Images classification based on the texton representation

Nicolás Morales Velandia
Universidad de los Andes
Carrera 1 # 19-27
n.morales10@uniandes.edu.co

Oscar Francisco Trujillo Puentes
Universidad de los Andes
Carrera 1 # 19-27
of.trujillo10@uniandes.edu.co

Abstract

There are many methods to classify images. One of the most important methods is classify an images using textons, were it is use the image texture to categorize the information inside the images. In this laboratory practice, we are use textons for image texture classification to categories an images from the Ponce Group of University of Illinois database. For this purpose, we are using to type of classifier, the k-nearest neighbors algorithm (KNN) and random decision forests.

1. Introduction

Each object has a specific color, shape and texture, visual characteristic that's what it defines. In this way, if an object is present inside an image, it is possible that this feature can be extract and apply to detect and classify a Image, using a group of path representation from the object in the image. Thus, the present laboratory aims to show the results obtained by classifying a set of images by using the information that gives the texture. With the use of algorithm that take the local descriptors and compare with a known and categorized database, the laboratory was used the texture information on the image. In this caso, the database used was from the Ponce Group, University of Illinois. The texture database feature 25 texture classes, 40 images samples each. In addition, the entire image are in gray-scale JPG format. [1] [3]

2. Methodology

2.1. Materials

2.1.1 Database

For this practice, were used the **texture database** from The Ponce Group (Computer Vision and Robotics of Beckman Institute, University of Illinois at Urbana-Champaign). This has 25 texture's classes; each class has 40 images and the image was a path of specifical material, that each material has a specific texture for each one.All the images are in 640 x 480 pixels, grayscale and JPG format. Among the categories is 3 types of bark, 3 types of Wood, water, granite. The database was divide in subgroups, 30 images for categorie in train and 10 in test, the anotation in this database was the names of the folders . In the Figure 1, it can see some examples of the database. [1]

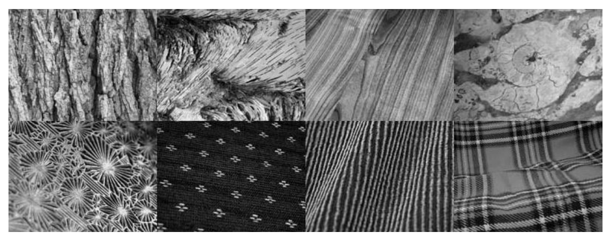

Figure 1. Examples of Texture Database

2.1.2 Codes

For the development of the different experiments, python was used as software to run experiments, in addition, the code present in the following link was modified `https://github.com/affromero/IBIO4680/tree/master/05-Textons`, with which the dictionary of textones of the images and the histogram was obtained. . For the test and training section, KNC (k-nearest neighbors classifier) and RFC (Random Forest Classifier) were used as classifiers.

2.2. Methods

For the classification of the database, the texture of the different objects was used, this is because the texture is a set of regular patterns placed irregularly. In this way, it is possible to estimate surface orientation or shape from image and compare to group image regions with consistent texture. So, the Textons are used. Bela Julesz definen a Texton as the principal units that predice a human in the texture perception, this represent that generally is required a human to look at the texture in order to decide what those fundamental units o repeated patterns are in the object, however, a texton can also be defined as analyze the texture in terms of statistical relationships between the fundamental elements.

For this reason, using the filter banks and methods of Leung and Malik, it develops a statistical approach is developed where textures are modelled by the joint probability distribution of filter responses. This distribution is the frequency histogram of each filter response with grouped in such a way that they allow obtaining the texture information of the object (clustering), this generates the texton. In this way, an image is classified by the use of textones when it presents the same or similar frequency histogram given the response of the different filters and resembles some of the textones that were previously calculated.

