EE368 Final Project: Face Detection

Group 6

Anthony Guetta
Michael Pare
Sriram Rajagopal

Introduction

The group project for the EE368 class involves the detection of face regions in a digital color image. The background of the problem is as follows. A set of images is taken with students (and professors) of the class standing in different configurations for the pictures. The scenario of the images, i.e. the background, is the same, while the lighting varied a slight bit due to cloud cover on that day. The problem defined is to detect all face regions in the image and to possibly classify them as males or females.

The approach followed is to segment the image into skin like and non-skin like regions based on the color information. Subsequently, morphological processing is applied to obtain expected face centers which are then validated by performing template matching and eigenface decomposition.

Initial Mask Creation: Color Segmentation Using RGB Vector Quantization

In order to greatly simplify further processing of a given input image, it is desired to segment the image such that only face pixels are retained. Two approaches were examined in an effort to achieve this goal. The first was global thresholding in HSV and YCbCr color space and the second, and subsequently adopted, was RGB vector quantization.

Our first task was to analyze the color space data in the given training images. All combinations of dimensions in each of RGB, HSV, and YCbCr color space were examined. Inspection of the results revealed that the clustering of face data points was highest in Cr vs. Cb space (Figure 1), while the least overlap between face and non-face points was apparent in V vs. H space (Figure 2).

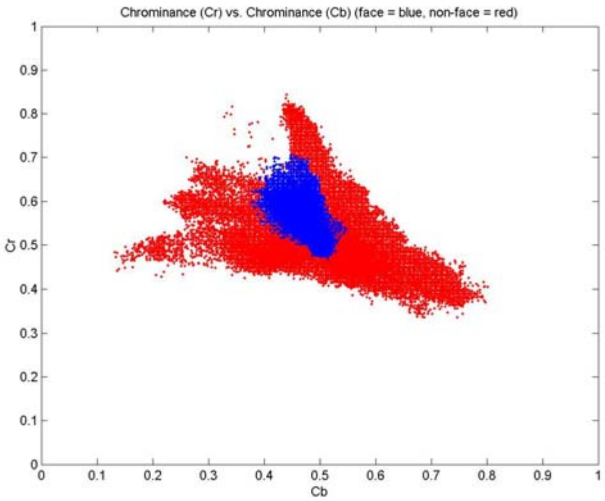


Figure 1: Clustering of training data in Cr vs. Cb space.

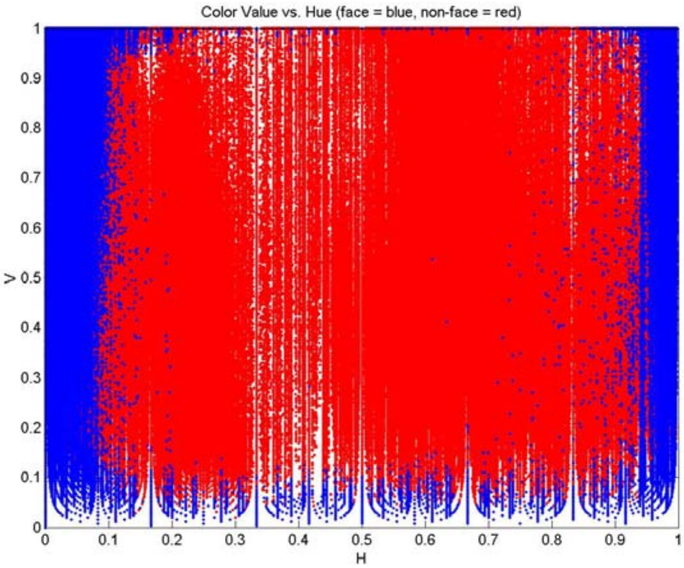


Figure 2: Clustering of training data in V vs. H space.

Although there is clearly significant overlap among face and non-face data in both of the above spaces, an attempt was made at bounding the face regions by simple linear equations. Global color segmentation could then be carried out by ensuring that a given pixel in the input image falls within the range of what are deemed as face pixels. Figure 3 shows an example mask image (binary image consisting of only zeros and ones which results from the thresholding) obtained from one of the training images. It was generated using the following bounding equations,

```
H < 0.07
H > 0.92
Cr > (-0.9000*Cb + 0.9100)
Cr < (-2.3855*Cb + 1.8368)
Cr < ( 1.4500*Cb + 0.1000)
Cr > ( 2.0133*Cb - 0.5754)
```

where H indicates hue and Cr and Cb indicate the respective chrominance components.

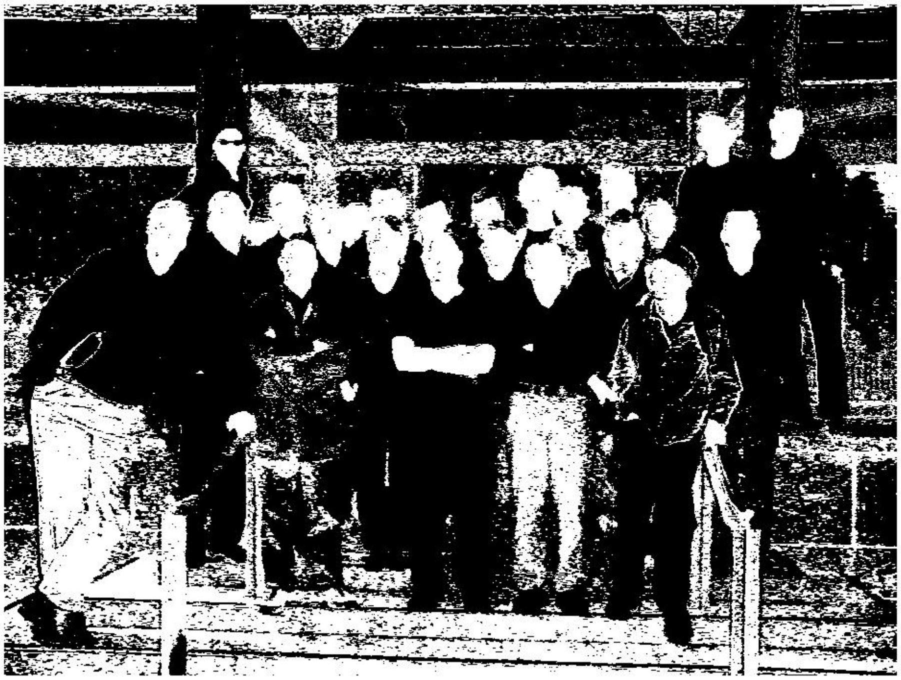


Figure 3: Mask image generated by global thresholding.

