Lab 5: Features

Cristian M. Amaya
Universidad de los Andes
cm.amaya10@uniandes.edu.co

1.. Introduction

Features is one of the most complex information we can get from an image, it's not intuitive how to approach problems with features as is not as simple as extracting pixel values. We can understand features as recurring patterns that have similar characteristics as orientation, gradient change and colors.

In this laboratory, we intend to classify a database into 25 feature categories training a KNN (K Nearest Neighbor) classifier with training data. To represent the training data, we create a filter bank of textons to generate histograms and train the classifier.

2.. Methods and Materials

2.1.. Database

The database has 1000 images, separated in two categories: training and test. The training sets contains 30 images for each of the 25 semantic categories while the test category only has 10 for each, the images are of the same size and are of different ripes of features at different orientations.

2.2.. Textons

Textons are the atoms of features, are the basic patterns that combine can generate more complex patterns, they are usually lines, dots and spots with different sizes and directions. Using textons we can identity the difference between local patters, using a similar technique of template matching, we can observe the response of filtering and decide which textons can better describe a pattern.

A library of functions for Matlab was used to generate the filter bank and generate the textons base in the training data.

2.3.. K Nearest neighbor

The nearest neighbor method is one of the most basic method in classification, in this case it takes into account the k nearest neighbors to decide the label of new data that enters the representation space. To compare between the training data, the Chi-square distance between histograms was used.

2.4.. Subsampling stategy

To optimize computation times, both sets were divide in groups and process in different orders. In the training group, 30 groups were formed taking one image of every categories each, with this groups different classifiers were trained. In the other side, the test data was divided in 10 groups of 25 image each, their annotation and predicted categories for each of the 30 classifiers were too and save for latter processing. all this sub sampling was necessary, as the servers and computers couldn't take the 750 images of training and assemble the filter banks and a unique classifier, or classify directly the set of test images.

2.5.. Code strategy

Three separated codes were develop to satisfy the subsampling problem, one would train the different classifiers, the other would classify the test images with the previous classifiers and filter banks, and lastly the las code would construct the confusion matrix based in the predictions. To allow communication between codes and avoid time loss do to Matlab problems, variables were constantly save in Matlab own file format (.mat).

3.. Results and Discussions

As it can be appreciated in the figures 1 through 3, the Average Classification Accuracy was of 39,2%, this was probably do to a problem in subsampling the training sets and how it was decided which category each image belong to.

With 30 classifiers, you obtained 30 prediction per test image, for time constraints, the mode of this 30 predictions was used as the label for each image. This didn't take into account the accuracy of each prediction, which could improve the result. Training one classifier with the right parameters could produce a more accurate prediction.

Additionally, there was some parallel work between building the classifiers and using them instantly to classify the test images, for this reason it was necessary to assemble the responds of different groups of classifiers in the final code (Confusion.m) to assemble the final confusion matrix.

4.. Conclusions

To construct a filter bank as big as we did, takes a big computational time, and based in the used of k means for the nearest neighbor method, it last even more as it tries to converge every time.

Perhaps the use of less images, the change and optimization of parameters and a more advance classification method would have produce a better result.

Images

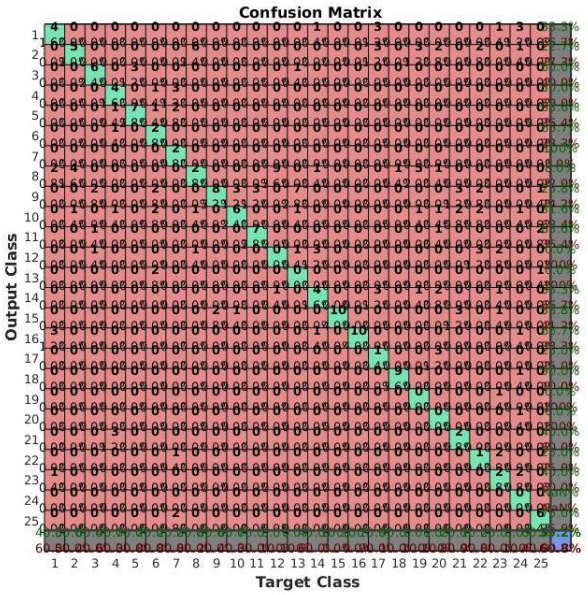

Figure 1. Full confusion matrix

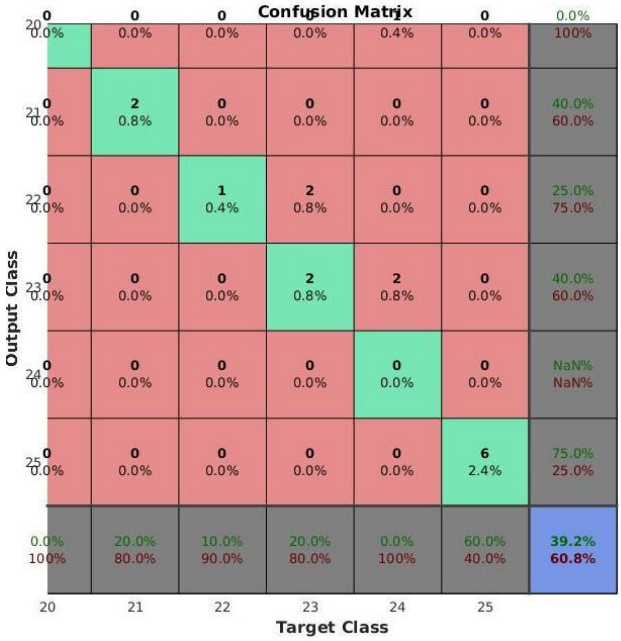

Figure 2. Closer look at the confusion matrix

Figure 3. Average Classification Accuracy