UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Environment for the analysis and comparison of texture descriptors

Alex Josué Flórez Farfán

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Alex Josué Flórez Farfán

Environment for the analysis and comparison of texture descriptors

Dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Odemir Martinez Bruno

**USP – São Carlos
December 2018**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

F632e Flórez Farfán, Alex Josué
Environment for the analysis and comparison of
texture descriptors / Alex Josué Flórez Farfán;
orientador Odemir Martinez Bruno . -- São Carlos,
2018.
121 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2018.

1. Computer Vision. 2. Texture Analysis. 3.
Texture Classification. 4. Texture Descriptor. 5.
Web Environment. I. , Odemir Martinez Bruno,
orient. II. Título.

Alex Josué Flórez Farfán

Ambiente para análise e comparação de descritores de textura

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Mestre em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. Odemir Martinez Bruno

USP – São Carlos
Dezembro de 2018

*This work is dedicated to my parents Claudio and Benigna.
Thank you for your loving support throughout my life.*

ACKNOWLEDGEMENTS

Thank you to:

God, for bringing me to this wonderful country Brazil.

My family for their unconditional love and continuous encouragement.

My brother Samuel for being a permanent example in my life.

My friends at church and laboratory for their friendship, faith, and companionship.

My advisor Odemir Bruno for his guidance, inspiration, and patience during this master's.

Instituto de Ciências Matemáticas e de Computação ICMC - USP, for the privilege to study in this institution.

Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq for the master's scholarship.

*“Is it possible for a man
to dream continuously
for seventy years?”
– Honi the Circle Maker*

RESUMO

FLÓREZ FARFÁN, A. **Ambiente para análise e comparação de descritores de textura.** 2018. 121 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

A análise de textura é uma área ativa de pesquisa que desempenha um papel importante em aplicações de visão computacional. A textura, juntamente com a cor e a forma, contém características importantes de uma imagem. A análise de textura permite caracterizar regiões dentro de uma imagem usando descritores. Esses descritores são aplicados no estudo de classificação de texturas, no qual o objetivo é identificar características que distingam uma determinada textura e atribuir um rótulo a uma imagem baseada nessas características. Devido à importância da análise de textura na visão computacional, os pesquisadores estão continuamente criando e desenvolvendo novos descritores, com o objetivo de melhorar o poder discriminativo dessas características em uma imagem. Uma tarefa difícil na análise de textura é comparar esses descritores e verificar quais são os mais adequados para cada tipo de tipo de imagem. A falta de uma boa revisão e comparação de descritores de textura pode fazer com que algumas aplicações não utilizem o descritor mais adequado para um tipo específico de textura. Portanto, nesta dissertação foi desenvolvida uma plataforma de pesquisa e colaboração para a análise e comparação de descritores de textura e conjuntos de dados de textura. A plataforma visa apoiar os pesquisadores na área de análise de textura, especificamente na classificação de texturas. A plataforma foi útil para realizar uma comparação extensiva de descritores de textura e vários conjuntos de dados de textura. Com essa plataforma, em alguns conjuntos de dados os resultados encontrados foram melhores que aqueles encontrados anteriormente na literatura. Os resultados indicam que a acurácia de classificação muda segundo o descritor e o classificador usado. Mudando os valores dos parâmetros dos descritores de textura foi possível obter acurácias diferentes e até melhores.

Palavras-chave: Visão por computador, Análise de textura, Classificação de textura, Descritor de textura, Classificação de textura, Ambiente web.

ABSTRACT

FLÓREZ FARFÁN, A. **Environment for the analysis and comparison of texture descriptors.** 2018. 121 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Texture analysis is an active area of research and plays an important role in computer vision applications. Texture, along with color and shape, contains important features of an image. Texture analysis allows to characterize regions inside an image by using descriptors. These descriptors are applied in the study of texture classification, in which the goal is to identify features that characterize a particular texture and assign a label to an image based on these features. Because of the importance of texture analysis in computer vision, researchers are continually devising and developing new descriptors, with the aim to improve the discriminative power of texture features of an image. A difficult task in texture analysis is to compare these descriptors and verify which are the most suitable for each type of image. The lack of a good review and comparison of descriptors cause that some applications do not use the most appropriate descriptor for a specific type of texture. Therefore, in this dissertation it was developed a research and collaboration platform for the analysis and comparison of texture descriptors and texture datasets. The platform aims to support the researchers in the area of texture analysis, specifically in texture classification. The platform was useful to perform an extensive comparison of texture descriptors and various texture datasets. Using the platform, in some datasets the results produced were better than those previously found in the literature. The results indicate that the classification accuracy varies according to the descriptor and classifier employed. By varying the parameters of texture descriptors it was possible to get different, yet better, classification accuracies.

Keywords: Computer Vision, Texture Analysis, Texture Classification, Texture Descriptor, Web Environment.

LIST OF FIGURES

Figure 1 – An example of textures found at University São Paulo, Campus São Carlos. The building presents a regular texture, with windows as its basic pattern, instead the natural elements present irregular textures, as is the case with the grass.	32
Figure 2 – A set of strawberries show <i>macro</i> texture properties. Taking a closer view at individual strawberries, another kind of texture is revealed, a <i>micro</i> texture. .	33
Figure 3 – A representation of dynamic textures as cubes. Dynamic textures coming from the movement of tree branches and marine algae.	34
Figure 4 – The two main processes that comprise texture classification: feature extraction and classification. Its input is an image dataset and its output are the classification results.	37
Figure 5 – The feature extraction process uses a texture descriptor to transform the input image into a feature vector.	38
Figure 6 – LBP operator applied on a 3×3 neighborhood.	40
Figure 7 – Comparison between cross-validation and stratified cross-validation. The label for each class is ordered.	44
Figure 8 – Sample texture images from the 1200Tex dataset.	47
Figure 9 – Sample texture images from the Brodatz dataset.	48
Figure 10 – Sample texture images from the CURET dataset.	48
Figure 11 – Sample images from the Kylberg texture dataset.	49
Figure 12 – Sample images from the KTH-TIPS2b texture dataset.	49
Figure 13 – Sample texture images from the Midrib dataset.	50
Figure 14 – Sample texture images from the Outex dataset.	51
Figure 15 – Sample texture images from the STex dataset.	51
Figure 16 – Sample texture images from the UIUC dataset.	52
Figure 17 – Sample texture images from the USPTex dataset.	53
Figure 18 – Sample texture images from the VisTex dataset.	53
Figure 19 – Design of the application showing the three layers of the architecture: Web Interface, Core Application, and Data Storage. The arrows signal the flow of data among the components of the application.	58

Figure 20 – The database design of the application. The feature extraction process is represented by the entities FeatureExtraction , TextureAlgorithm , Parameter , and ImageDataset . The classification process is represented by the entities Classification , Classifier , Validation , and Metrics	60
Figure 21 – The interface to make experiments presents to the user 4 drop down lists to choose values for Texture Algorithm, Image Dataset, Classifier, and Validation. The values of the parameters only appear when the user selects an algorithm.	62
Figure 22 – The first interface to upload source code of a texture algorithm. In the image the user provides the file CLBP.zip.	63
Figure 23 – The second interface to upload source code of a texture algorithm. In the image the user chooses the main file of the algorithm, and adds parameters specifying their names and values.	63
Figure 24 – The interface to upload texture datasets. In the image the user writes a unique name for the dataset, and specify the image format of the images. After the uploading the dataset is ready to be used in the application.	64
Figure 25 – The search interface helps to find previous classification results. The user has the option to leave the four boxes in blank, in which case the classification results will be shown in descending order, or can choose some of the four select boxes to view more specific results.	65
Figure 26 – Using the search interface to get classification results using the CLBP algorithm over the USPTex dataset with the RandomForest and LDA classifiers with a 10-fold cross validation scheme. The results are presented in descending order according to their accuracy.	66
Figure 27 – Metrics of classification results for an experiment that includes the descriptor LPQ, dataset UIUC, and classifier RandomForest.	67
Figure 28 – Confusion matrix showing the results of classification of LPQ over UIUC, using RandomForest with 10-fold cross validation. The cell clicked with red border indicates that 6 images were misclassified as <i>T14_brick1</i> instead of the actual class <i>T02_bark2</i> . The correspondent images appear at the right of the confusion matrix.	68
Figure 29 – The Web application adds tasks to the Queues <i>Features</i> and <i>Classify</i> . The Workers execute the tasks and store the generated data in the database. . . .	70
Figure 30 – Confusion matrices for the 1200Tex dataset. The confusion matrices correspond to the classification result using the Adaptive Hybrid Pattern and Complex Networks descriptors with the LDA classifier and 10-fold cross-validation scheme.	78
Figure 31 – Confusion matrix for the CURET dataset. It was used the AHP descriptor and the kNN classifier.	85

Figure 32 – Confusion matrices for the KTH-TIPS2b dataset. The confusion matrices correspond to the classification result using the BSIF and Fuzzy LBP descriptors with the kNN classifier and 10-fold cross-validation scheme.	89
Figure 33 – Classification results for LBP in Midrib dataset. Varying the parameters radius and points. Increasing the radius and maintaining 8 points the accuracy improves and it got its best result at <i>radius</i> = 3.	91
Figure 34 – Confusion matrix for the best result in Midrib dataset. It was used the CLBP descriptor and the kNN classifier.	93
Figure 35 – Confusion matrices for the Outex-10 dataset. The confusion matrices correspond to the classification result using the LFD and BSIF descriptors with the kNN classifier and 10-fold cross-validation scheme.	95
Figure 36 – Sample of texture images from STex dataset with most misclassified elements.	97
Figure 37 – Confusion matrices for the UIUC dataset. The confusion matrices correspond to the classification result using the LETRIST descriptor with the LDA and kNN classifiers and 10-fold cross-validation scheme.	100

LIST OF TABLES

Table 1	– Image datasets used in this work.	46
Table 2	– Summary of texture descriptors used in the platform.	74
Table 3	– Classification results in 1200Tex dataset using kNN, Random Forest and LDA with 10-fold cross validation.	75
Table 4	– Comparing classification results in 1200Tex dataset with texture descriptors not present in the platform.	76
Table 5	– The best 10 classification results in 1200Tex happens with the Adaptive Hybrid Pattern descriptor using LDA and 10-fold cross validation.	77
Table 6	– Classification metrics in 1200Tex dataset using the Adaptive Hybrid Pattern (AHP) descriptor and Complex Network (CN) descriptor using LDA classifier with 10-fold cross validation.	79
Table 7	– Classification results in Brodatz dataset using kNN, Random Forest and LDA with 10-fold cross validation.	80
Table 8	– The best 10 classification results in Brodatz dataset are obtained using LDA and 10-fold cross validation.	81
Table 9	– Classification metrics in Brodatz dataset coming from CLBP and AHP descriptors which produced the best results using LDA classifier with 10-fold cross validation.	81
Table 10	– Classification results in CURET dataset using kNN, Random Forest and LDA with 10-fold cross validation.	83
Table 11	– The best 10 classification results in CURET happens with the Adaptive Hybrid Pattern descriptor using kNN and 10-fold cross validation.	83
Table 12	– Classification results in Kylberg dataset using kNN, Random Forest and LDA with 10-fold cross validation.	86
Table 13	– The best 10 classification results in Kylberg dataset are obtained using kNN and Random Forest classifiers.	87
Table 14	– Classification results in KTH-TIPS2b dataset using kNN, Random Forest and LDA with 10-fold cross validation.	88
Table 15	– Classification metrics in KTH-TIPS2b dataset using the kNN classifier. The data comes from the BSIF and Fuzzy LBP descriptors.	89
Table 16	– Classification results in Midrib dataset using kNN, Random Forest and LDA with 10-fold cross validation.	90

Table 17 – The best 10 classification results in Midrib dataset comes from using the CLBP descriptor and the kNN and LDA classifiers.	92
Table 18 – Classification results in Outex-10 dataset using kNN, Random Forest and LDA with 10-fold cross validation.	94
Table 19 – The best 10 classification results in Outex-10 dataset comes from using different descriptors and parameters and kNN and Random Forest classifiers. . . .	95
Table 20 – Classification results in STex dataset using kNN, Random Forest and LDA with 10-fold cross validation.	97
Table 21 – Classification results in UIUC dataset using kNN, Random Forest and LDA with 10 fold cross validation.	98
Table 22 – Classification metrics in UIUC dataset using the LETRIST descriptor with LDA and kNN classifier and 10-fold cross validation.	101
Table 23 – Classification results in USPTex dataset using kNN, Random Forest and LDA with 10-fold cross validation.	102
Table 24 – The best 10 classification results in USPTex dataset are obtained using LDA and 10-fold cross validation.	103
Table 25 – Classification results in VisTex dataset using kNN, Random Forest and LDA with 10-fold cross validation.	104
Table 26 – Comparison of Texture Descriptors and Texture Datasets	105

CONTENTS

1	INTRODUCTION	25
1.1	Objective	27
1.2	Organization	27
2	TEXTURE	29
2.1	Introduction	30
2.2	Definition	31
2.2.1	<i>Types of Texture</i>	31
2.2.2	<i>Texture and color information</i>	33
2.3	Texture Analysis	34
2.4	Final Considerations	35
3	TEXTURE CLASSIFICATION	37
3.1	Feature Extraction	38
3.2	Texture Descriptors	39
3.2.1	<i>Importance of Texture descriptors</i>	39
3.3	Taxonomy of Texture Descriptors	40
3.4	Classification	42
3.4.1	<i>k Nearest Neighbors</i>	43
3.4.2	<i>Random Forest</i>	43
3.4.3	<i>Linear Discriminant Analysis</i>	43
3.5	Validation scheme	43
3.6	Final Considerations	44
4	IMAGE DATASETS	45
4.1	Texture Datasets in the platform	45
4.1.1	<i>1200Tex</i>	46
4.1.2	<i>Brodatz</i>	47
4.1.3	<i>CUReT</i>	47
4.1.4	<i>Kylberg</i>	48
4.1.5	<i>KTH-TIPS2b</i>	49
4.1.6	<i>Midrib</i>	50
4.1.7	<i>Outex</i>	50
4.1.8	<i>STex</i>	51

4.1.9	<i>UIUC</i>	52
4.1.10	<i>USPTex</i>	52
4.1.11	<i>VisTex</i>	52
4.2	Final considerations	53
5	TEXTURE EVALUATION PLATFORM	55
5.1	Requirements for Texture Descriptor	56
5.2	Application Design	56
5.2.1	<i>Architecture Design</i>	57
5.2.1.1	<i>Web Interface</i>	57
5.2.1.2	<i>Core Application</i>	59
5.2.1.3	<i>Data Storage</i>	59
5.2.2	<i>Database Design</i>	59
5.3	User Interfaces	61
5.3.1	<i>Make experiments of Texture Classification</i>	61
5.3.2	<i>Test new Texture Descriptors</i>	62
5.3.3	<i>Classify new Texture Datasets</i>	63
5.3.4	<i>Review Classification Results</i>	64
5.3.5	<i>Analyze metrics of classification</i>	65
5.3.6	<i>Interact with the Confusion Matrix</i>	66
5.4	Implementation Requirements	67
5.4.1	<i>Software details</i>	68
5.4.2	<i>Hardware used</i>	69
5.5	Final considerations	70
6	COMPARATIVE ANALYSIS OF TEXTURE DESCRIPTORS AND DATASETS	73
6.1	Analysis of Texture Datasets	73
6.1.1	<i>1200Tex dataset</i>	74
6.1.2	<i>Brodatz dataset</i>	77
6.1.3	<i>CUReT dataset</i>	82
6.1.4	<i>Kylberg dataset</i>	84
6.1.5	<i>KTH-TIPS2b dataset</i>	86
6.1.6	<i>Midrib dataset</i>	89
6.1.7	<i>Outex-10 dataset</i>	92
6.1.8	<i>STex dataset</i>	96
6.1.9	<i>UIUC dataset</i>	96
6.1.10	<i>USPTex dataset</i>	99
6.1.11	<i>VisTex dataset</i>	102
6.2	Global Comparison	103

6.3	Final Considerations	106
7	CONCLUSIONS	107
7.1	Contributions	108
7.2	Ideas for future works	108
	BIBLIOGRAPHY	111

INTRODUCTION

Vision is the most important sense in humans because it provides meaning and significance of the surrounding environment. Vision allows humans to understand their environment and 40% of the brain is dedicated to process information coming from vision (MARIEB; WILHELM; MALLATT, 2014). This kind of visual information is of interest to scientists in the field of Computer Vision, since these applications aim to extract meaningful information from images and videos, because they contain a rich source of data. Texture, along with color and shape, helps to understand the contents of an image, because texture represents important features of an image. Even though the understanding and perception of texture is natural for humans, the task of defining texture is not an easy one and so it has no formal definition.

Even if texture is not an easy concept to define, it is possible to represent an image by its visual texture in order to extract features to further processing such as image segmentation, texture classification, or texture synthesis. Texture is associated to the spatial arrangement of pixels in an image, in relation to their intensity or position. It also can be seen as a visual complex pattern composed of the interaction of elements that share some characteristics such as brightness, color, depth, and size. In the real world texture can occur at different scales, at arbitrary rotations, and it is affected by varying lighting conditions. Thus, it is normal that texture datasets contain images that present differences in illumination, position, noise, among other factors that make the feature extraction process a challenging task.

Texture analysis allows to identify an image or a region inside an image as texture, thus it helps in understanding the contents of an image. The analysis and recognition of texture is an active area of research playing an important role in computer vision and image processing applications, such as biomedical image analysis (FRANCIS *et al.*, 2014; SELVAN; DEVI, 2015; PINTOR *et al.*, 2016), industrial inspection (BIANCONI *et al.*, 2012; FELICIANO; LETA; MAINIER, 2015), analysis of satellite or aerial imagery (TANG *et al.*, 2015; CHOWDHURY; VERMA; STOCKWELL, 2015; WEI; HONG-YING, 2016), document image analysis (BERTOLINI *et al.*, 2013; HANNAD; SIDDIQI; KETTANI, 2016), and texture synthesis for

computer graphics or animation (GUO *et al.*, 2015; DONG *et al.*, 2016).

Because of the important role of texture analysis in computer vision, researchers are continuously devising and developing texture descriptors, with the aim to improve the discriminative power of texture features of an image. So, there are many texture descriptors devised in the literature in recent years with the purpose of assessing specific features of texture datasets.

A hard task in texture analysis is to compare these descriptors and verify which are the most suitable for each type of texture. The lack of a good review and comparison of texture descriptors in the literature cause that some applications do not use the most appropriate method for a specific type of texture. It is important to note that the discriminative power of a texture descriptor can vary among different image datasets. Therefore, it is not assumed that one texture descriptor should perform well for every image dataset, so it is important to test texture descriptors with different types of image datasets.

Therefore, this work proposes the development of a research and collaboration platform for the analysis and comparison of texture descriptors and texture datasets. The platform is a web-based application that aims to support the researchers in the area of texture analysis, specifically in texture classification. Among the functionalities that the platform provides are the ability to test the relative performance of new texture descriptors over benchmark datasets as well as to classify new image datasets with ready-to-use texture descriptors present in the platform. Using the platform, a researcher can easily modify the settings of an experiment of texture classification, which is comprised of the feature extraction and classification processes. For the feature extraction process, the platform contains a number of texture descriptors and image datasets ready-to-use. It provides means to upload new texture descriptors, so they can be compared with those already present in the platform. It also allows the uploading of images datasets to be analyzed by texture descriptors. For the classification process, the platform provides classifiers and a validation scheme. Different classifiers produce different results in the same set of texture features.

