# Texton's Lab

Esteban Galvis Gutierrez
Univesidad de los Andes
e.galvis10@uniandes.edu.co

## Abstract

*This lab has the purpose of extracting a specific set of features: Textons. These features are usually representations of textures, in other words, of patterns that repeat themselves in an image. When they are extracted, we want to classify the type of textons an image has. To classify we are going to use two popular algorithms: Random Forests and Nearest Neighbors*

## 1. Introduction

For the goal of the laboratory, we are going to use the Dataset from the Ponce Computer Vision Research group [1]. More specifically, we want to work with the Texture Database since or main goal is to extract some textons of a training set and later test with a classifier a new image and see if it corresponds to the image class of the training set. But first, a Texton is a vectorized representation of a texture in an image. Textures are just images that have patterns that repeat themselves. But what is the relationship between the pattern and the texture, what defines the pattern? What defines a texture therefore is what we call textons.

## 2. Materials and Methods

As we mentioned above, we are going to be working with the Ponce Texture Database. This database is composed of 25 texture categories. Each category has 40 images of that specific texture. To use this data set, we randomly selected 10 images of each category to build up a test set. With the dataset in place, an important recipe to obtain textures is through some filters. Textures are composed of edges, curves and specific pattern like shapes. Thus, there must be a set of filters that model these variations. A texture is represented then by the response to the orientation and spatial information of a filter. [2] A texture with a curve will have a better response with a filter that is also curvy. Please see Figure 1 and Figure 2

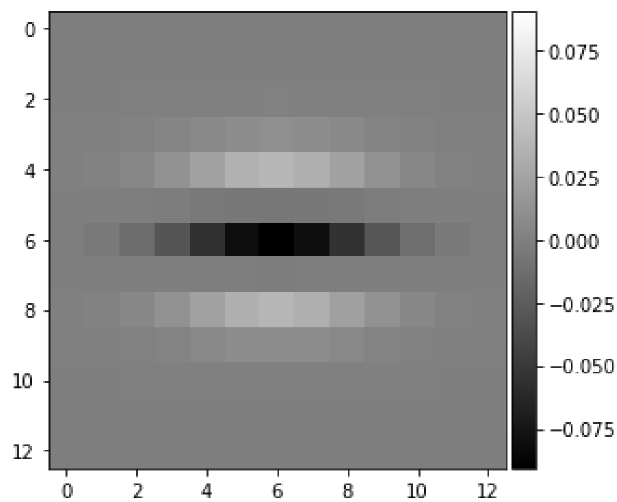


Figure 1: Example of a horizontal filter from the filter bank.

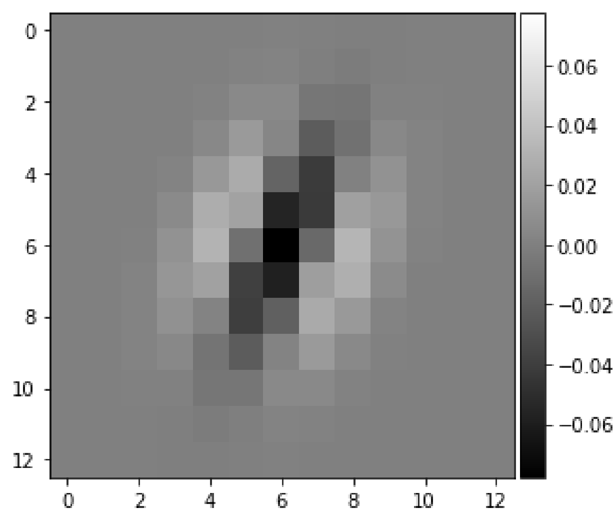


Figure 2: Example of a diagonal filter from the filter bank.

We are going to process each pixel of each training texture image with all the filters in the filter bank, which in this case there´re 16 filters.

Nonetheless, we are not taking the complete image size since it can be computation expensive. Thus, we restrict ourselves to a window of 30x30 pixels in each image to gather the texture information of the image. By doing this, we are sub-sampling our dataset and we can take this approach since a texture is an image of repetitive patterns. The patterns found in the 30x30 window, will be the same pattern that the whole image has.

When we have processed all pixels of the images with the filter bank, we gather all the responses that each pixel has and cluster the responses with an unsupervised algorithm as k-means, where our K will be the number of categories there is and obtained our textons.

With a baseline built, we can iterate through all the training and test samples and get their texton representations. Nonetheless, in this case we are interested in building a histogram representation of the image with a texton space instead of a color one. Since we have the labels of the images, we assigned the label to the corresponding texton histogram of the image. Now we have a training and test set with their corresponding texton´s histograms labeled ready to be fed to a classifier like Random Forest or Nearest Neighbors.

## 3. Results

Running a Nearest Neighbor classifier in the training data we obtain a normalized confusion matrix with an average classification error of 41%.

