Classification using texture patterns

Laura Munar
Universidad de los Andes
l.munar10@uniandes.edu.co

Maria Ana Ortiz
Universidad de los Andes
ma.ortiz1@uniandes.edu.co

1. Introduction

Texture classification is an active topic in image processing which plays an important role in many applications such as image retrieval, inspection systems, face recognition, medical image processing, etc. Textures are another feature that can help image processing because they can provide information about the spatial arrangement of the colors or intensities in an image. In this way, texture consists of texture primitives or texture elements, called *textons*. The texture analysis methods can be categorized in four groups: statistical methods, structural methods, filter-based and model based approaches [5].
Using textures to classify images consist in the selection of a filter bank. The filters are the patterns that are going to be searched in the image. After the cross correlation of the filters with the images, a texton dictionary is created. Then, a texton map is formed with the nearest label of the dictionary assigned to the test images. Finally, the texture classification is done with its histogram. It is important to notice that the each filter is looking for an specific pattern and the features are computed as the local energy of the filter responses. So, there are filters that only are going to find and specific form in the image. The important information is when all the filters are computed and the texton map is created because it can be seen the whole distribution of features in each image.
In the present work, a texture classification method is developed. The classifiers used are K Nearest Neighbor(KNN) and Random Forest (RF). In the first, a database is searched for the most similar elements to a given query element, with similarity defined by a distance function.The main goal of the algorithm is to decide the class of a new case based on the classed of the k most similar database elements. This method is very sensitive to its hyperparameters and it is known that the best combination of parameters values depends on the data sample. So, the choice must be done in an iterative way. It is very important to find a K value that describes the database and the distance measure used to classify too [1]. The second algorithm is a supervised learning algorithm. RF involves several hyperparameters controlling the structure of each individual trees and structure of the forest such as, node size, number of trees, among others. RF involves decision trees to make the predictions by combining each decision [4].
For the purpose of this work, the CIFAR10 dataset was used. This dataset consists of 60.000 images of 32x32 size. The entire dataset consists in ten classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck). There are 10.000 test images and were selected randomly from each class and the classes are mutually exclusive [3].

2. Materials and Methods

For this report different experiments where conducted to identification for getting the best value of the hyperparameters, the methodology follow for the experiments will be explained in the following items. Once the best hyperparameters where determined for the best results the final version for the challenge was applied.

2.1. Creating texton dictionary

For the creation of the filter bank use to create the dictionary the function provided was set for variables support in 2 and sigma in 0.6, and upon that different experiments where conducted for establishing the best K for each metric and additional experiments for defining hyperparameters that will lead to the best possible result on training and test set. From the filter created they were convolve over the set of train images and for deciding number of textons in te texton dictionary a experiment was conducted over 1000 as its explained in section 2.2. Finally the bank filters used are shown in the figure 1 and the map of text

2.2. Experiments

For the experiments different methods where approached:

- *Variation of K-textons*: The K from the K-means for grouping the textons was a variated in a series of experiments starting from 5 to 144. Variables such as number of train images, number of test images distance of KNN(K nearest neighbor),number of k nearest neighbors, number of trees where stateless in

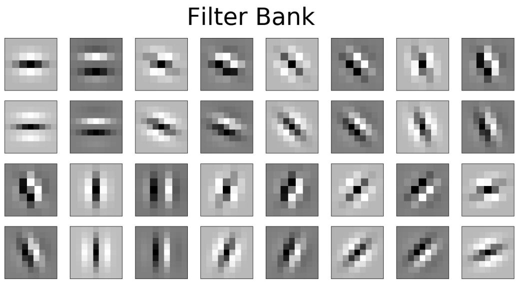

Figure 1. Filter Bank

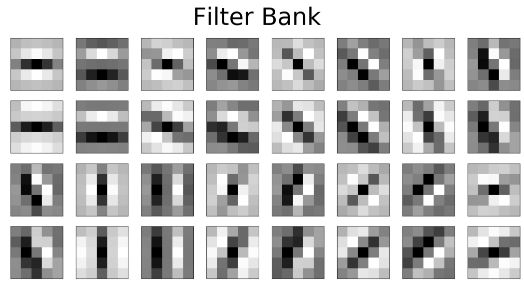

Figure 2. Filter Bank Best Method

1000, 1000,chi squared distance, 100,170 correspondingly.

