Texton representation and classification of the Ponce texture Database using K-nearest neighbor and random forests

Stephannie Jimenez Gacha
Universidad de los Andes
Cra 1 Este No 19A - 40 Bogotá, Colombia
s.jimenez16@uniandes.edu.co

Sergio Galindo León
Universidad de los Andes
Cra 1 Este No 19A - 40 Bogotá, Colombia
sa.galindo10@uniandes.edu.co

Abstract

A texton representation and classification of the PONCE group texture dataset was performed using K nearest neighbor and random forest classifiers. A 50 texton dictionary was created using 10 images per class and the texton representation of the database was performed using this dictionary. After that, both classifiers were trained using the texton histogram representation of images, varying the number and size of images in the training set and the classifiers parameters. The KNN classifier show a train ACA of 0.654 and a test ACA of 0.464 while the random forest with pure leaves and a probabilistic state classifier show a train mean ACA of 0,996 and a test mean ACA of 0.508, performing better than the KNN. Results might be improved by increasing the number of textons and a bigger training set size, however, it will require much more computational resources.

1. Introduction

Texton representation is the vectorial quantization of an image based on the responses of each pixel to a group of filters. In this representation, introduced by Malik in 1999, a clustering algorithm is used to find k centroids in the n-dimensional filter response space that correspond to the texton dictionary[2]. After that, the label of nearest centroid is assigned to each pixel according to their responses to the filter bank. The texton representation is based on the texture perception in which a texture can be represented by a repetitive local patterns. This means, the idea underneath the texton representation is to describe textures and therefore objects by a certain combination of responses[5].

Textons have multiple applications in computer vision. Besides vector quantization of images they can be used for classification and segmentation. If objects are assumed as texture-homogeneous entities, which might be valid for some elements, the texton representation for different instances of the same object must be similar while different objects will have significantly different representation[1]. Also, in the same scene, different objects must have different local patterns.

Classification of images based on texton representation can be achieved by the definition of an appropriate representation and distance notion definition. As images are represented by a collection of labels, the usage of texton histograms for representation and histogram distances for comparison is mainly used due to it's simplicity and coherence. For the classification task a classifier is trained using the selected representation and distance and the performance is evaluated in the test set. The most common classifiers used are SVMs, K-neighbor and random forests.

Finally, the classification performance will depend on multiple factors. One of them corresponds to the length of the dictionary, as a higher number of textons will produce better representation but an increased number will cause overfitting. Also the number of filters in the filter bank should be considered as n filters will produce a n-dimensional representation space that could be insufficient or more than sufficient for the problem. In addition, the number of categories and the dataset composition, extension and variety will affect the output for the algorithm.

In order to study and implement a texton representation and classification, a texton representation for the Ponce group texture database is proposed using 50 textons and a variable training dataset varying from 10 to 30 images for category. For the classification task the k nearest neighbor and random forest classifiers are proposed and evaluated while other parameters are considered in the improvement of the average classification accuracy.

2. Materials and Methods

To create a texton representation of a collection of images and training a classifier for them, we used the Ponce group texture database. This set contains 1000 images divided into 25 balanced classes of 40 images each, having between one to three different classes for texture. All images are in grayscale with a uniform size of 640x480 pixels and are found in JPG format. The annotations for the images correspond to the folder name they are located. The dataset also contains a complete set of python and Matlab functions to create a filter bank, obtain the responses of an image to the bank, compute a texton dictionary and display the confusion matrix. The dataset contains multiple classes for most of the semantic categories like *bark* and *wood* which increases the toughness of the classification task.

2.1. Texton dictionary and representation

In order to obtain a texton dictionary, we used 10 randomly selected images from each class and a linear filter bank, created by the function fbCreate. This function returns a collection of 16 filters with 8 different orientations equally distributed in the polar plane and 2 scales varying in a factor of 1.41. After that, the images were cropped into a size of 100x100 or 200x200 to improve computation. The image responses to the filters were obtained using the fbRun function over the concatenation of the train images selected. This function calculates the filter responses through frequency convolution using the FFT. Each pixel of each image represents a point in a 16-dimensional space (one dimension per filter).

