Textons: Laboratory 05

Sergio Steven Leal Cuellar*, y Mateo Rueda Molano†,
Departamento de Ingeniería Biomédica, Universidad de los Andes. Bogotá, Colombia
Email: *ss.leal10@uniandes.edu.co, †ms.rueda10@uniandes.edu.co,

***Abstract*—In this report the process to classify in 10 different classes several images from CIFAR10 dataset will be described. This will be done using a texton representation, which is a representation of local patterns that are being repeated all over the image. The results obtained from the texton classification were not acceptable enough due to the fact that the images have local patterns that are not repetitive all over the image and are not representative enough in order to classify the image. For future experiments it is important to make use of other forms of representation such as color and shape to obtain better results in the classification process.**

I. Introduction

Texture is a fundamental characteristic of natural materials and has the capacity to provide important information about the objects and the scene. Texture analysis plays an important role in pattern recognition and computer vision. In 2001, Leung and Malik introduce textons to achieve this problem and in computer vision they refer to an image which is constructed based on certain structures, such as spot-like features of various size, edges, orientations and scales [1]. However, the notion of textons was established many years ago by these authors. Julesz described the textons as fundamental micro-structures in natural images (and videos) and are considered as the atoms of pre-attentive human visual perception (Julesz, 1981). The notion of texton started in psychophysics, Julesz (1981) and colleagues discovered that pre-attentive vision is sensitive to some basic image features while ignoring other features. His experiments measured the response time of human subjects in detecting a target element among a number of distractors in the background [3]. This type of structures that are captured in a pre-attentive way such as elongated bars, points, crosses and terminators were called textons for the first time. In computer vision, textons are used as dictionaries, in order to achieve a faithful representation of various textures a finite big set of textons (texton dictionary) closely representing all posible local structures need to be obtain. However, this can cause high-dimensional models with great requirements of CPU and RAM [9].

Textons represent some texture features of an image and they are optimal to classify different structures with varied texture. In order to have a good performance using a classification algorithm using texton representation the dataset and the image objects must have important differences in texture for each one of the classes [9]. In this laboratory In this laboratory, 10 categories of images will be classified from the CIFAR10 database that contains 60000 images with 10 balanced classes. To do this, texture representation will be used in the map of textons and as a method of classification, algorithms such as those of the nearest neighbors and random forests. The dictionary of filters to be used is 2D, so it is necessary to transform the image to gray levels.

II. Materials and Methods

A. Database description

The CIFAR-10 dataset from the Canadian Institute for Advanced Research is a collection of images that are commonly used to train machine learning and computer vision algorithms. The CIFAR-10 contains 60000 32x32 color images in 10 different classes. There are 6000 images per class. The 60000 images are divided in 50000 training images and 10000 test images. The test contains 1000 randomly selected images for each class. The training set has 5 batches each with different number of images from one class to another [**?**].

B. Method and filters description

Textons are comprised of centroids from a k-means clustering of patch features. The elements belonging to a certain cluster are thus distributed in an n-D sphere, which might not reflect the intrinsic underlying distribution of the data in the feature space[4].

Repeated patterns which are the basis of textons are found with specific filters (spots, bars, raw patches, etc.). To find the patterns we consider the magnitude of response. The histogram of the texton map was used as The statistics for each of the images. To start the algorithm, a filter bank was created from the fbCreate function provided for the development of this laboratory. This function creates a filter bank and uses the number of filters (K) as a parameter and is applied to each of the images of both the training and the test. Then, the histograms of the filter responses (map) in each image are saved to obtain a matrix of representation of texture for each of the images. The texton algorithm works in the following way:
1. Build a texton dictionary with the instances of the train. (fbCreate function)
2. Given a test image, convolve it with the filter bank.
3. Represent each pixel as the vector with N answers

4. Assign each pixel to the text closest to its vector
5. The image of the labels is called texton map.
6. Histogram of the map representation.
7. Classify by supervised methods [5].

The algorithm of k nearest neighbors was used for the classification using the map of textons. This algorithm employs a distance (Minkowski, Euclidean, Chi-squared, etc.) to separate each one of the clusters created [6]. A classification model is created in the train set by fitting the class labels of each image with the histograms found by the texture representation. This is then applied to the test set and the predictions of the model are evaluated based on the true classes of each of the test images. Furthermore, Random Forest algorithm was used as a method of classification. This algorithm works a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [7]. Finally, the performance in the train and the test batches of both of those algorithms was tested by the confusion matrix and ACA values.

