Image classification using textons

Mauricio Neira
Universidad de los Andes
Cra 1 N^o 18A - 12, Bogotá - Colombia
m.neira10@uniandes.edu.co

Daniel Rodriguez
Universidad de los Andes
Cra 1 N^o 18A - 12, Bogotá - Colombia
da.rodriguez1253@uniandes.edu.co

Abstract

In this paper we classify the CIFAR 10 dataset using basic units of texture, known as textons. We generate these descriptors using k means on the feature space generated by the convolution of the dataset with a filter bank. We use two supervised methods of classification, k-Nearest Neighbour and Random Forests. We compare our classification results in terms of the main parameters of each sub-procedure.

1. Introduction

An image is described by its collection of boundaries, shapes and textures. Hence, in principle extracting the previous attributes enables us to classify a set of images. However, although boundaries and shapes are easy to define, textures defy a precise definition. Intuitively, textures are complex repeating patterns (although not perfectly symmetric nor necessarily complex). To study textures these complex patterns have been decomposed as the repetition of simpler structures representing units of texture. These units known as *textons* refer to fundamental micro-structures in natural images and are considered as the building blocks of preattentive human visual perception[3].

In this paper we classify the CIFAR 10 dataset using textons. First, we explain how we extract the texture features of our dataset. Then we describe the k-Nearest Neighbour and Random Forests classifiers employed. We compare our classification results in terms of the main parameters of each sub-procedure. Finally we discuss the main computational costs of our algorithm and the changes required to improve the accuracy of our classifier.

2. Images used

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. We use 1000 images composed of 100 random images of each class as training images and 10000 test images [1].

3. Procedure

We start with basic textures that describe horizontal, vertical edges. We rotate and scale the textures to obtain a filter bank of size 16. Each input image is convolved with the filter bank to extract local features. At all pixels, we obtain a 16-dimensional vector that stores the output of the convolution to each filter. Let n denote the size of our images and N the size of our training dataset. We denote our *feature space* as the $n \times n \times N$ 16-dimensional vector responses.

3.1. K-mean clustering in feature space

Assuming that local structures occur repeatedly in images of the same class we aim to identify the clusters formed by the vectors in the feature space. Following [2] we use clusters centers to build our dictionary of textons. Next we briefly describe our classification algorithms. In the results section we compare their performance based on their main parameters.

3.2. Nearest Neighbour

The k-Nearest Neighbour algorithm selects the class of an input by looking at its k-nearest neighbours in the feature space. Any given output of the classifier depends on the number of neighbours considered. The main assumption of this procedure is that features distance corresponds to their importance. Several variants have been proposed to properly scale features as the use of evolutionary algorithms.

3.3. Random Forests

The Random Forests algorithm selects the class of an input by building decision trees at the training phase and later choosing on a given statistic of these trees (e.g their mean or mode). The user is required to select the number of trees, the depth of the trees and the number of features or leaves to considered at each split. In 2002 it was pointed out that both Random Forests and Nearest Neighbours are weighted neighborhood schemes. Hence, for Random Forests, varying the stated parameters can be thought as tuning the com-

plex weighted function used to estimate the class of the input. Thus, we expect better classifying results using Random Forest vs k-Nearest Neighbours.

4. Results

In principle, we do not have a precise way to feed the best set of parameters for the two classifiers. Instead we run the k-Nearest Neighbours and Random Forest classifiers varying the neighbourhood size k and the number of trees M. We also run the k-means sub-procedure to see the influence of the number of textons. We select the average precision as our metric of performance since it is equivalent to the ACA ($\frac{\sum_1^n x_{i,i}}{n}$ where n is the number of classes) and the agreed metric for the challenge. In practical applications there is also a strict time constraint. Hence, we comment on the computation time of each algorithm later.

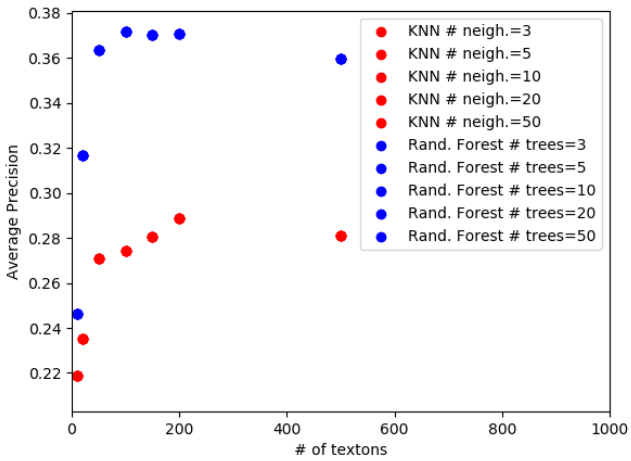

Figure 1. Average precision over classes against the number of textons (centroids in k-means). We plot curves for varying values of the neighbourhood size k and the number of trees M and obtain the same average precision.