To obtain the textons from this subset of images to create the dictionary, we used the computeTextons function with 50 centroids (ideally 2 per class and to posses better resolution) which returned the coordinates of the textons in the 16 dimensional space used. The computation of the textons is performed via k-means with k equals to the number of textons desired. The texton dictionary corresponds to the collection of the 50 centroids. Then, to obtain the texton representation of the complete dataset, the responses of all images to the same filter bank were obtained using the fbRun function. Finally, the nearest texton was assigned to each pixel of each image, using the smalled squared distance as criteria.

2.2. Classifiers training and performance

With the objective to train an appropriate classifier for the dataset a k-nearest neighbor and a random forest was trained using as input data 100x100 and 200x200 windows of 10, 20 and 30 images from each class in texton representation from the dataset, obtained previously. The k nearest neighbor was implemented with the normalized texton histogram of the images under the Minkowski distance and uniform weight distribution. On the other hand two different random forest classifiers were implemented: one under a probabilistic state and the gini impurity and the other under a max depth of 2, a deterministic state and gini impurity[3].For all classifiers the normalized texton histogram of the images was used as input data.

Finally, the classifiers performance was evaluated using the test subset, corresponding to 10 fixed images from each class. The confusion matrix for each classifier with different training window size, number of images and parameters was obtained.

3. Results

3.1. Texton dictionary representation

A 50 texton dictionary from 250 images equally distributed among classes was obtained. This number of textons was selected as it allows to have ideally 2 textons per class while keeping computational time low and an appropriate visual difference between classes, however, it should be variated to assure the best performance. Also, the texton representation of all the database, using the constructed dictionary was generated. Some examples of the images used for the dictionary construction and generated from it are shown:

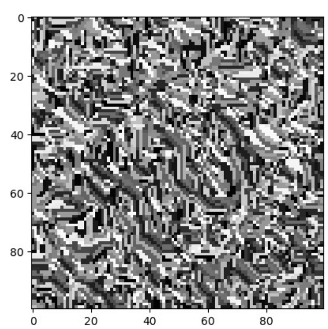

Figure 1. Train image of the texton dictionary belonging to class Granite in texton representation.

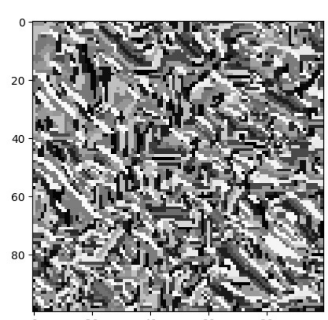

Figure 2. Test image of the texton dictionary belonging to class Granite in texton representation.

It can be seen that the texton representation of the images corresponds to a vector quantization, where filter responses and, by consequence, texture, determine the intensity in images. Despite grayscale, it is possible to notice differences between an image from *Granite*2 class and an image from

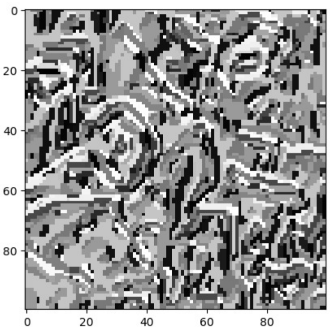

Figure 3. Test image for the texton dictionary, belonging to class floor 1 in texton representation.

the *Floor 1*3 class. Also, images belonging to the same class as 1 and 2 show similar patterns and intensity distribution.

3.2. K-Nearest Neighbor classifier

The k-nearest neighbor classifier was trained variating the train set characteristics. However, classifier parameters were fixed. The train images were cropped into two different sizes of 100x100 and 200x200 pixels and the number of images included varied from 10 to 30 images, representing a training set of equal, double and triple number of images respecting the test set. The Average classification accuracy and the confusion matrix for both the train and test performance was obtained and can be found in figure6,4 and 5. The best results were obtained by a window size of 200x200 and 30 training images with an ACA of 0.464.

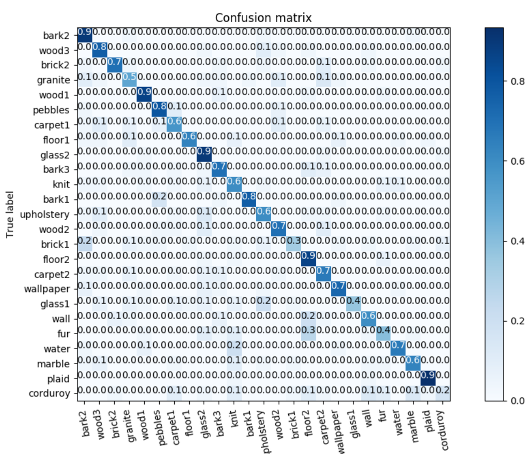

Figure 4. Confusion matrix for best KNN classifier performance over train set (Training images =30, window size =200x200).

