State University of Campinas
Institute of Computer Science
MO443 - Professor: Hélio Pedrini

Project 4

June 19, 2021

Pedro Henrique Vaz Valois
p265676@dac.unicamp.br

***Abstract*—Digital images are abundant in the current age. Be it as means of modern art or source of data, they have a high degree of importance in modern society. Thus, the need for processing, transforming and understanding such images has risen and a multitude of techniques were developed for these tasks. In this project, we analyze a variety of textures using local binary patterns and gray level co-occurrence matrices.**

I. Background

A monochromatic image is defined as a $M \times N$ matrix

$$I = (I_{mn}), 0 \leq m \leq M-1, 0 \leq n \leq N-1$$

where M is the image pixel width, N is the image pixel height and I_{mn} indicates the intensity in the $[0, 255]$ range for the pixel in position (m, n).

In this sense, a texture is a pattern contained within the image that has a degree of spatial repetition.

A. Local binary patterns

Local binary patterns (LBP) is an image transformation that highlights patterns based on the pixel surroundings in a binary fashion.

From the histogram of a LBP transformed image, it is possible to extract features such as flat shapes, edges and corners.

B. Gray level co-occurrence matrices

GLCM is a matrix histogram of co-occurring grayscale values at a given offset over an image. These values are measured for a fixed distance and angle between region within the image.

The GLCM has several properties such as contrast, entropy, energy and correlation that characterize the image under analysis.

II. Implementation

All code for this project has been developed and tested with `Python 3.7.9`, `numpy 1.20.2`, `matplotlib 3.4.1` and `scikit-image 0.18.1`

The software that implements the methods discussed in the last section is contained in the `script.py` file. It depends on `numpy` to perform the vectorized operations, `matplotlib` for reading and saving image files and `scikit-image` for convex hull extraction. Moreover, it uses python standard library's `os` module for checking if the output file already exists and the `argparse` module for parsing command line arguments.

Images and kernels are represented in the code as 2D `numpy` arrays in order to gain performance and also to simplify most math operations.

The `argparse` module provides a help text for using the script. By typing `python script.py --help`, a helper message is shown in the console.

The code was designed to deal with RGB files, and no tests were performed with monochromatic images. Despite that, all images are converted to grayscale upon input.

III. Results and Discussion

A. Local binary patterns

Initially, we apply LBP for 4 different textures setting 3 different levels for the number of points in the considered neighborhood: 1, 2 and 8. In figure 2, we can see that the LBP highlight specific patterns with granularity varying with the neighborhood size. In this sense, this may give a sense of direction because smaller neighborhoods can have different orientations that larger ones on the same image.

B. Local binary patterns histogram comparison

We apply LBP to four different textures and build the LBP histograms to compare each texture. The resulting LBP image and histograms for each texture are shown on figure 2 and the distances from texture 5 to the other textures is available at table I.

The results show that textures 5 and 6 are the closest whereas texture 7 is the most distinct. In truth, textures 5, 6 and 8 have well defined contours, borders, edges or corners while texture 7 is a diffuse mass. This fact results in a lbp that is quite different from the original texture 7, showing that it lacks proper patterns lbp highlight.

Table I: Distances between texture 5 and textures 6, 7 and 8

Texture	Euclidean	Bhattacharyya	Chi-square	Correlation
6	0.11	0.064	0.0057	0.79
7	0.38	0.27	0.052	0.33
8	0.16	0.13	0.012	0.55

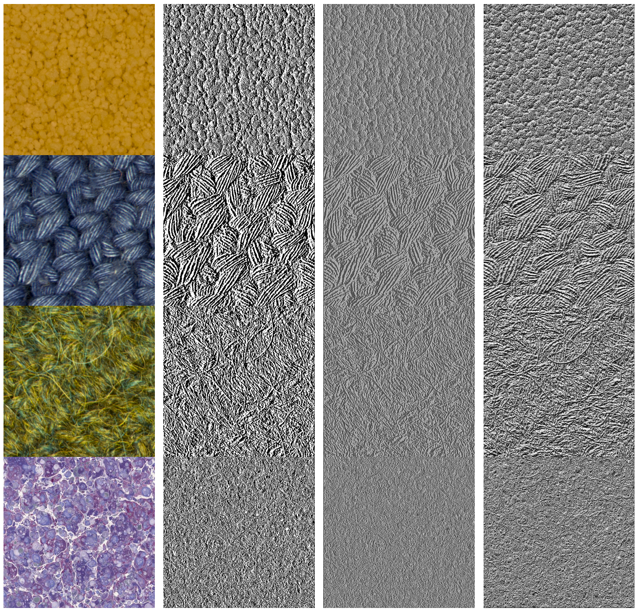

Figure 1: LBP applied to textures 1, 2, 3 and 4 (left) for neighborhoods of size 1, 2 and 8 (from left to right)

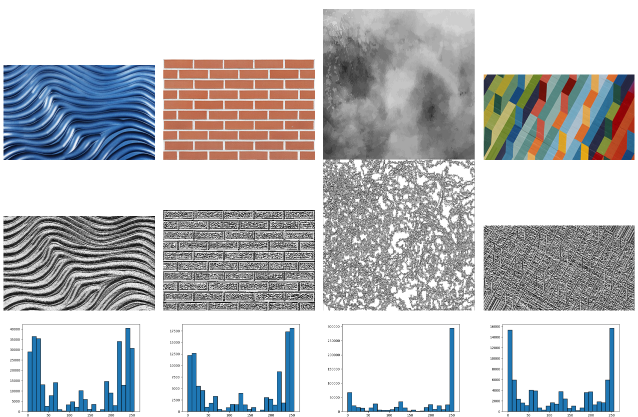

Figure 2: LBP applied to textures 5, 6, 7 and 8 (top) for neighborhood of size 1, 2 and 8 (from left to right)

C. *Co-occurrence matrices comparison*

Finally, we use GLCM metrics to characterize the textures on different aspects. Table II summarizes the textures in 5 different values. In this sense, we verify the idea of diffuseness in texture 7, showing it has the least level of momentum, contrast and correlation. On the other hand, we can check the lack of uniformity in the waves within texture 5 due to its high entropy and low energy.

Table II: GLCM metrics for textures 5, 6, 7 and 8

Texture	2nd-momentum	Entropy	Contrast	Energy	Correlation
5	369.27	1655.21	133.68	0.02	0.97
6	14.83	122.57	147.19	0.08	0.95
7	6.47	142.15	20.85	0.06	0.99
8	18.22	327.83	148.97	0.04	0.96

IV. Conclusion

In this project, local binary patterns and gray level co-occurrence matrices were built as means of analyzing textures. These tools have a wide range of applications, being able to extract a multitude of features to explain the nature of a textured pattern.