A web-based application represents an efficient medium to gather and share data. The scientific community benefits from using web-based tools to study, analyze, and share data coming from texture analysis. A web-based platform could promote research and collaboration among researchers, which provides facilities to test existing or new texture descriptors over benchmark image datasets or new image datasets. There are examples of web-based applications aimed to help in the analysis of images in fields such as biological, medical, and educational. Castañón *et al.* (2007) describes a web application to analyze photomicrographs of protozoan parasites using morphologic characteristics of shape and texture. Barnathan, Zhang and Megalooikonomou (2008) presents a web application for image analysis to assist researchers in the analysis of medical image data. Tramontan *et al.* (2011) proposes a web-based system to analyze images of the vascular structure of the eye fundus because it provides vital information to diagnose retinopathy. Garcia, Guzmán-Ramírez and Pacheco (2015) presents a tool that helps in the

teaching of the undergraduate course of electronics and image processing.

1.1 Objective

The objective of this master's is the analysis and comparison of texture descriptors and texture datasets through the use of a research and collaboration platform that simplify the creation of experiments in texture analysis, specifically in texture classification. With the goal to test the relative performance of new texture descriptors over benchmark datasets, and to classify new image datasets with ready to use texture descriptors.

1.2 Organization

The organization of this dissertation is as follows. [Chapter 2](#) describes basic concepts of texture analysis, highlighting the importance of texture in images. [Chapter 3](#) describes texture classification and the processes that comprise it: feature extraction and classification. [Chapter 4](#) presents a description of the texture datasets used in the platform. [Chapter 5](#) explains the design, features, and functionalities of the proposed platform, Texture Evaluation Platform (TEP). [Chapter 6](#) presents the results of the analysis and comparison of texture descriptors and texture datasets produced using TEP. Finally, [Chapter 7](#) presents the conclusions of this dissertation.

TEXTURE

The study of texture is important in research areas of image processing and computer vision. In fact, applications of texture are present in many fields of science and industry, such as medicine, to detect malignant tissues in breast (FRANCIS *et al.*, 2014), to classify oral cysts (FLORINDO; BRUNO; LANDINI, 2017), in biology, to diagnose the nutritional status in maize (ROMUALDO *et al.*, 2014), to identify forage species (FLORINDO *et al.*, 2014), to classify wood through their transverse cross-section (SILVA *et al.*, 2017), in industrial settings, to monitor metal corrosion (FELICIANO; LETA; MAINIER, 2015), to characterize nanostructured materials (FLORINDO *et al.*, 2013), in surveillance systems, to detect crowds (FAGETTE *et al.*, 2014), among others. These sample applications indicate that texture provides useful information about regions of interest that share similar characteristics. Texture does not have a formal definition in the scientific community. However, texture is widely used, as the examples just seen, because people have a natural understanding about what texture means.

Vision is the main sense that humans use to understand their environment. Around 40% of cerebral cortex is dedicated to process visual information, and the brain gives meaning to the visual information by analyzing color, shape and movement (MARIEB; WILHELM; MALLATT, 2014). Computer vision applications try to emulate the ability of the human visual system to understand scenes and the environment, with the goal to extract information and understand the contents of images and videos, even in non-optimal conditions, such as poor spatial resolution, nonuniform lighting, geometric distortions, or noise (MARQUES, 2011).

This chapter introduces texture as a visual attribute of a digital image. Section 2.1 introduces the notion of texture in a general sense. Section 2.2 defines texture as a visual and tactile attribute, and gives some definitions of texture in the scientific community. Section 2.3 explains areas of application of texture analysis. Finally, Section 2.4 discusses the ideas presented in this chapter.

2.1 Introduction

A peculiarity of human beings is how they perceived things. Our visual system allows us to see and recognize an isolated object, and if more than one object are spatially grouped, our mind perceives it as if they were one ([LANDY; MOVSHON, 1991](#)). Considering a single coffee bean and a set of coffee beans over a table. In the former there is an individual coffee bean, and in the latter the spatial arrangement of the coffee beans are an entity of their own and with different characteristics when grouped.

A broad definition given by [Macmillan \(2018\)](#) describes texture as “the effect that is produced when different things combine”, unsurprisingly, it is expected to find the word texture in varied contexts. For example, in music texture refers to the sound quality that is the result of relationships of melodies and harmonies of a piece of music ([SCHMIDT-JONES, 2012](#)). In poetry the concept of texture is related to the way that the words interact in a poem and the sensations that this interaction produce ([Poetry Foundation, 2007](#)). In image analysis and computer vision, texture is a set of complex visual patterns composed of basic elements that share some common characteristics like intensity, color, size.

The studies of texture perception were first made by [Julesz \(1962\)](#), who contributed to the theory of human visual perception of texture. Julesz suggested that two texture images will be perceived the same when their appropriate statistics were similar. In a later work, [Julesz \(1981\)](#) defined texture as a composition of basic elements, *textons*, that are repeated in an organized way, and tried to identify those *textons* and the way to group them together to form a texture. In 1973, the work of [HARALICK; SHANMUGAM; DINSTEIN](#) was the precursor of texture analysis. In this work it was developed a set of textural features with the goal to categorize the contents of images. The features were derived in the spatial domain using statistical measures from a co-occurrence matrix, a matrix that describes spatial relationship of pixels.

In nature, texture is present on the surface of things, for example, in feathers of a bird, in grains of sand on the beach, in cotton threads on a shirt, etc. Natural surfaces exhibit patterns that help to understand the relationship among the surfaces to the surrounding environment. Texture is an important property present in all surfaces, and can be perceived by visual and tactile senses. Visual texture is an attribute easily perceived by humans, with just a glance it is possible to get an idea about the type of texture of a surface, because it contains information about the spatial composition of the surface. In a different way, tactile texture can be felt by actually touching a surface. Every surface of anything exhibits specific texture characteristics that can be identified by touching them. And so, people have developed a rich lexicon to describe visual and tactile texture. For example, a tiled wall, a wrinkled t-shirt, a cracked surface, and the rough bark of a tree.

2.2 Definition

Texture, as a characteristic of a surface, can be perceived by tactile and visual means. Texture relates to the amount of smoothness or roughness of a surface. Most definitions on dictionaries correspond to tactile texture, for example, the online Oxford dictionary defines texture as: “The feel, appearance, or consistency of a surface or a substance.” (Oxford Press, 2018). In this way, texture is felt by actually touching a surface. On the other hand, texture, as a visual attribute, is how the human visual system perceives a surface, as if it were touched. In Computer Vision and Image Processing texture is studied as a visual attribute of a digital image. In this work, the term texture refers to visual texture, thus, the analysis is made in images whose content represents a kind of texture.

The scientific community does not provide a single definition of visual texture. Instead, it gives various consistent definitions of texture, sharing some common attributes. Materka, Strzelecki *et al.* (1998) define texture as a visual complex pattern composed of elements that have brightness, color, depth, and size. The patterns that share these characteristics correspond to texture. Similarly, Russ (2016) state that images that contain regions characterized by a variability or spatially repeated patterns are called texture.

Texture images in the real world present variations originated by capturing conditions, that is, rotation, illumination, scale, and viewpoint changes. In this way, the texture of a surface is dependent on the scale and the geometric conditions to capture the image. The texture captured as an image of a surface can be represented as (PETROU; SEVILLA, 2006):

- The variation of the albedo that characterizes the material of the surface. Albedo is the ratio of reflected to incident light, also known as the reflective power.
- The variation of the shape of the surface. For example if the surface is smooth, it has a regular surface, or if the surface is rough, it has an irregular surface.

To capture a texture it is necessary to take a snapshot of the surface of a material. In this process the contents of texture may appear different if multiple texture images are captured from the same surface under different capturing conditions. Nonetheless, the acquired texture provides useful information to understand the contents of an image, to distinguish the surface of objects, to get a notion of the degree of deformation of objects, to recognize similar regions, and to discriminate texture.

2.2.1 Types of Texture

Texture can be classified based on the regularity of the patterns present, another way it is based on the scale in which the patterns are observed, and whether or not they present a temporal component. Thus, texture can be categorized as regular/irregular, macro/micro textures, and static or dynamic textures.

Regular and Irregular. Texture as a visual pattern of repeated elements present some amount of variability in element appearance and relative positions. Regular textures present a consistent appearance, most of them are created by humans, and they are also known as artificial textures. Irregular textures present stochastic patterns, they are usually found in nature, and that is why they are called natural textures.

- **Regular textures:** The texture is composed of a regular repetition of an element over a surface. These sets of texture elements or patterns are called *textons*. It can be assumed that the texture was generated by transforming the *texton* appearance and geometry, and they are placed on a regular grid. In a real image the grid of *textons* are going to be transformed in some affine way. For near-regular textures, the grid is generally warped. In a regular texture the local statistical properties are the same.
- **Irregular textures:** The texture is difficult to represent in terms of *textons*, the details that generate the texture are unknown. The parts that compose a texture should have something in common in terms of texture features.

For example in [Figure 1](#), the window represents a local pattern of the building and it presents a regular texture. On the other hand, one predominant texture in this image is the grass, which being natural presents an irregular texture.

Figure 1 – An example of textures found at University São Paulo, Campus São Carlos. The building presents a regular texture, with windows as its basic pattern, instead the natural elements present irregular textures, as is the case with the grass.

Source: Elaborated by the author.

Macro and micro textures. Texture can be seen as macro-texture and micro-texture. Macro-textures are textures composed of large primitive patterns. Micro-textures are textures composed of very detailed patterns ([PETROU; SEVILLA, 2006](#)). Thus, the perception of texture can change according to the observation point, a set of objects may appear as a texture according to the

resolution employed, for example, a number of cities in a satellite image, a group of buildings in a city, a set of walls in a building, an arrangement of bricks in a wall, even the clay that compose the bricks, they all form a texture. Figure 2 shows an example of images presenting macro and micro texture. The grouping of strawberries makes them look like they have macro texture. Each individual strawberry presents another type of texture in itself, which can be considered micro texture.

Figure 2 – A set of strawberries show *macro* texture properties. Taking a closer view at individual strawberries, another kind of texture is revealed, a *micro* texture.

Source: Adapted from [Unsplash \(2018\)](#).

Static and Dynamic. Texture can be categorized as static and dynamic. Static textures represent patterns present in a single image, such as grass in a field, bark in a tree, leaves in a forest, etc. Instead, dynamic textures are patterns that present repetition in space and time ([GONÇALVES, 2013](#)). Dynamic textures add a temporal component to represent the texture. They are formed by sequences of texture images that represent moving objects and present certain stationary properties in time ([DORETTO et al., 2003](#)). The work of [Ribas \(2017\)](#) represents dynamic textures as cubes, in which the third dimension corresponds to the movement of the dynamic texture along time. Some examples of dynamic textures are the smoke caused by fire, the flow of water in a river, the movement of tree branches, and marine algae, see [Figure 3](#).

2.2.2 Texture and color information

The use of color in texture analysis can improve the tasks of texture classification and pattern recognition. Texture and color information has been used in applications like face recognition ([CHOI; RO; PLATANIOTIS, 2012](#)), material recognition ([SANDID; DOUIK, 2016](#)), and skin detection.

An image can be analyzed using its binary, intensity (gray scale), or color information. A binary image only needs two values and it is the simplest way to represent an image. In a color image, each pixel represents three components in the standard color space RGB (Red, Green and Blue). A color space is an organization of colors in an image. Human color perception is quite subjective as regarding perceptual similarity. To design formal color descriptors, it is necessary

Figure 3 – A representation of dynamic textures as cubes. Dynamic textures coming from the movement of tree branches and marine algae.

Source: Adapted from Ribas (2017).

to specify a color space, its partitioning, and how to measure similarity between colors. The colors in an image can be distinguished better in a certain color space.

Some texture descriptors extend its properties from gray level to color images. The color texture analysis can be divided into two categories: methods that process color and texture information separately, and those that consider color and texture together (MäENPää; PIETIKÄINEN, 2004). The approach by BACKES, CASANOVA and BRUNO (2012) uses the information of each channel individually and all channels together. Casanova *et al.* (2016) mapped each color channel into a surface and explored the influence of texture and color channels over a region of pixels. The change in spectral qualities of the environment affects the performance of color texture algorithms (KANDASWAMY; SCHUCKERS; ADJEROH, 2011).

2.3 Texture Analysis

Texture analysis is essential for applications that deal with texture, in which its goal is to characterize regions of images using their texture contents. Texture analysis focuses on extracting a set of distinctive features to characterize texture, in order to represent an image texture in a vector of measurable characteristics, which eases the subsequent processes of classification, segmentation or synthesis of textures.

Texture analysis is used in applications such as shape analysis, image segmentation, object recognition, image retrieval, texture synthesis, classification of biological and medical image datasets, and aerial and satellite surveys (BARLEY; TOWN, 2014). In the analysis of materials, texture plays an important role because many properties of materials are dependent on their texture (HUMPHREYS, 2001).

Throughout the years, texture analysis have been successfully used in a variety of applications. In order to capture texture properties, different texture descriptors have been

developed to describe texture regions. These descriptors are applied in the study of texture classification, texture segmentation, and texture synthesis.

Texture Classification. Texture classification seeks to identify features that characterize a particular texture. The purpose is to assign a class or label to an image based in these texture features. The approach is to get the features of an image dataset, compute a measure of similarity or distance between features, and to perform a classification (BARLEY; TOWN, 2014). A scheme of classification requires a training set to learn the classes, and a testing set to check the accuracy of the classifier. Two fundamental issues in texture classification are how to characterize textures using derived features and how to define a robust distance/similarity measure between textures, which remain elusive despite considerable efforts in the literature (THAKARE; PATIL; SONAWANE, 2013).

Texture Segmentation. Segmentation is the process of identify and divide an image into regions. Each region share some similar characteristic, such as color, shape, or texture. In segmentation using texture, each identified region corresponds to one kind of texture, thus, adjacent regions have different textures.

Texture Synthesis. The purpose of texture synthesis is to build an image based on features extracted from a sample texture. This new texture should appear as realistic as if it were generated by the same stochastic process (WEI; LEVOY, 2000). An application of texture synthesis is to rebuild damaged or lost parts in images and video, a process called *inpainting*.

2.4 Final Considerations

In the real world, texture is a characteristic present in every surface and it is both tactile and visual. A visual texture is an attribute of an image that describes its inner properties, and it is related with tactile texture, but they differ in the way how they are perceived. In the scientific community, texture has different definitions, but generally is defined as the presence of patterns in a region. In computer vision applications, texture is treated as a visual attribute present in images, thus, the analysis of texture provides useful information about regions in an image.

Texture can be categorized as regular or stochastic. A regular texture can be generated from basic elements and placed in a region following a pattern. Stochastic textures do not follow a defined path in their generation. Usually, regular patterns are the result of human intervention, stochastic patterns are naturally found in nature.

The next chapter deals with texture descriptors in the scene of texture classification. It describes texture descriptors as a process of transformation of an image, that aims to find important characteristics of texture in an image to produce as output a feature vector that represents that image.

TEXTURE CLASSIFICATION

Texture classification is an active area of research in computer vision. The goal in texture classification is to properly assign categories to a set of images based on their features. It requires two processes, which are feature extraction and classification. In feature extraction the aim is to obtain a robust and compact representation of a texture image in the form of a feature vector, and in classification the goal is to assign a category to each feature vector using a classifier.

Texture classification can be seen as a sequence of two processes, as shown in [Figure 4](#). The Feature Extraction process receives as input an image dataset, a set of texture images identified with its corresponding classes. It applies a texture descriptor to an image dataset, and it produces as a result a set of features that can be stored as a feature file, a set of feature vectors with the classes of the images. The Classification process uses a classifier to analyze the data contained in the feature file and classifies it according to its features. It receives a feature file that contains the feature vector and the category to which it belongs to, and gives as a result a set of measures that represent the classification results.

Figure 4 – The two main processes that comprise texture classification: feature extraction and classification. Its input is an image dataset and its output are the classification results.

Source: Elaborated by the author.

This chapter describes the main processes needed to perform texture classification, the

importance of texture descriptors, the texture descriptors and the classification methods used in this work. [Section 3.1](#) describes the feature extraction process. [Section 3.2](#) explains the importance of texture descriptors and describe some desired properties of texture descriptors. [Section 3.3](#) outlines the taxonomy of texture descriptors. [Section 3.4](#) describes the classification process and the classifiers used in this work. [Section 3.5](#) presents the validation process used in the work. Finally, [Section 3.6](#) discusses the ideas presented in this chapter.

3.1 Feature Extraction

The process of feature extraction is of fundamental importance in computer vision and image processing applications, and it has the greatest impact in the problem being solved. Feature Extraction is a process of transformation that aims to represent interesting characteristics of a whole image or parts of it to produce as output a compact representation in the form of a feature vector. Feature Extraction process can be seen as a dimensionality reduction technique. A sample texture image from Brodatz, a benchmark texture dataset, for example has 200×200 pixels, which means 40 thousands points of raw data. This amount of data is reduced using feature extraction, so that the classifiers work with less data.

The feature vector is useful for other image processing tasks such as pattern recognition, classification or segmentation. In texture analysis, a texture image is an image whose main content is texture, and Feature Extraction process refers to find important texture characteristics in a texture image. The goal of feature extraction in the context of classification is to obtain a set of features from an image that will be given to a classifier to categorize the image in a set of previously defined classes.

[Figure 5](#) depicts the feature extraction process in computer vision. The main component in the feature extraction process is the texture descriptor, which is the algorithm responsible for finding the texture characteristics of an image. For classification purposes, the input is an image and its output is a feature vector, thus feature extraction is a transformation from a pictorial to a numerical representation. The feature vector is a compact representation of a texture image.

Figure 5 – The feature extraction process uses a texture descriptor to transform the input image into a feature vector.

Source: Elaborated by the author.

3.2 Texture Descriptors

Texture descriptors are algorithms specifically designed to analyze and extract texture features. The features describe the main attributes or properties of a texture image for further processing in the classification stage (MARQUES, 2011). The characteristics produced should reflect intensity, roughness, uniformity, regularity, and repetitive patterns present in the texture. In the literature, many compact and discriminative texture descriptors have been proposed (SIMON; UMA, 2018).

A texture descriptor works on an image or regions inside it, and it produces texture features in the way of a feature vector to be used in the classification process. A texture descriptor usually operates in a local neighborhood, because texture is perceived as a visual pattern of elements with variability in appearance and positions, usually located in a local neighborhood.

It would be ideal to have texture features that produce high classification rates independently of the classifiers used. Thus, the features obtained by the texture descriptor influences the performance of the classification process.

Texture descriptors should be robust and compact. Robust means that the texture descriptor can extract features even though the input image undergoes some variations. Compact refers to the fact that the feature vector should be of low dimension (HU; LONG; ALREGIB, 2017).