From Figure 1 it is clear that Random Forest performs better than k-Nearest Neighbours for all numbers of textons. This out-performance is expected since Random Forests use a more complex weighting function to classify images. We can visualize this weighting function as a metric in k-Nearest Neighbours which gives non-circular neighbourhoods. Both methods show a decrease in performance for long values of the number of textons. This common phenomenon displays that a large number of descriptors may capture textures that are more common between classes than inside classes. Notice also that the precision average does not depend on the size of the neighbourhood nor the number of trees. Hence, we do not require the full power of the classifiers employed. We reason that data is distributed such that increasing the complexity of the classifier function does not change the output.

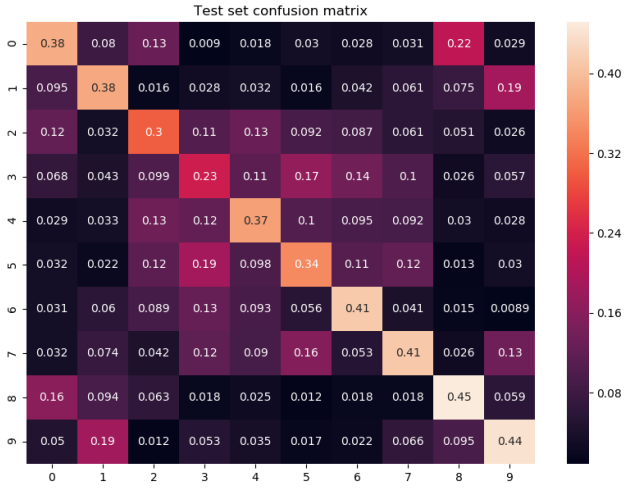

Figure 2. Confusion matrix of test set classification using the Random Forest procedure with 3 trees and 100 textons.

We discuss our results for individuals classes based on the Random Forest classifier with 3 trees and 100 textons 2. As a first observation, the confusion matrix for the training set shows perfect classification. We argue that this result is due not to overfitting but a small training dataset (100 images per class per 10 classes). The confusion matrix for the test shows different degrees of precision. The *ship* class was best classified with a precision value of 0.45. We reckoned that ship images have large bodies of water or sky with uniform texture surrounding them, and, in addition, ships are painted with few colours and simple geometric patterns. In fact, the 4 best classified classes (airplanes, automobiles, ships, trucks) correspond to machines or vehicles that exhibit the same attributes: large uniform backgrounds and simple textures. On the contrary animal classes (birds, cats, deers and dogs) exhibit a great variety in coat colours, patterns, textures and lengths. We obtain the lowest performance for such animal classes. Additionally our classifier methods tend to confuse between these classes since some of these species exhibit similar fur. Relating to our sensorial experience, it appears to us that humans mainly focus on shape cues to categorize these animals. Notice that horses are the most homogeneous of this species of animals. Hence, for the horse class we obtain a precision comparable to that of vehicles.

5. Conclusions

In this paper we have classified the CIFAR 10 data set using textons and supervised classifiers. We obtain a best average precision of 0.37 using Random Forest with 3 trees and 100 textons trained on only 1000 images (100 images per class). In this setting, Random Forests give a better performance than k-Nearest Neighbours under similar conditionsç. The best classifier exhibits a high performance in vehicle classes with large uniform backgrounds and simple textures given the small training set. However, the algorithm fails to distinguish between animal classes with complex textures and varied backgrounds. A possible enhancement of our procedure involves adding filters, like dots, that resemble the animal coats. However, extracting the feature space is the most computational demanding subprocedure in our algorithm. As humans we use with a varying degree attributes such as boundaries, shapes and textures depending on the classification task. Another approach is then to aid the classification with geometric features that extract the shape of the objects that represent the class. However, animal classes exhibit similar shapes (think of dogs and cats) so there are many limitations still to hold.

References

[1] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[2] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1010–1017. IEEE, 1999.

[3] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *International Journal of Computer Vision*, 62(1-2):121–143, 2005.