It can be seen that the training accuracy is quite below expected for the training set. This could be related to the number of textons used to represent the images or to the nature of the classifier (criteria=minimum distance) that derive into an inappropriate distance metrics to separate the images in the feature space. As expected, the average classification accuracy for the classifier improved with the number of train images and window size.

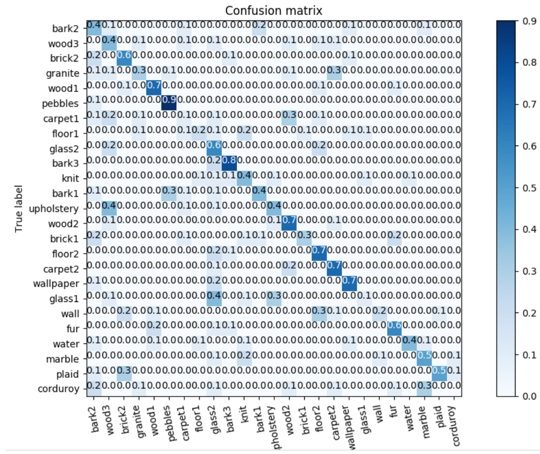

Figure 5. Confusion matrix for best KNN classifier performance over test set (Training images =30, window size =200x200).

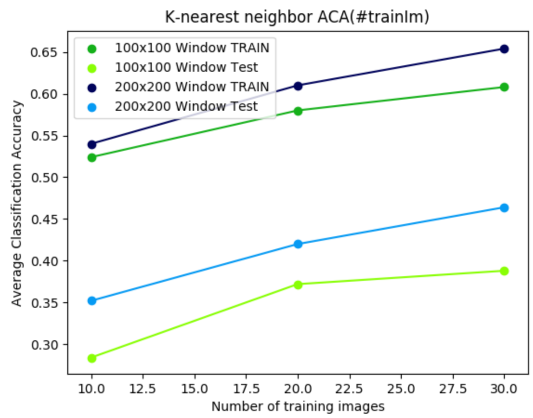

Figure 6. ACA of KNN classifier according to number of training images and window size.

3.3. Random Forest classifier

The first random forest classifier was trained using a max tree depth of 2 and a deterministic state with a variable number of training images, from 10 to 30 images. For this classifier an average classification accuracy of 0.156 was obtained. The confusion matrix shows a preference of the classifier to classify all classes into floor2 category, which make evident the incorrect selection of the parameters7. After that, the second random forest classifier was trained using a probabilistic state and pure leaves with a variable number of images from 10 to 30, utilizing the same training set used for the KNN and the previous classifier. The average classification accuracy and the confusion matrix for both the train and test were obtained. For this classifier, the best results correspond to a window size of 200x200 and 30 images in

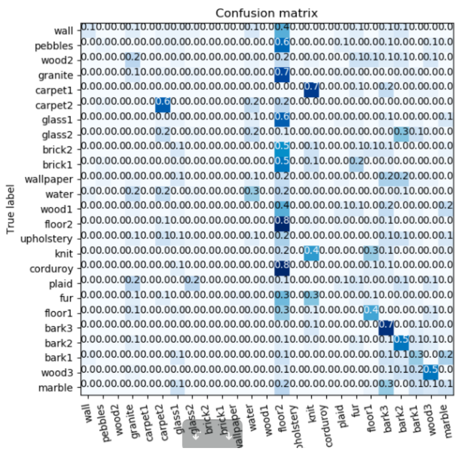

Figure 7. Random forest classifier confusion matrix over test set using a max depth of 2 and a deterministic state.

the training set with an ACA of 0.508.