3.2.1 Importance of Texture descriptors

The features extracted from an image are greatly influenced by the purpose of the application. A common requirement is that the texture descriptors should produce feature vectors invariant to rotation, scaling and translation. This means that the application will be able to recognize texture patterns at various angles, with different sizes, and in other positions (MARQUES, 2011). Rotation invariance denotes features that allow to identify textures rotated at different angles (PIETIKÄINEN *et al.*, 2011). Scale invariance means that the features extracted of images do not change if the distance between the images and the camera is altered (UHL; WIMMER, 2015). Translation invariance refers to features extracted that do not vary with the location of the region analyzed (NIXON, 2008).

The goal of a texture descriptor is to obtain a vector of characteristics that best discriminates classes, getting the most effective features for class separability. Texture descriptors try to get 'better' features in order to improve the accuracy of texture classification systems. The most convenient approach is to pay a lot of attention in choosing image features so that the classes are linearly separable.

The size of the feature vector varies according to the texture descriptor used. A texture descriptor produces a feature vector that depends on the application. The number of features influences the performance of the classifiers and it is especially important for real time applications

(SIMON; UMA, 2018).

The parameters of texture descriptors must be adjusted for each specific application, in order to find values that produce feature vectors with discriminative power. The researcher usually test a range of values for their parameters, and highlights those values that provide the best results for each dataset in a given experiment.

3.3 Taxonomy of Texture Descriptors

Many approaches have been proposed in the literature to analyze texture. An approach to categorize texture descriptors is based on how the texture features are computed. They are divided in four categories: statistical-based, structural-based, signal processing-based, and model-based features (TUCERYAN; JAIN, 1998; MIRMEHDI; XIE; SURI, 2008).

Statistical-based features. The spatial distribution among pixels determines the texture features of an image. The features computed vary according to the intensity and the number of pixels used. Some representative descriptors that use statistical features are Local Binary Pattern (LBP), and those derived from LBP such as Completed LBP (CLBP) and Adaptive Hybrid Pattern (AHP).

- The Local Binary Pattern (LBP) operator is a simple and efficient texture descriptor. The basic version of the LBP operator works in a 3×3 neighborhood (OJALA; PIETIKÄINEN; HARWOOD, 1996). In Figure 6 each pixel neighbor whose value is greater than the central pixel gets a 1, if it is less than or equal to the central pixel gets a 0. The binary values around the central pixel correspond to the new value resulting from the LBP operator.

Figure 6 – LBP operator applied on a 3×3 neighborhood.

Source: Elaborated by the author.

- The Completed LBP (CLBP) descriptor uses information from the central pixels, the sign of the difference and the magnitude of the difference. CLBP works on a

local region using a transform operation of the sign and magnitude (Local Difference Sign Magnitude Transform, LDSMT). The central pixel is binary coded after the thresholding operation as CLBP Central (CLBP_C). The LDSMT operation produces two sets of features CLBP Sign (CLBP_S) and CLBP Magnitude (CLBP_M). The sign component encodes better the local difference than the magnitude component. The feature vector combines CLBP_S, CLBP_M and CLBP_C, making CLBP a rotation invariant descriptor (GUO; ZHANG; ZHANG, 2010a).

- The Adaptive Hybrid Pattern (AHP) descriptor takes an hybrid approach using local patterns and global spatial structures. AHP is a noise robust texture descriptor that uses an adaptive quantization algorithm. AHP has been tested with images that contain independent zero mean noise, independent non zero mean noise and dependent noise (ZHU *et al.*, 2015).

Structural-based features. The structural features try to find basic elements that represent a texture pattern. The basic elements are called *texels* (texture elements) or *textons*, and are useful for creating artificial textures. In this way, texture is seen as a spatial order of basic texture elements. The basic elements or micro-textures are spatially arranged to form a larger pattern or macro-texture (BHARATI; LIU; MACGREGOR, 2004). The arrangement of the basic elements represent a proper description for regular patterns, but can be inefficient to describe complex patterns. In this work we are not using this kind of descriptors because they need to preprocess the texture dataset in order to find the basic elements.

Signal processing-based features. In the signal processing approach the features are obtained after applying a transformation to an image. The transformation can be a filtering or a filter bank decomposition over an image, and then computing the energy of the response, in the spatial, frequency or joint spatial-frequency domain. Some common approaches are based in Gabor and Wavelet transform.

- The Gabor descriptor uses filter banks for image classification and segmentation. The filter banks are obtained for specific scales and orientations. In practice, Gabor filters are the result of applying the Fourier transform and a Gaussian window over an image. Each filter is the result of a sinusoidal wave modulated by a Gaussian envelope. The frequencies considered in the Gaussian windows cover all the frequency bands of the image, thus, getting several versions of the original image. Gabor is widely used because it is invariant to rotation, scaling and translation (WANG; DING; LIU, 2005).
- The Dual-Tree Complex Wavelet Transform (DTCWT) descriptor. The Discrete Wavelet Transform (DWT) extracts information from a signal in the frequency domain at different scales by passing the signal through low pass and high pass filters. The DTCWT uses two different real DWTs to provide the real and imaginary parts

of the Complex Wavelet Transform. It uses wavelets to capture texture information because its decomposition provides information about the structure of the image, strong persistence across scales and compressible properties ([ANANTRASIRICHAIBURN; BULL, 2014](#)).

Model-based features. These models represent an image using stochastic or generative models, and they are capable to model natural textures, the parameters of the model are used as texture descriptors. Among these kind of descriptors we have the Bouligand Minkowski that is fractal based, and another in which an image is modeled as Complex Network.

- The Bouligand Minkowski descriptor uses fractal information to extract features from an image. A fractal is a geometric structure that presents self-similarity at any scale, and it is complex in nature because different details can be observed at any scale. The fractal dimension is the most important measure of a fractal, it represents how the structure changes according to the scale. A fractal descriptor is a set of fractal dimensions extracted from the entire image or by recursive decomposition of the image. The Bouligand Minkowski descriptor represents a gray scale image as a three dimensional surface, then this surface is dilated by spheres to get the fractal descriptor ([SILVA *et al.*, 2015](#)).
- Complex Networks. The approach by [Backes, Casanova and Bruno \(2013\)](#) treats a texture image as a complex network. This descriptor uses measures of complex networks to form the texture features. The complex network attempts to characterize the information among image pixels and their neighbors. The feature vector is a set of measures resulting of a degree histogram analysis of the complex network.

3.4 Classification

Texture classification is the process of categorizing images into separate classes based on textural properties. The choice of a classifier is very important in texture classification. We used different classifiers in the experiments, and we found that each classifier can produce different results over the same data.

In the platform it was favored the use of classifiers that provide good results in terms of performance, computational cost, and ease of use. The classifiers used in the platform are k Nearest Neighbors (kNN), Random Forest (RF), and Linear Discriminant Analysis (LDA). SVM (Support Vector Machines) was not employed because it requires parameter customization to achieve its best results.

3.4.1 *k* Nearest Neighbors

k-Nearest Neighbors (kNN), also known as instance based learner (IBk), is a simple and effective classification algorithm that uses a distance metric to classify data. Its purpose is to assign a label to an unknown instance data looking at similar data. The classification is performed when an instance of the testing set is compared against each instance of the training dataset. Then it takes the *k* most similar ones, which are the *k* nearest neighbors. It looks at their labels and according to a majority vote decides what is the label for this instance (MARSLAND, 2015).

3.4.2 Random Forest

In Classification, a Random Forest (RF) is an ensemble of Decision Trees. Ensembles combine several classifiers to create more powerful classifiers (MÜLLER; GUIDO, 2017). A decision tree makes a recursive partitioning of the data using a series of questions over the features to categorize the data, in the process builds a decision boundary which separate the instances into their respective classes. A Random Forest consists of various Decision Trees with the purpose of reducing the overfitting that happens in Decision Trees, each Decision Tree is built different from the others (BREIMAN, 2001).

3.4.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) seeks to find a linear transformation of the data that can help to discriminate the classes. After the transformation, the classification is made using some metric such as Euclidean distance or cosine measure (LI; ZHU; OGIHARA, 2006). LDA finds a projection of the features, a feature space, with the goal to maximize the between-class measures and minimize the within-class measures in the process of classification. This version of LDA does not use PCA (Principal Component Analysis) to reduce the number of features.

3.5 Validation scheme

The platform uses cross-validation as a means to evaluate the performance of the classifiers. In a single split of data, the data is split in training set and test set, first the classifier works in the training set and then it evaluates the test set. Instead, in a **cross-validation** scheme the data is split into *k* folds, each time the classifier uses *k* – 1 folds as training set and 1 fold as testing set. Hence, all the data is part of the testing set exactly once.

A better approach to split the data is by using a **stratified cross-validation** scheme, in which each fold contains approximately the same number of samples per class in each fold (MÜLLER; GUIDO, 2017). A common value for the number of folds is 10, this of course means that a dataset has at least 10 samples per class. For example, Figure 7 shows how a dataset

with 3 classes and 50 elements per class is split in 3 folds using cross-validation and stratified cross-validation, the gray bars corresponds to the test set. In cross-validation the first split will have classes 1 and 2 as training set and class 0 as test set. Instead, in stratified cross-validation the test set contains elements from each class, thus, stratified cross-validation provides a better division of data for the classifier.

Figure 7 – Comparison between cross-validation and stratified cross-validation. The label for each class is ordered.

Source: Müller and Guido (2017).

3.6 Final Considerations

Defining a generally good texture descriptor is inherently difficult, due to the vague definitions of texture. However there are a few criteria that should be met by a texture descriptor. In essence, a texture descriptor should be a method that produces a feature vector that (1) is invariant to rotation, scale, intensity and translation, (2) is robust to noise, (3) has a small number of features, and (4) is computationally efficient.

Some texture descriptors use spatial information from local neighborhoods, such as the LBP descriptor and their many improvements. LBP has the ability to encode fine details. Other texture descriptors work in the frequency domain to capture additional details of texture, such as those based on the Fourier transform. They are able to deal with noisy images.

Texture features should be computed at an appropriate scale according to the image being analyzed. In real time applications, texture descriptors should be robust in the face of changes in the capturing process and efficient in terms of computational complexity.

The representation of an image texture as a set of features has important challenges because of the random nature of patterns present in nature.

IMAGE DATASETS

A texture dataset is a collection of images whose main content is texture information. There are benchmark and domain specific datasets. Benchmark datasets are those specifically created to test the performance of texture descriptors. Most benchmark datasets are balanced, that is, they present an equal number of samples per class. Domain specific datasets contain images that correspond to an area of application, for example material textures, microscopic leaves images, or those resulting from medical scans. Normally, these datasets are imbalanced, that is, the number of samples per class is not equal, and the size of each image is larger when compared to images of benchmark datasets.

It is important that researchers spend time analyzing their image datasets. In a real application it is important to know the image dataset and which properties may be relevant to classify them. In most real world applications of texture analysis it is necessary to take into consideration non-ideal conditions outside the controlled, ideal laboratory environment ([KANDASWAMY; SCHUCKERS; ADJEROH, 2011](#)). The work made in ([HOSSAIN; SERIKAWA, 2013](#)) categorizes the texture datasets into four areas, according to the majority of images presented in each dataset: medical images, natural textures, dataset of materials, and dynamic textures. There are also image datasets in which each class contains images related by some visual attribute of texture ([CIMPOI *et al.*, 2014](#)).

[Section 4.1](#) describes the texture datasets used in the platform, and [Section 4.2](#) summarizes this chapter.

4.1 Texture Datasets in the platform

A texture dataset can be organized, according to the application, into two ways: as a single directory or a root directory with subdirectories. As a single directory, all the images of the dataset are in a directory that is named after the dataset. The filename of each image

helps to identify the classes and the samples of each class. The filename follows the pattern *class_sample.extension*, in Brodatz, a benchmark texture dataset, for example, the image with filename *D31_08.bmp* represents class *D31*, sample 08, and extension *bmp*. As a root directory with subdirectories, the root directory identifies the dataset, the name of each subdirectory identify the classes, and the images inside a subdirectory are the corresponding samples.

Table 1 presents a summary of the characteristics of the texture datasets used to evaluate the texture descriptors in the platform. The number of images goes from 606 in Midrib to 7616 in STex. The range of classes varies from 24 in Outex-10 to 476 in STex. Each class in every dataset has at least 10 samples, this value is important in the classification process to apply a cross-validation scheme with 10-fold. The Midrib dataset is imbalanced, the number of samples is variable from 10 to 20 images per class. In the other datasets, the number of samples is equal to the number of images divided by the number of classes.

Table 1 – Image datasets used in this work.

Name	Images	Classes	Samples/class	Type	Size
1200Tex	1200	20	60	color	128×128
Brodatz	1110	111	10	gray scale	200×200
Curet	5612	61	92	color	200×200
Kylberg	4480	28	160	gray scale	576×576
KTH-TIPS2b	4752	11	432	color	200×200
Midrib	606	50	10 ~ 20	color	variable
Outex-10	4320	24	180	gray scale	128×128
STex	7616	476	16	color	128×128
UIUC	1000	25	40	gray scale	640×480
USPTex	2292	191	12	color	128×128
VisTex	3904	61	64	gray scale	64×64

Source: Elaborated by the author.

4.1.1 1200Tex

The 1200Tex, created at University of São Paulo, is a dataset of plant species. The texture images correspond to 20 species of Brazilian flora. The process to collect the samples consisted of scanning a leaf and manually extracting windows. Because of the manual scanning the images can contain noise, or shadows, thus there is variability in texture of the same species (DALCIMAR; JOACI; MARTINEZ, 2009).

The 1200Tex dataset is divided into 20 classes, with 60 samples per class, making a total of 1200 color images. The images are *png* files of size 128×128 pixels (Scientific Computing Group, 2012a). A sample of 1200Tex textures is presented in Figure 8.

Figure 8 – Sample texture images from the 1200Tex dataset.

Source: Elaborated by the author.

4.1.2 Brodatz

The Brodatz texture dataset is a widely used benchmark dataset used to assess texture descriptors. It contains gray scale images of natural textures that were photographed under controlled lighting conditions. The Brodatz album contains 112 images collected for artists and designers ([BRODATZ, 1966](#)). The Brodatz texture dataset was created from the photographs of textures in the Brodatz album ([RANDEN, 2007](#)).

The Brodatz texture dataset is composed of 111 classes with 10 samples per class, making a total of 1110 images. The images are *bmp* files of size 200×200 pixels. Many publications only use a subset of this dataset ([HOSSAIN; SERIKAWA, 2013](#)). A sample of Brodatz textures is presented in [Figure 9](#), the images correspond to the whole texture before being split in samples, and the text under each texture represents the class and the name of the texture according to the work of [BRODATZ \(1966\)](#).

4.1.3 CURET

The Columbia-Utrecht Reflectance and Texture (CURET) dataset is another famous benchmark. It was created by the collaboration of the Columbia University and the Utrecht University. The CURET dataset contains textures of real world surfaces captured under changing illumination and viewing directions, presenting geometric and photometric properties, such as specular surface (aluminum foil), diffuse surface (concrete), isotropic surface (cork), anisotropic surface (corn husk), surface with large height variation (crumpled paper), natural surfaces (lettuce, moss) ([DANA et al., 1999](#)).

CURET is divided in 61 classes, with 92 samples per class, making a total of 5612 images. The images are *png* files of size 200×200 pixels ([Computer Vision Laboratory, 2016](#)). A sample of CURET textures is presented in [Figure 10](#).

Figure 9 – Sample texture images from the Brodatz dataset.

Source: Elaborated by the author.

Figure 10 – Sample texture images from the CURET dataset.

Source: Elaborated by the author.

4.1.4 *Kylberg*

The Kylberg texture dataset was created at the Uppsala University, Sweden. The Kylberg dataset includes images of fabrics, grains, and natural surfaces. Each image was captured under the same conditions of light, direction, and distance. Each image is normalized to reduce effects of exposure settings and have the same mean gray value, $\mu = 127$, and standard deviation, $\sigma = 40$ (KYLBERG, 2011).

Kylberg is divided in 28 classes, with 160 samples per class, making a total of 4480 images in gray scale. The images are *png* files of size 576×576 pixels. A sample of Kylberg textures is presented in Figure 11.

Figure 11 – Sample images from the Kylberg texture dataset.

Source: Elaborated by the author.

4.1.5 KTH-TIPS2b

The KTH-TIPS2b dataset was created at the KTH University with the purpose of getting textures with variations in scale, pose and illumination. The KTH-TIPS2b dataset includes images of 11 texture materials. The KTH-TIPS2b is a balanced dataset, instead the version KTH-TIPS2a is an imbalanced dataset (CAPUTO; HAYMAN; MALLIKARJUNA, 2005).

KTH-TIPS2b is divided in 11 classes, with 432 samples per class, making a total of 4752 color images. The images are *png* files of size 200×200 pixels. A sample of KTH-TIPS2b textures is presented in Figure 12.

Figure 12 – Sample images from the KTH-TIPS2b texture dataset.

Source: Elaborated by the author.

4.1.6 Midrib

The Midrib dataset was created at University of São Paulo with the purpose to identify plant species using microscopic images of the cross-sections of leaf midrib. The images were collected at the Cerrado biome in central Brazil, at IBGE (Brazilian Institute of Geography and Statistics) Ecological Reserve. The images contain only the region of the midrib, the background was manually removed (SILVA *et al.*, 2015).

The Midrib is divided in 50 classes (plant species), with a variable number of samples per class, ranging from 10 to 20, in total there are 606 images. One difference with respect to other datasets is the size of the images, ranging from 2 MB to 130MB, with a resolution from 500×500 pixels to 9500×8200 pixels, the images are *png* files (Scientific Computing Group, 2012b). A sample of Midrib textures is presented in Figure 13.

Figure 13 – Sample texture images from the Midrib dataset.

Source: Elaborated by the author.

4.1.7 Outex

Outex is a framework for texture analysis designed to test texture classification and segmentation algorithms, developed at Oulu University. It has natural scenes and surface textures captured under variations in terms of illumination direction, surface rotation, and spatial resolution (HOSSAIN; SERIKAWA, 2013). There are 16 test suites used for classification. The suite Outex-10 dataset is frequently used to test texture descriptor algorithms because contains images captured at nine angles (0, 5, 10, 15, 30, 45, 60, 75, 90 degrees) and with constant illumination. (TIMO *et al.*, 2002).

Outex-10 is divided in 24 texture classes, with 180 samples per class, making a total of 4320 images. The images are *ras* files of size 128×128 pixels (Machine Vision and Media

Processing, 2007). A sample of Outex-10 textures is presented in Figure 14.

Figure 14 – Sample texture images from the Outex dataset.

Source: Elaborated by the author.

4.1.8 STex

The Salzburg Texture Image Database, STex, was created at the Salzburg University. The images were captured around the Salzburg city. It contains natural and material images, they are fairly homogeneous.

STex dataset is divided in 476 images, each image corresponds to a class, with 16 samples per class, making a total of 7616 color images. The images are raw Netpbm format *pnm* of size 128×128 pixels (Multimedia Signal Processing and Security Lab, 2008). A sample of STex textures is presented in Figure 11, each image represents the whole image before being split in samples.

Figure 15 – Sample texture images from the STex dataset.

