Textons

Jaime Enrique Cascante Vega
Universidad de los Andes
je.cascante10@uniandes.edu.co

Angela Castillo Aguirre
Universidad de los Andes
a.castillo13@uniandes.edu.co

Abstract

Texture is one of the characteristics that can represent an object or structure within an image. In the case of image analysis, a texton is considered as the fundamental structure that is capable of describing a whole texture. For description of images by means of the textures in it, it is important to use filters that allow to see the response of each pixel to these, so that it may be possible to determine a representative pattern for the texture. The aim of this study is to develop a classification of images with different textures, through the use of unsupervised classification, such as K Neighbors classification and Random Forest method. As for the results, it can be concluded that the variation of the parameters in the classification methods may affect the final result. However, it is reported that the RF method turned out to be the best for the experiments carried out.

1. Introduction

Texture is one of the features that represents objects on an image. The variation of the regular textures, can bring information about the local surface orientation. A texture can be described as a pattern that is repeated throughout the object or structure. However, the automatic identification of the shape by means of a texture, is a process that requires a number of steps to be achieved. Some of the steps to complete this prosecution, include the extraction of repeated patterns to determine local affine deformations. Also, it is considered a stage in which it is infered the local surface orientation through different algorithms. [1] Regarding the analysis tasks of a texture, it was introduced the concept of a *texton* by Julesz, which consists in the basic element of pre-attentive perception of a texture [2]. A texton became the most basic reprensetation of a pattern that describes a whole texture. Starting from this, many studies have been carried out to analyze images through the assessment of characteristics in the textons of a texture, to take out great variety of information from the image. Because of this, the aim of this study is to look at the behavior of different textures due to the enforcement of various filters to get its textons representarion in order to decide a classification of these, in an unsupervised way.

2. Materials and Methods

2.1. Database

The database was provided by the Ponce Group from the datasets for computer vision research [3]. The database is composed of 25 different classes each class with 40 grayscale images for a total of 1000 images . Each class correspond to one specific texture, the resolution of each image is 640×480. Among the textures the Ponce's group add textures of water, granite, marble, pebbles, wall, unholstery, wallpaper, fur, kint corduroy, plaid, and different types of brick, glass, carpet, floor, bark and wood. In figure above is shown a random image of each texture class.

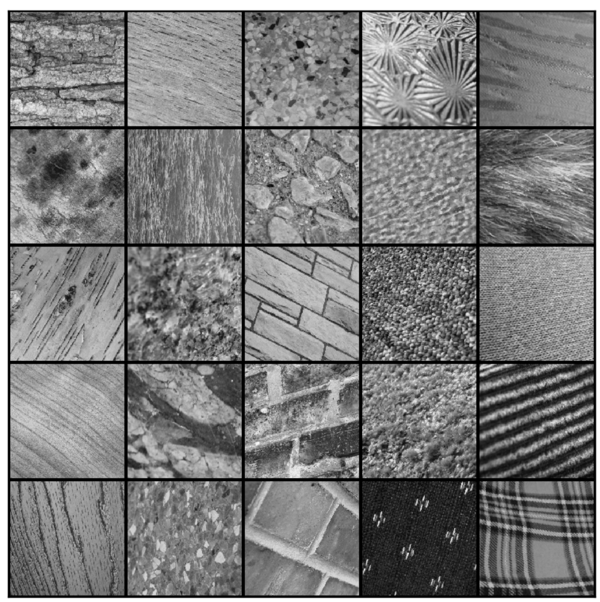

Figure 1: Random image for each texture class.

Let $\mathcal{S}$ be the whole dataset. We divide $\mathcal{S}$ into to subsets that correspond to the train set $\mathcal{S}_{train}$ and to the test set $\mathcal{S}_{test}$ such that, $|\mathcal{S}| = |\mathcal{S}_{train}| + |\mathcal{S}test|$. We randomly select 10 images of each class and assign those image to the test set $|\mathcal{S}test| = 25 \cdot 10 = 250$ (the 25% of the dataset), then the remaining images are used in the train set $\mathcal{S}_{train}| = 25 \cdot 30 = 750$ (the remaining 75% of the dataset). All our experiments use the same train and test set, we only split the whole dataset once.

2.2. Preprocessing, Filtering and Clustering

For reducing the computational cost we initially only consider a subset of each image of resolution 256×256 cropped from the center of each image. Aditionally we subsample the train set, due to computational storage memory, to 15 images. Hence we randomly select 15 images of the train set and we use those to create the texton dictionary. In figure 2 is shown the filters used for computing the texton map, we consider 8 orientation in both directions and 2 scales for having a total of 16 filters.