Global thresholding clearly retains face pixels and rejects a portion of non-face (background) pixels. However, it is obvious that this technique's advantages in simplicity do not outweigh its shortcomings in results. Much of the background has been passed by the threshold, being mistaken as face data. This is especially true of brighter colored areas like the stairs, beams, and pants. We thus conclude that an alternate approach to global thresholding is required.

In the preceding analysis, we completely neglect the RGB color space, and for good reason. If we inspect the figure below, we see that there is a very large amount of overlap of face and non-face data. As a result, global thresholding in RGB space yields poor results.

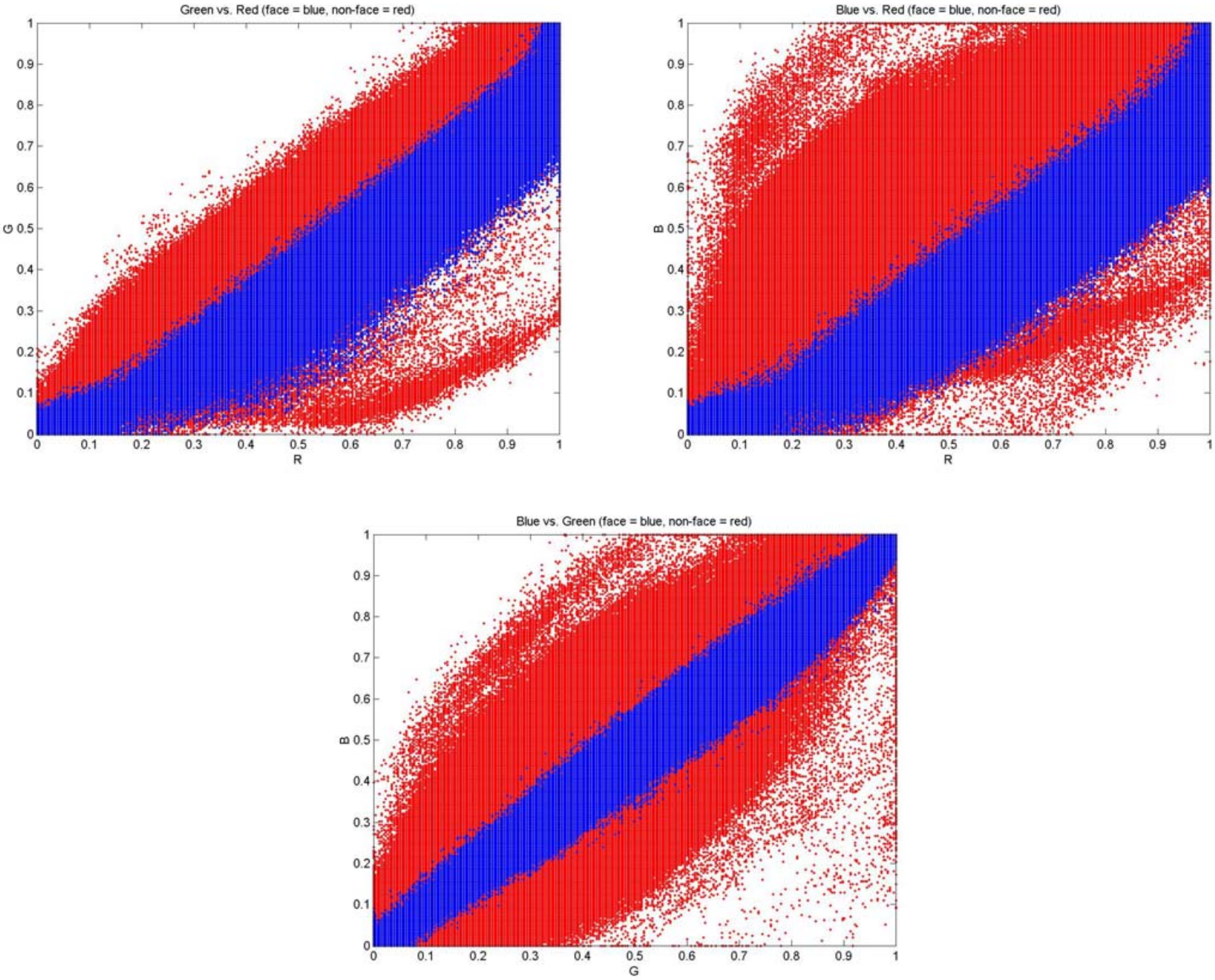


Figure 4: Clustering of training data in all RGB dimensions.

At first glance, one would think that color segmentation based on RGB values holds no promise. And, as mentioned above, this indeed holds true for global thresholding. But what if we imagine a situation where we limit the number of possible RGB triples? In other words, we consider quantizing RGB space, where we turn our attention away from thresholding and instead view a given set of RGB values as vectors in 3-space. Can we find a way to quantize RGB 3-space such that face pixels tend toward certain quantization levels while non-face pixels tend toward different levels? With a vector quantization algorithm developed by Linde, Buzo, and Gray known simply as the LBG algorithm [2,3,4], we can achieve exactly this.

Given a desired number of quantization levels, N, and a (preferably large) set of input vectors, the LBG algorithm partitions k-space (where k is the length of the input vectors) into N distinct regions each with a single quantization level, or code vector. The code vector is ideally the average of all the input vectors in its partition. Furthermore, the partitions are such that any input vector in a given partition is closer to the code vector of that partition than to the code vector of any other partition. The set of all final code vectors is termed the code book.

Using the LBG algorithm, we devised separate code books for both face and non-face data, with the respective RGB pixels obtained from the training images as our input vectors. "Face space" was quantized to 8 levels and "non-face space" to 32 levels. Given the code books and an RGB vector, we could then determine whether a pixel corresponds to a face or a non-face simply by computing the minimum distance between the RGB pixel vector and the code vectors in each code book. Figure 5 shows the result of segmenting the image of Figure 3 using the RGB vector quantization approach.

