Image classification algorithm based on texton representation

Saul Gomez
Universidad de Los Andes
sc.gomez11@uniandes.edu.co

Diego Valderrama
Universidad de Los Andes
df.valderrama@uniandes.edu.co

Abstract

An algorithm was implemented to address image classification in 10 categories according to the textons representation. The CIFAR-10 database was used. It was employ a filter bank of 16 filters with two sizes each, it was calculated the optimum number of textons (K), images used for the creation of textons dictionary and the classifiers' hyperparameters. Results were evaluated using Average Classification Accuracy (ACA) and discussed in order to identify strengths and flaws in the developed algorithm.

1. Introduction

Within the problems tackled by computer vision, the representation of images plays a fundamental role when performing their analysis and processing. For this reason, and based on the type of problem being addressed, it is convenient to use different representations of color, such as texture or shape. Texture is always present in natural images and constitutes an important visual cue for a variety of image analysis applications like image segmentation, image retrieval, and shape from texture [1]. Textons refer to fundamental micro-structures in natural images and are considered as the atoms of pre-attentive human visual perception, related to texture [2]. These can be understood as a series of patterns that are repeated in an image object. Texture classification is a fundamental issue in computer vision and image processing, playing a significant role in a wide range of applications that includes medical image analysis, object recognition, content-based image retrieval, and many more [1].

Based on the above, the classification of images by their texture is possible because it is expected to find that these patterns of textons are very similar in images that correspond to the same category. Thus, when extracting the structures of a set of images and calculating their respective map of textons, it should be found that the texton histograms present similar distributions, so that by using classifiers like K-Nearest Neighbor or Random Forests, the images under analysis can be categorized with an acceptable room for error.

2. Materials & Methods

2.1. Database

The images were taken from the database CIFAR-10 which is a collection of images that are commonly used to train machine learning and computer vision algorithms. The database is composed of 60.000 32x32 color images in 10 classes, with 6000 images per class distributed in five batches for the training set each one of 10000 images for a total of 50000 and a test set with another 10000 images.

The classes correspond to images of planes, cars, birds, cats, deer, dogs, frogs, horses, boats and trucks labeled with a class number of 0-9, respectively. Thus, the general idea is to identify the structures that make each of the classes different and extract the corresponding patterns by the textons representation to implement an algorithm that automatically performs the classification of these images.

2.2. Methods

2.2.1 Texton dictionary

First, a bank of 16 filters of two different sizes (7x7 and 9x9) was selected. Then, the images were concatenate into a single a image and filtered with the filter bank previously created. Later, the textons were created using K-means. To do this, we select K vectors from each databases class as a textons. Finally, the texton dictionary was built by joining the textons of all classes.

In addition, it should be noted that some filters from the filter bank created may be more discriminative than others, depending on the images you want to analyze. In terms of the database used in this study, the shape of the objects of the classes analyzed will have a better response as they have more structures that resemble some of the filters. For example, in the car class, they are expected to have a better response to vertical and horizontal edge filters, which would make these types of filters more discriminatory for this kind of images.

2.2.2 Clasifiers

As a first approximation to classify the images we use a nearer neighbor classifier (KNN) to discriminate the textures of the train set in order to obtain a prediction model of the categories of the images. To do this, the the filter bank was applied to the train set and the histogram of the filter responses was created. The hyperparameter used was the number of neighbors with which the model was obtained. Additionally, random forest was used as a classification method. For this classifier, the most important hyperparameter is the number of trees in each forest which was modified in the experiments.

2.3. Experiments

The experiments were made in order to create the best textons dictionary that represents database classes and a good model that allow to predict the class of each image of the test set. To do this, the following parameters were taken into account: number of clusters or textons, number of images per class to create the textons dictionary and hyperparameters of the classifiers used.