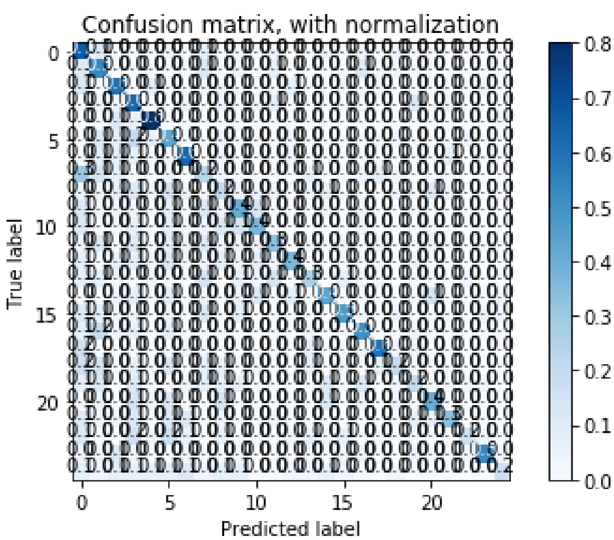


Figure 3: Normalized nn training confusion matrix.

Running a Nearest Neighbor classifier in the test date gets as an average classification error of 22%

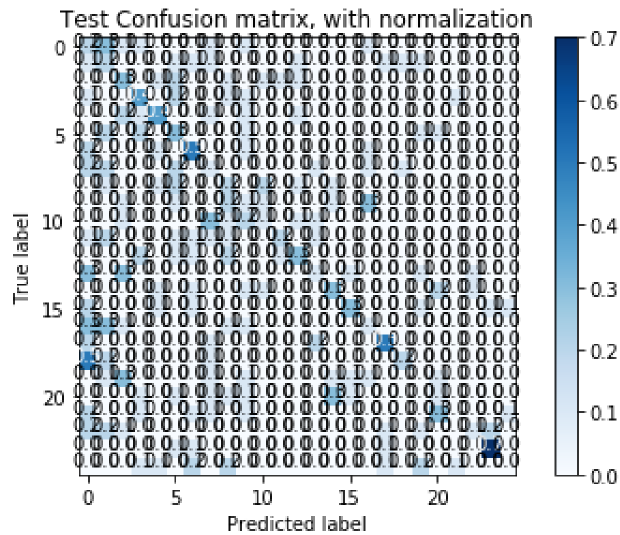


Figure 4: Normalized nn testing confusion matrix.

Running a Random Forest to the training set with 10 estimators has as an average classification error of 99%.

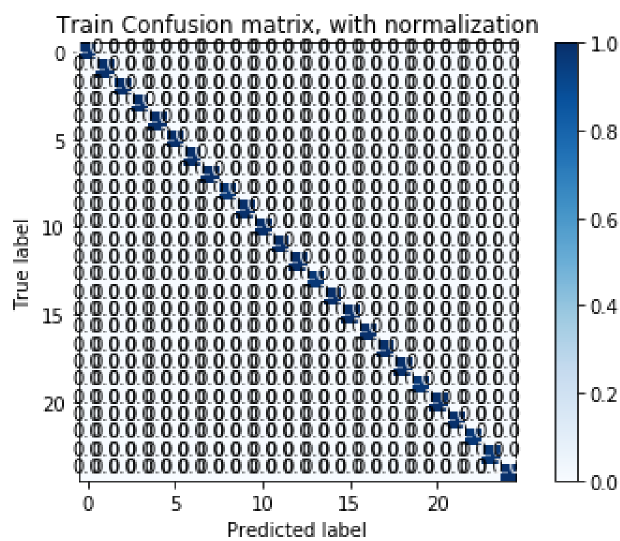


Figure 5: Normalized random forest training confusion matrix

Running a Random Forest to the test set with 10 estimators has as an average classification error of 20.7%.

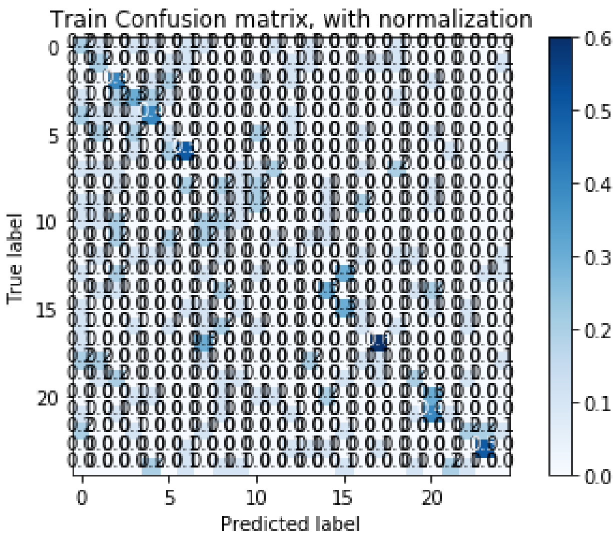


Figure 6: Normalized random forest testing confusion matrix

To conclude, our models are not having excellent results. This might be due to the small window we fixed at the beginning of our pipeline. However, we see that Random Forest performs better in the training set and it is really close to obtaining similar results in the testing set as NN. The respective confusion metrics of our experiments tells us how well our models predicted the target label.

## References

[1] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A Sparse Texture Representation Using Local Affine Regions. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1265-1278, August 2005.

[2] Th. Leung and J. Malik, Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons, ICCV 1999.