We divide the training images into two parts, one for feature extraction and one for feature selection. The ration is 1:1.

We resize the images to around 30000 pixels each. And for each window we calculate the eight statistics of glcm and the 6 statistics of the color. Then use k-means to cluster (是每一个image cluster 100个吧？)the 14 dimension windows to get 100 clusters means for each image.

Here are the comparisons of using 5 clusters and 100 clusters. Clearly, 100 clusters contain much more information. The cluster means are the feauters, which we use to form the dictionary.

We calculate the Euclidean distance between the 100 cluster means in each learn\_set images with the words in the dictionary. If the distance is smaller than 1, we considered them as the same features. As a mention, we first scale the 14 variables so that they have the same weight when calculating the distances. Now we get a huge matrix of the numbers of words in dictionary contained in each image in the learn\_set.

Input the matrix into random forest do get the selected features. We select the features which can give a more than 0.03 mean decresement of Gini index. Get about 2500 features for 1000 images.

For the model training part, we use SIFT and GBM to get about 0.72 accuracy, but the feature extraction is very slow. For GLCM and Color and GBM we get around 80% accuracy. And it takes about 20 minutes. For GLCM and Color and neural networks, the accuracy is also around 80% but it’s unstable, (when you input a matrix with a column of all 0 numbers, can’t get the answer 括号这句对不对？) For GLCM and Color and PCA-based Support vector machines or using random forest directly, the accuracies are less than 75%.

At last this little test of only using colors give us an inspiration to combine GLCM and colors.