First of all, it was selected and arbitrary number of images per class, neighbors for KNN classifier and trees for Random Forest classifier. Once these values were set, the algorithm was trained by varying the number of textons in order to find the value that maximizes performance in the classification. The optimal number of clusters was then set and the algorithm was iteratively executed but this time it was changing the number of images per class to create the dictionary of textons. Finally, from these two values the performance was compared with the two classifiers employed and the one with the highest performance was selected. Once this was done, the algorithm was executed by varying the hyperparameters of the chosen classifier, storing the textons dictionary and the model to predict test classification of given classifier that had the better performance after 100 iterations. This last procedure was developed because the algorithms of both classifiers have a random initialization, so the same results are not always obtained from their execution.

2.4. Algorithm

The first step was to load the images and the training labels. Then, 100 images per class and their labels were selected. In the next step, the filter bank and the textons dictionary were created using K = 256 since this is the number of gray levels of an image. Later, the textons were assigned to each image and the histogram of textons was created with which the prediction model was defined using random forest as a classification method with 175 trees. The textons dictionary and the model obtained were also saved in order to use them with the test images. Finally, ACA and confusion matrix were used as evaluation methods for the train and the test set.

3. Results

First, it was calculated the optimum number of textons setting arbitrary numbers for images per class and neighbors in KNN classifier, 30 and 3, respectively. According to the figure 1, this value corresponds to 44 which maximizes the performance in train set classification. Subsequently, experiments were carried out to find the optimum number of images per class and neighbours in KNN. As can be seen in figures 2 and 3, the values found were 9 and 3, respectively. Likewise, based on the data found for the KNN classifier, experiments were carried out to determine the value of the parameters that optimize the algorithm response in the classification of the training set but now using Random Forests classifier (See figures 4, 5, 6, 7). This process was repeated several times to obtain parameters that better fit the predictor model of Random Forests. Figure 8 shows the results for ACA performance in train set varying the number of trees in Random Forests. The number of clusters and images per class set was 32 and 20, respectively. According to these results, it is clear that Random Forests based model presents better performance classifying train set and provide us a wide range of parameters to train model and evaluate in test set, thus this classifier was selected. Based on the parameters mentioned previously, the performance of the algorithm developed in the test set was evaluated. Figure 10 shows the confusion matrix obtained on the algorithm classification performance in the test set. Also, ACA was approximately 29,16%. These results indicate a very poor class prediction accuracy.

Now, since the initial results obtained were unsatisfactory, it was necessary to define new parameters. Rethinking the results presented before it was notice that we still have several options to choose new parameters that fit better to the images and provides a better performance. Based on the results showed in figure 5 it was decided to set the number of images per class used in the textons dictionary construction in 50, in order to provide the model a wide variety of images per class and make a more complete dictionary. The next step, was to figure out what number textons could fit better to the model. Then, since an evaluation was already carried out using a low number of clusters, it was intended to employ a much larger number that offered more information in the representation of texts of the structures in the images. To do this, it was considered that employing a number of textons similar to the number of intensities on the grayscale could give sufficient information to the classifier to identify unique patterns between the different classes. Based on this, and taking into account the results presented in the figure 8, the number of textons was set in 256.

Finally, a new experiment was developed in order to find the number of trees that fit better to the new parameters set. Thus, the algorithm was trained varying the number of trees and it was found that for values greater than 100, the algorithm performance was maximum in the training set. For this reason, the number of trees in Random Forests model was set in 175. Then, the algorithm was evaluated in test set again. The results are provided by figure 12 and it was obtained an ACA equals to 38,2%. This represents a considerable improvement on the results initially presented.

4. Discussion

Based on the results obtained, the random forest is a better classification than KNN. This may be due to the fact that the neighbors in KNN are constant, but in a random forest, the variability of the pixels that are part of the same forest allows some features to be taken into account which KNN ignores. However, using random forest as a classification method with several number of trees produces an over-learning of the features of train set used, so when we use the model with new images some patterns of the different image classes are omitted making wrong predictions.

Fig 12 shows that the classes that cause the most confusion are planes, boats, trucks and cars. This could happen due to objects presented in these classes have different form patterns but have similar texture so the model confused them because the information regarding the shape of the objects is ignored. This situation also happens with cat and dogs. It is important to know that the main limitation of the model is the number of images used for the training since it was 5% of the images used in the test. Also, as mentioned above, ignore the patterns of the shape of the objects with similar texture make that the model does not have enough information to predict correctly. Finally the time required to create the textons dictionary of the model used is about 6 minutes.