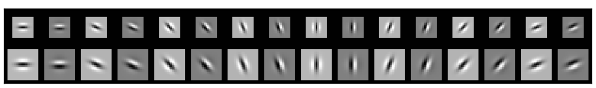

Figure 2: Initial filterbank for computing textons.

Each filter is convolve to each image in the subsampled version of the train set, so for each pixel we now have a response vector of $16 \times 2 = 32$ entries, where each one correspond to the response to each filter applied. After the filter responses of the train dataset are computed the textons are computed given k number of desired cluster. The textons are no more than the centers of the Voroni cells after k-means is used for clustering the filter responses. We use a sparse range of k for having better classification accuracy, we select $k \in \{25, 40, 60, 80, 100\}$. Basically k-means compute the *best* centroid such that the distance of each input is minimized with it's centroid, so given a set of vector of filter responses for a pixel $\mathbf{x} \in \mathbb{R}^{32}$ $k - means$ select the k clusters such that similar textures are clustered together.

Intuitively as k increase a higher mapping is assigned to each vector, hence the representation in the higher dimensional of the vector improve as k increase, however the risk of overfitting in the posterior classification task also does increase as k increase. This is also refereed as the *Bias-Variance* trade off or *Gauss-Markov* Theorem, as the complexity of the representation map increase the classifiers will learn more complex features, patterns of the data and have poor generalization (good performance on unseen data). However the classifiers used in this work specially the $K-$Nearest neighbourhood classifier is not a complex class of model, thus the risk of overfitting is not a big deal.

A new image is classified computing it's filter response with the filter bank (figure 2) and then finding the cluster each pixel belongs to. After finding that response (textons assignment) a histogram is made for representing the frequency of pixel that fall in each one of the k cluster and then a classifier is training such that it maps a frequency estimation into a class.

2.3. Textons map representation and Classification

The textons map representation uses the centroids of the computed clusters. Each centroid is the representations of a different texton. Then, the bank of filters is applied to the training images set and the test images set, so each pixel will have a response to the enforced filters. Each response is compared to the centroids to determine the centroid that most resembles to that pixel. Labels will be assigned according to the filter response of each pixel. So, the response of the pixel that looked closest to the centroid X received the label of that centroid. So at the end, each pixel will have a label associated to a centroid, generating a textons map.

After that, the representation for each textons map was computed as the histogram of the map, where the number of bins described the amount of clusters chosen for the k-means method applied to each experiment. The resultant histogram is normalized by the size of the textons-map matrix. The histograms were normalized by the size of the input image. The normalization, consider the histogram a simple vector in $\mathbb{R}^k$ can be seen as:

$$x_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}$$

For the classifiers it were considered the *K Neighbours Classifier* (KNN) and the *Random Forest Classifier* (RF) methods. [4] Respecting to the parameters of the K Neighbours method, it was chosen a $K = 3$, representing 3 neighbours of a point, with uniform weights for each neighbour and an Euclidean distance for the power parameter of the Minkowski metric. The training stage was performed by fitting the model using the data for train and the known targets of this data. On the other hand, regarding the parameters of the Random Forest classifier, it was selected the default number of trees, which is 10, the maximum depth of the tree was selected as 2 and the seed used by the random number generator was settled at 0, which establishes the deterministic behaviour during fitting. The classifiers were tested first in train data, and then were used to classify the test set.

After that, some of the parameters were changed to analyse the performance of the classifiers subjected to other conditions. Regarding the K Neighbours method, the number of neighbours was settled at 5 while the weights were changed to a *distance* type, in which the weights of the neighbours of a query point are greater if they are closer, than when they are further. Moreover, the Random Forest method parameters were adjusted to enhance the results. The number of trees was increased to 15 and the nodes of the tree were expanded until the leaves were pure by setting the maximum depth parameter as *None*.

3. Data Analysis

For the analysis of results, it was developed the implementation of the confusion matrix [4] where the different classes are related. With this, it was tested the performance of the classifiers. For this purpose, it was computed the predictions done by the classifiers against the real classes of the test data. It was also calculated the Average Classification Accuracy (ACA), that indicates the average of correct predictions done by the classifiers. The analysis by means of this method allowed to see the relationship between the clusters and the predictions made for each classifier. [1]

4. Results

The textons maps were applied over the total set of images available in the dataset. The distribution of the textons allowed to determine the responses of every pixel to the chosen filter bank. The figure 3 shows the textons map for a random image in the test set. From here, it is possible to see the texture represented by the classification obtained from the clustering stage. On the other hand, the histogram of the textons map, became the representation of each image to train the classifier. Figure 4 represents the histogram distribution of the selected textons map (Fig. 3). Here, it is possible to see that the texton 40 has the highest response over the whole image. It will be expected that the images from that class would have a similar response to the textons, so the histograms of this class should be alike.

