CS 479/679 Pattern Recognition
Spring 2019 – Prof. Bebis
Programming Assignment 2 - Due: 3/26/2019

1. In the previous assignment, you designed a Bayes classifier assuming the following 2D Gaussian distributions:

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

   In this assignment, you will assume that you do **not** know the **true** parameters of the Gaussian distributions and that you need to estimate them from the training data using the Maximum Likelihood (ML) approach.

   a. Using the same 200,000 samples from the previous assignment, estimate the parameters of each distribution using ML. Then, classify all 200,000 samples assuming P (ω_1) = P (ω_2); count the number of misclassified samples and compare your results to those obtained in assignment 1.

   b. Repeat experiment (1.a) using 1/100 of the samples from each distribution (randomly selected) to estimate the parameters of that distribution using ML. Then, classify all 200,000 samples assuming P (ω_1) = P (ω_2); count the number of misclassified samples and compare your results to those obtained in experiment (1.a).

2. Repeat problem 1 using the samples (same as in Assignment 1) from the following 2D Gaussian distributions:

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}$$

3. Face detection using **skin color** is a popular approach. While color images are typically in RGB format, most techniques transform RGB to a different color space (e.g., chromatic, HSV, etc.). This is because RGB values are more sensitive to changes of brightness due to illumination changes.

   a. Implement the skin-color methodology of [Yang96 “A Real-time Face Tracker”] which uses the chromatic color space. To build the skin color **model**, use *Training_1.ppm* (and *ref1.ppm*), shown in Figure 1, which are available from the course’s webpage. To **test** your method, use *Training_3.ppm* (and *ref3.ppm*) and *Training_6.ppm* (and *ref6.ppm*), which are also available from the course’s webpage. To quantitatively evaluate the performance of your method, generate **ROC** plots (i.e., false positives (FP) vs false negatives (FN)) by varying the skin-color threshold. A FP would be a non-face pixel which was classified as skin-color while a FN would be a face pixel which was classified as non-skin color. To compute the FPs and FNs for each test image, use the corresponding reference images.

b. Repeat (3.a) using the YCbCr color space In the YCbCr color space, the luminance information is contained in Y component; and, the chrominance information is in Cb and Cr. Therefore, Y should not be used in the skin color model. The RGB components can be converted to the YCbCr components using the following transformation:

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = -0.169R - 0.332G + 0.500B$$
$$Cr = 0.500R - 0.419G - 0.081B$$

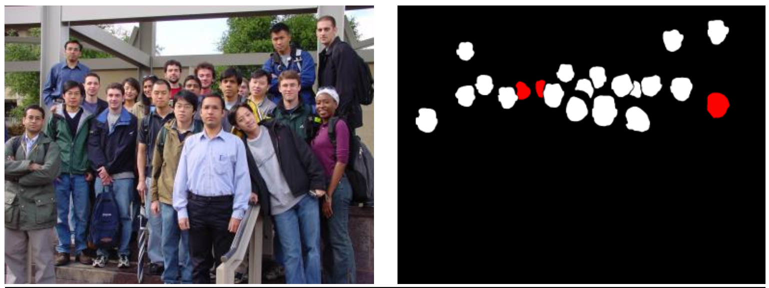

Figure 1. Training_1.ppm and *ref1.ppm* images.

For comparison purposes, plot the ROC curves in the same graph.

Note: Irfanview is a nice tool for image display/manipulation. Sample code to read/write color images in PPM format can be found in my CS 302 webpage:

http://www.cse.unr.edu/~bebis/CS302/

Information on the PPM image file format can be found here:

http://paulbourke.net/dataformats/ppm/

http://www.cse.unr.edu/~bebis/CS302/Lectures/IP.ppt