5. Conclusion

As demonstrated, texture representation is not always an optimal mechanism for analyzing all types of images. It is necessary to carry out a rigorous analysis of the problem to be addressed and of the images that are available to make a better approach to the solution. In terms of the current study could be more profitable to addressed this problem using color or shape representations.

Likewise, as stated in previous section it is necessary to train the algorithm with a large number of images in order to achieve better results. Since only 500 images were used to train the algorithm and it was tested in 10000 images, the model does not learn well the patterns that differentiates each category, so consequently the classification accuracy was not satisfactory.

On the other hand, it should be added that the database used in this study has several categories that share considerable similarities among themselves, that although the human eye is able to perceive differences, these can become a great challenge for the computer vision, and even more when one takes into account the representation by textures, which showed that generated a lot of confusion for the implemented algorithm.

References

[1] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[2] S.-C. ZHU, C.-E. GUO, Y. WANG, and Z. XU. What are textons? *International Journal of Computer Vision*, 62(1):121.

6. Images

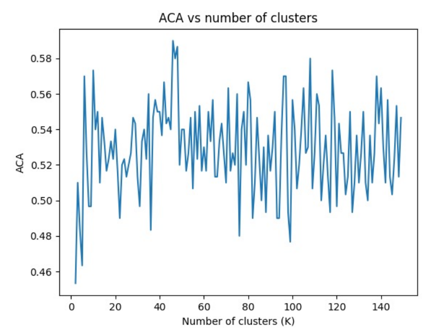

Figure 1. ACA performance in train set varying the number of clusters setting an arbitrary number of images and neighbors

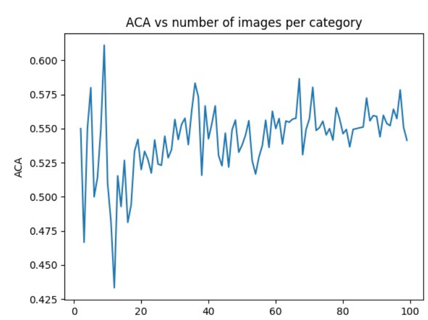

Figure 2. ACA performance in train set varying the number of images per class setting number of clusters in 44 an arbitrary number of neighbors

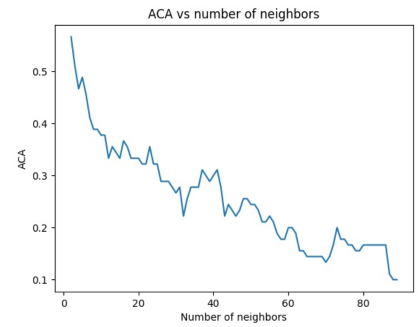

Figure 3. ACA performance in train set varying the number of neighbor for KNN setting number of textons in 44 and number of images per class in 9

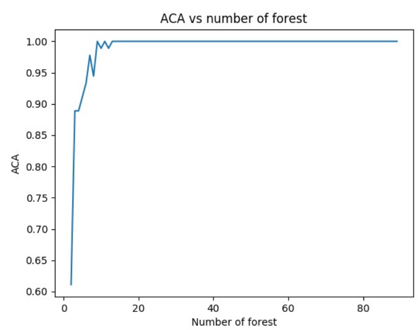

Figure 4. ACA performance in train set varying the number of trees for Random Forests setting clusters in 44 and the number of images per class in 9

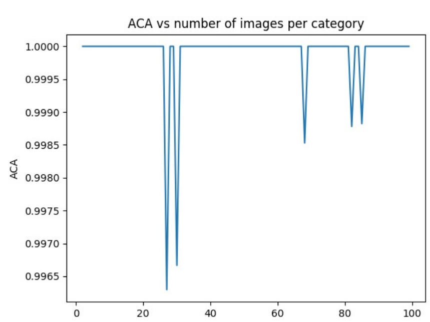

Figure 5. ACA performance in train set varying the number of images per class setting clusters in 44 and number of trees in 20

