CSCI Project: Test 1 – Labeling Paths and No paths in an image

In this test you will use Support Vector Classification and K-Nearest Neighbor to learn a classifier for paths in an image. The code that you will need to implement is in the ProjectCode directory. You will need to implement your own version of K-Nearest Neighbor. Please use the matlab version of LIBSVM. The following link contains all LIBSVM information:

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

The matlab version can be obtained from a link on this page.

To debug your code, please run the file Run_Test_1_Code_Log.m. This will let you label an image with path and no path areas, send the data to your learning algorithm, and then use the resulting model on all future images.

By the due date, please upload your ProjectCode directory (zipped version) to moodle.

There are two main matlab functions that you will need to implement

1. Build_Model.m: This builds either an SVM model or a K-Nearest Neighbor Model given the training data. You need to experiment with various kernels and K values. You should use the following steps for choosing the best model (keep in mind the time it takes to build a model - 15 minutes at most - and the time it takes to evaluate it when choosing which model you will use):

    a. Take half of your labeled data and use it for learning, and the other half for testing.

        i. For picking and SVM kernel and kernel parameters (i.e. the learning parameters), divide the learning data equally into a training set and validation set. Use the training set to build the model given a set of learning parameters, and the validation set to pick the learning parameters that give the lowest error (on the validation set). Using these learning parameters on the entire learning set to build a single model. Then calculate the error rate of this SVM model on the test set. Please report which learning parameters you tried and why you tried these.

        ii. For K-Nearest Neighbors, use N-fold cross validation on the learning data (where N is the total number of learning examples) and calculate the error on the test set. What range of K did you search an why?

    b. Choose the model (either SVM or K-Nearest Neighbor) that has the lowest error rate on the test set. Report which model you choose and why?

2. Apply_Model.m: This algorithm applies the model you constructed above to successive images. Note that this model should be fairly fast to evaluate—Try to construct one that runs at a few frames per second.