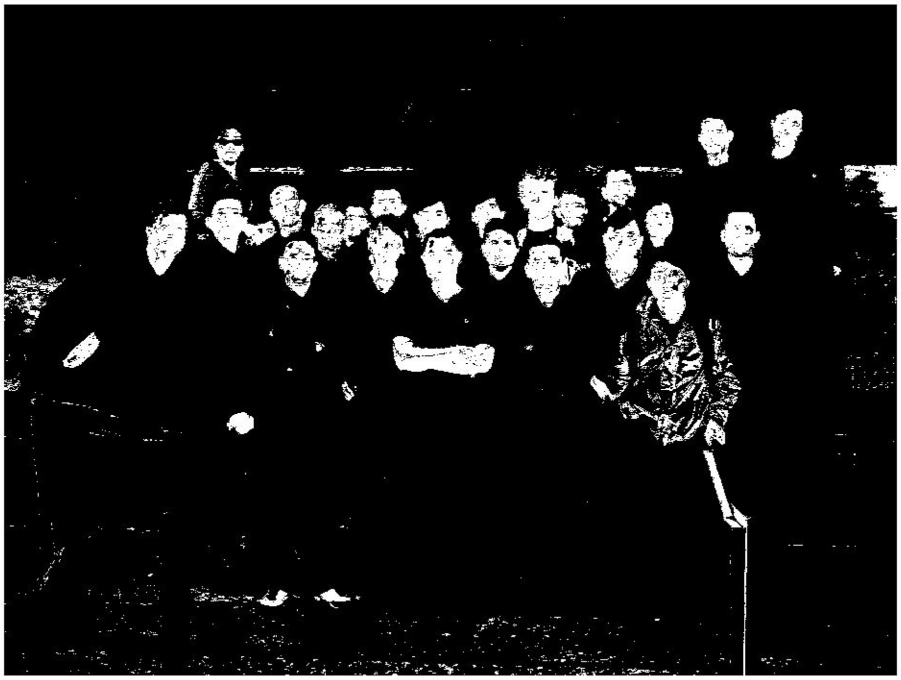


Figure 5: Mask image generated by RGB vector quantization.

The background rejection using the RGB vector quantization approach is much greater than that of simple global thresholding. However, two minor issues remain. First, the faces are somewhat spotty. This problem can easily be solved by morphological processing (which is discussed further in the next section). The second problem results mainly from bright colored clothing such as purple, green, and red. Certain regions of such colors are not fully rejected by the initial quantization. Since these colors are not "face-like", however, we expect that generating separate vector code books for them using the LBG algorithm (where 4 quantization levels are chosen) will solve the problem. As can be seen in the following figure, which shows the final masked grayscale image, it clearly does.

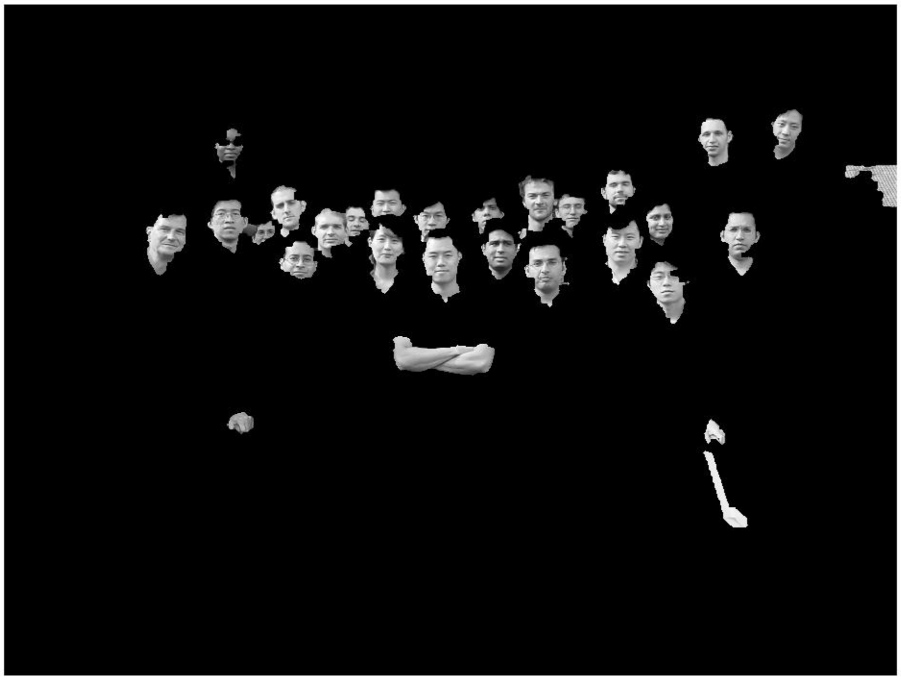


Figure 6: Masked grayscale original image obtained by RGB vector quantization with problem area code books and subsequent morphological processing.

Processing the Initial Mask Image

The first step in processing the mask is an initial “cleaning”. Image opening operations are performed to get rid of "salt and pepper" noise and closing operations to fill in holes in the faces. The mask image obtained from the initial color segmentation also includes a few non-skin parts of the background image. However, most of these are distinguishable from flesh-like regions by their shape. The flesh regions mostly consist of faces, arms, and fists. The shape of faces and fists is usually oval. The non-flesh regions, on the other hand, were typically elongated in the horizontal or vertical direction. Such regions consisted mostly of the bright edges of the clothes and the wall edges. Hence, for each blob in the initial mask, the ratio of width to height is found. The regions which are elongated in either the horizontal or vertical direction by a factor of 2.9 are rejected from being a face candidate.

The only remaining concern can be the rejection of a face region when there are overlapping faces. People standing on the different tiers of the steps will not contribute to a vertically elongated shape because the faces will usually be separated by the hair on the lower person’s head and hence will be recognized as separate face candidates. The face regions of people standing on the same row are always separated at least by the length of a shoulder. Therefore, it is expected that the aspect ratio threshold will not discard any genuine face candidates. The color segmentation also breaks up non-face regions into smaller speckled regions. These are cleared up by thresholding the area of the blobs. Regions with small area are rejected. In this manner, a cleaner mask image is generated for further morphological processing, the goal of which is to pinpoint face centroids. Figure 6 (above) shows the result of this preprocessing of the initial mask image.