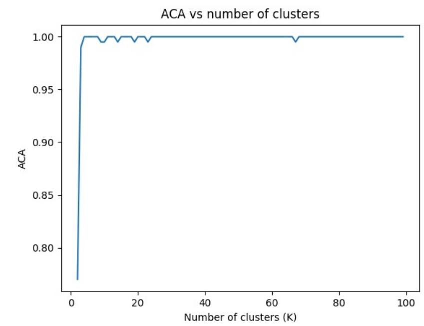

Figure 6. ACA performance in train set varying the number of clusters setting the number of images per class and trees in 20

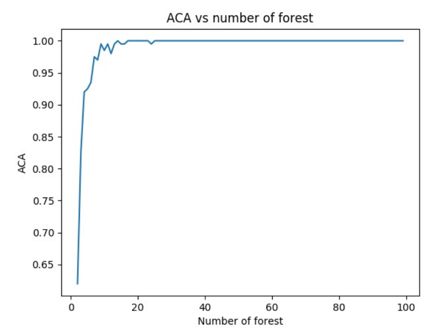

Figure 7. ACA performance in train set varying the number of trees for Random Forests setting clusters in 14 and the number of images in 20

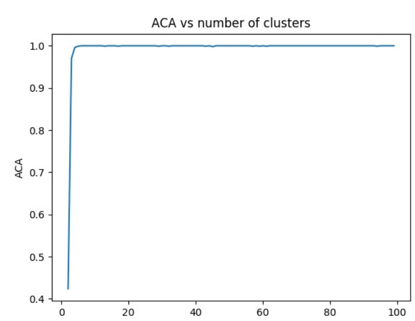

Figure 8. ACA performance in train set varying the number of clusters setting the number of images and trees in 20 and 40, respectively

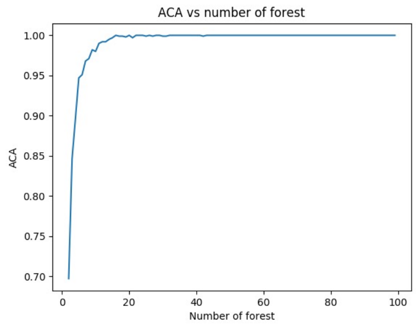

Figure 9. ACA performance in train set varying the number of trees in Random Forests setting clusters in 32 and the number of images in 20

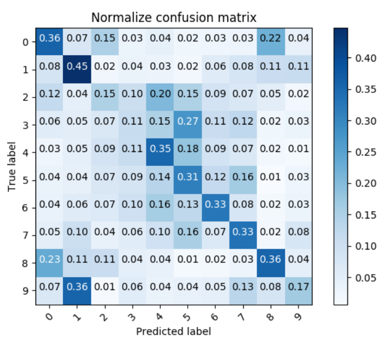

Figure 10. Confusion matrix in test set with 32 clusters, 20 images per class and 50 trees for Random Forests predictor model

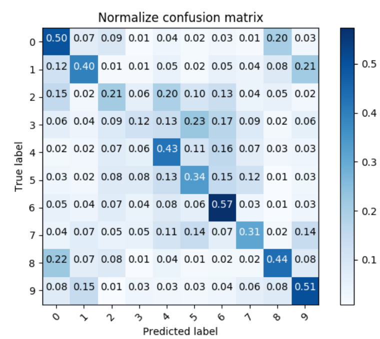

Figure 12. Confusion matrix in test set with 256 clusters, 50 images per class and 175 trees for Random Forests predictor model

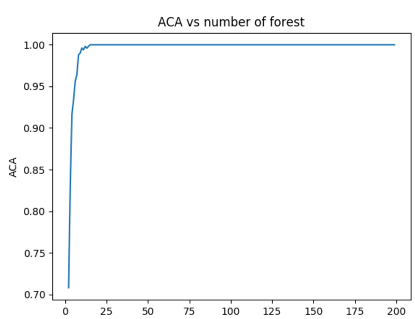

Figure 11. ACA performance in train set varying the number of trees in Random Forests setting clusters in 256 and the number of images in 50