- *Variation of train images*: The number of train images was variated to observe computational time and variation in accuracy scores for KNN and Random Forests clasification methods, the variable was variated from 1000 to 10000 with increments of 1000. Variables such as number of test images distance of KNN(K nearest neighbor),number of k nearest neighbors, number of trees where stateless in 1000, 1000,chi squared distance, 100,170 correspondingly
- *Variation of K nearest neighbord*: K was varieted on clasificator KNN from 1 to 200, remaining static the other variables: 8000 train images, 10000 test images and chi square distance.
- *Variation of Number of Random Forests*: The number of random forests was variated and the other variables of the experiment where remained stateless.
- *Variation of distance for K nearest neighbor*: The distance for evaluating proximity was change to intersection kernel
- *Variation of max features of Random Forests*: Max features was varieted to see its effect on classification, other variables where remained stateless.

2.3. Best Method

After analyzing the results of all the previously mentioned experiments, the final approach was to establish the sorter as random forests with 0.2 max features and 170 trees, additionally with 144 textures and a filter bank as shown in figure 2, this best method was train in 1000 images and tested on 10000.

How many textons are you using? Why? Description of the classifiers, hyperparameters and distance metrics Did you apply any adjustments or preprocessing to the data? why?

3. Results

According to the previous section, the first thing done was proving different values of K to have an idea of the number that could describe best the problem.

Figure 3. Variation of the K parameter in the performance for both classifiers

Then, the number of images in the train set was evaluated with both classifiers,

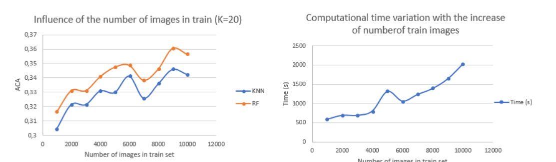

Figure 4. Influence of the number of images in the train set in the performance of the classifiers

Each classifier was also evaluated separately and their hyperparameters were taken into account.

3.1. Best Method

The best method results confusion matrix are shown in figures 7 and 8 corresponding to train an test, the result of accuracy score is in table 2

The time enlapsed on the test and train between the nearest neighbor and the random forest is largely different, with more than 100 seconds of difference.

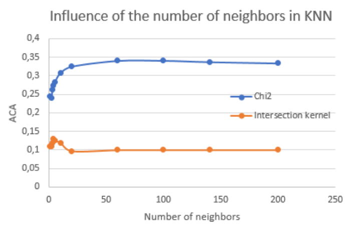

Figure 5. Influence of the number of neighbors in KNN with two distance measures

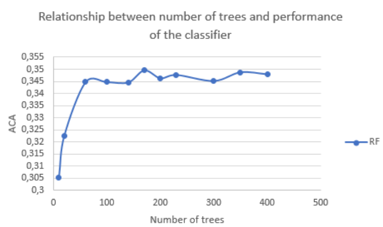

Figure 6. Influence of the number of trees in RF

Table 1. Influence of the maximum features hyperparameter in RF performance

Max features	**ACA**
0,1	0,3325
0,2	0,335
0,3	0,3329
0,5	0,3282
0,6	0,3289
0,8	0,3161
0,9	0,2819
1	0,3316
5	0,3338
10	0,3267
16	0,2391

ACA Train	100%
ACA Test	39.23%

Table 2. Accuracy Score Best Method

	Time
Computing Time of Textons	41.6460
Time Assing Textons	109.6809
Time KNN	103.2605
Time Random Forest	0.9110

Table 3. Time enlapsed on each method

4. Discussion

The figure 3 shows the influence of K in the performance of the classifiers. As it can be seen, this value also depends on the classifier used. For KNN the best value performance was 0.267 with K = 16 and in RF the best ACA was 0.369 with K = 144. This could be explained with the fact that KNN needs also to adjust the K number of neighbors