Morphological Erosion of the Mask Image: Finding Centroids

Once the final binary mask image has been obtained, the goal is then to differentiate among the various flesh-colored objects which happen to be connected in the mask. When two faces are next to each other, for example, their corresponding white areas in the mask will be connected. Doing a simple labeling of connected regions in the image would then count the two close faces as just one face. This is the problem we need to avoid. We want to separate the flesh regions of the mask into smaller regions, each corresponding to a single flesh-colored object in the image. This will produce a new binary image with which we can then use to determine the centers of each flesh-object.

The main method used for this separation process is the erosion operation for binary images. However, this method is highly sensitive to the shape and size of the structuring element used. A very large structuring element will entirely erase some mask regions, whereas a very small element will have little effect at all on the mask. Also, the aspect ratio of the element determines whether more erosion happens in the vertical or horizontal directions. A wide element will reduce the horizontal width of a flesh-object mask much more than it will reduce the vertical span, and the opposite holds for taller structuring elements. If two flesh-objects are stacked on top of one another in the mask image, then an erosion with a wider element will separate the mask into two distinct regions. A taller element used in that case may not separate them as one would desire.

In order to strike a balance between separating horizontally and vertically connected mask regions, we did extensive experimentation on the training set. Noticing that there is in general more horizontal connectedness between faces, there is a preference for taller structuring elements. But at the same time, it must be wide enough to separate flesh-objects that are connected vertically. It seemed under our training conditions that the best ratio of vertical to horizontal was 5:2, a little taller than the general shape of a face.

As for determining the actual size of the structuring element, it is most crucial not to lose entire regions of the mask. Therefore, we implemented an iterative approach to erosion. That is, we used a relatively small element, but eroded the eroded image until we finally achieved the separation we were looking for. Because the smaller mask regions will be erased much sooner than the larger ones in this process, whenever after an iteration a given connected region of the mask contains less than a certain number of pixels, it is saved in a separate mask image. This way all of the small mask regions — each only corresponding to a single flesh-object — are stored in a new mask image.

The image resulting from the above process is a mask where each white region is a small 'blob' in the center of a flesh-object. As we want to pinpoint the center of each face, the next step is to simply compute the centroids of each blob. This is done by averaging the set of row values and column values for points in that region. We can then use these centroids as the defining center-point for each flesh object.

In some instances, there may be multiple centroids calculated within a given face. This can happen if the erosion process splits a single mask-region into two pieces before it is small enough to be saved to the new mask. This occurs when the jagged edges of the flesh object erode in towards each other and touch before the central blob in considered a representative of just one object.

Consequently, we must discard the extra centroids which we consider to be derived from the same face. As there are usually only one to three centroids in a face, all quite clustered, the discarding is effectively accomplished by checking for centroids within a certain radius of each other and taking the one that represents the largest central blob. This is a sound method, because in general, the larger the blob, the more central its location within the face.

Once the parsing of the centroids is complete, we are left with a set of single points, each at the center of a flesh-object. Now we need to decide if those points are the centers of faces or just the centers of some other flesh-colored objects.

Face Template Matching and Thresholding

The basic idea of face template matching is to find regions which are similar to the template. Hence, generating a template which looks like a face is important. The template must be such that it rejects the non-face regions in the image. The template that was used for matching is shown in figure 7. This is the mean image found from all the 145 men and 19 female faces in the given set of training images.

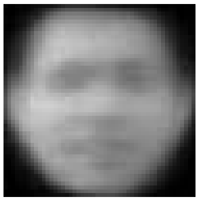


Figure 7: Mean-face used as template (enlarged)

The size of the actual template is 50x50 pixels. Larger templates yield low correlation for smaller faces, while the smaller template is able to yield high correlation for both the small and large faces.

In an effort to achieve optimal results, multi-resolution template matching was attempted. Every pixel in the mask image was considered as the center of a candidate face. Multiple size images were then extracted using that pixel as the center of such an image and correlation was performed with the template. The size of the extracted image, which matches the size of the underlying face, was expected to give the maximum correlation. However, in extracting larger sizes, parts of nearby faces are included in the image. This tends to affect the correlation and yields higher values for larger sizes irrespective of the underlying content in the image. As a result of this, only a 50x50 template is used for the final template matching scheme.

Implementation of template matching is similar to convolution. Most faces yield good correlation results. However, some common problems were identified over the set of training images. For example, the people in the top part of the picture were usually in the shade and, as a result, have lower correlation values. But the correlation is able to pick up the shape of the faces even in the shade. Some bright objects on the left and right edges of the image actually give large correlation values. These values are localized and are not due to the shape of a face, but have more to do with complexity of information in the region. For example, the corrugated roof in the right corner of some of the training images experiences a large correlation value. Similarly, crossed arms in some images yield large correlation values. Finally, we note that, although in general the face template successfully rejects hands, those that are not rejected can be expected to be in the lower part of the image.

Most of these problems are generic across all training images and hence a spatial weighting function for the correlation values is suggested. The values in the upper part of the image are boosted and those in the right, left, and bottom parts are attenuated. The 2-dimensional weighting function is shown in figure 8 below. A fourth order function is used to boost poorly correlated values at the top and a sixth order decay to attenuate values at the bottom. The correlation values in the central part of the image are retained. This function takes care of the illumination adjustment for the images. A sample correlation with the template is shown in figure 9. It is evident that the face regions are picked up by the correlation operation. The morphological processing yields good results for expected face centers. The correlation values at those locations are used to decide between the face and non face regions. This includes some neck regions and fists. Hence, an intelligent threshold is set to accept the face regions and reject the rest. The results of this operation are very consistent and are exact nearly always.

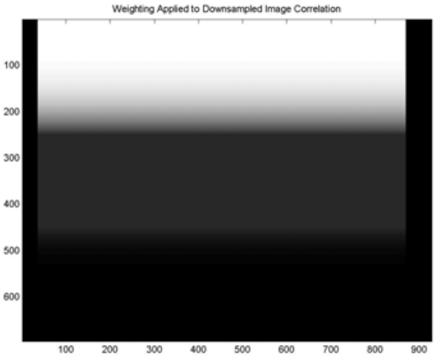


Figure 8: Weighting function for correlation values.