First, to classify images using textons, it necessary to generating the texton dictionary. The dictionary is make up from merging the textons of all the texture classes. Later, a set of group of unregister images of a particular texture class are convolved from the group of train, are convolved with the texton filter bank. The result from the filter responses are combined and grouped into textons using a K-Means algorithm. Second, starting from the filter responses categories generating a model, that it is known as texton map. Once you construct the texton map, you create the histogram of textons, which is the frequency with each texton occurs in the labelling. Finally, using another image from the test group is classified by forming its histogram and the using a classifier. The new image is avowed as belonging to the texture class of the closest model. [2]

Taking into account that the texts are histograms of frequency before the application of the different filters and there were 25 categories, at least 50 clusters were used, this was determined by the Nyquist-Shannon sampling theorem, where an analog type response can be reconstructed , without error, of samples taken in the same way, the sampling ratio must be equal, or greater, to twice its frequency, since it is 25 categories, it is expected to use at least 50. It should be noted that there are no filters more discriminated than others, this depends on the image, where a filter affects in greater quantity depending on the pattern of texture present, this is due to the inclinations, shapes, thicknesses and luminosity.

2.3. Classifiers and preprocessing of data

To classify the images within the different categories, two classifiers were used. First, k-nearest neighbors classifier (KNC) is the simplest machine learning technique. First, consider the Voronoi partition of the representation space induced by the train data and assign to each test point the label of its Voronoi cell. Next, Assign to each test point and assign the label the category with the greatest presence to its surroundings nearest neighbors among the train data. Second, Random Forest classifier (RFC). A random forest is a classificator that fits a number (decision tree classifiers) on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but these change according to the answer with each node of the tree, the default value to continue is that the answer is true.

In each of these methods, it is necessary to define certain parameters, in the case of KNC the number of desired neighborhoods is defined, in the case of the test performed, a number of 25 neighborhoods was used since it was the number of categories present, in addition , it was left by default that all the neighborhoods had the same weight and that the distance between test and train was linear. In the case of RFC, we define the number of desired trees which were greater than 25 given the number of categories, the maximum depth which was taken randomly and random state that were also random.

Before developing the dictionary of textones, it was necessary to perform a preprocessing of the data, in this case a readjustment of the original size of the image was applied to values of 100 x 100 pixels and 200 x 200 pixels, besides, the amount of iamgenes de train from 30 to 20.15 and 10 iamgenes by category. These changes were made for two reasons, by using a greater number of images and by having a larger size, the processing time would be very high and this would lead to a greater RAM consumption, stopping the procedure because it does not have enough memory.

3. Results

The method to evaluate the classification problems are the confusion matrices and the ACA, this corresponds to the average of the correct response images that are the diagonal of the matrix, divided by the number of iamgenes used in the test. Next, some examples of confusion matrices are presented and a table with the parameters and differents ACA.

3.1. Results and Confusion matrix Train set

To evaluate how effective the training and classification model was, the training set on the trained model was evaluated.

Classifier	# Train images	ACA
KNC	10	24.4%
RFC	10	97.6%
KNC	15	22.9%
RFC	15	95.5%
KNC	20	26.8%
RFC	20	100%