Source: Elaborated by the author.

4.1.9 UIUC

The UIUC dataset was created at University of Illinois at Urbana-Champaign. UIUC dataset contains texture surfaces coming from albedo variations, 3D shapes, and a combination of both, captured with uncontrolled lighting conditions. Textures in UIUC dataset appear with changes in rotation, scale and viewpoint, also presents a high intra-class variation (LAZEBNIK; SCHMID; PONCE, 2005).

UIUC dataset is divided in 25 classes, with 40 samples per class, making a total of 1000 gray scale images. The images are *jpg* of size 640×480 pixels (The Ponce Group, 2006). A sample of UIUC textures is presented in Figure 16.

Figure 16 – Sample texture images from the UIUC dataset.

Source: Elaborated by the author.

4.1.10 USPTex

The USPTex dataset was created at the University of São Paulo and is maintained by the Scientific Computing Group. USPTex dataset contains a variety of color images of textures found on a daily basis, such as rice, vegetation, clouds, etc. They were captured with a digital camera and divided in non overlapping windows.

USPTex dataset is divided in 191 classes, with 12 samples per class, making a total of 2292 texture color images (BACKES; CASANOVA; BRUNO, 2012). The images are *png* files of size 128×128 pixels (Scientific Computing Group, 2012c). A sample of USPTex textures is presented in Figure 17.

4.1.11 VisTex

The Vision Texture dataset, VisTex was created by the Vision and Modeling MIT Media Lab. The VisTex dataset contains textures captured under real world conditions. In the case of

Figure 17 – Sample texture images from the USPTex dataset.

Source: Elaborated by the author.

lighting conditions the images were exposed to daylight, artificial-florescent or artificial incandescent light, so some noise, shadows, or depths appear in the textures (HOSSAIN; SERIKAWA, 2013).

VisTex is divided in 61 texture classes, with 64 samples per class, making a total of 3904 images. The images are *pgm* files of size 64×64 pixels (Vision and Modeling Group, 1995). A sample of VisTex textures is presented in Figure 18.

Figure 18 – Sample texture images from the VisTex dataset.

Source: Elaborated by the author.

4.2 Final considerations

Texture datasets are developed to test different properties of texture descriptors, such as noise robustness or rotation invariance. In benchmark datasets usually synthetic noise is added to

test robustness against noise. In texture datasets that are used to assess rotation invariance the images are rotated at certain discrete angles, for example the Outex-10 dataset present images rotated at 9 angles (0, 5, 10, 15, 30, 45, 60, 75, 90 degrees).

The process of acquisition of texture images affects the final image texture. The creators of the texture datasets need to describe the process of acquisition of the images, thus, it is important to spend time analyzing the inner features of a texture dataset in order to choose an appropriate texture descriptor.

Most of benchmark datasets consists of an equal number of samples per class, they are balanced datasets. In contrast, in real datasets is common to have imbalanced datasets, that is, the number of images is not equal among classes, the Midrib dataset has a variable number of samples per class. Another important issue concerns to the size of the images, in real datasets the image size is usually larger than those images found in benchmark datasets.

In texture classification, the number of samples in a class is important in order to use a cross-validation scheme. For example, the Brodatz dataset has 10 samples per class, so it is possible to make 10-fold cross-validation, in this case the train set contains 9 samples and the test set contains 1 sample. However, if a dataset contains less than 10 samples, let's say 9 samples, it can not be possible to make a 10-fold cross-validation, because the train set will have 9 samples, and the test set will be left with nothing.

The next chapter explain the platform developed in this master's, its features, functionalities, and technical details about its construction.

TEXTURE EVALUATION PLATFORM

This chapter describes the research and collaboration application Texture Evaluation Platform (TEP) to study and analyze the performance of texture descriptors and image datasets. In the scientific community it is important to study, analyze and share data among researchers. The scientific community benefits from knowledge sharing. A web application has some features that differentiates it from traditional applications. It runs on any browser on multiple platforms independent of the underlying operating system, all the users interact with the same version of the application, and does not require a process of installation. The present work is a web-based application in the areas of computer vision and image processing that aims to ease the tasks of texture analysis.

This platform was built with the purpose of analyzing and comparing the performance of texture descriptors. The platform provides texture descriptors, texture datasets, and classifiers to perform experiments of texture classification. From now on, we use the word “user” as synonym of “researcher”. The interaction of the user with the platform provides ways for the user to:

- make experiments of texture classification,
- test new texture descriptors by uploading source code to the platform,
- classify new texture datasets by uploading them to the platform,
- review classification results by searching the data stored,
- analyze metrics of classification, and
- interact with the confusion matrix

[Section 5.1](#) presents the requirements that texture descriptors should meet in order to be employed in the platform. [Section 5.2](#) describes the design of the platform, explaining the architecture design and the database design. [Section 5.3](#) presents the functionalities of the

platform showing their interfaces for the interaction with the user. [Section 5.4](#) describes the technical details of construction and hardware of the platform. Finally [Section 5.5](#) summarizes main points of this chapter.

5.1 Requirements for Texture Descriptor

The texture descriptors used in the platform meet the following requirements, they refer to the number and type of parameters, the type and size of the images, the output of the descriptors, and pre and post processing steps.

1. **Number and type of parameters.** The texture descriptor has at least two parameters. The first parameter corresponds to an input image, the second parameter is specific to the algorithm, if there are others, they are also specific to the algorithm.
2. **Type and size of images.** The texture descriptor works with gray scale and color images, when necessary the input image is converted to gray scale. Ideally, there is no restriction in the size of the input images, the descriptor should work with images of different sizes. The descriptor handles the following image formats *bmp*, *ras*, *pgm*, *pnm*, *png*, *jpg*, *tif*.
3. **Output of texture descriptors.** The output is a set of numbers representing the feature vector of the input image. The feature vector must contain all valid numbers for each attribute, that is NaN (not a number) and Inf (infinite) are not allowed. This feature vector is used to build a file of features that will be used in the step of classification.
4. **Pre processing** The texture descriptor processes an image at a time. This means that there is no pre-processing step in which the descriptor works with a subset of the images to extract some parameter values that will influence the posterior feature extraction process.
5. **Post processing.** There are no steps of post processing, like those aiming to reduce the number of features over the whole set of features. All features are used in the classification stage, because we want to test how the length of the feature vector influences the classification results.

5.2 Application Design

The design of the application consists of three main components: web interface, core application and data storage. This design contains the main components of the platform and constitute the design principles for the architecture and database of the application.

The web interface consists of a set of interfaces in which the user interacts with the platform and presents the results and metrics of classification. The core application performs the feature extraction and classification in sequence. The feature extraction module receives as

input an image dataset, a set of images identified with its corresponding classes, and returns as output a feature file, a set of feature vectors with the classes of the images. The Classification module classifies the data contained in the feature file, it uses a training and testing set in a cross-validation scheme, and it returns the classification results. The data storage is dedicated to save the data originated by the user's interaction with the application. The following sections describe the architecture and database design of the platform.

5.2.1 Architecture Design

The software architecture helps to define the basic characteristics of an application, as well as the interactions between components of the application (RICHARDS, 2015). The Texture Evaluation Platform (TEP) is based on a three layers architecture. In a layered architecture, each layer performs a specific function. The layers of TEP are the Web Interface, the Core Application, and the Data Storage.

Figure 19 depicts the design of the application TEP with its three layers, it shows the relationships among the components and the flow of data between them. The flow of data starts in the Input Selection module in which the user chooses values to perform an experiment of texture classification. This data is processed by the Core Application that performs the feature extraction and the classification. Finally, the data generated by the classification results are stored in the database and presented in the web.

The user interacts with the Web Interface creating experiments by choosing the texture algorithms, image datasets, classifiers, and validation schemes. The Web Interface also presents the classification results to the user. The Core Application performs the process of texture classification, running in sequence the texture extraction and the classification. The Data Storage stores the data generated by the application, this includes the data of the experiments, feature files, and the classification results.

Each layer in the application, the Web Interface, the Core Application and the Data Storage is described in what follows.

5.2.1.1 Web Interface

As a web based application, the Web Interface provides a familiar environment to use. The Web Interface is the entry point of the platform. The user interacts with the application through the Web Interface, making experiments and getting results. The setup of an experiment contains the values of texture descriptors, image datasets, classifiers, and validation. This setup is sent to the Core Application that processes it. After the processing, the user gets the results of the classification.

The Web Interface is composed of the **Input selection** and the **Output results** components. The **Input selection** provides the entry point for the setup of experiments, the testing of

Figure 19 – Design of the application showing the three layers of the architecture: Web Interface, Core Application, and Data Storage. The arrows signal the flow of data among the components of the application.

Source: Elaborated by the author.

new texture descriptor and new image datasets. The **Output results** component provides a place to present the results.

The **Input selection** component serves as an interface with the following purposes:

- **Customize experiments.** The researcher enters the data to perform experiments of texture classification. In the interface the user selects a texture descriptor and enters its parameter values. The user also selects the image datasets, the classifiers, and the validation scheme. The data is sent to the Core Application and as a response it presents a message to the Web Interface indicating the status of the experiments.
- **Test new texture descriptors.** The researcher can upload source code of texture descriptors. The researcher needs to provide the parameters of the algorithm. The application registers the new texture descriptor to make experiments.
- **Test new image datasets.** The researcher can upload images and test them with existing texture descriptors. The images should be packed in a *zip* format file. The application only considers files that are images, any other file types are not considered. The application presents this new image dataset in the interface to make experiments.

The **Output results** shows the metrics and results of classification to the user. The results are presented as a table detailing the values used in the experiment: texture descriptor and its parameters, image dataset, classifier, validation scheme, accuracy and standard deviation of classification, and a link for the detailed metrics.

5.2.1.2 Core Application

The Core application is responsible for performing the texture classification based on the setup of an experiment made by the user. It consists of the **Feature extraction** and the **Classification** modules.

The **Feature extraction** module performs the texture analysis, therefore, it applies each one of the texture algorithms selected to the image datasets selected. The result of each processing is a feature file that contains the feature vectors of every image of a dataset. The application runs in an independent process the feature extraction, and the intermediate results are stored in the database application. For example, if the user selects 3 texture algorithms and 2 image datasets, the application starts 6 processes to perform the feature extraction, each process creates a feature file. These 6 feature files will be used by the **Classification** module.

The **Classification** performs the texture classification of the feature files applying the classifiers and the validation scheme selected by the user. The application runs in an independent process the classification, the results of the classification are stored in the database application.

5.2.1.3 Data Storage

The goal of the Data Storage is to store and retrieve data of the application, and to have access to historical data. It stores internal data, data generated by the user, and by the application itself, it also keeps record of the image datasets used, and the feature files created. The internal data consists of data related to the configuration of the application, and default values for the parameters of the texture algorithms, paths to image datasets, names of classifiers, and validation schemes.

The user generates data every time that makes an experiment, including new values for the texture algorithms with its parameters, image datasets, classifiers, and validation schemes. The data that the application generates includes the metrics and accuracy of the classification results, timestamps of beginning and ending of processes, and the time lapse of each process.

5.2.2 Database Design

The database design determines how the data is organized and how they are related. The use of a database helps to integrate the data, reduce the redundancy of data, and allows complex operations to be performed in the data (TEOREY *et al.*, 2008). The design of the database reflects the main processes of the application, feature extraction and classification. Figure 20 shows the

entity relation model of the application. The names of the entities help to identify the data they store. Each entity has a unique identifier, its primary key. The lines that join the entities represent the relationship between entities.

Figure 20 – The database design of the application. The feature extraction process is represented by the entities **FeatureExtraction**, **TextureAlgorithm**, **Parameter**, and **ImageDataset**. The classification process is represented by the entities **Classification**, **Classifier**, **Validation**, and **Metrics**.

Source: Elaborated by the author.

To make an experiment of texture classification, we need first to extract features and then to classify these features. In the database design the main entities that compose an experiment are **FeatureExtraction** and **Classification**.

The entities that correspond to the feature extraction process are **FeatureExtraction**, **TextureAlgorithm**, **Parameter**, and **ImageDataset**. The attribute of **FeatureExtraction** is *feature_file*, which represents the filename of the feature file created as a result of the feature extraction process. The attributes of **TextureAlgorithm** are *name* and *number_parameters*, *name* is the filename of the texture descriptor algorithm and *number_parameters* is the number of parameters of the algorithm. The **Parameter** entity has *name*, *value*, and *order* attributes that help to identify each parameter of the algorithm. The **ImageDataset** entity has *name* and *path* attributes, *path* contains the full path of the dataset identified with the *name* attribute.

In terms of relationships, one **FeatureExtraction** consists of one **TextureAlgorithm** and one **ImageDataset**, and in turn one **TextureAlgorithm** has one or many **Parameters**. The

result of the feature extraction process is stored in *feature_file* in the **FeatureExtraction** entity.

The entities that correspond to the classification process are **Classification**, **Classifier**, **Validation**, and **Metrics**. The **Classification** entity has *accuracy* and *std* attributes, *accuracy* stores the mean accuracy, and *std* is the standard deviation of the classifications for one experiment. The **Classifier** entity has the *name* attribute, that helps to identify the classifier shown in the interface. The **Validation** entity has *name* and *value* attributes, *value* stores the number of folds or a special value for Leave One Out in a cross validation scheme. The **Metrics** entity contains details of the closest value to the classification result stored in the attribute *accuracy* of **Classification**. The attributes *kappa*, *mean_absolute_error*, *number_correct*, *number_incorrect* contain classification metrics. The attribute *confusion_matrix* contains data to recreate the confusion matrix and to identify each image correctly classified or misclassified. The attributes *true_positive*, *false_negative* and *false_positive* contain data to compute the precision, recall, and f_measure for each class.

In terms of relationships, one **Classification** consists of one **FeatureExtraction**, one **Classifier**, and one **Validation**. It is important to notice that one *feature_file* is used to make one or more classifications, in this way, one **FeatureExtraction** has one or many **Classifications**. The result of an experiment of classification is stored in *accuracy* and *std* attributes in the **Classification** entity. In addition, one **Classification** result is stored in one **Metrics**.

5.3 User Interfaces

As a web based application, the platform provides a familiar environment to use and share information between people all around the world using a standard browser. The user interacts with the application to make experiments of texture classification, to test new texture descriptors, to classify new texture image datasets, to review classification results, to analyze metrics of classification, and to interact with the confusion matrix. The following subsections details these interfaces.

5.3.1 Make experiments of Texture Classification

The platform facilitated the realization of the experiments of texture classification. An experiment of texture classification consists of executing two processes in sequence: feature extraction and classification. The feature extraction process is executed once using a texture descriptor applied over a texture dataset. The classification process consists of applying one classifier over a set of texture features. To evaluate the performance of the classifier the platform uses a stratified 10-fold cross validation scheme. The classification process is executed 20 times in order to obtain the mean value and the standard deviation of the accuracy.

To setup an experiment for texture classification the user chooses the texture descriptors, the texture datasets, the classifiers, and the validation schemes. The interface presents 4 input

boxes to select multiples values. When choosing the texture algorithms the user can enter ranges of values (start, step and stop) for its parameters. The start and stop values correspond to a half open interval $[start, stop)$, thus, using *step* the values represented are $start, start + 1 * step, \dots, start + n * step < stop$. It was chosen a half open interval to deal with stop values of real numbers.

Figure 21 shows an example of using the experiment interface. For the feature extraction process, the user selected 1 image dataset (Outex-10), and 1 texture algorithm (CLBP.m) and customized its parameters, the parameter **radius** is set to 2, the parameter **points** has $start=8$, $step=4$, and $stop=17$, so the parameter **points** has three values $\{8, 12, 16\}$. With this setting the user has at once 3 different values for CLBP, that is, (2, 8), (2, 12), and (2, 16). For the classification process, the user selected 3 classifiers (kNN, RandomForest, and LDA), and 1 validation scheme (10-fold). Thus, at the end of the experiments the user will get 9 classification results, which comes from 3 texture descriptors, 1 dataset, 3 classifiers and 1 validation scheme, that is $3 \times 1 \times 3 \times 1 = 9$.

Figure 21 – The interface to make experiments presents to the user 4 drop down lists to choose values for Texture Algorithm, Image Dataset, Classifier, and Validation. The values of the parameters only appear when the user selects an algorithm.

Texture Evaluation Platform Experiment Algorithms Datasets Search Task Progress

Making an experiment

Texture Algorithm: x CLBP.m

Image Dataset: x Outex-10

Statistical Classifier: x kNN, x RandomForest, x LDA

Cross Validation: x 10-fold

CLBP.m

radius: 2

points: 8, 4, 17

Create experiments

Source: Elaborated by the author.

5.3.2 Test new Texture Descriptors

The platform TEP allows the uploading of source code to test new texture descriptors. The source code can be in the following programming languages: Matlab, Python or Java. Figure 22 show the interfaces to upload an algorithm in the platform. In this interface the user chooses the file to upload, in this case a compressed file *CLBP.zip*. After the uploading, the application shows another interface to define the parameters of the algorithm, see Figure 23. In

this interface the application shows a message indicating that the file *CLBP.zip* has successfully been uploaded. The user needs to choose the main file which will be the entry point for the texture algorithm. Each time the user clicks the button *Add parameters* the interface presents two input boxes to enter the name and the value of a parameter. To register the algorithm in the platform the user clicks in *Register algorithm!*. After completing this step the algorithm is ready to be used in the application.

Figure 22 – The first interface to upload source code of a texture algorithm. In the image the user provides the file *CLBP.zip*.

Source: Elaborated by the author.

Figure 23 – The second interface to upload source code of a texture algorithm. In the image the user chooses the main file of the algorithm, and adds parameters specifying their names and values.

Source: Elaborated by the author.

5.3.3 Classify new Texture Datasets

Lots of researchers working in a specific field has their own image datasets, that can be of plant textures, medical images, or even from nanostructure materials. The platform TEP has the option to receive new image datasets. This allows the researcher to classify their image datasets

with ready to use texture descriptor. Figure 24 shows the interface to upload a texture dataset. The name of the dataset is unique, and there are two types of image format: class_sample.ext and dataset/classes. The allowed file type is a compressed **zip** file. When the application finishes uploading the dataset it appears a message informing to the user that the dataset is correctly uploaded, and then the new dataset is ready to be used.

Figure 24 – The interface to upload texture datasets. In the image the user writes a unique name for the dataset, and specify the image format of the images. After the uploading the dataset is ready to be used in the application.

The screenshot shows the 'Texture Evaluation Platform' interface. At the top, there is a navigation bar with links: 'Texture Evaluation Platform', 'Experiment', 'Algorithms', 'Datasets', and 'Search'. Below this, the section is titled 'Upload your Image Dataset'. It contains three input fields: 'Name your dataset' with the value 'Outex_13', 'Image format' with the value 'class/sample.bmp', and 'Image files' with a file selection button labeled 'Examinar...' and a file named 'Outex_13.zip'. At the bottom of this section is a blue button labeled 'Upload your dataset!'.

Source: Elaborated by the author.

