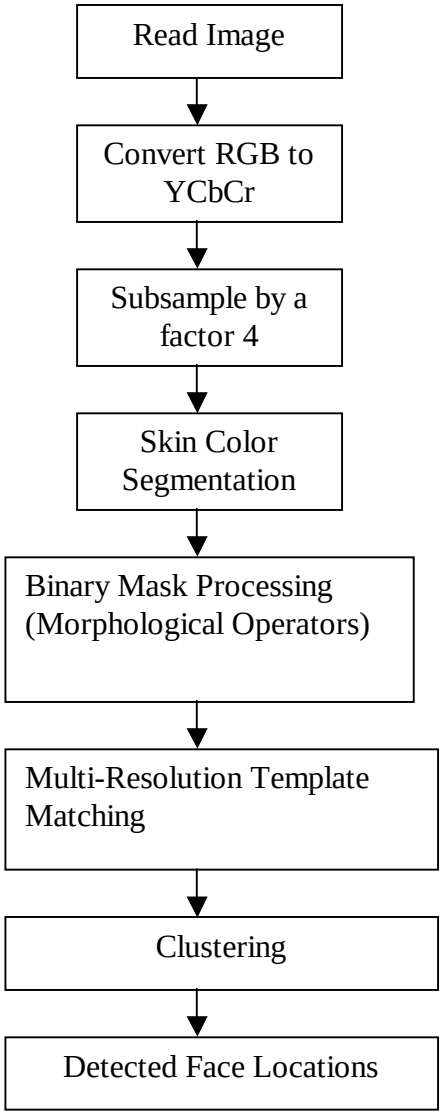


Fig. 9 - Grouping of correlation peaks
a) Peaks in the image after correlation and thresholding
b) Grouping of peaks after morphological processing
c) Detected faces

c)



4. The Algorithm Flow Chart



4.1 Final Results

The final results for the seven test images are shown below.

Image	1	2	3	4	5	6	7
Score	20	22	24	22	24	24	21
Total	21	24	25	24	24	24	22

5. Schemes That Didn't Quite Make It: Fisher Linear Discriminant, Eigenfaces and SVM

Apart from connected pixel analysis a few other techniques were tested to reject false positives like hands, arms and legs.

5.1 Fisher's Linear Discriminant

Fisher's linear discriminant technique is a popular scheme to segregate patterns that are linearly separable. It is based on projecting the pattern vectors from a large dimensional space onto a line thereby reducing the N-d classification problem to a 1-d classification problem.

A database of 500 samples each of 60x60 face images and non-face images were created and subsampled to 24x24. The samples were generated by extracting windows from the original data set. The non-face images largely comprised regions from the images that showed up as faces post template matching. The linear discriminant is computed as follows [4]:

$$S_1 = \sum (X - M_1)(X - M_1)^T$$
$$S_2 = \sum (X - M_2)(X - M_2)^T$$

$$S_w = S_1 + S_2$$

$$M_1 = \text{mean}(X)$$

$$M_2 = \text{mean}(Y)$$

$$W = S_w^{-1}(M_1 - M_2)$$

The X's are the face vectors, the Y's are the non-face vectors and S_w is the within Scatter Matrix. S1 and S2 are correlation matrices for the two classes of vectors and W is the fisher's linear discriminant.

If the projected component of the object on Fisher's linear discriminant ($W^T X$) exceeds a certain threshold then the object is classi fied as a face. Otherwise it is rejected.

This technique had a very poor performance in rejection of false positives possibly because detected non-face regions and face regions are not linearly separable.

5.2 Eigenfaces

A variation of template matching was also attempted in order to reject the false positives. All faces were extracted using the provided ground truth data on the first 4 training images. These face images were then interpolated onto a uniform 95x85 cartesian grid. Using Sirovich and Kirby procedure [5] the first 94 eigenimages were computed. Ten of these eigenfaces corresponding to the 10 largest eigenvalues captured most of the energy and hence were used in the analysis. This process was repeated at two more scales of 124x111 and 67x59.

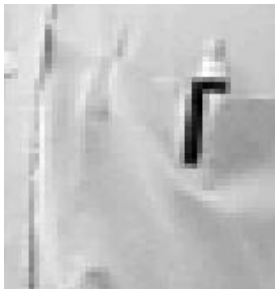
Two different schemes were attempted, to reject non-face objects obtained after template matching. The first one was to compute the total energy contained along the first ten eigenvectors and reject objects that were below a certain threshold. The other scheme was to maintain a database of face and non-face projections and use the nearest neighbor criteria in rejecting false positives. Both schemes gave results similar to template matching and at an increased computational cost and hence were discarded.