[1]The code to plot the confusion matrices was adapted from: `https://goo.gl/ny8TKh`

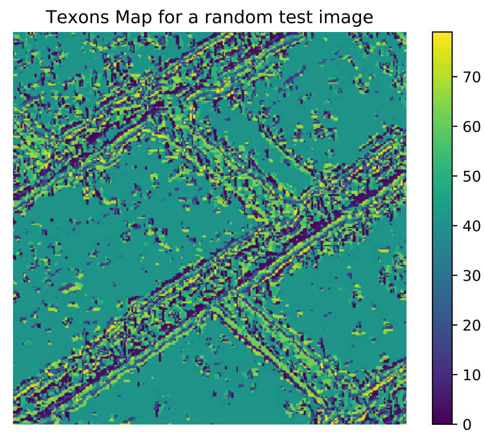

Figure 3: Textons Map for a Random Image

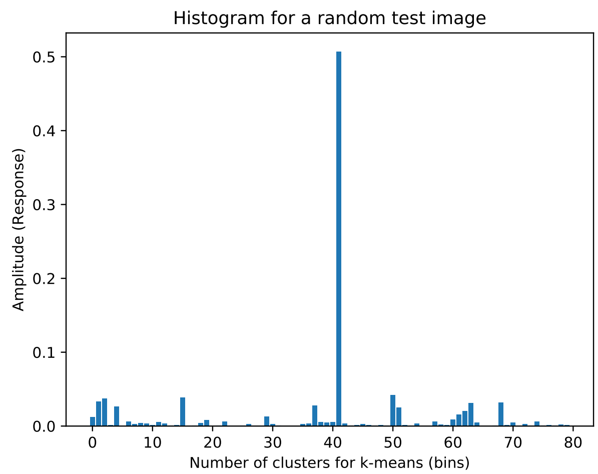

Figure 4: Histogram of a Textons Map from a Random Image

After that, the classifiers were trained with default parameters, to see their performance by testing them using the same train data. Figure 5 represents the relationship between the number of clusters and the ACA for each classifier. For this experiment, the averages of the classifiers were 0.72325 and 0.362 for KNN and RF, respectively. Which means that on average the KNN classification method was 0.3612 above the RF method. The best result for this experiment was when there were 60 different clusters to group the pixels-response and the KNN method was applied. Because of this, figure 6 shows the confusion matrix for this result. As it is possible to see, the classifier failed more when it was sorting out the first 12 classes, while the remaining 13 had fewer mistakes. Nevertheless, the best classification was performed with *wood2* class.

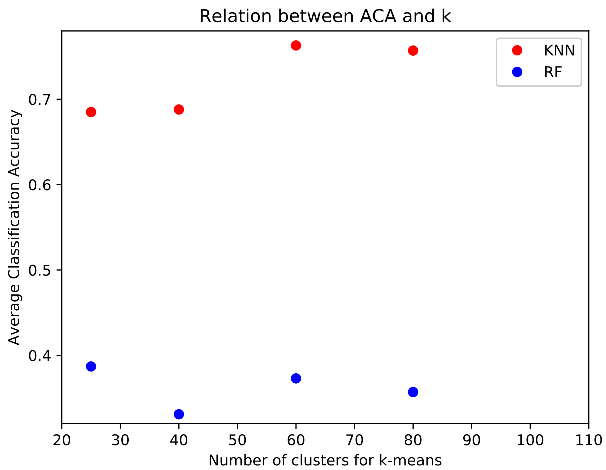

Figure 5: Relationship between ACA and number of clusters for k-means method for the training set. KNN = K Neighbors Classifier, RF = Random Forest Classifier