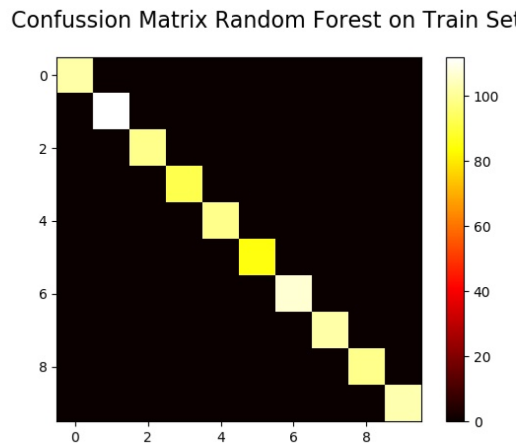

Figure 7. Heatmap of Random Forest train

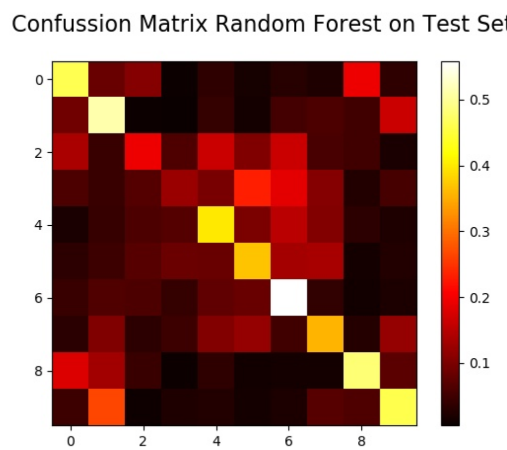

Figure 8. Heatmap of Random Forest test

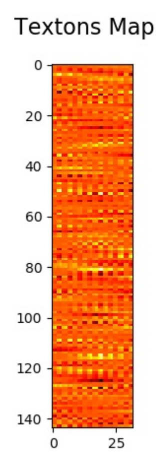

Figure 9. Heatmap of textons

hyperparameter. Here was only considered as a variable the K value for the creation of the dictionary. However, RF presents better performance because this method is more robust than KNN.

On the other hand, the number of images which the dictionary is created is very important because the problem has to be described in its best way to train a good model. The figure 4 shows that increasing the number of the train set improves the performance of both classifiers but it also increases considerably the computational time of the algorithm. In order to go from a ACA of 0.304 to 0.3425 with KNN the algorithm, 1439 more seconds are needed. Then, the decision of how many images are needed in train also depends on the time that the creation of the dictionary of textons take.

The choice of a Kernel depends on the problem at hand because it depends on what it is trying to be modeled. A polynomial kernel, for example, allows to model feature conjunctions up to the order of the polynomial. In figure 5 is shown that the kernel Chi square fits better than the kernel Histogram intersection kernel to the representation of the problem. This can be explained because the latter gives the minimum distance within the points but not considers the distribution of the data. This figure also presents that the best performance was 0.34 with 100 number of neighbors in the method. When K is small there is a restriction of the region of a given prediction and is forcing the classifier to be more blind to the overall distribution.

Moreover, a random forest algorithm trains each decision tree with a different subset of training data. Each node of each decision tree is split using a randomly selected attribute from the data. This element of randomness ensures that the Machine Learning algorithm creates models that are not correlated with one another. As a result, potential errors are evenly spread throughout the model and are cancelled out by the majority voting decision strategy of the model. The figure 6 presents the influence of the number of trees in this problem. It shows that the best number of trees is 350 with a performance of 0.3488 but after 100 trees the performance does not change significantly. This can be explained on what was mentioned before, because some errors may be spreading through the decision tree.

On the other hand, the maximum features parameter is the size of the random subsets of features to consider when splitting a node. According to table 1, it can be seen that 5 maximum features gives the best ACA result of 0.3338. According to Bernard et.al the strength of randomization in the tree induction is led by this hyperparameter which plays an important role for building accurate RF classifiers. There is no information on what value of max features use in an algorithm but some studies support the approximation of setting the maximum features as the square root of the size of the original feature set. Also, it was found that extreme values (K=1 o = size of the original feature) are not advised to be used. The choice of this hyperparameter actually leans on a compromise between two needs: to force via randomness the tree induction process to diversify choices of splitting criteria in order to induct trees different from each other; and, to choose relevant features for splitting criteria in order to induct trees performant enough. The maximum features parameter acts as a trade off for balancing performance and diversity of trees in the ensemble [2].

After analyzing the results it was define that for the best method random forests works better, it may be because of the intrisict randomness in the k-means, another factor is that distance used for K Nearest Neighbord wasn't descriptive enough for defining closeness between categories, additionally that a pattern of performance was identify in which more textons affected KNN method and less textons affected Random Forest method.

According to the confussion matrix reported on figure 8, the class that cause the most confusion is number 4, which is category for cats

5. Conclusions

This method could be improved by changing metrics or sort, because image information or representation is very powerfull, following these creating a texton dictionary can be very memmory expensive for compute, and processing all available cores. The ACA scores are low, since the classification process is not the most powerfull, and as said before the process could be improved.

References

[1] D. Batista, G. Fustado. How k nearest neighbor parameters affect its performance. *Instituto de Ciencas Matemticas y de Computacin*. Available: http://conteudo.icmc.usp.br/pessoas/gbatista/files/asai2009.pdf.

[2] L. A. S. Bernard, S. Heutte. Influence of hyperparameters on random forest accuracy. *MCS Springer*, 5519(1):171–180, 2009.

[3] A. Krizhevsky. The cifar 10 dataset. Available: https://www.cs.toronto.edu/ kriz/cifar.html.

[4] M. B. A. Probst, P. Wright. Hyperparameters and tuning strategies for random forest. *Research Gate*, 2018.

[5] W. X. z. Zhu, S. Yizhou. What are textons? *International Journal of Computer Vision*, 62(1):121–143, 2005.