5.3 Support Vector Machines (SVM)

A support vector machine is a learning computer algorithm used in pattern classification problems. The principle behind SVM is to find an optimal hyperplane such that the two different regions are separated as far as possible to minimize errors. In our face recognition application, we want to construct a function $f(\mathbf{x})$ such that $f(\mathbf{x}) = 1$ for a face region and $f(\mathbf{x}) = -1$ for a non-face region. Therefore, $f(\mathbf{x})$ can be constructed in the form $f(\mathbf{x}) = \sum y_i * a_i * k(\mathbf{x}, \mathbf{x}_i) + b$. We input training samples in the form of $\{\mathbf{x}_i, y_i\}$ where the $\mathbf{x}_i$'s are the training input vectors and the y_i's are input values taking on either 1 or –1. The $k(\mathbf{x}, \mathbf{x}_i)$ function is called a kernel. There are different types of kernels ranging from the linear function (dot product of x and x_i), a polynomial function, and the radial basis function involving Gaussian distributions. Basically, the algorithm calculates the scalar values a_i's and b based on the training samples given and thus the $\mathbf{x}_i$'s associated with the nonzero a_i's are called support vectors.

Here are two examples of the training face images we input into the SVM:

And here are two examples of the training non-face images we input into the SVM:



In our project, we implemented a SVM with a radial basis function (RBF) kernel. The function $f(\mathbf{x})$ has the form $f(\mathbf{x}) = \sum w_i * G(\mathbf{x}; \mathbf{c}_i, \sigma_i^2) + b$ where G is the i^{th} Gaussian basis function with center $\mathbf{c}_i$ and variance σ_i^2. The w_i's are weights and b is the bias term to produce a scalar output. Based on the training samples and the number of basis functions that are supplied, the SVM calculates all the parameters in $f(\mathbf{x})$. To train the SVM, we generated 470 face regions and 500 non-face regions from the first 4 training pictures, each of size 49x55 pixels. We chose various basis function numbers of 1, 8, 16, 20, and 36. After the parameters of $f(\mathbf{x})$ are generated, we then tested it out in our program after skin segmentation and template matching. We hoped that this rejection method would further eliminate all non-face regions to produce an improved face recognition result. However, after implementing and adjusting different factors such as size of training regions and number of training samples, we did not achieve satisfactory results. This is because the SVM produced decision regions that are too tightly bound to the training face samples and were not able to classify the faces in the other 3 training pictures. We considered generating more training samples, but we realized that the time it would require to train the SVM would be prohibitively long. Since we had already achieved adequate results from skin segmentation and template matching, we decided that including the SVM in the program would only slow down our runtime and would not produce noticeable improvements.

6. Conclusions

Because large databases of face and non-face regions were not available, our implementations of techniques such as FLD and SVM worked about as well as template matching. In addition, eigenface methods yielded no noticeable improvement over template matching. Ideally, neural networks, FLD, SVM, or MRC (maximal rejection classification) could be used after color segmentation and template matching to reject the false positives, non-faces classified as faces. However, the absence of large databases to train these classifiers degraded their performance. Instead of these techniques we used a highly heuristic and scale-specific method of clustering correlation peaks.

7. References

1. M.Hunke and A. Waibel, “Face Locating And Tracking for Human Computer Interaction”,in *Proc. Conf. Signals Systems and Computers*, Nov 1994, vol2 pp 1277-1281.

2. D. Cahi and K.N. Ngan, “Face segmentation using skin color map”, IEEE *Trans. On Circuit and Systems for Video Technology*, vol 9, no 4, pp. 551-564, Jun. 1999.

3. Abbas El Gamal, “Introduction To Statistical Signal Processing”, Lecture Notes.

4. Richard Duda, Peter Hart and David Stork, “Pattern Classification”, 2nd ed., John Wiley and Sons.

5. Sirovich, L. and Kirby, M.: 1987, Low dimensional procedure for the characterization of human faces, *J. Opt. Soc. Am. A* **4**, 519–524.

6. Gender classification with support vector machines, www.merl.com/reports/docs/TR2000-01.pdf

7. Support Vector Machines (SVM), http://www.site.uottawa.ca/~stan/csi5387/SVM-transp.pdf