Classification of Cifar10 Database with Texture

Natalia Andrea Durán Castro
Universidad de los Andes
na.duran@uniandes.edu.co

Ana M. Velosa Orduz
Universidad de los Andes
am.velosa@uniandes.edu.co

Abstract

In this paper is shown a method to classify the images of cifar10 dataset. This method is based on transform the images to texture space and use its histograms to develop the classification. In the classification part, there is a contrast between the results with KNN and Random Forest classifiers.

1. Introduction

Texton are the equivalent of atoms but in image processing, or how Béla Julesz in 1981 describe them, ”the putative units of pre-attentive human texture perception.”[5]. Since then, textons have acquire a lot of importance in the application of computer and machine vision, especially trying to find better solution to the segmentation and classification problems.[1]

In this paper we are going to focus on the classification problem. This problem is about identifying to what class an unknow object is member, based on different cues like similarity, symmetry, common fate,illusobry or subjective contours or even proximity [1].

A method to develop this is with the use of classifiers like K-Means, if we don’t know what we want to find, Super Vector Machines where you know what you want and you map the data in to a higher ideational space, where it is possible construct an optimal separating hyperplane in this space or use K nearest neighbor where you classified point using the distance between them [4]. Also, it is possible to use the classifier of Random Forest that is a supervised learning algorithm. This classifier builds an ensemble of decision trees. In this way, it can improve the result with the combination of learning models [3].The use of any of this classifiers depends of the number of categories and type of the dataset. For this reason, any classifier is considered better than other in a objective way.

The evaluation methodology to study this problem is the confusion matrix and ACA. In this matrix is presented a supervised learning, due to the presence of annotations, with the instances of predicted class and actual class. Generally, the rows corresponding to the predicted class and the columns to the actual class. In this case, it is possible to normalize this matrix with regard to the columns. The ACA index is the mean of the diagonals in this normalized confusion matrix. These diagonals presents the true positives values, which are the probability of classification to each category, where the ideal value is 1 [2].

In this project the dataset called CIFAR-10 was used, which have 60’000 32x32 color images divided in 10 classes. This dataset it divided also in 50’000 images for training and 10’000 for testing, which are chosen 100 randomly from each one of the class. Those images are organized in to batches, where you will find 5 training batches and 1 testing bath that has the images already explained before.[6]

2. Methodology

The dataset was downloaded with the functions of os library. Then, the script cifar10.py was used to read the images in pythons. The training images were divided in 5 parts, where each one had 10000 images. Even if these parts were unbalanced to each category, the complete train group was balanced.There is ten categories in this dataset: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

To convert the images to texture, it was necessary to use 16 filters in a way to recognize the spatial information. Then, it was used k-means to create a textons dictionary. The variable k was choosed with regard to the result of ACA in the train group.

After obtain the texton map, it was used to pass each image to texture space. To do this, it was used the function *assigntextons* .This functions use the euclidean distance to assign each pixel of each image to a category in the textons dictionary.Due to the time process was a variable to taking account, the dictionary of textons was building with the firsts 500 images of the trainig group. Then, these histograms of train images were used with the labels to train the classifier model.An example of the histogram without normalize is shown in the figure 1. However, it is important to know that the histograms that are going to be used for the classification are normalized.

In the other hand, we do not jet know which one of the classifier algorithms is better for this problem, so we decided to use Random Forest and KNN to take a decision based on the results we obtain. The SVM classifier was ruled out because it has not good results when there is more than two categories.

Fist we used KNN to classified the data, in this method were realized several tests where the variable k was changed. Then, it was choosed the k where the best ACA value was evidence. In the case of Random Forest, the value of k was the one found on the KNN tests. In this case, the value of estimators was varied. Those estimators indicate the number of trees in the forest, wich mean the number of combinations of learning models.

3. Results

The results of the algorithm with knn are shown in the figures 3 and 2. The figure 3 is the result of each test with KNN vary the value of k.In this, it is possible to observe that with the increase of the quantity of categories in the textons dictionary, there is a increase in the ACA or the test images. However, the value of ACA is less than 20 percent. This is much less that the chance. On the other side, the figure 2 shows the time processing of the algorithm with the KNN classifier taking account the value of categories in the textons dictionary.It does not has a lineal behavior.