After several runs of experiments to find out the optimal number of images in the training set, the parameters for KNN and Random Forest were kept constant and the k (number of textons) was changed iteratively. After several tries the optimal k was determined in a range from 112 and 160. So, the value k=160 was taken arbitrarily. Depending on the images, some filters may be more discriminative than others (e.g. in a landscape a filter with horizontal components will be more discriminative than a vertical one in images like a truck or with a high amount of horizontal components). In this case horizontal components are more important due to the orientation of the texture in the image. Which would not happen in a image of a building for example.

C. Hyperparameters and distance metrics

The hyperparameters used for KNN are the numbers of neighbors and the metric. The number of neighbors determine the number of points or neighbors for the queries and the metric determines the type of metric for the distance measured for the prediction. First, uniform weights were used, all points in each neighborhood are weighted equally. Meanwhile, another experiment with weight = 'distance' was done (weights are assigned proportional to the inverse of the distance from the query point, so further points weight far more less) which in addition with the parameter p = 1, (Manhattan distance) and applying a chi2 test to the data, gave good results.

In a first experiment, using the parameter weigth = 'uniform', the number of neighbors used in the model was 2 because in the figure 1 it is shown the best ACA (0.659) of all our experiments with k= 112 filters. In addition, after several runs of experiments it was determined that larger numbers of neighbors cause worst results and lower ACA (figures 2 3 4). Also, this little number of neighbors makes the method much more sensible to the noise but improves the obtained ACA in train. In addition, the distance used was Minkowski, because in literature many text classification articles use that distance [8]. Additionally, this distance is more robust than the Euclidean one because it turns out to be a generalization of it. Nevertheless, when running several images(more than a 1000) on the test dataset, the noise caused a very low ACA. So, the parameters that gave a slightly better results were the weigth = distance, chi2 test and 50 neighbors with a Manhattan distance (in Minkowski distance 50 neighbors actually gaves one of the worst results). But, the actual difference isn't higher than 3% on the ACA.

The important hyperparameter used in this case of Random Forests are the max depth of the random trees, in the documentation in Python is told that better results are obtained when the max depth of the tree is not define ('None'). However, a experiment with this parameter was also done and gave an ACA of 1 in depth= 10 (figure 7, we expect this method memorized the train set. Nevertheless, it takes a far more RAM memory and with a max depth of 7 or 8 satisfactory results were obtained. Another parameters used is the number of estimators (the number of trees in the forest). In this case, the larger this parameter the better directly proportional to the increase of the size of images in the train set. So a value for this parameters was set on 100 for a small slice of the training model. For the 1000 images training set 500 max depth was established in order to maintain the same ACA, however taking some more RAM.