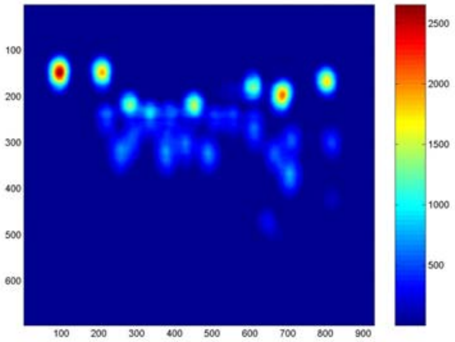


Figure 9: Sample correlation result.

Eigenface-Based Detection

Eigenfaces form a basis for decomposing the faces in the images provided. As a result, every given face can be expressed as a linear combination of these eigenfaces. The technique of generating the eigenfaces is well known and an efficient method of implementation, the Sirovich and Kirby method, is used [5]. The advantage of the eigenfaces is that they pack maximum energy in the least number of coefficients. Hence, only the first 8 eigenfaces were used for face projections. They are shown in figure 10.

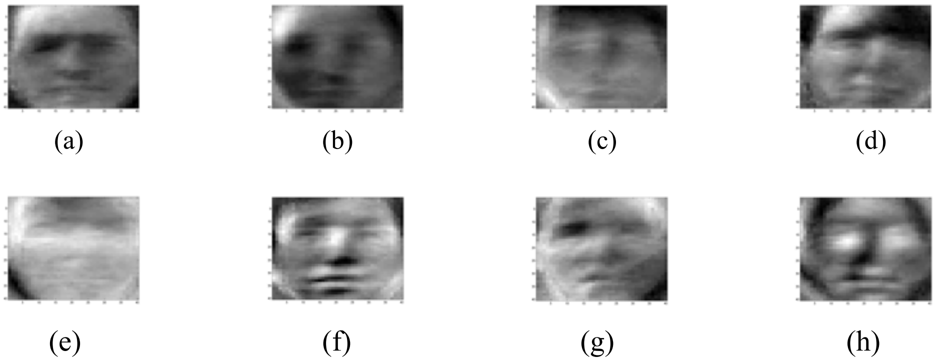


Figure 10 (a-h): Eight eigenfaces.

The faces were used from the training set provided. The content of the faces used for generating the eigenfaces was varied. For example, a 50x50 image, which fitted each face from forehead to chin and ear to ear was used initially. The problem now lies in detecting faces from the mask image. When a candidate face is cut out from the image, it will include parts of other objects and when multi-resolution operation is performed, the candidate image is completely distorted. Eigenface decomposition is highly sensitive to the data provided to it. Thus, small spatial variation in the candidate face will throw the vectors far away from the expected face vectors. Therefore, a conservative approach is used in extracting the candidate faces from the mask image. This region will include a square from above the eyebrow to below the lips and extending to the edge of each eye.

Figure 11: Candidate Face extraction (a) Conservative and (b) Including surrounding regions

For each extracted image, the eigencoefficients are found and the distance to the closest face is calculated. This distance is used as a metric for determining a face or non-face region. When a large size candidate face is extracted from the image, the non-face regions have large distances to the face vectors. However, some of the connected faces also have the same problem because of distortion form the nearby faces. When a conservative approach is used, the face regions have a lower distance but the fists also get mapped to the face region because their shape also fills the image and the knuckles look like eyes.

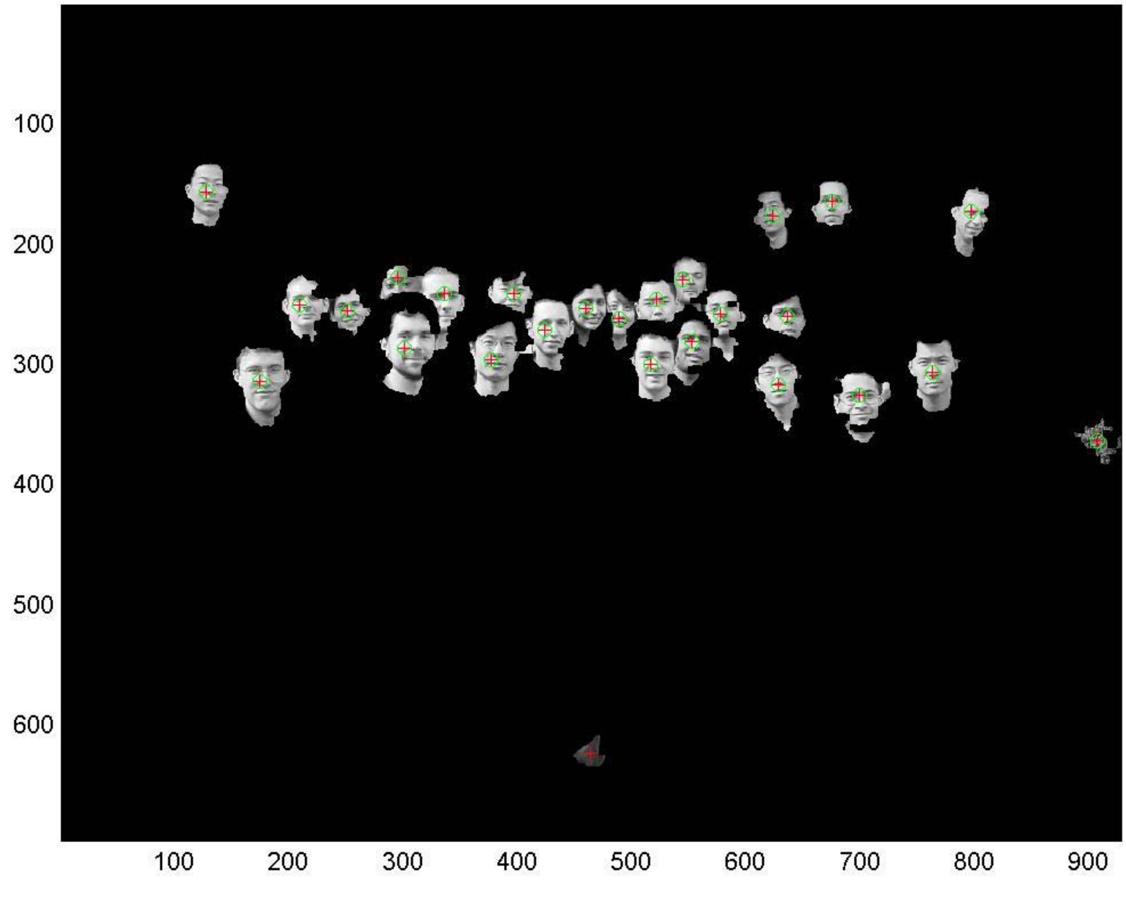


Figure 12: Sample result of Eigenface-based detection. Red '+' are estimated centroids from morphological operation. Green 'O' are detected faces.