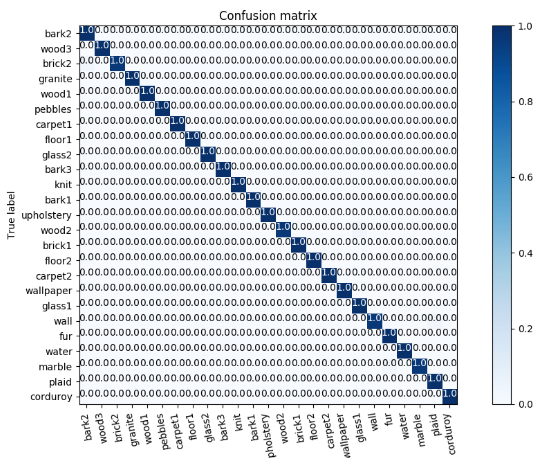

Figure 8. Confusion matrix for best RF classifier performance over train set (Training images =30, window size =200x200)

The random forest classifiers implemented have an ACA very close to 1 with a mean of 0,996 for the training set, also, the training ACA show no dependency respecting the train set characteristics which evidences a correct implementation and robustness for the training phase. On the other hand, the performance over the test set shows a little improvement respecting the KNN classifier.

4. Discussion

Random forest provide a better classification accuracy for texton representation compared to k nearest neighbor. This might be caused by the type of distance considered by each classifier. KNN uses a generalized euclidean distance that might be inappropriate for certain similar classes as random forest uses the Gini impurity, a probabilistic metric about random incorrect labeling. Similar textures can be

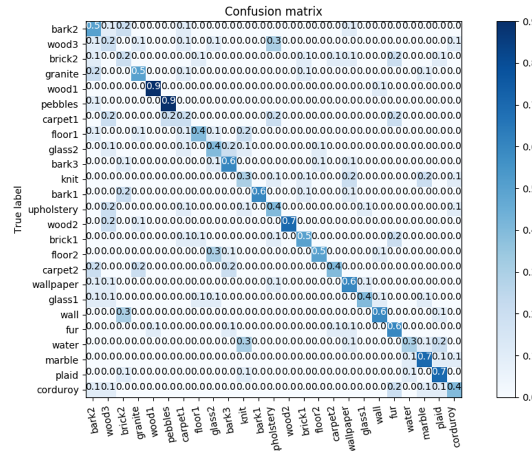

Figure 9. Confusion matrix for best RF classifier performance over test set (Training images =30, window size =200x200)

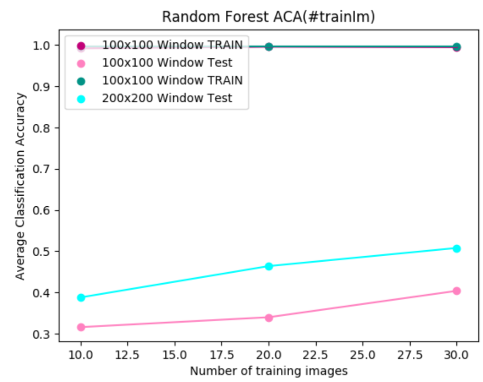

Figure 10. ACA of RF classifier according to number of training images and window size.

constructed from same number and type of textons, leading to small histogram distance. In KNN the spacial information of texton distribution is less important.

Also, the usage of a uniform weight function in the KNN classifier can affect the performance as distance should not be considered as the only significant parameter. On the other hand, the usage of a deterministic model with a determined random state could contribute to misclassification in random forest by eliminating probability labeling as well as the tree maximum depth. The metric and weight function in KNN like the classification criterion, tree depth and random state in random forest can be changed easily in the given functions which is expected to give different results[4][3]. This parameters can be tuned theoretically, however, experimental tuning is recommended. Starting from the best initial results an experimental tuning can be performed. This process could be time consuming but might offer performance tendencies and a practical method to optimize the ACA with multi-parameter optimization.

Respecting the texton dictionary it takes an average of 2 hours to create with a window size of 200x200 and an approximate of 40 minutes with a size of 100x100, using 10 images. This is due to the n dimensional k-means clustering of 2.5 to 10 million points which can be computational expensive and take a considerable number of iterations to converge in all centroids simultaneously. However the texton representation takes much less time to perform as well as the classifiers training and execution.

Both classifiers take approximately 15 seconds to train and 7 seconds to classify the test set. This is due to the small number of points used for training, one for each train image, which represents a total of 250 points as maximum. In addition, the classification algorithms of both classifiers uses simple metrics to predict the categories of the input data.