5.3.4 Review Classification Results

To review and examine the data previously generated by the platform, the user enters to the Search interface. Figure 25 shows the interface to search classification results. The interface is similar to the interface to make an experiment. There are four select boxes to choose among the texture descriptors, texture datasets, classifier, and the value of validation. The classification results appear at the bottom, the data presented is: texture descriptor and its parameters, texture dataset, classifier, validation, accuracy and standard deviation of classification, and a link to the details of metrics.

It is possible to leave the four select boxes in blank, in this case the application will present the best 200 classification results in descending order. It is better to choose either a texture descriptor or an image dataset to get more specific results. The application presents the results of classification as a table in descending order according to the classification accuracy. Each row presents the setup of an experiment, and consists of the following values: texture descriptor and its parameters, texture dataset, classifier, validation scheme, accuracy and standard deviation, and a link for the metrics. For example, Figure 26 presents the classification results after the user chooses CLBP, USPTex, RandomForest, LDA, and 10-fold in the select boxes. In this case, the results shows that using the CLBP algorithm varying the radius between 1 and 2,

Figure 25 – The search interface helps to find previous classification results. The user has the option to leave the four boxes in blank, in which case the classification results will be shown in descending order, or can choose some of the four select boxes to view more specific results.

Source: Elaborated by the author.

and points between 6 and 8, the accuracy achieved is over 90% using the RandomForest and LDA classifiers.

5.3.5 Analyze metrics of classification

The application presents metrics to evaluate the classification results in the Metrics interface. The values correspond to the closest accuracy to the mean accuracy of an experiment of 20 executions. The Metrics interface is accessed through the Search interface and presents data from the setup of an experiment, data of feature extraction and results of classification. The data from an experiment are the descriptor, dataset and classifier. The descriptor refers to the texture algorithm and its parameters. The dataset refers to the name, number of images and number of classes. And the classifier refers to the name and value of validation. The data of feature extraction is the number of attributes. The data of classification results are the confusion matrix, accuracy, mean absolute error, kappa, number of correct classifications, number of incorrect classifications, precision, recall, f-measure, TP (true positive), FN (false negative), and names of classes. The sum of TP and FN gives the number of samples in a class.

Figure 27 shows the metrics of classification results after applying the LPQ algorithm over the UIUC dataset with the Random Forest classifier. The UIUC has 1000 images divided in 25 classes, with 40 samples per class. The accuracy in this case is 72.2 that correspond to the division of number of correct instances over the number of images in the dataset. At the bottom of the image next to the last line, analyzing the class T10_floor1, it is seen that the number of FN is greater than the TP, and so the precision, recall and f-measure are less than 0.5 for this class.

Figure 26 – Using the search interface to get classification results using the CLBP algorithm over the USPTex dataset with the RandomForest and LDA classifiers with a 10-fold cross validation scheme. The results are presented in descending order according to their accuracy.

Search Classification Results

Texture Algorithm

Image Dataset

Statistical Classifier

Cross Validation

Show entries

Filter:

Algorithm	Dataset	Classifier	k-fold	Accuracy	Metrics
CLBP(radius:1, points:6)	USPTex	LDA	10-fold	93.67 (0.19)	USPTex
CLBP(radius:2, points:8)	USPTex	LDA	10-fold	92.90 (0.27)	USPTex
CLBP(radius:2, points:6)	USPTex	LDA	10-fold	92.54 (0.16)	USPTex
CLBP(radius:1, points:8)	USPTex	LDA	10-fold	92.35 (0.24)	USPTex
CLBP(radius:1, points:6)	USPTex	RandomForest	10-fold	91.77 (0.35)	USPTex
CLBP(radius:2, points:6)	USPTex	RandomForest	10-fold	90.58 (0.35)	USPTex
CLBP(radius:2, points:8)	USPTex	RandomForest	10-fold	90.12 (0.35)	USPTex
CLBP(radius:1, points:8)	USPTex	RandomForest	10-fold	90.05 (0.40)	USPTex
CLBP(radius:3, points:8)	USPTex	LDA	10-fold	88.01 (0.29)	USPTex
CLBP(radius:3, points:8)	USPTex	RandomForest	10-fold	87.44 (0.34)	USPTex

Showing 1 to 10 of 14 entries

Source: Elaborated by the author.

5.3.6 Interact with the Confusion Matrix

A confusion matrix is a tool that helps to visualize the results of a classification model. It provides visual feedback about the performance of a classifier over multiple classes. It is a matrix of size $n \times n$ where n is the number of classes. The rows correspond to the true classes and the columns to the predicted classes. Elements correctly classified lie on the main diagonal of the confusion matrix, while elements incorrectly classified as another class lie outside the main diagonal. If all elements lie in the main diagonal we have a classifier with perfect accuracy.

The application presents an interactive confusion matrix to identify the images correctly classified and those misclassified. In this way the user have a better idea of the results of classification. The confusion matrix can be accessed through the link that appears in the metrics interface of the classification results. Each element in the matrix has a color, the stronger the color the more number of successes of classifications. Those elements with zero classifications are of white color.

Figure 28 shows an example that corresponds to the confusion matrix for the UIUC

Figure 27 – Metrics of classification results for an experiment that includes the descriptor LPQ, dataset UIUC, and classifier RandomForest.

Texture Evaluation Platform

Experiment

Algorithms

Datasets

Search

Task Progress

Metrics

Algorithm	Parameters			Dataset	Classes	Images	Matrix	
LPQdesc.m	winSize: 3, decorr: 1, freqestim: 1			UIUC	25	1000	UIUC	
Classifier	k-fold	Attributes	Accuracy	Mean absolute error		Kappa	Correct	Incorrect
RandomForest	10-fold	256	72.2	5.3214		0.7104	722	278
precision	recall	f-measure	TP	FN	Name classes			
0.619	0.65	0.634	26	14	T01_bark1			
0.553	0.525	0.538	21	19	T02_bark2			
0.795	0.775	0.785	31	9	T03_bark3			
0.659	0.675	0.667	27	13	T04_wood1			
0.667	0.8	0.727	32	8	T05_wood2			
0.615	0.6	0.608	24	16	T06_wood3			
0.773	0.85	0.81	34	6	T07_water			
0.872	0.85	0.861	34	6	T08_granite			
0.577	0.75	0.652	30	10	T09_marble			
0.5	0.475	0.487	19	21	T10_floor1			
0.696	0.8	0.744	32	8	T11_floor2			

Source: Elaborated by the author.

dataset using the LPQ descriptor (LPQdesc.m in the image) applying the Random Forest classifier. A summary of the classification result is shown at the top. The cells of the diagonal are more bluish indicating a greater number of successes of classification. To interact with the confusion matrix the user can move the mouse over the cells of the matrix, a description will be exhibited corresponding to the actual class, the predicted class, and the number of instances classified. Making a click at a cell in the confusion matrix at the right will appear the correspondent images.

5.4 Implementation Requirements

In this work it was necessary to define the appropriate set of software tools to develop the platform. In the same way, it was defined the hardware in which the platform would run. The following subsections explain the software and hardware used to deploy the platform.

Figure 28 – Confusion matrix showing the results of classification of LPQ over UIUC, using Random-Forest with 10-fold cross validation. The cell clicked with red border indicates that 6 images were misclassified as *T14_brick1* instead of the actual class *T02_bark2*. The correspondent images appear at the right of the confusion matrix.

Source: Elaborated by the author.

5.4.1 Software details

The platform is designed to meet the purposes of this work, that is, to ease the research and collaboration among researchers. The platform was mainly developed in the Python programming language. Python was chosen because it contains a useful set of libraries that ease the creation of programs to make data science, web programming, among others. In the application Python simplified the integration of scientific code with web code. Python contains packages that integrates nicely in the development process, and to interface with other programming languages needed for the platform: JavaScript, Matlab, and Java.

JavaScript helps to develop interactive web applications, in the platform it is used in the client side sending and receiving *JSON* data from the server. Matlab is widely used in the scientific community, it has a rich set of tools for image processing, in the platform most of the texture descriptors are written in Matlab. Java is used to access the machine learning algorithms of Weka (Waikato Environment for Knowledge Acquisition) that works in a Java environment.

The main python packages used in the application are Flask, ‘python weka wrapper’, sci-kit learn, SQLAlchemy, and Celery.

- The microframework **Flask** provides tools to build web applications. It is a minimal framework that can be easily extended. It uses a template engine that helps in the creation of web interfaces. In the platform, Flask helps in the interaction between the Web Interface and the Core Application.
- The '**python weka wrapper**' is a package that allows Python programs to interact with Java applications. It is used in the platform to access the classification algorithms of Weka.
- The **scikit-learn** package is a machine learning toolkit written in Python, it contains tools for machine learning, data mining and data analysis (PEDREGOSA *et al.*, 2011).
- The **sqlalchemy** package is a database toolkit designed for efficient database access. It is used to persist objects to a relational database using a familiar programming model. It helps to deal with the changes in the database in the development process of the platform.
- The **Celery** package is an asynchronous task queue based on distributed message passing. Celery works in an asynchronous fashion distributing tasks over workers to run in the background, thus the user is not affected in its interaction with the application. A worker can be a computer, or a node in a cluster (SOLEM, 2018). In the application, the feature extraction process and the classification process are time-consuming tasks, so they are managed by Celery. There are two queues in the platform: *Features* and *Classify*. The *Features* queue is responsible for managing tasks that extract texture features, and the *Classify* queue is responsible for managing tasks that make the classification.

5.4.2 Hardware used

It was possible to develop the application on one platform and deploy it on another platform thanks to the use of programming languages that require minimal changes to work on different platforms.

The development process was made in a notebook running Windows 7, of 64 bits, with a processor Intel core i5 and 8 GB of RAM. The application was deployed on the Linux cluster of the Scientific Computing Group (SCG). The cluster consists of 5 nodes, each node has 8 CPUs with 16GB of memory RAM. The main process runs in the root node and is responsible of managing the other nodes, collecting data and storing it in the database. The other nodes run the processes to extract features and make the classification. The feature extraction process runs on 3 nodes, and the classification process runs on 2 nodes.

To use the application the user needs a modern compatible browser such as Firefox or Chrome, and the application can be accessed from any device, such as a notebook or a smartphone.

Figure 29 shows a view of the Texture Evaluation Platform, its components and interactions between them. The Web application adds tasks of feature extraction and classification to

the Queues *Features* and *Classify*. The workers process the tasks, after the processing is done the workers store the generated data in the database. The web application explore the database to display classification results.

The Web Application and Database run in the main node in the cluster. The Queues are managed by Celery also running in the main node. The Workers are distributed and run in different nodes in the cluster.

Figure 29 – The Web application adds tasks to the Queues *Features* and *Classify*. The Workers execute the tasks and store the generated data in the database.

Source: Elaborated by the author.

5.5 Final considerations

The application provides a familiar environment to use. As a web based application it is easily accessible using a standard browser. The researcher can analyze and compare the relative performance of various texture descriptor algorithms in different image datasets.

The interfaces should enable a meaningful interaction with the user, thus, there are opportunities for improving the user interaction with the platform.

The researcher can focus in a texture descriptor that gets the best classification results in a specific image dataset and the researcher can analyze the texture descriptor that performs better among the available image datasets.

In comparison to traditional applications, there are additional requirements that a web based application needs to meet. The application needs to be secure, reliable and scalable.

Moreover, it should conform to principles of accessibility.

The application takes advantage of the Python programming language because it was possible to develop the platform in an operating system (Windows) and deploy it on another operating system (Linux) with minimal changes to the developed code.

For processes that consume many resources, time or memory, it is necessary that they run in independent workers. The use of a task queue in the platform helps to distribute the tasks among the nodes in the cluster.

As an application that interacts with users, the time response is important. Making some ‘minor’ changes in code result in an overall improvement in time response, for example, in the Search interface after adding an index to a column in the database the page renders in a proper time.

The next chapter presents a comparative analysis of texture descriptors and texture datasets. It describes the setup for the experiments and the classification results obtained through the use of the functionalities of TEP presented in this chapter.

COMPARATIVE ANALYSIS OF TEXTURE DESCRIPTORS AND DATASETS

The platform is a useful tool to analyze the relative performance of texture descriptors and texture datasets. It allows the researcher to perform texture classification experiments using texture descriptors and datasets present on the platform. The platform is running at the cluster of the Scientific Computing Group and it can be accessed through the site <http://scg.ifsc.usp.br>. Using the platform we performed a set of experiments with the purpose to compare different texture methods on different datasets.

In this chapter experiments and data generated using the platform to study texture descriptors and datasets are presented and discussed. [Section 6.1](#) analyzes the interaction among texture descriptors over texture datasets. [Section 6.2](#) presents a comparison of results found using the experiments. [Section 6.3](#) presents information gained through the use of the platform.

6.1 Analysis of Texture Datasets

In the experiments we consider 29 texture descriptors ([Table 2](#)), 11 texture datasets ([Section 4.1](#)), 3 classifiers ([Section 3.4](#)) and 1 validation scheme ([Section 3.5](#)).

For comparison purposes with texture descriptors not present in the platform, we consider those works that employ texture datasets with the same characteristics, such as size of images, number of classes and samples per class, since a dataset can be modified to test certain properties of a texture descriptor. As an example, the UIUC dataset has images of size 640×480 and some recent works employed a cropped version of the images of size 256×256 ([FLORINDO; BRUNO, 2018](#)). The following subsections describe the results of the experiments.

Table 2 – Summary of texture descriptors used in the platform.

N	Descriptor	Acronym	Reference
1	Local Binary Pattern	LBP	(OJALA; PIETIKÄINEN; HARWOOD, 1996)
2	Uniform LBP	LBP ^{u2}	(OJALA; PIETIKÄINEN; MAENPAA, 2002)
3	Rotation Invariant LBP	LBP ^{ri}	(PIETIKÄINEN; OJALA; XU, 2000)
4	Uniform Rotation Invariant LBP	LBP ^{riu2}	(OJALA; PIETIKÄINEN; MAENPAA, 2002)
5	Rotated LBP	RLBP	(MEHTA; EGAZARIAN, 2013)
6	Complete LBP	CLBP	(GUO; ZHANG; ZHANG, 2010a)
7	Fuzzy LBP	FLBP	(IAKOVIDIS; KERAMIDAS; MAROULIS, 2008)
8	Shift LBP	SLBP	(KYLBERG; SINTORN, 2013)
9	Adaptive LBP	ALBP	(GUO <i>et al.</i> , 2010)
10	LBP Variance	LBPV	(GUO; ZHANG; ZHANG, 2010b)
11	Local Phase Quantization	LPQ	(AHONEN <i>et al.</i> , 2008)
12	Local Ternary Pattern	LTP ^{riu2}	(TAN; TRIGGS, 2010)
13	Binarized Statistical Image Features	BSIF	(KANNALA; RAHTU, 2012)
14	Gray Level Co-occurrence Matrix	GLCM	(PATHAK; BAROOAH, 2013)
15	Locally Encoded TRansform feature hISTogram	LETRIST	(SONG <i>et al.</i> , 2017)
16	Segmentation-based Fractal Texture Analysis	SFTA	(COSTA; HUMPIRE; TRAINA, 2012)
17	Bouligand Minkowski	Minkowski	(SILVA <i>et al.</i> , 2015)
18	Multi Fractal Spectrum	MFS	(XU; JI; FERMÜLLER, 2009)
19	LBP Histogram Fourier	LBP-HF	(AHONEN <i>et al.</i> , 2009)
20	Local Configuration Pattern	LCP	(GUO; ZHAO; PIETIKÄINEN, 2011)
21	Local Contrast Pattern reduced riu2	LCP ^{re}	(SONG <i>et al.</i> , 2014)
22	CLBP Histogram Fourier Signal Magnitude	CLBP-HFSM	(ZHAO <i>et al.</i> , 2012)
23	Adaptive Hybrid Pattern	AHP	(ZHU <i>et al.</i> , 2015)
24	Local Frequency Descriptor	LFD	(MAANI; KALRA; YANG, 2013)
25	Fourier Angular	Fourier	(SILVA <i>et al.</i> , 2016)
26	Gabor	Gabor	(WANG; DING; LIU, 2005)
27	Dual-Tree Complex Wavelet Transform	DTCWT	(ANANTRASIRICHA; BURN; BULL, 2014)
28	Undecimated DTCWT	UDTCWT	(ANANTRASIRICHA; BURN; BULL, 2014)
29	Complex Networks	CN	(BACKES; CASANOVA; BRUNO, 2013)

6.1.1 1200Tex dataset

The 1200Tex texture dataset contains images of 20 plant species of Brazilian flora, there are 60 samples per specie, making a total of 1200, hence the name of this dataset, the images are of size 128×128 . Table 3 shows the classification results in the 1200Tex dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest (RF) and LDA with 10-fold cross validation. The best results correspond to the Adaptive Hybrid Pattern (AHP) descriptor across the three classifiers. In this dataset the classifier is influencing the classification results, because the first 8 descriptors using the LDA classifier achieved an accuracy over 80%, that are close to the best results using kNN and Random Forest. Thus, for 1200Tex dataset the LDA classifier discriminates better the classes improving the classification results. The best result correspond to the Adaptive Hybrid Pattern (AHP) descriptor with the LDA classifier with an accuracy of 87.45% (0.36). From the 1200 images, AHP correctly identifies 1049 images and incorrectly 151 images. The second best result corresponds to the Complex Network (CN) descriptor with 84.90% (0.36). CN correctly identifies 1019 images and incorrectly 181 images. It is worth noting that the Complex Network descriptor using kNN got 65.31% (0.60) and with Random Forest got 69.20% (0.46), this indicates that the characteristics extracted from an image can be better discriminated according to the classifier used.