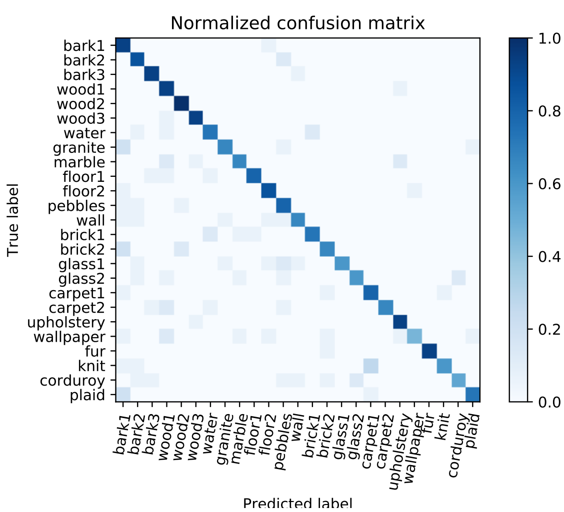

Figure 6: Confusion matrix for the best result of the training set

Then, it was proceeded to apply these same classifiers on the evaluation set. Figure 7 comparison of the method due to different amount of clusters for the k-means stage. Here, the results followed the same pattern as that observed in the test with the training data. Nevertheless, as was expected the average measure between the two measures decreased. The average of the ACA for the KNN method was 0.576 and the one for the RF method was 0.287. The decreasement was perceived either in the individual average as in the difference between the averages of the methods, been 0.289. Figure 8 respresents the confusion matrix for the best result in this experiment. Here, the KNN method was not that accurate as in the fisrt measure, as it was expected. In this case, the best classification was with *bark1* class.

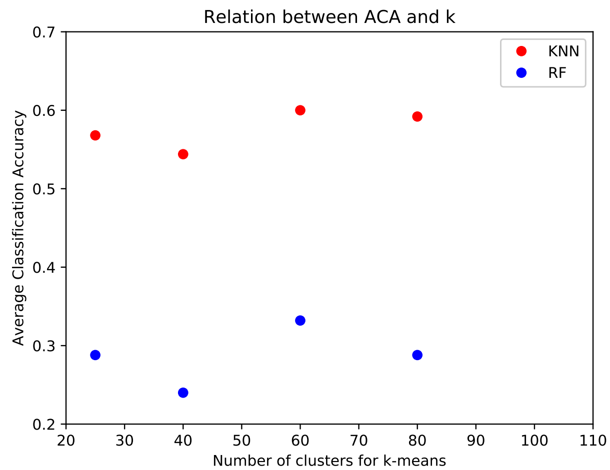

Figure 7: Relationship between ACA and number of clusters for k-means method for the first experiment. KNN = K Neighbors Classifier, RF = Random Forest Classifier

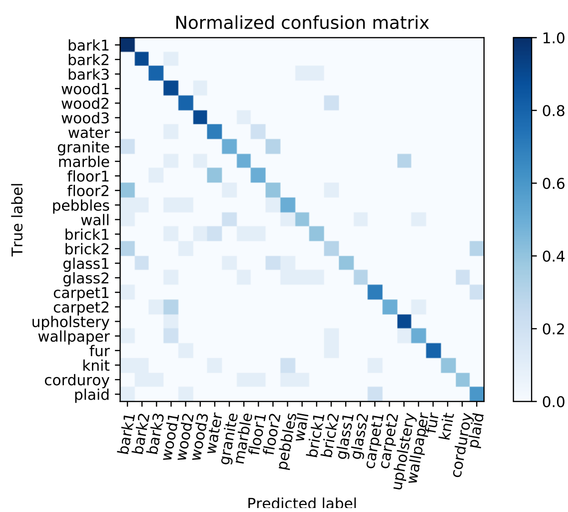

Figure 8: Confusion matrix of the first experiment best result

For the last experiment, the parameters of the classifiers were changed to determine how this can vary the results. The variation allowed to see an increase in the ACA for both methods. Besides, an improvement was identified regarding the performance of the RF classifier. Figure 9 shows the pattern that is possible to identify in the first three points for both classifiers. Nevertheless, the RF method improves the performance for the last number of clusters. On the other hand, the figure 10 represents the best result that was achieved through the done experiments. Besides, it is possible to see that the classifier make fewer mistakes especially in the first classes, which was where the error was most important in past experiments. The ACA for this classification was 0.764, indicating that the best classifier was the Random Forest method.

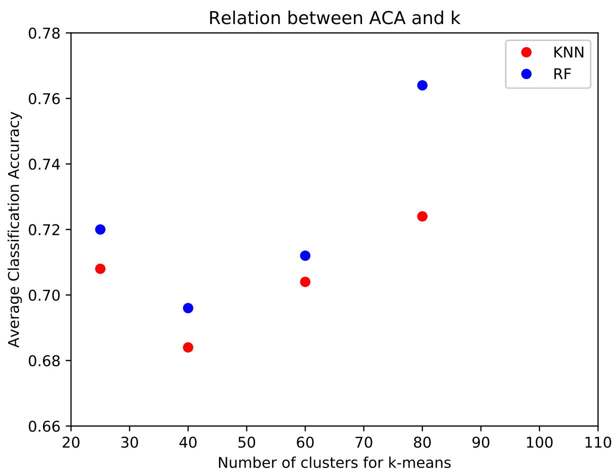

Figure 9: Relationship between ACA and number of clusters for k-means method for the second experiment. KNN = K Neighbors Classifier, RF = Random Forest Classifier