Table 1. Parameters and ACA in train set

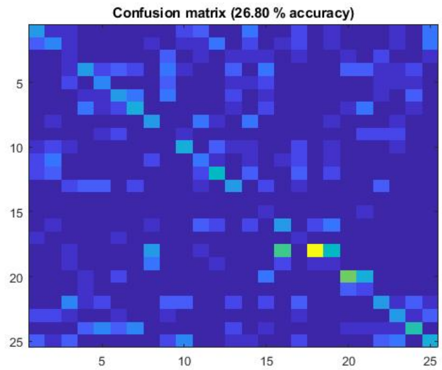

Figure 2. KNC for a model with 20 images of train

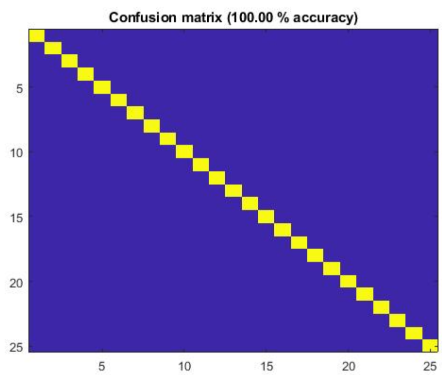

Figure 3. RFC for a model with 20 images of train

Through the response of testing the model on the same training set, it was possible to evaluate how efficient the classifier used was, in this way, it was observed that the classifier by nearest neighbor was not very accurate given that it gave lower indexes to the 30% and which indicates that it is not accurate.Additionally, it can be observed that the classifier with the highest number of used textones is precisely the one that has the most creter value, this can be explained because the k used for the classifier depends fundamentally on the data. On the other hand, Random Forest Clasifier is very efficient since it has indexes greater than 95%, showing that the classifier with the highest confidence is RFC

3.2. Results and Confusion matrix Test set

To evaluate how effective the test model was, the test set on the trained model was evaluated.

In figure 2, the confusion matrix can be observed, for the model with 20 images of the KNC classifier. This model obtained a result of 26.8% accuracy in the ACA.

In figure 3, the confusion matrix can be observed, for the model with 20 images of the RFC classifier. This model obtained a result of 100% accuracy in the ACA.

Classifier	# Train images	ACA
KNC	10	12.8%
RFC	10	8.8%
KNC	15	14.8%
RFC	15	10.8%
KNC	20	13.6%
RFC	20	12.4%

Table 2. Parameters and ACA in test set

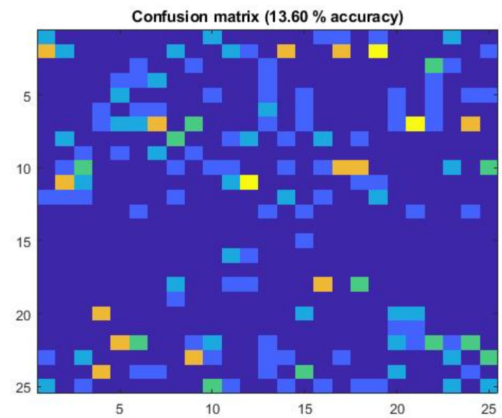

Figure 4. KNC for a model with 20 images of test

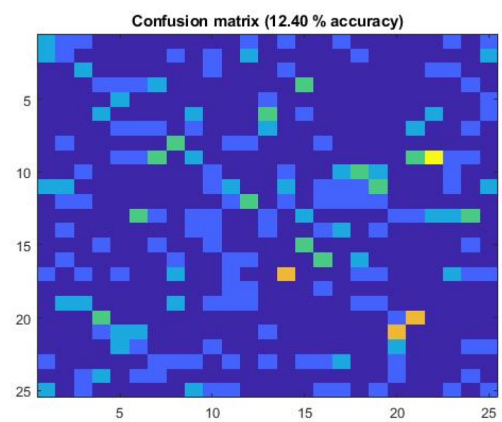

Figure 5. RFC for a model with 20 images of test

4. Discussion

As can be seen from the results obtained with the two methods used. The KNC classifier, in general, obtains better performance compared to the RFC classifier. These data are opposed to the results obtained for training, in addition to the performance of both classifiers is below the performance obtained above, mainly for RFC.

For the RFC classifier, the low performance of 95% for train to 11% on average. This means that the data extracted for train and test generate a model with very large overfitting in the data. This over fit in the model means that the chosen value of k for the classifier is inadequate and the number of images chose for the model is to small, because the model shows a lot of noise with the data, this can be seen in the confusion matrices obtained for the different models created.The effect of noise in the classification can be observed mainly in figures 11 and 13, corresponding to the confusion matrix for the RFC classifier with the 10 and 15 images models, due to the fact that high responses are observed in different categories. they are not on the diagonal of the matrix. Accordingly, these models present the lowest ACA result among all the obtained results, of 8.8% and 10.8% for the model of 10 and 15 images respectively.

The classification with KNC is the method that presents the best performance for the data test, however it can be observed that the model with 15 images presents the highest ACA in the test validation. However, it does not agree with the ACA's highest performance training model. This result validates the fact that the model presents overfitiing in the data, as described above.

