Optical Character Recognition and Pattern Classifiers

Optical character recognition is the process of imaging a document, recognizing individual letters and then outputting the text of the original document (for example in ASCII format). OCR software is common, but is far from perfect. This project will explore some of the issues associated with OCR processing. These issues are common in machine vision systems, and include: segmentation and blob formation, feature extraction, and pattern matching. To make this project simple, we will use a single blob feature (your choice). Model a class $i$ with a mean, $m_i$, and variance, $v_i$. Use the Mahalanobis distance to classify a measurement, $f$, into a class $i$: $d_i = (f - m_i)^T v_i^{-1} (f - m_i)$.

In this project you should try to distinguish two letters. Use your first and last initials, ‘F’ and ‘D’, for example. Create three images: two for class modeling and one for testing. The images used for modeling should contain examples of a single kind of letter (all ‘D’s, for example). The testing image should contain examples of both kinds of letters. Vary the typeface (bold, italic) and font size for the letters. When creating a test image, include 15+ occurrences of a given letter (this can be done with ‘Paint’ for example).

Prior to modeling or classifying letters, your program should segment the letters from the background via a threshold, then isolate all the blobs and compute a feature for each one. As we will only use measurements with one feature, you will only need to find a mean and variance in order to model each letter class. Once the classes are modeled, you can then identify each letter in the testing image. After you get your program working, then test it again with a more challenging pair of letters, ‘O’ and ‘D’, for example.

Your program should generate an output that indicates the letter classification in the testing image. Do this by setting the gray level for each letter, based on its classification.

Report
1) Program listing for ee528.cpp, with comments, and file header giving your name, course and project number.
2) Brief report including:
   a) Your name, the course title, and number of this project.
   b) Hardcopies of output images.
   c) Describe your feature.
   d) Report the mean and standard deviation for each class. Report the success rate of the classifier.
3) Answers to questions.
4) An executable version of your program, with input file(s).

Questions
1) Given the following patterns for a class: $\{ [1,1]^T, [1,4]^T, [2,1]^T, [2,4]^T \}$ Find the mean vector, $m$, and covariance matrix, $C$, for the class. Find the inverse of the covariance matrix, and then compute the distance, $d$, to the class for the measurement $f = (1,2.5)$ using $d = (f - m)^T C^{-1} (f - m)$.
2) Use the method of Lagrange multipliers to find the largest possible circle, centered at (0,2) that fits inside a triangle with vertices (-2,0) (2,0) (0,4). Hint: begin by drawing a picture!
3) Given a binary edge image consisting of a square (100x100 pixels) that is randomly oriented, what is the anticipated pattern of peaks on the Hough space.
4) Given that binary data needs to be processed by a median filter, suggest a more efficient algorithm than the method based on a sort, discussed in class.