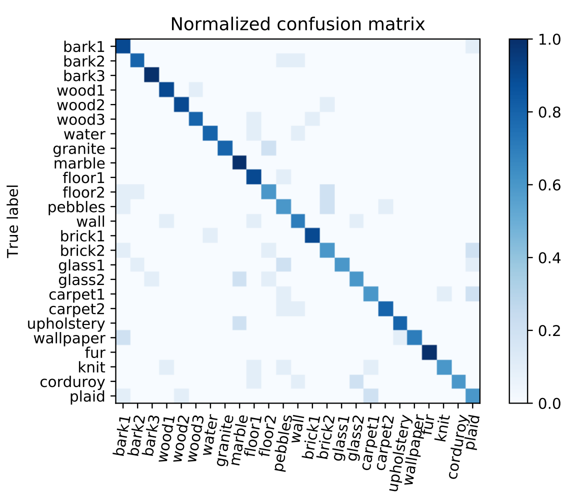

Figure 10: Confusion matrix of the second experiment best result

5. Discussion

According to the results it is possible to say that the use of textons can represent a certain pattern presented as a texture. Nevertheless, there are some considerations to take into account for the moment of the processment of the images to work with. First it is important to identify the right amount of information to process, because this technique requieres a lot of memory capacity on the machine as well as RAM. Also, it is possible to analyze that the success of the method it is closely related to the amount of filters that are applied to the images, as well as the forms that are chosen for these filters.

On the other hand, it is possible to say that once the data is ready for the classification stage, the number of possibilities for its analysis can vary in huge range if the parameters are not chosen correctly. From the results, it is clear that the method can even converge to a possitive result, as the amount of clusters increased, as was the case of RF method. Or the method can oscillate as was the case of the KNN method.

Let the empirical error probability be defined as

$$\widehat{L_n}(h) = \frac{1}{m}\sum_{i=1}^{m} I\{h(x^{(i)}) \neq y^{(i)}\} \tag{1}$$

Where $h(x)$ are the classifiers (KNN or RF). More generally we can assume $\mathcal{C}$ as class of hypothesis $h(x) : \mathcal{R}^d \rightarrow \{1, 2, ..., |C|\}$, where d is the input dimension. The goal is to obtain the function $h \in \mathcal{C}$ with smallest error probability [5]. A well known result in statistical learning is that if the class of models $\mathcal{C}$ has finite VC-dimension then the error probability of the selected hypothesis is:

$$L_n(h^*) \leq \mathcal{O}\left(\sqrt{\frac{k \log m}{m}}\right) \tag{2}$$

Where k is the VC-dimension of the selected hypothesis $h \in \mathcal{C}$. The class of hypothesis defined for example by K-NN are classifiers with VC-dimension equal to K which is less than dimension of the input instances so it can be bounded as eq. 2. This result shows not only that the confidence on the prediction of the KNN classifiers is bounded, but also that it must perform better than the chance.

About the random forest classifiers they are just bagging of weak learners [6], then there are no theoretical guarantee about bounds on the error, however it have been shown the strong-ability of aggregating weak learners. The VC dimension of a RF model is not bounded, hence they can approximate more non-smooth functions, that's why we think the RF model is better that the KNN that have a finite VC-dimension.

6. Conclusions

- Even the methods implemented in this work have in general good performance, a lot of information in the image is not used, textons are not considering spatial information on the image. More over, the textures learned in this work are fixed due to the filter bank used, this is in general a common approach followed, however we are limited to the detecting features, patterns that match the ones in the filter bank. This is a clearly limitation as the filter bank is not varied in this work, additionally due to the computational time and memory the experimental accuracy have in general no statistical confidence, we are always using the same images. However, some authors suggest that as patterns are repetitive non-homogeneous maps they can be seen as *bootstraped* data that in general have good performance in generalization [7, 6].

- As can be seen in figure 3 the textons maps have a lot on noise on textures that are not associated with the texture desired, in this case brick, it have high response on edge that are not in the edge of the bricks.

References

[1] R. Szeliski, *Computer vision*. 2011.

[2] B. Julesz, "Textons, the elements of texture perception and their interactions," *Nature*, vol. 290, pp. 91–97, May 1981.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, May 2005.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Applications of Mathematics*. Springer, corrected 2nd ed., 1997. missing.

[6] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, Aug. 1996.

[7] L. Breiman, "Statistical modeling: The two cultures," *Statistical Science*, 2001.