Table 3 – Classification results in 1200Tex dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	AHP	336	80.75 (0.43)	AHP	324	81.66 (0.44)	AHP	432	87.45 (0.36)
2	CLBP	128	78.56 (0.57)	DTCWT	38	79.75 (0.38)	CN	108	84.90 (0.33)
3	DTCWT	50	76.62 (0.33)	CLBP	128	79.44 (0.75)	DTCWT	86	83.51 (0.32)
4	FLBP	256	75.08 (0.51)	LETRIST	352	78.57 (0.61)	CLBP	128	83.04 (0.42)
5	LETRIST	352	74.34 (0.45)	FLBP	256	77.85 (0.31)	SLBP	64	81.20 (0.33)
6	BSIF	256	73.08 (0.42)	SLBP	256	76.71 (0.62)	BSIF	256	80.90 (0.30)
7	SLBP	64	73.06 (0.43)	CLBPHF-SM	46	76.35 (0.60)	CLBPHF-SM	76	80.24 (0.26)
8	LTP ^{riu2}	16	73.04 (0.37)	LCP	81	76.32 (0.62)	LPQ	256	80.05 (0.40)
9	LBP ^{u2}	59	70.53 (0.48)	LBP ^{u2}	59	75.70 (0.57)	LCP	81	79.77 (0.36)
10	LFD	276	68.94 (0.51)	LTP ^{riu2}	20	75.66 (0.57)	LBP ^{u2}	59	79.21 (0.31)
11	LPQ	256	67.43 (0.52)	LFD	276	75.53 (0.63)	FLBP	256	78.83 (0.62)
12	LCP ^{re}	36	67.38 (0.51)	LBP-HF	38	74.39 (0.61)	LETRIST	315	78.33 (0.63)
13	URLBP	135	66.20 (0.50)	LBP	64	73.22 (0.71)	LFD	276	78.25 (0.60)
14	LBP	64	65.37 (0.59)	LCP ^{re}	36	73.08 (0.55)	LBP-HF	38	77.37 (0.43)
15	CN	108	65.31 (0.60)	URLBP	59	73.03 (0.68)	GLCM	48	77.30 (0.36)
16	Gabor	49	64.80 (0.52)	BSIF	256	72.70 (0.62)	LBP	64	77.20 (0.38)
17	LBP-HF	27	64.00 (0.48)	Gabor	30	71.10 (0.57)	URLBP	59	75.97 (0.39)
18	LBPV	10	63.91 (0.47)	CN	108	69.20 (0.46)	LTP ^{riu2}	20	75.86 (0.33)
19	SFTA	33	63.67 (0.39)	SFTA	45	69.09 (0.48)	LCP ^{re}	36	75.00 (0.51)
20	ALBP	10	62.65 (0.48)	LBP ^{ri}	36	68.15 (0.60)	UDTCWT	6144	74.39 (0.43)
21	LBP ^{riu2}	10	62.32 (0.42)	LPQ	256	67.99 (1.10)	Fourier	80	74.07 (0.45)
22	CLBPHF-SM	46	59.66 (0.52)	LBPV	10	67.28 (0.65)	SFTA	45	71.52 (0.56)
23	LCP	64	58.60 (0.63)	LBP ^{riu2}	10	67.05 (0.63)	LBP ^{ri}	36	71.00 (0.49)
24	Fourier	80	57.42 (0.65)	ALBP	10	66.28 (0.61)	LBPV	10	68.61 (0.36)
25	LBP ^{ri}	14	56.87 (0.52)	MFS	26	65.17 (0.64)	LBP ^{riu2}	10	67.58 (0.32)
26	GLCM	48	52.17 (0.43)	Fourier	80	63.14 (0.77)	Gabor	49	66.97 (0.38)
27	MFS	26	49.84 (0.42)	GLCM	48	59.82 (0.74)	ALBP	10	66.19 (0.37)
28	UDTCWT	384	48.52 (0.51)	UDTCWT	384	57.96 (1.00)	Minkowski	142	62.08 (0.42)
29	Minkowski	54	35.48 (0.50)	Minkowski	42	39.95 (0.60)	MFS	26	48.27 (0.44)

Table 3 also shows how the length of the feature vector influences the classification results of the classifier used. That is the case of the AHP descriptor, for kNN the best result corresponds to values of AHP that generated 336 features, for Random Forest different values of AHP generated 324 features, and for LDA another set of values generated 432 features. Thus, with a greater number of features LDA performs a better discrimination among classes which allows to obtain the best classification result in the 1200Tex dataset. Another results interesting to examine are those found with the CLBP descriptor. The number of features generated using CLBP are 128 and the classification results are 78.56% (0.57), 79.44% (0.75), and 83.04% (0.42) using kNN, Random Forest and LDA respectively. These results indicate that using the same features the classification accuracy varies, and again LDA distinguishes better among classes in the 1200Tex dataset.

Minkowski and MFS were the descriptors that got the worst classification results, and using LDA most of the descriptors with lower classification results also have a smaller number of features with the exception of UDTCWT and Minkowski. The descriptor LBP-HF with 38 attributes got 77.37% (0.43) and this value continues decreasing until the descriptor MFS with 26 attributes got 48.27% (0.44).

Table 4 shows the best classification results using the platform and three results of recent texture descriptors not present in the platform. In the work that presented this dataset the classification accuracy got 84% using Gabor Wavelets and LDA with Leave One Out cross validation scheme (DALCIMAR; JOACI; MARTINEZ, 2009). The fractal texture descriptor by Florindo, Assirati and Bruno (2016) achieved 86.25% and 0.03 of error. They use Principal Component Analysis (PCA) to reduce the number of features, and LDA as classifier and cross validation, even though they do not indicate the number of folds. The Discrete Schroedinger descriptor by Florindo and Bruno (2017) got 85.33% and 0.03 of error, using PCA, with LDA and 10-fold cross validation scheme. The Complex Network descriptor achieved an accuracy of 84.90% (0.33) using the classifier LDA with 10-fold cross-validation scheme. The use of the AHP descriptor and the LDA classifier produced a better result than those found in the literature for the 1200Tex texture dataset with an accuracy of 87.45% (0.36).

Table 4 – Comparing classification results in 1200Tex dataset with texture descriptors not present in the platform.

Descriptor	Accuracy	Features	Post	Classifier	Cross-validation	Author
Gabor Wavelet	84.00	20	-	LDA	LeaveOneOut	(DALCIMAR; JOACI; MARTINEZ, 2009)
Fractal Non-additive entropy	86.25	-	PCA	LDA	k-fold	(FLORINDO; ASSIRATI; BRUNO, 2016)
Discrete Schroedinger	85.33	-	PCA	LDA	10-fold	(FLORINDO; BRUNO, 2017)
Complex Network	84.90 (0.33)	108	-	LDA	10-fold	(BACKES; CASANOVA; BRUNO, 2013)
Adaptive Hybrid Pattern	87.45 (0.36)	432	-	LDA	10-fold	(ZHU <i>et al.</i> , 2015)

Source: Elaborated by the author.

Table 5 presents the 10 best classification results in 1200Tex dataset using the classifier LDA with 10-fold cross-validation. It happens that the 10 results correspond to the Adaptive Hybrid Pattern (AHP) descriptor. The parameters of AHP are radius, points and quantization, the values of the parameters change the number of features generated. The parameters indicate that in this dataset the best results happens when radius varies from 2 to 4 using from 12 to 16 points. The accuracy varies from 86.20% (0.27) until 87.45% (0.36), these 10 results from AHP are better than the result found with Complex Network 84.90% (0.33). The first three results corresponds to the number of features of AHP for each classifier in Table 3, 432 (LDA), 336 (kNN) and 324 (Random Forest).

Figure 30 shows the confusion matrices for the best classification results for the 1200Tex dataset using the Adaptive Hybrid Pattern (AHP) and the Complex Network descriptors with the LDA classifier and 10-fold cross-validation scheme. In the confusion matrix, the vertical axis corresponds to the actual classes and the horizontal axis corresponds to the predicted classes. The classes with the largest number of incorrectly classified elements are *c14* and *c16* in both confusion matrices, out of 60 elements they have 43 and 34 elements correctly classified using AHP, and 38 and 32 elements correctly classified using CN. It can be noted that these two classes share the largest number of misclassified elements in both confusion matrices. In the confusion matrix coming from AHP, there are 11 elements of *c14* misclassified as *c16* and there are 10

Table 5 – The best 10 classification results in 1200Tex happens with the Adaptive Hybrid Pattern descriptor using LDA and 10-fold cross validation.

Parameters (radius, points, quantization)	Length	Accuracy
(2, 16, 9)	432	87.45 (0.36)
(3, 12, 9)	336	87.21 (0.35)
(3, 16, 7)	324	87.06 (0.33)
(2, 16, 7)	324	87.02 (0.35)
(2, 12, 9)	336	87.02 (0.34)
(3, 16, 9)	432	86.92 (0.39)
(3, 12, 7)	252	86.91 (0.32)
(4, 16, 9)	432	86.43 (0.36)
(2, 12, 7)	252	86.37 (0.33)
(3, 16, 5)	216	86.20 (0.27)

Source: Research data.

elements of *c16* misclassified as *c14*, also the class *c16* has 6 elements misclassified as class *c07*. The class *c12* has 9 elements misclassified as class *c07*. The other class that has more misclassified elements is the *c18* with 8 elements misclassified as *c19*. The other classes have 4 or fewer items misclassified as another class. The larger number of images that belong to one class but were classified as another are *c01* with 15, *c07* with 19, *c14* with 22, *c18* with 13 and *c19* with 13, for example, the class *c19* has all its 60 elements classified correctly, but another 13 elements were classified as *c19*. In the confusion matrix coming from CN, a class to note is *c03* that has 9 elements incorrectly classified as *c18*, instead using AHP all 60 elements of *c03* were correctly classified. Thus, the features extracted from each descriptor helps to discriminate better different elements in each dataset.

Table 6 presents the precision, recall, the number of true positive (TP), and false negative (FN) for each class in the 1200Tex dataset, from the Adaptive Hybrid Pattern and Complex Network descriptors. The class with the lower recall in both cases are *c16*, with values close to 0.5 which means that almost 50% of images were incorrectly classified as another class. From the data of AHP, the number of incorrectly classified images is 151, most of the images incorrectly classified are in the classes *c01*, *c12*, *c13*, *c14*, *c15*, *c16*, *c17* and *c18*. From the data coming from descriptor CN, in addition to these classes most of the images incorrectly classified are in classes *c03* and *c09*.

6.1.2 Brodatz dataset

The Brodatz dataset used in the experiments contains 111 classes with 10 samples per class, the images are gray scale of size 200×200 , in total there are 1110 images. Table 7 shows the classification results in the Brodatz dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. With 10 images per class, it is important to note that in the 10-fold cross validation

Figure 30 – Confusion matrices for the 1200Tex dataset. The confusion matrices correspond to the classification result using the Adaptive Hybrid Pattern and Complex Networks descriptors with the LDA classifier and 10-fold cross-validation scheme.

Source: Elaborated by the author.

Table 6 – Classification metrics in 1200Tex dataset using the Adaptive Hybrid Pattern (AHP) descriptor and Complex Network (CN) descriptor using LDA classifier with 10-fold cross validation.

AHP						CN				
precision	recall	f-measure	TP	FN	Classes	precision	recall	f-measure	TP	FN
0.75	0.75	0.75	45	15	c01	0.797	0.783	0.79	47	13
1	0.917	0.957	55	5	c02	1	0.917	0.957	55	5
0.984	1	0.992	60	0	c03	0.943	0.833	0.885	50	10
1	0.967	0.983	58	2	c04	1	0.983	0.992	59	1
0.903	0.933	0.918	56	4	c05	0.967	0.983	0.975	59	1
0.983	0.95	0.966	57	3	c06	1	0.983	0.992	59	1
0.733	0.917	0.815	55	5	c07	0.949	0.933	0.941	56	4
0.933	0.933	0.933	56	4	c08	0.836	0.85	0.843	51	9
0.95	0.95	0.95	57	3	c09	0.781	0.833	0.806	50	10
0.95	0.95	0.95	57	3	c10	0.912	0.867	0.889	52	8
0.95	0.95	0.95	57	3	c11	0.836	0.933	0.882	56	4
0.885	0.767	0.821	46	14	c12	0.721	0.817	0.766	49	11
0.857	0.8	0.828	48	12	c13	0.852	0.767	0.807	46	14
0.672	0.717	0.694	43	17	c14	0.633	0.633	0.633	38	22
1	0.833	0.909	50	10	c15	0.93	0.883	0.906	53	7
0.607	0.567	0.586	34	26	c16	0.571	0.533	0.552	32	28
0.862	0.833	0.847	50	10	c17	0.881	0.867	0.874	52	8
0.787	0.8	0.793	48	12	c18	0.672	0.717	0.694	43	17
0.822	1	0.902	60	0	c19	0.868	0.983	0.922	59	1
0.934	0.95	0.942	57	3	c20	0.883	0.883	0.883	53	7

Source: Research data.

process 9 images will be used for training and only one image will be used for testing in each pass of the classification process. The best results for kNN and LDA classifiers correspond to the CLBP descriptor with 98.22% (0.06) and 98.76% (0.09) respectively. The best result for Random Forest is 97.79% (0.15) using the LETRIST descriptor, and it is a lower value than those found using LDA. It can be seen that each classifier provides different classification results for the same descriptor. The first four descriptors in LDA are derived from the LBP descriptor. This indicates that the Brodatz benchmark dataset can be discriminated using local features, since each image contained in the album correspond to a specific texture captured under controlled conditions. The results in kNN and Random Forest of CLBP descriptor have 200 features, and in LDA has 128 features. The second overall best result comes from AHP using the LDA classifier with 98.46% (0.11). The results in kNN of AHP descriptor have 336 features, and in Random Forest and LDA have 252 features.

The LBP Variance (LBPV) descriptor got 91.45% (0.20) of accuracy using the LDA classifier, this descriptor has the minimum number of features to discriminate among the 111 classes of the Brodatz dataset. Therefore, its discriminative power can be improved combining this descriptor with another, or even applying a multi-scale approach to extract texture features. As in the case of the 1200Tex dataset, the descriptors Minkowski and MFS got the lowest classification results among the three classifiers.

The Brodatz dataset is vastly used in the literature, and so there are modified versions of

Table 7 – Classification results in Brodatz dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	CLBP	200	98.22 (0.06)	LETRIST	352	97.79 (0.15)	CLBP	128	98.76 (0.09)
2	BSIF	256	98.00 (0.09)	CLBP	200	97.33 (0.17)	AHP	252	98.46 (0.11)
3	LETRIST	637	97.88 (0.09)	AHP	252	97.00 (0.21)	FLBP	256	98.36 (0.13)
4	AHP	336	97.64 (0.07)	BSIF	256	96.85 (0.26)	SLBP	256	98.32 (0.19)
5	SLBP	256	97.45 (0.14)	LCP	81	96.45 (0.34)	LETRIST	352	98.28 (0.16)
6	LPQ	256	97.34 (0.11)	LPQ	256	96.24 (0.40)	BSIF	128	98.14 (0.08)
7	DTCWT	38	96.78 (0.11)	CLBP-HFSM	76	96.04 (0.31)	LBP	128	98.01 (0.16)
8	URLBP	135	96.37 (0.16)	DTCWT	86	96.01 (0.36)	LPQ	256	97.88 (0.08)
9	FLBP	256	96.23 (0.12)	SLBP	256	95.95 (0.37)	CLBP-HFSM	76	97.79 (0.13)
10	LBP	128	95.86 (0.15)	FLBP	256	95.44 (0.21)	CN	108	97.75 (0.17)
11	LBP ^{u2}	59	95.47 (0.10)	LBP	256	95.44 (0.34)	LCP	81	97.73 (0.09)
12	LCP ^{re}	36	94.99 (0.16)	URLBP	135	95.34 (0.34)	LBP ^{u2}	59	97.05 (0.18)
13	CLBP-HFSM	76	94.97 (0.20)	LBP ^{u2}	59	95.09 (0.30)	DTCWT	62	96.70 (0.13)
14	LTP ^{riu2}	20	94.91 (0.15)	LCP ^{re}	36	94.86 (0.30)	URLBP	243	96.66 (0.25)
15	LBP-HF	27	94.63 (0.09)	LFD	276	94.54 (0.29)	GLCM	48	96.02 (0.17)
16	ALBP	10	94.60 (0.15)	LTP ^{riu2}	16	94.27 (0.40)	LBP-HF	27	95.61 (0.17)
17	LFD	276	94.28 (0.19)	LBP-HF	27	93.84 (0.28)	LBP ^{ri}	36	94.89 (0.29)
18	LCP	81	93.31 (0.24)	LBP ^{ri}	36	92.61 (0.41)	LTP ^{riu2}	20	94.84 (0.17)
19	LBP ^{ri}	36	93.23 (0.16)	ALBP	10	92.29 (0.35)	UDTCWT	1536	94.75 (0.36)
20	Fourier	80	91.92 (0.27)	GLCM	48	90.61 (0.35)	LFD	276	94.62 (0.28)
21	LBPV	10	91.78 (0.12)	Gabor	49	90.36 (0.46)	LCP ^{re}	36	93.35 (0.21)
22	GLCM	48	91.68 (0.17)	SFTA	45	89.28 (0.35)	SFTA	45	93.14 (0.28)
23	CN	108	91.41 (0.17)	Fourier	80	89.15 (0.40)	Fourier	80	93.10 (0.28)
24	SFTA	45	91.15 (0.16)	UDTCWT	384	89.06 (0.66)	ALBP	10	92.27 (0.21)
25	LBP ^{riu2}	10	89.85 (0.19)	LBPV	10	88.65 (0.45)	LBPV	9	91.45 (0.20)
26	UDTCWT	384	89.71 (0.17)	CN	108	88.04 (0.34)	LBP ^{riu2}	10	88.56 (0.21)
27	Gabor	49	89.56 (0.21)	LBP ^{riu2}	10	86.93 (0.51)	Gabor	35	88.06 (0.31)
28	MFS	26	81.33 (0.24)	MFS	26	84.03 (0.48)	Minkowski	142	86.76 (0.26)
29	Minkowski	69	59.81 (0.38)	Minkowski	102	56.03 (0.59)	MFS	26	82.10 (0.33)

Source: Research data.

the Brodatz dataset in order to reach specific objectives of a descriptors. For example, the work by [Florindo and Bruno \(2016\)](#) used a Brodatz dataset with 111 classes and 16 samples per class, making a total of 1776 images of size 128×128 , and got 91.56% (0.01) of accuracy using the LDA classifier. In another work by [Backes and Junior \(2017\)](#) used a Brodatz dataset with 40 classes and 10 samples per class, the images have size of 200×200 , and got 99.25% of accuracy using the LDA classifier.

[Table 8](#) shows the 10 best classification results in Brodatz dataset. All the results correspond to the LDA classifier. It is noted that most of the descriptors are derived from LBP, this indicates that for the Brodatz dataset extracting local patterns produced better discrimination among classes. The best two results correspond to CLBP with radius 2 and using 6 and 8 points, normally it is used 8 points, for this dataset using 6 points results in a better data discrimination. On the other hand, the AHP descriptor gets its best results using 12 or 16 points as it was in the case of 1200Tex dataset. Out of 1110, the first two results coming from the CLBP descriptor misclassified 14 and 16 elements respectively. From the third to the tenth classification result the number of misclassified increases from 14 until 19, this indicates that it is possible to use any of

Table 8 – The best 10 classification results in Brodatz dataset are obtained using LDA and 10-fold cross validation.

Descriptor	Length	Accuracy	Misses
CLBP(radius:2, points:6)	128	98.76 (0.09)	14
CLBP(radius:2, points:8)	200	98.52 (0.10)	16
AHP(radius:2, points:12, quantization:7)	252	98.46 (0.11)	17
FLBP(fuzziness:6)	256	98.36 (0.13)	18
AHP(radius:3, points:16, quantization:7)	324	98.34 (0.08)	18
SLBP(radius:1, points:8, limit:3)	256	98.32 (0.19)	19
CLBP(radius:1, points:6)	128	98.32 (0.10)	19
AHP(radius:2, points:16, quantization:9)	432	98.31 (0.07)	19
SLBP(radius:1, points:8, limit:5)	256	98.30 (0.17)	19
LETRIST(Ls:3, Lr:4)	352	98.28 (0.16)	19

Source: Research data.

Table 9 – Classification metrics in Brodatz dataset coming from CLBP and AHP descriptors which produced the best results using LDA classifier with 10-fold cross validation.