Gender Recognition: The Headband Template

At first, using either the eigenfaces technique or a correlation with a gender-specific template seemed the most logical approach to determining face-gender. However, after seeing some of the poor correlations with just an average face-template, there was no way we could differentiate genders based on those techniques. We needed to find another feature which could give us a more definite gender distinction, and that is exactly what we found in the headband.

Given that we had relatively few females in our training images, a specific feature such as the wearing of a white headband could allow us to distinguish a large percentage of the women in the group. Therefore, we simplified the problem of locating females to locating a white headband.

Doing a correlation with a zero-mean headband template seemed the best technique. The template is designed using the average of the headband regions from the given training set, and is scaled to best approximate the expected size in the ¼ scale image we are correlating with. There are a few problems encountered, however. First, the correlation value at the actual headband location is only a local maximum, and not the best correlation in the image. This is because there are many areas of the image which have a white area with a darker area beneath it, and this is exactly what the template is looking for.

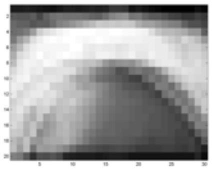


Figure 13: The headband template.

To solve this problem, we created a mask image using a correlation value threshold. This created a white region anywhere the correlation was above a certain value. Most areas that were not near the headband could be easily discarded by simply labeling the connected regions and removing those with a size larger than that expected for the headband location itself. Large mask regions corresponding to light-colored beams with dark areas below them are thrown out, leaving only the small local maximum regions of correlation.

The next step is to see if any of these local maximum locations are just above one of the face-centroids found earlier. If so, we can then distinguish that face as female. So, we first use as a reference point the centroid of each correlation-maximum region in the thresholded mask image. From that point we simply define a small rectangular region below it in which to look for face-centroids. If we find one, we assign it as female.

In order to avoid a handful of false-positive classifications as female, we had to ensure that the threshold of correlation was high enough to overlook the same type of pattern elsewhere in the image. We were conservatively high in thresholding, because there were many more males in the images. Also, the upper right side of the image often posed a problem, because anyone with dark hair standing there would have a light sky behind them, leading to a strong correlation just above their face. Consequently, any local maximum which occurs there must be instantly ignored to be on the safe side.

After completing the gender-recognition using the headband template, we now have all of the face centers pinpointed with genders assigned. The result is ready for display.

Conclusion

The task of face detection in a digital image is a well established problem. There are many approaches which all try to achieve the same end result: efficiently detecting all human faces in a given image and rejecting everything that is not a face. Given the constraint for the runtime of the detection algorithm (about 7 minutes), we decided to implement computationally inexpensive methods. The initial step of vector quantization of the RGB values yields excellent results for the subsequent processes to work on. The morphological processing gives an estimate of the centroids of the faces regions, even for connected faces in most cases. The illumination corrected template matching yields near perfect results. The results of the eigenfaces were found to be comparable to template matching, but inferior in some cases due to its strong dependance on spatial similarity to the known face dataset. The results for the seven training images are tabulated below. We were able to obtain roughly 95% accuracy and an approximate average running time of 75 seconds.

Training Image	Final Score	Detect Score	Number Hits	Num Repeat	Num False Positives	Distance	Runtime	Bonus
1	22	21	21	0	0	15.9311	71.91	1
2	22	21	23	0	2	13.6109	82.96	1
3	25	25	25	0	0	9.8625	80.48	0
4	22	22	24	0	2	11.3667	81.15	0
5	24	24	24	0	0	9.5960	69.59	0
6	23	23	23	0	0	11.5512	80.25	0
7	22	21	21	0	0	14.1537	71.52	1

References

[1] Gonzales, R. & Woods, R. (2002). Digital Image Processing. Prentice Hall, Inc., pp. 331-335

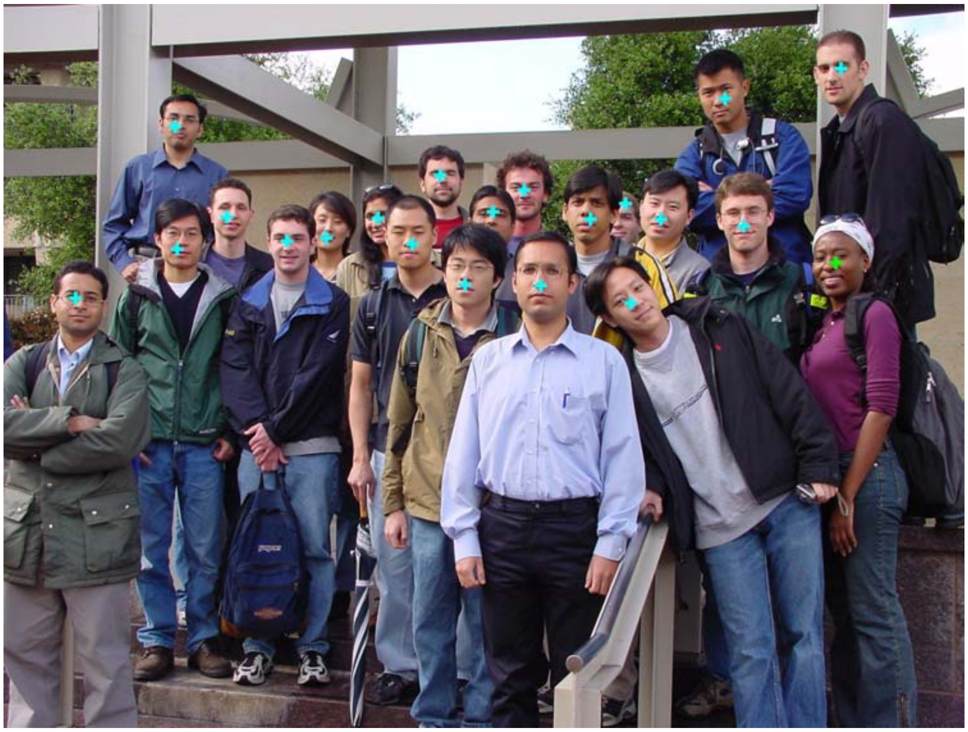
[2] Linde, Y., Buzo, A. & Gray, R. (1980). An Algorithm for Vector Quantizer Design. IEEE Transaction on Communications, 28(1), pp. 84-94