The figures 4 and 5 shows the confusion matrix of Random forest. In comparisson with KNN, the group of train has a increase in its ACA. However, the test group does not. It was used as estimators 10 times the number of categories(10). Also, the k was who had the best result in the test of KNN.

4. Discussion of the results

The no linear behavior showns in the figure 2 is due to the quantity of process running in the GPU. So, there is not a complete information to conclude about the time that take the process with regard to k. To make a better analysis of time, it is necessary to taking account the process that were executed at the same time in the GPU used.

The results shows a better ACA with the method of KNN, but this is not the result expected. Also, both classifiers have not a good ACA, because it is too little. For this reason, it is better to evalue another method to classify these images.

References

[1] D. Charalampidis. Texture synthesis: Textons revisited. *IEEE Explore*, 15(1):777 – 787, 2006. 1

[2] L. Daza and P. Arbeláez. Ibio3470: Bases de datos en visión por computador, 2018. Universidad de los Andes, Bogotá, Colombia. 1

[3] N. Dondge. The random forest algorithm. 2018. 1

[4] J. V. J.A.K. Suykens. Least squares support vector machine classifiers. *Katholieke Universiteit Leuven*, (1):293–300, 1999. 1

[5] B. Julesz. ”textons, the elements of texture perception, and their interactions. *Nature*, 290(1):91–97, 1981. 1

[6] Krizhevsky.A. The cifar-10 dataset. último acceso 28 Febrero 2019. 1

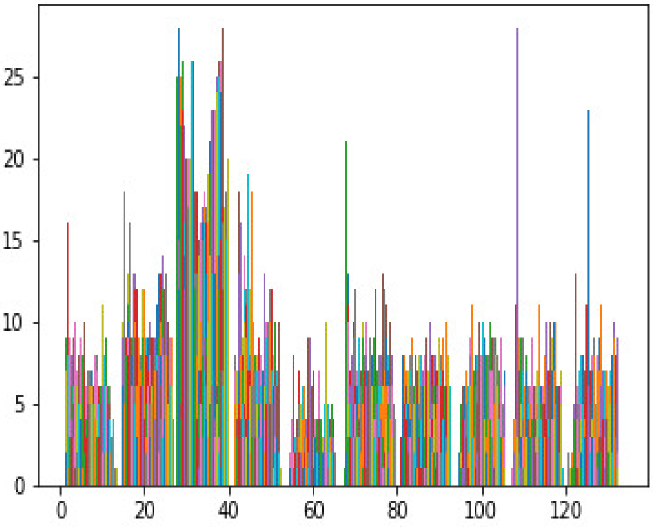

Figure 1. Histogram without normalizing with K= 135

KNN.PNG

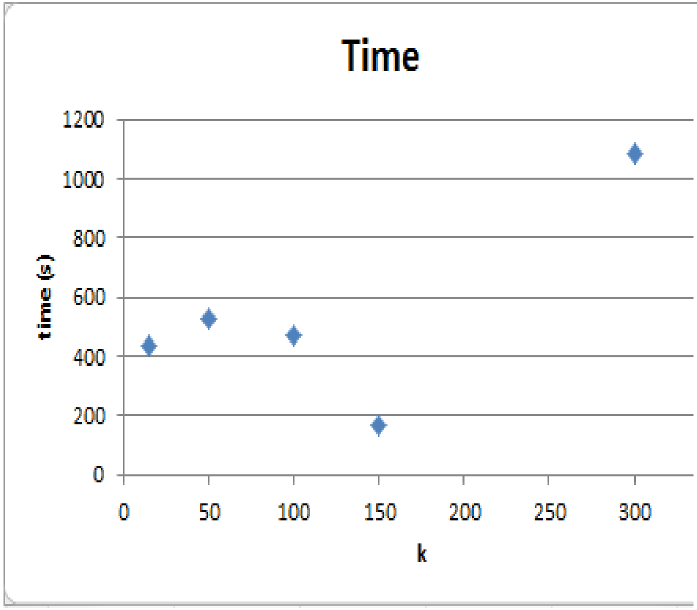

Figure 2. Time taking to process the algorithm with KNN with regard to the value of K