CLBP						AHP				
precision	recall	f-measure	TP	FN	Classes	precision	recall	f-measure	TP	FN
1	0.7	0.824	7	3	D7	0.889	0.8	0.842	8	2
1	0.9	0.947	9	1	D13	0.909	1	0.952	10	0
1	0.9	0.947	9	1	D38	1	1	1	10	0
1	0.9	0.947	9	1	D42	1	0.9	0.947	9	1
1	0.9	0.947	9	1	D43	0.9	0.9	0.9	9	1
1	1	1	10	0	D45	0.9	0.9	0.9	9	1
1	1	1	10	0	D58	0.9	0.9	0.9	9	1
1	0.9	0.947	9	1	D59	1	0.8	0.889	8	2
0.769	1	0.87	10	0	D60	0.9	0.9	0.9	9	1
1	0.9	0.947	9	1	D62	1	0.9	0.947	9	1
1	0.8	0.889	8	2	D69	1	1	1	10	0
1	0.9	0.947	9	1	D72	1	0.9	0.947	9	1
1	1	1	10	0	D88	0.9	0.9	0.9	9	1
1	1	1	10	0	D90	1	0.8	0.889	8	2
0.9	0.9	0.9	9	1	D91	1	0.8	0.889	8	2
1	0.9	0.947	9	1	D100	1	0.9	0.947	9	1

these descriptors for a similar dataset and the decision to choose one over other it will depend on another factors, such as processing time to extract features.

Table 9 shows metrics of the classification results of the best two descriptors in Brodatz dataset. The data was reduced to highlight those classes that contain misclassified elements. There are 14 misclassified elements using CLBP and 17 misclassified elements using AHP. CLBP has 11 classes in which some elements were misclassified whereas that AHP has 13 classes. All these classes have at most 3 misclassified elements, there are 14 misclassified elements in the results from CLBP and class *D7* has 3 misses and class *D96* has 2 misses, and there are 17 misclassified elements in the results from AHP and classes *D7*, *D59*, *D90* and *D91* have 2 misses each class.

6.1.3 CURET dataset

The CURET dataset contains 61 classes with 92 images per class, each image of size 200×200 , making a total of 5612 images. CURET presents real world textures captured in different imaging conditions, such as illumination and viewing angle. The images were converted to gray scale images before extracting texture features. One difference of CURET with respect to the previous datasets is that in CURET the samples come from different capture conditions, whereas in 1200Tex and Brodatz the samples are extracted from a single image. Table 10 shows the classification results in the CURET dataset. It is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best classification results correspond to the Adaptive Hybrid Pattern (AHP) descriptor across the three classifiers. The best overall result is for kNN with 98.06% (0.07), out of 5612 images AHP correctly identified 5503 images and incorrectly classified 109 images. The same descriptor AHP but with a different number of features got its best result for Random Forest with 97.26% (0.13), while in LDA the result of classification decreased to 93.42% (0.09). It can be seen that on this dataset the results coming from LDA are lower compared to those obtained using kNN and Random Forest. The second overall best result corresponds to DTCWT descriptor, the DTCWT is based in Wavelet transform, with 97.73% (0.05), in this case it correctly classified 5485 images and incorrectly classified 127 images. As in the case of the 1200Tex and Brodatz dataset, the descriptors Minkowski and MFS got the lowest classification results among the three classifiers. The variation of the classification results with MFS is notorious regarding the use of the classifier, with kNN it got 68.93% (0.24) and with RF it increased to 75.75% (0.28), but with LDA it decreased to 21.76% (0.27)

Table 11 presents the 10 best classification results in CURET dataset using the classifier kNN with 10-fold cross-validation. It happens that the 10 results correspond to the Adaptive Hybrid Pattern (AHP) descriptor. The parameters of AHP are radius, points and quantization, the values of the parameters change the number of features generated. The parameters indicate that in this dataset the best results happens when radius varies from 2 to 3 using from 12 to 16 points. The accuracy varies from 97.79% (0.09) until 98.06% (0.07), these 10 results from AHP are better than the result found with DTCWT 97.73% (0.05).

Comparing results from descriptors in the literature not present in the platform we have the work of Cernadas *et al.* (2017) in which they analyze the CURET dataset with its intensity and its color information. Its best result achieved 97.7% combining intensity and color information in an integrative approach, and 98.3% using pure color features and intensity features. The work of Pan *et al.* (2017) proposed a descriptor based on LBP and achieved 97.12% for CURET using intensity information with the kNN classifier. For the classification process it uses a training and test set repeated 100 times. From these results, it can be seen that the AHP descriptor produced a result that is closer to those found in literature in the CURET dataset with an accuracy of 98.06% (0.07) using only intensity information. This result shows that using only intensity information

Table 10 – Classification results in CURET dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	AHP	324	98.06 (0.07)	AHP	432	97.26 (0.13)	AHP	432	93.42 (0.09)
2	DTCWT	62	97.73 (0.05)	CLBP	200	96.39 (0.15)	UDTCWT	1536	92.91 (0.12)
3	CLBP	200	96.99 (0.13)	LETRIST	413	96.31 (0.10)	LETRIST	413	91.99 (0.14)
4	BSIF	256	96.63 (0.13)	DTCWT	62	96.08 (0.11)	CLBP	200	90.51 (0.13)
5	LETRIST	413	96.63 (0.11)	LCP ^{re}	36	95.01 (0.17)	LBP	256	90.47 (0.12)
6	LBP	256	96.35 (0.14)	LFD	276	94.98 (0.15)	DTCWT	62	90.25 (0.10)
7	SLBP	256	96.19 (0.17)	CLBP-HFSM	76	94.96 (0.13)	LFD	276	90.03 (0.12)
8	LFD	276	96.07 (0.12)	LCP	81	94.41 (0.14)	SLBP	256	90.02 (0.11)
9	LCP ^{re}	36	95.98 (0.10)	BSIF	128	94.30 (0.19)	FLBP	256	88.24 (0.18)
10	LBP ^{u2}	59	95.19 (0.12)	LBP ^{u2}	59	92.97 (0.21)	CLBP-HFSM	76	88.21 (0.13)
11	UDTCWT	384	95.11 (0.11)	LBP ^{PHF}	38	92.30 (0.16)	LCP	81	86.22 (0.14)
12	FLBP	256	94.49 (0.15)	LBP	256	92.24 (0.21)	BSIF	256	85.83 (0.12)
13	Fourier	80	94.13 (0.13)	Gabor	25	92.20 (0.14)	Fourier	80	84.61 (0.16)
14	LPQ	256	93.98 (0.12)	SLBP	256	92.10 (0.19)	LBP ^{u2}	59	84.00 (0.11)
15	LCP	81	93.50 (0.13)	FLBP	256	91.96 (0.19)	LPQ	256	83.66 (0.10)
16	CLBP-HFSM	76	92.40 (0.13)	Fourier	80	91.89 (0.21)	LBP ^{PHF}	38	81.29 (0.13)
17	URLBP	59	91.95 (0.14)	URLBP	59	91.64 (0.15)	CN	108	80.97 (0.11)
18	Gabor	25	90.79 (0.15)	LPQ	256	90.03 (0.20)	URLBP	59	80.91 (0.14)
19	SFTA	45	90.05 (0.16)	LBP ^{ri}	36	89.98 (0.18)	GLCM	32	75.95 (0.14)
20	LTP ^{riu2}	20	89.91 (0.17)	LTP ^{riu2}	20	89.57 (0.15)	LCP ^{re}	36	74.19 (0.13)
21	ALBP	10	88.91 (0.15)	SFTA	45	89.01 (0.13)	Gabor	25	73.83 (0.14)
22	LBP ^{PHF}	38	88.30 (0.20)	ALBP	10	88.76 (0.18)	LBP ^{ri}	36	73.20 (0.17)
23	LBP ^{ri}	36	87.89 (0.21)	UDTCWT	384	88.56 (0.26)	SFTA	45	71.52 (0.17)
24	GLCM	32	86.57 (0.15)	GLCM	32	87.29 (0.19)	LTP ^{riu2}	20	66.99 (0.17)
25	CN	108	85.03 (0.18)	CN	108	86.03 (0.27)	ALBP	10	63.52 (0.16)
26	LBP ^{riu2}	10	84.01 (0.23)	LBP ^{riu2}	10	84.45 (0.27)	LBP ^{riu2}	10	60.45 (0.18)
27	LBPV	10	83.65 (0.15)	LBPV	10	82.35 (0.17)	LBPV	10	57.19 (0.20)
28	MFS	26	68.93 (0.24)	MFS	26	75.75 (0.28)	Minkowski	142	53.01 (0.23)
29	Minkowski	142	54.93 (0.23)	Minkowski	142	54.21 (0.34)	MFS	26	21.76 (0.27)

Source: Research data.

Table 11 – The best 10 classification results in CURET happens with the Adaptive Hybrid Pattern descriptor using kNN and 10-fold cross validation.

Parameters (radius, points, quantization)	Length	Accuracy
(2, 16, 7)	324	98.06 (0.07)
(3, 12, 7)	252	98.05 (0.07)
(2, 16, 9)	432	98.05 (0.07)
(3, 12, 9)	336	98.02 (0.08)
(3, 16, 9)	432	97.94 (0.11)
(2, 16, 5)	216	97.88 (0.08)
(2, 12, 9)	336	97.88 (0.06)
(3, 16, 7)	324	97.87 (0.10)
(3, 12, 5)	168	97.85 (0.07)
(2, 8, 5)	120	97.79 (0.09)

Source: Research data.

the AHP descriptor provides characteristics that allow to differentiate the classes one from each other, and it can be possible to improve the results of CURET adding color information to the feature extraction with AHP.

Figure 31 shows the confusion matrix of the best classification result in CURET dataset using the Adaptive Hybrid Pattern (AHP) with the kNN classifier and 10-fold cross-validation. The number of correctly classified images in each class across the diagonal are above 80 images, and the class *Linen* is the one with the most misclassified images. The confusion matrix shows that class *Linen* misclassified 4 elements with class *Polyester*, the other classes have 3 or fewer items misclassified as another class.

6.1.4 Kylberg dataset

The Kylberg dataset consists of 28 classes with 160 samples per class of size 576×576 making a total of 4480 images, in the experiments we use the version with no rotated patches. Table 12 shows the classification results in the Kylberg dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best classification results correspond to the CLBP descriptor across the three classifiers. The best result for kNN gets 100% of accuracy, using Random Forest gets 99.97% (0.02) and using LDA 99.92% (0.01). In this case the results on the three classifiers corresponds to the same number of features. The number of misclassified images are 1 and 4 in Random Forest and LDA respectively. It can be noted that the second best result corresponds to the Local Contrast Pattern (LCP^{re}), with 99.96% (0.01) in kNN, with 99.93% (0.02) in Random Forest, and slightly decreases with LDA and got 98.64% (0.05). In this dataset, almost all the descriptors got results higher than 90% using the three classifiers, with the exception of MFS with LDA. It is also noted that there are no results for the LETRIST descriptor, this is because the descriptor did not manage to extract characteristics to the dataset, due to the number of samples in the dataset and the size of the images.

Comparing the performance of the texture descriptors in the platform with others not present in the platform, we found that the work by Hanbay *et al.* (2016) proposed an improved HOG (Histogram of Gradients) descriptor and used the Kylberg dataset to assess its performance. It achieved 100% of accuracy using the kNN classifier, the feature vector has a length of 128 elements. It used a training and testing set for the classification process. Another work by Hiremath and Bhusnurmath (2017) proposed the local directional binary patterns (LDBP). Its result in Kylberg with this descriptor also achieved 100% of accuracy. They use Principal Component Analysis (PCA) plus LDA to reduce the number of features and to improve the discrimination among classes, with kNN as classifier and 10-fold cross validation. It is noticed that Kylberg presents such set of images that can be effectively classified. In this dataset most descriptors got classification results close to 100% therefore is not an adequate benchmark dataset for texture descriptors.

Figure 31 – Confusion matrix for the CURET dataset. It was used the AHP descriptor and the kNN classifier.

Source: Elaborated by the author.

Table 12 – Classification results in Kylberg dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	CLBP	200	100.00 (0.01)	CLBP	200	99.97 (0.02)	CLBP	200	99.92 (0.01)
2	LCP ^{re}	36	99.96 (0.01)	UDTCWT	384	99.95 (0.02)	AHP	120	99.90 (0.01)
3	UDTCWT	384	99.94 (0.02)	LCP ^{re}	36	99.93 (0.02)	UDTCWT	1536	99.78 (0.02)
4	URLBP	135	99.94 (0.02)	DTCWT	50	99.86 (0.03)	URLBP	243	99.68 (0.03)
5	AHP	120	99.92 (0.01)	AHP	120	99.81 (0.04)	LBP	256	99.49 (0.03)
6	LPQ	256	99.92 (0.02)	LPQ	256	99.79 (0.05)	SLBP	256	99.44 (0.03)
7	DTCWT	62	99.86 (0.03)	CLBP-HFSM	76	99.71 (0.04)	CLBP-HFSM	76	99.44 (0.03)
8	LTP ^{riu2}	20	99.56 (0.06)	URLBP	135	99.67 (0.05)	DTCWT	62	99.44 (0.03)
9	SLBP	256	99.53 (0.05)	LCP	81	99.44 (0.07)	LCP	81	99.35 (0.04)
10	LBP	256	99.52 (0.05)	FLBP	256	99.21 (0.07)	FLBP	256	99.30 (0.04)
11	CLBP-HFSM	76	99.42 (0.05)	LBP-HF	38	99.16 (0.07)	LCP ^{re}	36	98.64 (0.05)
12	ALBP	10	99.28 (0.07)	LBP	256	99.10 (0.06)	CN	108	98.57 (0.04)
13	LBP ^{u2}	59	99.25 (0.07)	SLBP	256	99.06 (0.05)	LFD	276	98.45 (0.05)
14	LBP-HF	38	99.22 (0.07)	BSIF	256	99.02 (0.10)	Fourier	80	98.33 (0.06)
15	SFTA	45	99.09 (0.06)	LTP ^{riu2}	20	98.96 (0.09)	LPQ	256	98.29 (0.04)
16	CN	108	99.08 (0.08)	LFD	276	98.91 (0.07)	LBP ^{u2}	59	98.18 (0.06)
17	LCP	81	99.04 (0.05)	LBP ^{u2}	59	98.87 (0.09)	LBP-HF	38	98.08 (0.05)
18	LBP ^{ri}	36	99.04 (0.07)	Gabor	25	98.69 (0.09)	LBP ^{ri}	36	97.87 (0.05)
19	BSIF	256	98.96 (0.08)	SFTA	45	98.61 (0.11)	SFTA	45	97.84 (0.05)
20	FLBP	256	98.94 (0.06)	LBP ^{ri}	36	98.21 (0.08)	LTP ^{riu2}	20	97.19 (0.05)
21	LBP ^{riu2}	10	98.51 (0.07)	CN	108	98.21 (0.08)	BSIF	256	96.97 (0.06)
22	LFD	276	98.51 (0.04)	ALBP	10	98.17 (0.11)	Minkowski	102	96.44 (0.04)
23	LBPV	10	97.71 (0.11)	Fourier	80	97.45 (0.10)	GLCM	32	95.88 (0.05)
24	Fourier	80	97.54 (0.07)	LBP ^{riu2}	10	96.59 (0.13)	ALBP	10	93.39 (0.06)
25	Gabor	25	97.45 (0.11)	GLCM	32	96.57 (0.12)	LBPV	10	93.24 (0.07)
26	GLCM	32	97.21 (0.11)	MFS	26	96.49 (0.09)	LBP ^{riu2}	10	92.88 (0.07)
27	MFS	26	95.47 (0.14)	LBPV	10	96.44 (0.13)	Gabor	25	89.33 (0.11)
28	Minkowski	69	95.42 (0.16)	Minkowski	69	93.26 (0.17)	MFS	26	80.10 (0.13)
29	LETRIST	-	-	LETRIST	-	-	LETRIST	-	-

Source: Research data.

Table 13 shows the 10 best classification results in Kylberg dataset. The results correspond to the kNN and Random Forest classifier, in this case the results obtained with LDA were less than those obtained with kNN. The descriptors are derived from LBP and based in Wavelet transform. The descriptor that have less attributes is Local Contrast Pattern (LCP^{re}) with only 36 features. The number of misclassified images varies from 0 to 3 images using these descriptors. It can be noted that as the images are of size 576×576 using the descriptors based in LBP the best results are obtained with a radius of 3, but maintaining the standard number of points to 8.

6.1.5 KTH-TIPS2b dataset

KTH-TIPS2b dataset contains textures collected under different conditions of lighting, scale and pose, there are 11 classes with 432 samples per class, making a total of 4752 color images of 200×200 . Table 14 shows the classification results in the KTH-TIPS2b dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best result for kNN classifier correspond to the BSIF descriptor with 98.12% (0.08). The best results for Random Forest

Table 13 – The best 10 classification results in Kylberg dataset are obtained using kNN and Random Forest classifiers.

Descriptor	Length	Classifier	Accuracy	Misses
CLBP(radius:3, points:8)	200	kNN	100.00 (0.01)	0
CLBP(radius:3, points:8)	200	RF	99.97 (0.02)	1
LCP ^{re} (radius:3, points:8)	36	kNN	99.96 (0.01)	2
UDTCWT(wlevels:4, startlevel:2)	384	RF	99.95 (0.02)	2
CLBP(radius:2, points:8)	200	kNN	99.95 (0.01)	2
UDTCWT(wlevels:4, startlevel:2)	384	kNN	99.94 (0.02)	3
URLBP(radius:3, points:12)	135	kNN	99.94 (0.02)	3
URLBP(radius:2, points:8)	59	kNN	99.93 (0.02)	3
LCP ^{re} (radius:3, points:8)	36	RF	99.93 (0.02)	3
URLBP(radius:3, points:8)	59	kNN	99.93 (0.02)	3

Source: Research data.

and LDA classifiers correspond to the CLBP descriptor with 95.96% (0.15) and 91.69% (0.10) respectively, these are lower values than those found using kNN. For KTH-TIPS2b the best classification results corresponds to the descriptors using the kNN classifier, it can be noted that the 20 first results from kNN and Random Forest are over 90%, that almost reaches the best results that were found using LDA. In this dataset the LDA classifier got the lowest classification results, compared to the results already seen in previous datasets. From the data using the kNN classifier, the best result is 98.12% (0.08) using the BSIF descriptor, the number of misclassified images are 89. The second best result is 98.02% (0.09) using the Fuzzy LBP descriptor, the number of misclassified images are 94. In this dataset the descriptors UDTCWT and LETRIST did not have results because they failed to extract features to this dataset, this because the number of images present in the dataset.

Comparing results from descriptors in the literature not present in the platform we have the work by [Ji *et al.* \(2018\)](#) who proposed a multi-scale Local Ternary Pattern (LTP) on gray scale images. On KTH-TIPS2b dataset got 98.71% (0.60) with Support Vector Machine (SVM) as classifier, and 98.58% (0.65) with kNN as classifier. They consider a multi-scale approach by concatenating the results of applying LTP with 3 radius and 3 different filter windows. The work by [Bianconi *et al.* \(2017\)](#) proposed a color texture descriptor derived from LBP, for the classification process they used a training and testing set repeated 100 times and achieved 98.5% with kNN as classifier. It is noted that these works got their best results using the kNN classifier, as it happened with the experiments in the platform. These results show that the BSIF and Fuzzy LBP descriptor produced classification results closer to those found in literature for the KTH-TIPS2b dataset with an accuracy of 98.12% (0.08) and 98.02% (0.09) using only intensity information.