From the results obtained, it was observed that the best classifier was RFC, this is due to the results obtained by proving the training data on the model, obtaining an almost 100 % result, this is due to the fact that it receives a greater amount of Parameters that allows to generate a better model as a greater number of trees, with greater depth.

The biggest problem with the method is its processing time at the time of making the dictionary of textones, this varies between 1800 seconds to 3800 seconds, this is due to the process of applying the bank of fltros on each image and rearranging it, this caused a memory expenditure that at the time of decreasing capacity was spent more time. On the contrary to the training, after performing the histograms to train the models, it took between 1 second and 8 seconds, demonstrating the difficulty of the texton method.

The categories that caused most confusion was glass 2 and brick 1, this is because there are very similar categories at the time of applying the filters, which at the time of training were faci that went to brick 2, wall or pleebes. This causes a limitation in the method apart from the RAM and processing time, the great resemblance between categories generates difficulties and throws similar responses in the filters.

Finally, it is recommended to use filters that have elements more similar to those of each category, such as the edges of bricks or the shapes of the different types of glass. Aditionally, use another method of clustering as GMM, which would allow a better form and increase the number of clusters.

5. Conclusion

As a result of the results obtained, it can be concluded that the methods are not efficient to classify with the use of textones. Additionally, because the results show an overadjustment of the data in all the models. That is why it is necessary to review the models created, as well as the classifiers used. Therefore, it is necessary as a first measure to use different images for training and to observe the performance of each model with a smaller number of images for training, to observe if there is noise reduction on the classification, and wich model is better.

Finally, because the methods used in this practice are linear classification methods, it is recommended to use other type of classification, as nonlinear classification methods such as the distance x^2 between histograms of textones.

References

[1] Ponce research group: Datasets. =http://www-cvr.ai.uiuc.edu/ponce$_grp/data/$, 2006.

[2] Y. Javed and M. M. Khan. Image texture classification using textons. In *Emerging Technologies (ICET), 2011 7th International Conference on*, pages 1–5. IEEE, 2011.