The main categories presenting confusion for the KNN classifier are wood 1 with pebbles and fur with brick 2 with normalized values of 0.4. This can be explained by the long straight patterns present in both pairs of images that could be represented by the same set of textons. Respecting the random forest classifier the main categories presenting confusion are marble with bark 1, floor 2 with pebbles and bark 2 with wallpaper. The confusion between floor 2 and pebbles can be explained due to the similar rounded patterns with similar scales while the other classes might be confused because of the similar contrasts. It is quite surprising that there is little confusion between classes belonging to the same object, however, it can be explained because of the different rotations and scale between these classes.

Texton representation allows a proper representation of images according to local patterns, or textures, however, it requires a high RAM an CPU requirements for dictionary construction as it involves n dimensional clustering. Also, as the spacial information is neglected by using the histogram to train the classifiers and predicting, the number of textons used to represent the images correspond to a critical parameter because an increased number of textons could mitigate the lost spacial information.

One of the main limitations of the method corresponds to texture rotation as the responses of the image with the filter bank might present significant differences respecting other orientations of the texture. A texture classifier with training data composed of a single rotation of the textures will performed inaccurate for slightly different dataset. For this, a more complete filter bank including different scales and multiple rotations is recommended for the texton dictionary creation.

Respecting the filter bank, it is possible to say that the $45\deg$ oriented filters, without considering the scale, are more discriminative than the rounded filters and other filter orientations. These filters are capable to respond to most of the intermediate orientations and behave well for small and large scales. In addition, it must be seen that a small filter can offer a proper response to large scales features with successive small and correct responses while a large filter might not behave well with small image features, averaging the responses of them.

As seen from the confusion matrices, both classifiers tend to perform progressively better as the training set is more representative. This can be done by a bigger, more variate and better resolution dataset, achieved in our experiments by an increasing number of training images and window size. Also, the parameters of the classifier should be taking into account with a better performance when using random forest with gini impurity, a probabilistic state and no limitation in tree depth. In addition, the usage of a bigger number of textons, resulting in redundant or close centroids should improve the classification as representation space is more accurate.

For this reasons, the best results were obtained for the random forest classifier using a 200x200 window and 30 images per class for training. Ideally, the best classification performance should be obtained by using the entire set of images and a bigger k to obtain the texton dictionary combined with a large training set that uses 30 full resolution images per class to train. Finally, one of the disadvantages of the texton base-classification is that performance is strongly related with both the dictionary generation as well as with the classifier hyper-parameters and training set.

5. Conclusions

Texton representation allows the representation of images based on local patterns or texture, however, this representation is highly sensitive to the images used to construct the texton dictionary, the number of textons selected and the filter bank size, scale and diversity. The dictionary construction corresponds to a slow process that must be performed in a high processing and high memory device, however, it could be done once.

Classification of images according to texture using the texton histogram and a K nearest neighbor classifier show a medium performance. This classifier has an inadequate training ACA with a maximum of 0.654 and a test ACA of 0.464. This might be due to a small number of textons used to represent the images combined with the distance metrics of the classifier. The minimum distance could be inadequate in order to differentiate visual related classes.

Respecting classification of images according to texture using texton histogram and a random forest classifier, an improved performance respecting the KNN classifier was obtained. The random forest classifier show an almost perfect performance for the training phase, independent of the set characteristics. This classifier performed better with pure leaves and a probabilistic state.

Both classifiers show an ACA proportional to the train set size as expected, for this, the best classification was obtained using a random forest classifier with pure leaves , a probabilistic state, a training size of 30 images and a window size of 200x200 pixels (ACA=0.508). To obtain a better performance, the number of textons should be increased as well as the filter bank variability to avoid rotation related misclassification or poor representation differences between related classes.

References

[1] T. Leung and J. Malik. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[2] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: cue integration in image segmentation. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 918–925. IEEE, 1999.

[3] scikit documentation. 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier scikit-learn 0.19.1 documentation.

[4] scikit documentation. sklearn.neighbors.KNeighborsClassifier scikit-learn 0.19.1 documentation.

[5] S.-C. Zhu, C.-e. Guo, Y. Wang, and Z. Xu. What are Textons? *International Journal of Computer Vision*, 62(1/2):121–143, 4 2005.

6. Code and Images

The code is available at the team's repository at *https://github.com/steff456/IBIO4680/tree/master/05-Textons/Answers*