[3] Patane, G. & Russo, M. (2001). The Enhanced LBG Algorithm. Neural Networks, vol. 14 no. 9, pp. 1219-1237

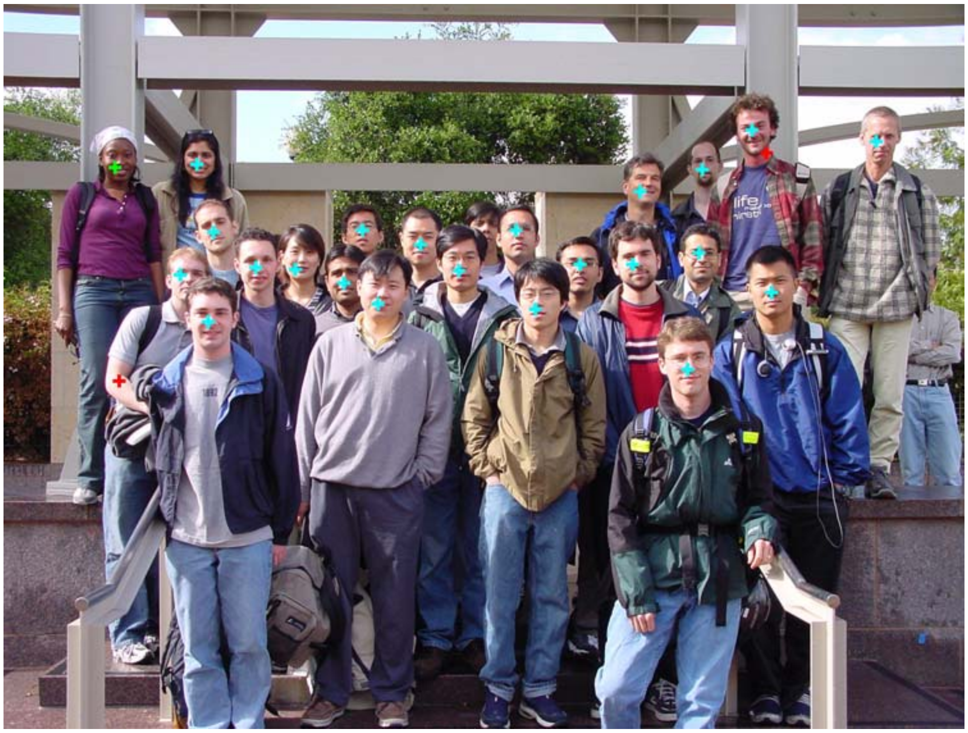
[4] Phamdo, Nam. Vector Quantization. www.data-compression.com/vq.htm