D. Hyper parameters in train

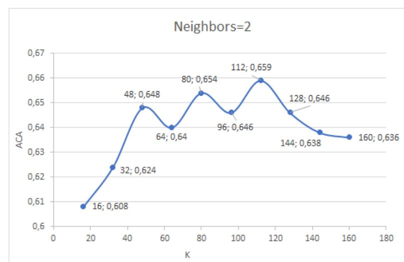

Figure 1. ACA vs k - KNN: 2 neighbors

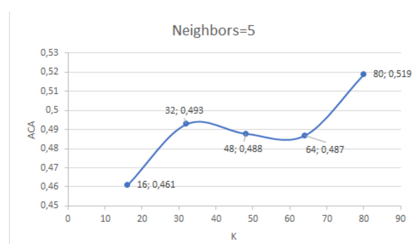

Figure 2. ACA vs k - KNN: 5 neighbors

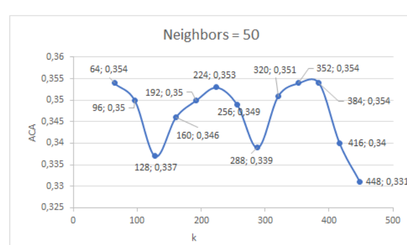

Figure 3. ACA vs k - KNN: 50 neighbors

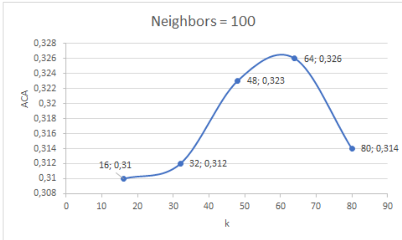

Figure 4. ACA vs k - KNN: 100 neighbors

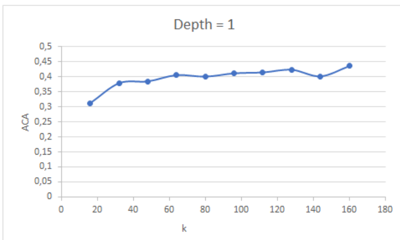

Figure 5. ACA vs k - Random Forest: depth= 1

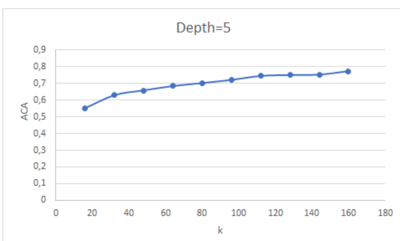

Figure 6. ACA vs k - Random Forest: depth = 5

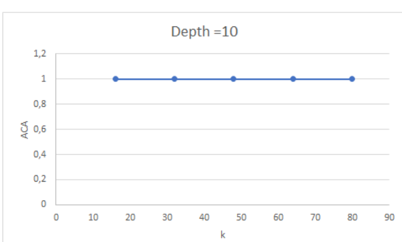

Figure 7. ACA vs k - Random Forest: depth = 10

Finally, it is important to mention that in the original database a rgb to gray process was applied, so that every slice of the data obtained was a image and not a of R,G or B channel as the documentation of the database CIFAR10 stated. [2].

III. Results

A. Confusion matrix and ACA - Train batch

Below is the confusion matrix of each of the classifiers (KNN and RandomForest) for the train group

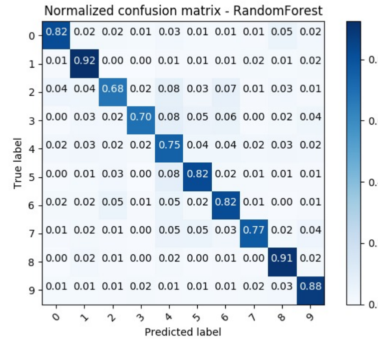

Figure 8. Confusion matrix (KNN classifier) train

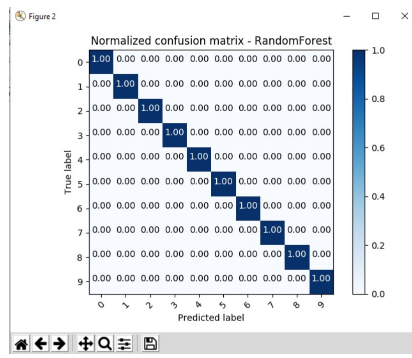

Figure 9. Confusion matrix (Random Forest classifier) train

The ACA in KNN classifier is 0.659 and in Random Forest is 1.0.

B. Confusion matrix and ACA - Test batch

Below is the confusion matrix of each of the classifiers (KNN and RandomForest) for the train group