[3] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

6. Attachments

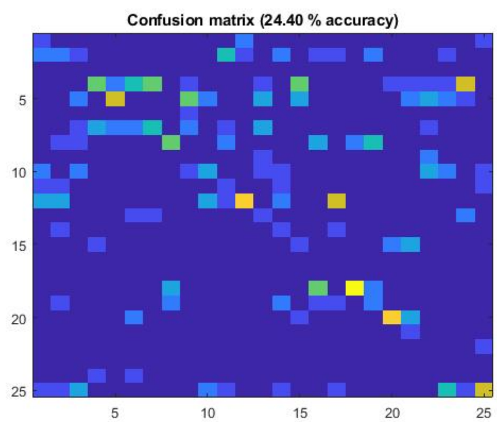

Figure 6. KNC for a model with 10 images of train

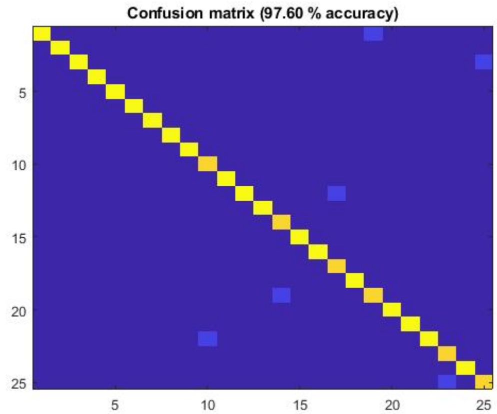

Figure 7. RFC for a model with 10 images of train

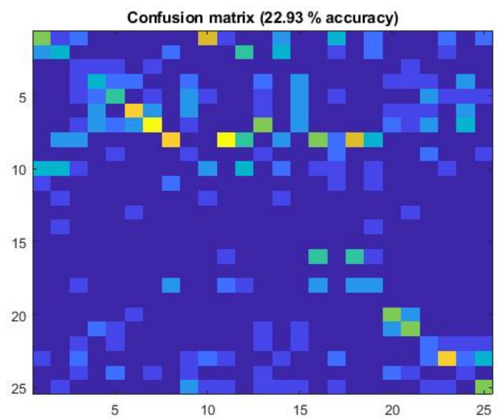

Figure 8. KNC for a model with 15 images of train

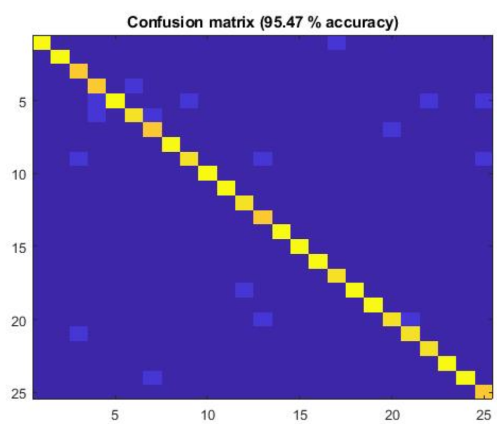

Figure 9. RFC for a model with 15 images of train

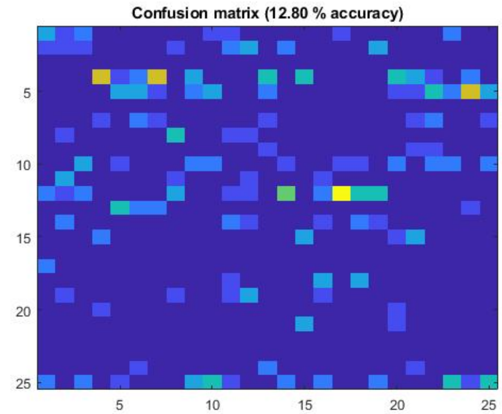

Figure 10. KNC for a model with 10 images of test

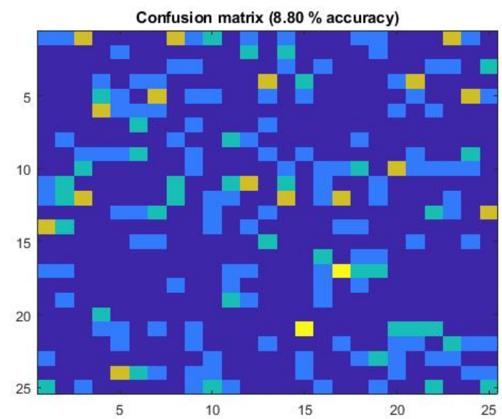

Figure 11. RFC for a model with 10 images of test

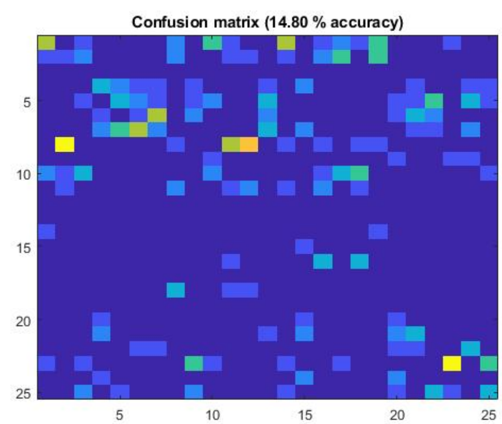

Figure 12. KNC for a model with 15 images of test

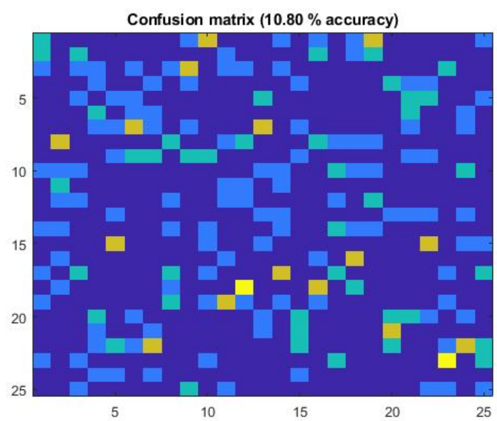

Figure 13. RFC for a model with 15 images of test