test.png

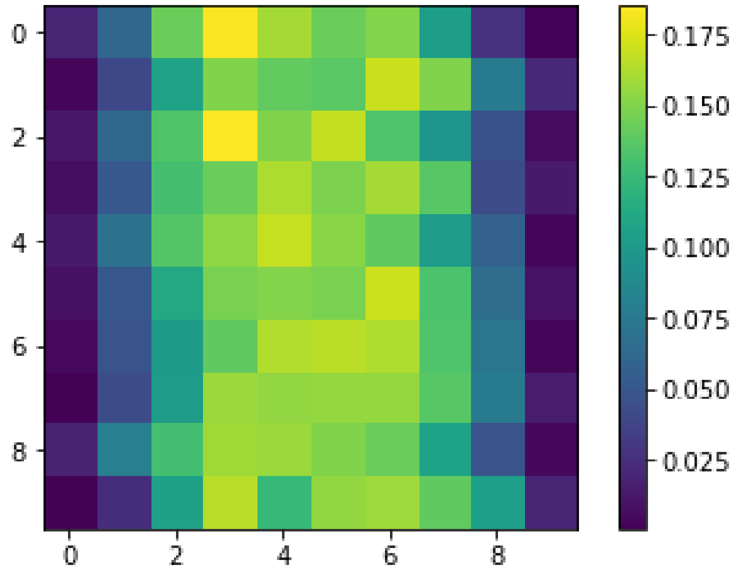

Figure 4. Random Forest’s confusion matrix with test group

KNN.PNG

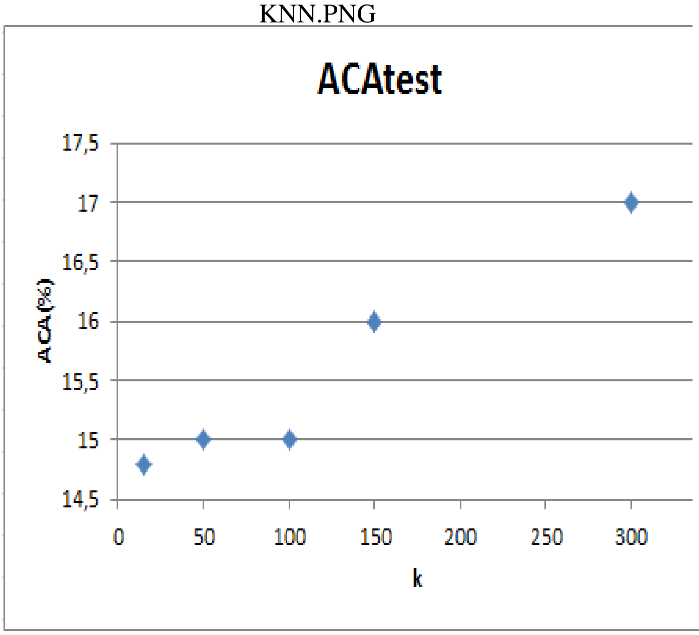

Figure 3. ACA of the algorithm with KNN with regard to the value of K

train.png

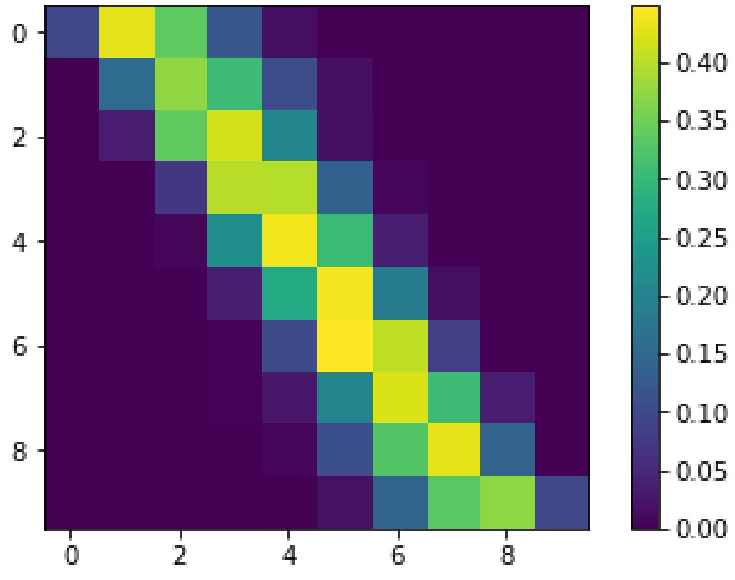

Figure 5. R
andom Forest’s confusion matrix with train group