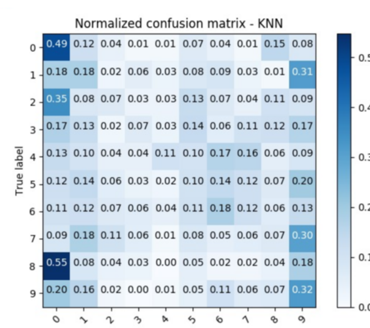

Figure 10. Confusion matrix (KNN classifier) test

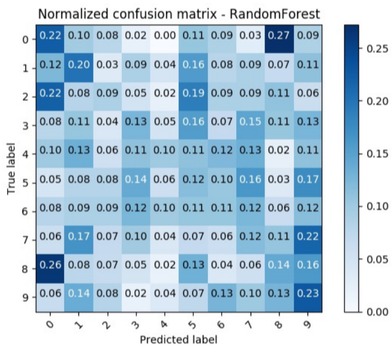

Figure 11. Confusion matrix (Random Forest classifier) test

The ACA in KNN classifier is 0.19 and in Random Forest is 0.17 ± 0.05

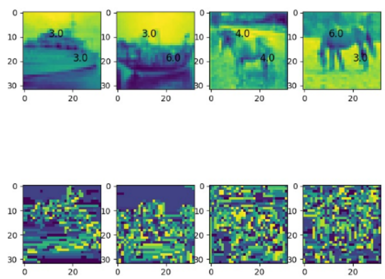

Figure 12. Visual representation of the classification test

IV. Discussion

The results were not as expected since sufficient ACA was not achieved to classify the images in this database. However, when looking at the texton maps, it is very difficult to distinguish 3 of them from the 4 images 12. This may suggest that this method of classification by textures is not effective when classifying images from this database. For example, images of animals have similar textures in terms of hair and differ more by shape and color.The categories that cause the most confusion are those classes with several features that are no that obvious. For example, some categories of animals were indistinguishable when looked through their texton maps and had a lot of redundant or noise texture information that came form the background, like in the case of some images of animals like the birds, cats, deers or dogs. This may explain the low values that can be seen in the diagonal (confusion matrix) for those classes. (most of the cases this classes are being misclassified)

In train, the best classifier corresponds to random forests but this is due to the fact that this method tends to over fit the data of the training group and does not provide reliable information of how much efficiency it performs when classifying the model. This can be seen in the great difference that the random forest model presents in the train a and in the test batch. It can be asserted that there was a huge over fitting of the models. This situation can be due to the fact that the textons differ greatly in each image and do not provide relevant information for the representation matrix.

The training model applying both classifiers at the time using textons method took 10 minutes for 1000 images(100 per class) and training for 2500 images(250 per class) took almost 25 minutes approximately. Testing, takes almost 20 minutes for 1000 images and 40 to 60 minutes to test 2500 images . The reason behind this long waiting time is because it is required to convolve each image with each of the bank's filters and store it. This can take a long time when the number of images is large due to the huge increase in the number of operations due to convolution. Then, k-means process takes time for the amount of images in the dataset. For databases with larger size of images, you must have very broad CPU and RAM requirements to carry out this task in an acceptable time.

V. Conclusions

The limitations of this method are that it only takes into account the texture aspect as a classifier. Although it is important for the process of vision and recognition in humans, it does not take into account aspects of color and shape that are very important as characteristic aspects of each object. For example, airplanes, tractors and cars in this dataset have similar patterns of texture and materials, however, they show great differences in their shape. This important aspect in several categories is not taken into consideration using the textones method.

There are ways to improve the result of the classification method for the CIFAR10 database. As a first aspect, the representation matrix can be expanded by adding more representation dimensions such as the color histogram of the images using the Pixelwise method. However, the limitations are the sensitivity to the discretization of the color space as well as ignoring spatial information of the image. Another way could be increasing the dictionary of textons and specify in the filter bank unique patterns of texture for each of these classes, however this is not specific to classify other types of images. Also, histograms of the gray scale images can be concatenated to complement the numerical information of this representation and have more information to fit the classifier. Finally, all the images in the dataset must be used to have greater reliability in the model created in train.

References

[1] T. Leung and J. Malik, "Recognizing surfaces using three-dimensional textons," Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999

[2] A. Alex Krizhevsky, "The CIFAR-10 dataset," Variable Naming Conventions, 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html.

[3] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu, "What are Textons?," International Journal of Computer Vision, vol. 62, no. 1/2, pp. 121–143, 2005.

[4] Jie Yang, Xinyang Feng, Angelini ED, Laine AF. Texton and sparse representation based texture classification of lung parenchyma in CT images. Conf Proc IEEE Eng Med Biol Soc. 2016;2016:1276-1279.

[5] M. Zucker, "Texture classification with textons." Swarthmore College, United States, 2016.

[6] Pedregosa, F, et al. Scikit-learn: Machine Learning in Python -K near neighbors. Journal of Machine Learning Reasearch; 2011: 2825–2830
[7] Pedregosa, F, et al. Scikit-learn: Machine Learning in Python -Random Forest. Journal of Machine Learning Reasearch; 2011: 2825–2830
[8] J. Yang, X. Feng, E. D. Angelini, and A. F. Laine, "Texton and sparse representation based texture classification of lung parenchyma in CT images," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016.
[9] I. Bloch and J. R. M. Cesar, Progress in pattern recognition, image analysis, computer vision, and applications: 15th Iberoamerican Congress on Pattern Recognition, CIARP 2010, Sao Paulo, Brazil, November 8-11, 2010: proceedings. Berlin: Springer, 2010.