Extra Credit

Your grade will not be lowered if you skip this assignment!

Assigned: 12/03/02

Due: 12/15/02

Skin Pixel Detection

This problem uses support vector machines (SVMs) to build classifiers. You may use any available SVM software package to prepare your solutions; for a list of options, see

http://www.kernel-machines.org/software.html

If you've never used SVM's before, a good starting point would be Dr. Gavin Cawley's Matlab SVM toolbox, available at

http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox/

Specify which SVM package you use, and submit any code that you write yourself to determine your solutions.

a) Write a function which allows you to manually select subsets of pixels from images. One easy way to do this is to use Matlab's *ginput* function to select rectangular regions.

b) Extract 10,000 (or more) RGB triples corresponding to skin pixels. Try to use a wide range of skin colors and lighting conditions. You can use the `faces.jpg` image on the web site, or any other convenient images. Divide these examples into equally sized training and test sets.

c) Find at least 10,000 RGB triples corresponding to non-skin pixels. You can easily do this by subsampling pixels from images which don't contain people. You may use the `forest.jpg` and `clutter.jpg` images, or any others that are convenient. Divide these examples into equally sized training and test sets.

d) Fit a single Gaussian density to each of the training sets from parts (b) and (c). Use these density estimates to build a classifier as in Forsyth & Ponce, Algorithm 22.2. Using the test sets, calculate and plot the receiver operating curve (ROC) for this classifier.

e) Using the same training sets, build a standard linear SVM classifier. Most SVM packages allow you to specify the amount of weight placed on errors with respect to the two different classes. By retraining the SVM using different weight settings, construct an ROC curve for the linear SVM. Compare this curve to the Gaussian classifier of part (d).

f) Repeat part (e) using third-order polynomial and radial basis function kernels (this should only require trivial modifications to your code). Again construct the ROC curves, and discuss your results.

g) Find an image which contains people, as well as interesting background clutter. Run your classifiers from parts (d)-(f) on this image. Choose a point on the ROC curves based on the approximate percentage of skin pixels, assuming all errors are equally costly. For each classifier, plot a binary image where skin pixels are white and background pixels black. Which types of pixels are most difficult for the classifiers?