[5] EE368 Lecture Notes, Professor Bernd Girod, Stanford University

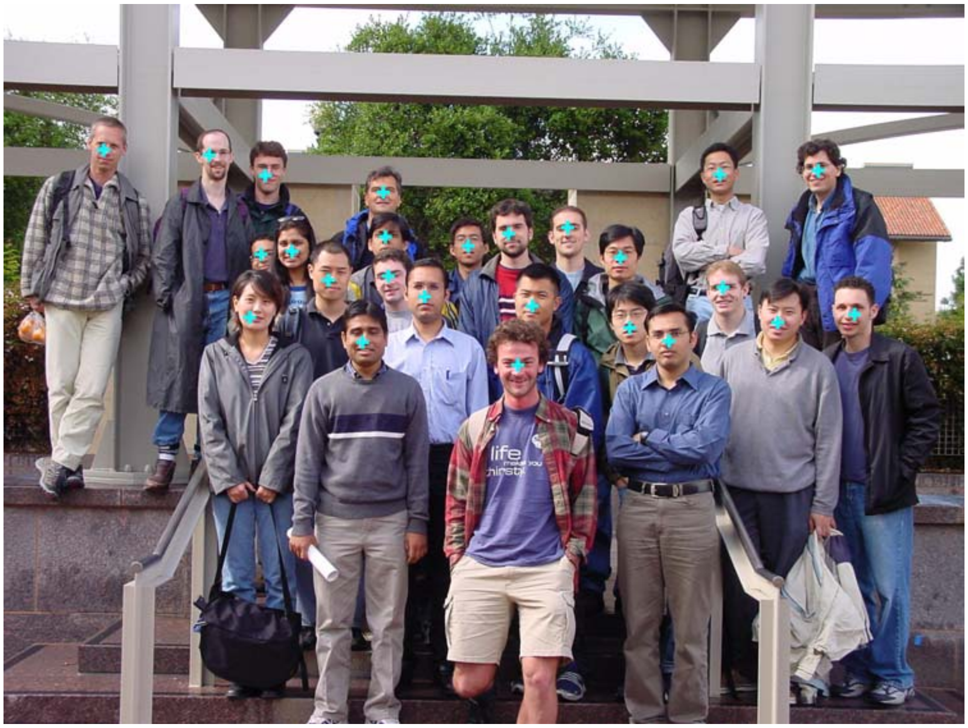
Results of Testing



Training 1

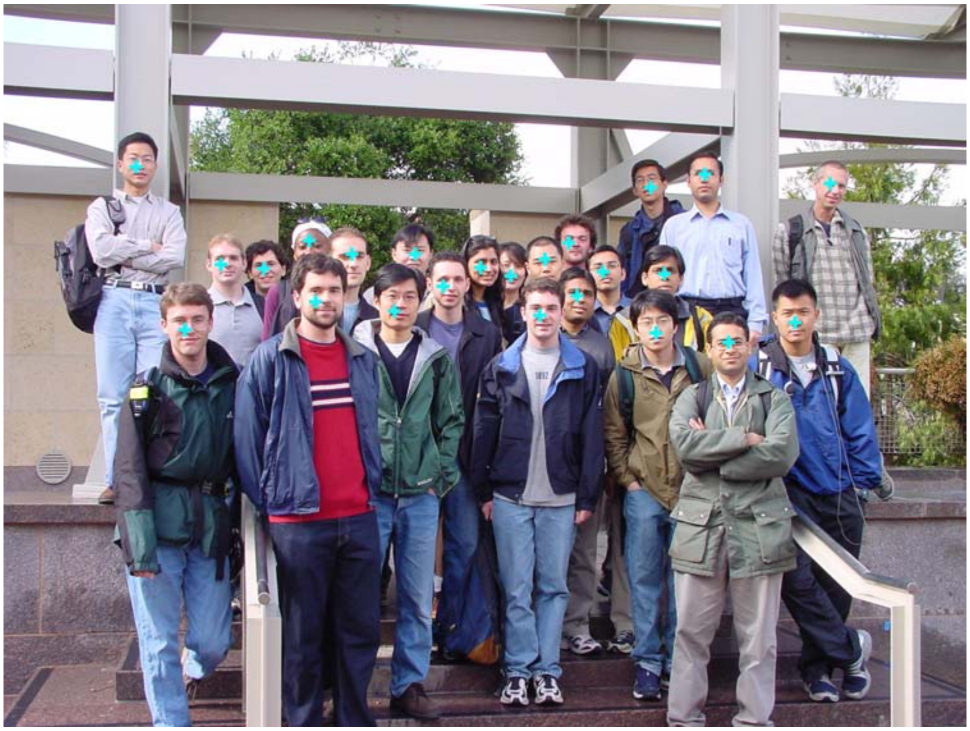


Training 2

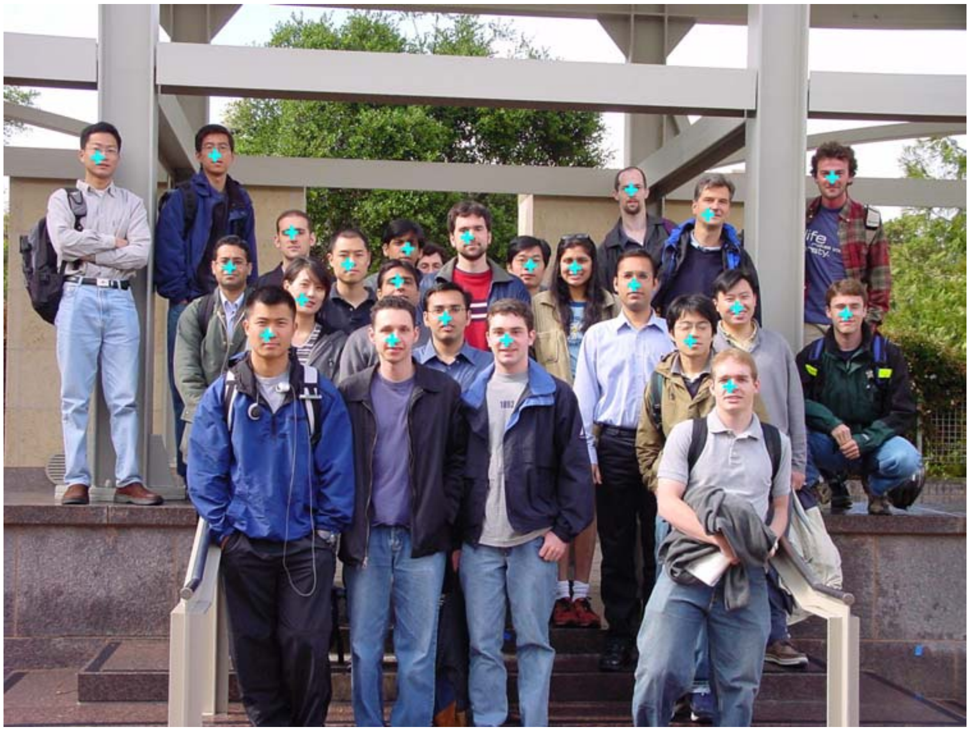


Training 3

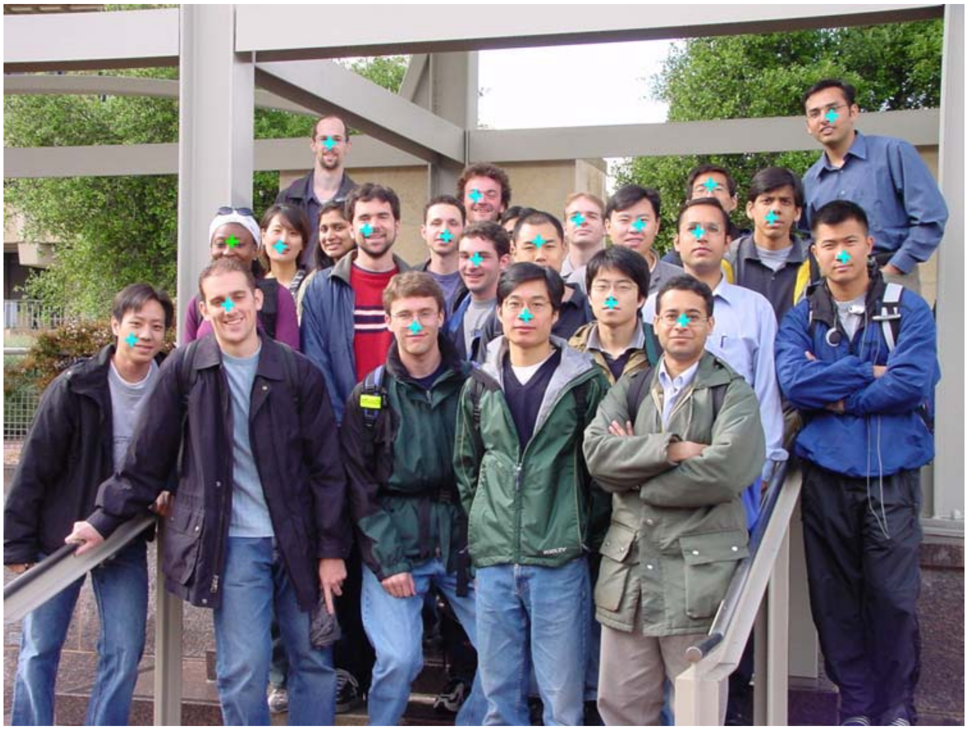
Training 4



Training 5



Training 6



Training 7