[Table 15](#) presents the precision, recall, the number of true positive (TP), and false negative (FN) for each class in the KTH-TIPS2b dataset, from the BSIF and Fuzzy LBP descriptors. There are 432 samples per class and none of the classes got 100 percent hits, which is reflected

Table 14 – Classification results in KTH-TIPS2b dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	BSIF	256	98.12 (0.08)	CLBP	200	95.96 (0.15)	CLBP	200	91.69 (0.10)
2	FLBP	256	98.02 (0.09)	DTCWT	62	95.85 (0.15)	SLBP	256	91.59 (0.13)
3	SLBP	256	97.83 (0.12)	BSIF	256	95.82 (0.11)	LBP	256	90.97 (0.10)
4	DTCWT	62	97.82 (0.09)	AHP	252	95.79 (0.17)	AHP	252	90.10 (0.13)
5	LBP	256	97.82 (0.09)	FLBP	256	94.99 (0.13)	FLBP	256	88.73 (0.13)
6	LPQ	256	97.76 (0.08)	CLBP-HFSM	76	94.87 (0.20)	CLBP-HFSM	76	88.02 (0.13)
7	LBP ^{u2}	59	97.34 (0.08)	LBP ^{u2}	59	94.82 (0.16)	BSIF	256	87.59 (0.09)
8	CLBP	200	97.02 (0.10)	LBP	256	94.41 (0.18)	DTCWT	74	85.98 (0.13)
9	AHP	252	96.28 (0.10)	SLBP	256	94.25 (0.18)	LPQ	256	85.92 (0.10)
10	LFD	276	95.78 (0.15)	LCP	81	93.93 (0.19)	LCP	81	84.28 (0.10)
11	CLBP-HFSM	76	95.74 (0.12)	LCP ^{re}	36	93.86 (0.18)	LBP ^{u2}	59	82.66 (0.13)
12	LCP ^{re}	36	94.95 (0.11)	LPQ	256	93.48 (0.22)	URLBP	59	82.30 (0.12)
13	URLBP	59	94.57 (0.17)	LFD	276	92.78 (0.18)	LFD	276	80.47 (0.16)
14	LBP-HF	38	93.83 (0.16)	URLBP	59	92.75 (0.18)	LBP-HF	38	80.23 (0.09)
15	LCP	81	92.93 (0.17)	LBP-HF	38	92.12 (0.19)	CN	108	79.13 (0.11)
16	Fourier	80	92.71 (0.15)	Gabor	30	91.78 (0.16)	SFTA	45	76.17 (0.13)
17	SFTA	45	92.11 (0.14)	Fourier	80	90.61 (0.19)	Fourier	80	75.97 (0.18)
18	Gabor	36	90.94 (0.12)	SFTA	45	90.41 (0.21)	LCP ^{re}	36	75.76 (0.11)
19	LTP ^{riu2}	20	90.08 (0.17)	LTP ^{riu2}	20	89.76 (0.24)	GLCM	48	73.91 (0.12)
20	ALBP	10	89.55 (0.13)	LBP ^{ri}	36	89.55 (0.19)	LBP ^{ri}	36	73.83 (0.10)
21	GLCM	48	89.50 (0.14)	GLCM	48	88.21 (0.23)	LTP ^{riu2}	20	71.72 (0.09)
22	LBP ^{ri}	36	88.65 (0.17)	ALBP	10	88.05 (0.21)	LBPV	10	66.67 (0.09)
23	CN	108	87.07 (0.19)	CN	108	87.59 (0.23)	Gabor	36	65.16 (0.18)
24	LBPV	10	86.33 (0.15)	LBPV	10	85.60 (0.23)	ALBP	10	64.64 (0.11)
25	LBP ^{riu2}	10	85.01 (0.19)	LBP ^{riu2}	10	84.38 (0.23)	LBP ^{riu2}	10	63.55 (0.10)
26	MFS	26	78.20 (0.20)	MFS	26	83.15 (0.23)	Minkowski	142	59.66 (0.09)
27	Minkowski	142	64.81 (0.21)	Minkowski	69	70.17 (0.38)	MFS	26	50.51 (0.14)
28	LETRIST	-	-	LETRIST	-	-	LETRIST	-	-
29	UDTCWT	-	-	UDTCWT	-	-	UDTCWT	-	-

Source: Research data.

in the recall measure. The class with the most misclassified elements is *lettuce_leaf* with 19 and 21 elements misclassified using BSIF and Fuzzy LBP respectively. From the data coming from BSIF the classes with more misclassified elements are *cork*, *cotton*, *cracker*, *lettuce_leaf*, and *wool*. And from the data coming from Fuzzy LBP the classes with more misclassified elements are *corduroy*, *cotton*, *lettuce_leaf*, and *wool*. The other classes have less than 10 elements misclassified in both descriptors.

Figure 32 shows the confusion matrices for the best classification results for the KTH-TIPS2b dataset using the BSIF and Fuzzy LBP descriptors with the kNN classifier and 10-fold cross-validation. Comparing the confusion matrices, the *corduroy* class is the one that improves its success rate, from 15 in Fuzzy LBP to 7 in BSIF. It is also noted that the *lettuce_leaf* class is the one that is misclassified with most other classes. In BSIF *lettuce_leaf* is classified as *BrownBread* with 5 images, and as *WhiteBread* with 6 images. In Fuzzy LBP the *lettuce_leaf* is classified as *BrownBread* with 6 images, and as *cork* with 5 images. In BSIF, the *cork* class has the highest number of misclassified elements with class *cracker* with 9 images. The other class with the highest number of misclassified images is *cotton* as *wood* with 7 images. In Fuzzy

Table 15 – Classification metrics in KTH-TIPS2b dataset using the kNN classifier. The data comes from the BSIF and Fuzzy LBP descriptors.

BSIF						FuzzyLBP				
precision	recall	f-measure	TP	FN	Classes	precision	recall	f-measure	TP	FN
0.995	0.993	0.994	429	3	aluminium_foil	0.995	0.995	0.995	430	2
0.977	0.995	0.986	430	2	brown_bread	0.973	0.988	0.98	427	5
0.991	0.984	0.987	425	7	corduroy	0.99	0.965	0.978	417	15
0.981	0.97	0.976	419	13	cork	0.966	0.979	0.972	423	9
0.984	0.972	0.978	420	12	cotton	0.981	0.965	0.973	417	15
0.95	0.972	0.961	420	12	cracker	0.959	0.986	0.973	426	6
0.983	0.956	0.969	413	19	lettuce_leaf	0.983	0.951	0.967	411	21
1	0.993	0.997	429	3	linen	0.998	0.993	0.995	429	3
0.975	0.991	0.983	428	4	white_bread	0.982	0.988	0.985	427	5
0.979	0.993	0.986	429	3	wood	0.979	0.995	0.987	430	2
0.979	0.975	0.977	421	11	wool	0.977	0.975	0.976	421	11

LBP these same classes are misclassified in almost the same number of images, the *cork* class as *cracker* with 7 images, and the *cotton* class as *wood* with 8 images.

Figure 32 – Confusion matrices for the KTH-TIPS2b dataset. The confusion matrices correspond to the classification result using the BSIF and Fuzzy LBP descriptors with the kNN classifier and 10-fold cross-validation scheme.

Source: Elaborated by the author.

6.1.6 Midrib dataset

The Midrib dataset contains images that are larger than typical benchmark datasets, such as Brodatz or Outex-10. The purpose of studying this dataset is to identify plant species using the

midrib cross-section of leaves. It consists of 50 classes (plant species), with a variable number of samples per class, ranging from 10 to 20, making a total of 606 images. Table 16 shows the classification results in the Midrib dataset. It is presented the best results for 16 different texture descriptors using the kNN, Random Forest and LDA classifiers with 10-fold cross validation scheme. The best results correspond to the CLBP descriptor across the three classifiers. The CLBP descriptor got its best result with 92.62% (0.35) using the kNN classifier, it got results over 86% using both Random Forest and LDA, even though they come from different set of parameters because of the different number of attributes. The results coming from the CLBP descriptor show that it uses 648 attributes in kNN and Random Forest, but it is not shown in the values of LDA because only got 76.40% (1.09), instead it appears a better result of the CLBP descriptor with 200 attributes and 88.44% (0.65) of accuracy, this result indicates that the number of features also influences the classification process. The CLBP, URLBP, and LBP^{u2} descriptors got results over 86% using KNN, a value that is better than the accuracy achieved by Silva *et al.* (2015), in that work they used 5 texture analysis methods to classify the species and got 83.67% of accuracy with a fractal descriptor.

Table 16 only presents the classification results from 16 descriptors because the other descriptors failed to extract features in the Midrib dataset, these texture descriptors are ALBP, AHP, UDTCTWT, FLBP, Gabor, GLCM, LBPV, LETRIST, LFD, LPQ, Minkowski, LCP^{re}, and CN. They were not able to compute the textures features mainly because of the size of the images in the Midrib dataset. It is also noted that the LTP^{riu2} and CLBP-HFSM descriptors do not have values for the LDA classifier.

Table 16 – Classification results in Midrib dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	CLBP	648	92.62 (0.35)	CLBP	648	87.19 (0.64)	CLBP	200	88.44 (0.65)
2	URLBP	243	86.86 (0.50)	LCP	81	80.92 (0.53)	LCP	289	83.10 (0.66)
3	LBP ^{u2}	243	86.43 (0.58)	URLBP	243	79.67 (0.84)	DTCWT	62	73.72 (0.69)
4	LBP	256	83.77 (0.66)	BSIF	1024	78.50 (0.75)	URLBP	243	73.04 (0.89)
5	LTP ^{riu2}	36	83.14 (0.61)	CLBP-HFSM	76	77.95 (1.64)	BSIF	512	69.72 (0.70)
6	BSIF	1024	81.61 (0.47)	LBP ^{u2}	243	76.37 (1.10)	SFTA	21	69.65 (0.70)
7	LBP ^{riu2}	18	80.44 (0.68)	LTP ^{riu2}	36	72.93 (0.98)	LBP	4096	66.11 (0.97)
8	LBP ^{ri}	36	77.05 (0.50)	LBP ^{riu2}	26	72.78 (0.99)	LBP ^{riu2}	26	63.76 (0.74)
9	SFTA	21	70.03 (0.43)	LBP	256	72.50 (1.27)	LBP ^{ri}	36	62.43 (0.77)
10	LCP	289	69.54 (0.57)	SFTA	21	71.34 (0.36)	LBP ^{u2}	59	59.45 (0.82)
11	DTCWT	62	68.30 (0.67)	LBP ^{ri}	36	67.94 (0.94)	LBP ^{PHF}	38	58.40 (0.63)
12	MFS	26	67.11 (0.49)	DTCWT	62	67.55 (1.19)	MFS	26	54.27 (0.83)
13	CLBP-HFSM	76	66.44 (0.65)	MFS	26	63.67 (0.72)	SLBP	256	48.09 (0.81)
14	LBP ^{PHF}	38	56.78 (0.79)	LBP ^{PHF}	38	59.59 (0.93)	Fourier	80	37.15 (0.82)
15	SLBP	256	54.43 (0.61)	SLBP	256	54.44 (1.21)	LTP ^{riu2}	-	-
16	Fourier	80	49.87 (0.57)	Fourier	80	40.18 (1.27)	CLBP-HFSM	-	-

Source: Research data.

It is interesting to examine the results obtained with the LBP descriptor, because it reached the best result in the work of Silva *et al.* (2015). They used the LBP operator and achieved an

accuracy of 49.12%. Their work does not indicate the values of *radius* and *points*, or the number of attributes of the feature vector. With standard values of *radius* = 1 and *points* = 8 the LBP descriptor gave an accuracy of 55.51%. As the images are large, it is reasonable to increase the *radius* in the experiments. Taking 8 points with radius larger than 1 in this dataset they capture better the texture information of the images. Figure 33 shows the classification results using the LBP descriptor with different values of points and radius. The best accuracy is obtained with *radius* = 3 and *points* = 8 using the kNN classifier with 83.77% (0.66) of accuracy. This result implies that with large images, the texture descriptor used needs to cover larger regions to capture better texture features. Using the kNN classifier, with 8 points the feature vector has 256 attributes, changing the number of points to 12, the feature vector goes to 4096 attributes, however the classification accuracy does not improve as it is also shown in the data presented in Table 16.

Figure 33 – Classification results for LBP in Midrib dataset. Varying the parameters radius and points. Increasing the radius and maintaining 8 points the accuracy improves and it got its best result at *radius* = 3.

Source: Elaborated by the author.

Table 17 presents the ten best classification results in Midrib dataset, the data comes from using the CLBP descriptor and the kNN and LDA classifiers. In Midrib dataset the images are larger than images in Benchmark datasets, and the values found indicate that the images in Midrib dataset require a larger region to capture texture details. The first three results uses 16 points, and their classification accuracy increases conforming the radius increases from 4 to 6. The same happens when using 8 points and increasing the radius from 2 to 4, the classification accuracy increases.

Figure 34 shows the confusion matrix of the best result in the Midrib dataset that corresponds to the CLBP descriptor with the kNN classifier. Midrib dataset is an imbalanced

Table 17 – The best 10 classification results in Midrib dataset comes from using the CLBP descriptor and the kNN and LDA classifiers.

Descriptor	Length	Classifier	Accuracy	Misses
CLBP(radius:6, points:16)	648	kNN	92.62 (0.35)	45
CLBP(radius:5, points:16)	648	kNN	92.17 (0.55)	47
CLBP(radius:4, points:16)	648	kNN	90.86 (0.55)	55
CLBP(radius:4, points:12)	392	kNN	90.59 (0.50)	57
CLBP(radius:4, points:8)	200	kNN	90.57 (0.33)	57
CLBP(radius:3, points:8)	200	kNN	89.83 (0.38)	62
CLBP(radius:3, points:12)	392	kNN	89.46 (0.37)	64
CLBP(radius:3, points:16)	648	kNN	89.36 (0.40)	64
CLBP(radius:2, points:8)	200	kNN	89.08 (0.55)	67
CLBP(radius:4, points:8)	200	LDA	88.44 (0.65)	70

Source: Research data.

dataset and has from 10 to 20 images per class, this can be seen in the values of the main diagonal. Out of 606 images in Midrib, CLBP with kNN correctly classified 561 images and incorrectly classified 45 images. In the matrix the names of the classes are alphabetically ordered. The class *MyracrodruonUrundeuva* has the largest number of incorrectly classified elements, out of 14 elements it has 4 misclassified elements, each one misclassified with a different class. The class *DilodendronBipinnatum* has 8 elements correctly classified out of 11 elements, and it has its 3 elements misclassified as *BauhiniaUngulata*. The other classes have 2 or 1 elements misclassified as another class.

6.1.7 Outex-10 dataset

Outex-10 dataset is part of a framework of image datasets intended to test texture classification and segmentation. Outex-10 contains 24 classes of homogeneous texture images rotated at 9 angles, with 180 images per class, in total there are 4320 images of size 128×128 . Table 18 shows the classification results in Outex-10 dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best classification results correspond to LFD and BSIF descriptors using the kNN classifier with 99.98% (0.02), which means that only 1 image was misclassified as another class. The best result for Random Forest got 99.91% (0.03) by using LPQ descriptor, in LDA got 99.88% (0.02) by using LFD. The descriptors with the best result have been developed to be rotation invariant and noise robust, as is the case of LFD and LPQ, these descriptors works in the frequency domain being robust to rotations. It can be noted that all the descriptors got an accuracy over 90%, even the Minkowski descriptor with LDA got 90.82% (0.09) of accuracy. These classification results indicate that the Outex-10 dataset is a well established benchmark and it is expected that most texture descriptors can discriminate well among their classes.

Table 19 presents the 10 best classification results in Outex-10 dataset using kNN and

Figure 34 – Confusion matrix for the best result in Midrib dataset. It was used the CLBP descriptor and the kNN classifier.

Source: Elaborated by the author.

Table 18 – Classification results in Outex-10 dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	LFD	276	99.98 (0.02)	LPQ	256	99.91 (0.03)	LFD	276	99.88 (0.02)
2	BSIF	256	99.98 (0.02)	LFD	276	99.90 (0.03)	CLBP	200	99.80 (0.02)
3	LPQ	256	99.96 (0.02)	BSIF	128	99.90 (0.03)	CLBP-HFSM	76	99.79 (0.02)
4	FLBP	256	99.94 (0.02)	CLBP-HFSM	76	99.87 (0.02)	LETRIST	637	99.79 (0.02)
5	DTCWT	38	99.92 (0.02)	LETRIST	637	99.87 (0.04)	SLBP	256	99.68 (0.04)
6	CLBP	200	99.88 (0.02)	DTCWT	62	99.83 (0.04)	AHP	120	99.63 (0.02)
7	AHP	120	99.87 (0.04)	CLBP	200	99.79 (0.04)	LBP	256	99.53 (0.04)
8	SLBP	256	99.83 (0.04)	FLBP	256	99.76 (0.06)	LCP	81	99.47 (0.03)
9	LBP ^{u2}	59	99.75 (0.04)	LBP-HF	38	99.71 (0.05)	FLBP	256	99.33 (0.05)
10	LBP	256	99.70 (0.05)	AHP	120	99.69 (0.02)	LBP-HF	38	99.07 (0.03)
11	Gabor	30	99.64 (0.05)	SLBP	256	99.69 (0.07)	URLBP	59	99.05 (0.04)
12	LCP ^{re}	36	99.46 (0.06)	LCP	81	99.68 (0.05)	BSIF	256	98.95 (0.06)
13	URLBP	59	99.41 (0.04)	LBP	256	99.63 (0.05)	DTCWT	50	98.90 (0.04)
14	CLBP-HFSM	76	99.38 (0.04)	LBP ^{u2}	59	99.62 (0.07)	UDTCWT	1536	98.79 (0.06)
15	LCP	81	99.21 (0.05)	LCP ^{re}	36	99.61 (0.05)	LPQ	256	98.68 (0.06)
16	LTP ^{riu2}	20	99.03 (0.08)	Gabor	30	99.49 (0.08)	Fourier	80	98.61 (0.06)
17	CN	108	98.64 (0.09)	URLBP	59	99.17 (0.05)	LCP ^{re}	36	98.60 (0.04)
18	Fourier	80	98.60 (0.07)	LTP ^{riu2}	20	98.74 (0.09)	LBP ^{u2}	59	98.42 (0.06)
19	LETRIST	637	98.53 (0.05)	LBP ^{ri}	36	98.47 (0.10)	CN	108	97.47 (0.05)
20	LBP ^{ri}	36	98.52 (0.07)	ALBP	10	97.92 (0.11)	SFTA	45	96.85 (0.04)
21	LBP-HF	38	98.49 (0.06)	Fourier	80	97.83 (0.11)	LBP ^{ri}	36	96.79 (0.07)
22	ALBP	10	98.37 (0.07)	CN	108	97.73 (0.10)	LTP ^{riu2}	20	96.63 (0.04)
23	LBPV	10	97.91 (0.09)	LBPV	10	97.35 (0.10)	glcm	32	94.96 (0.08)
24	UDTCWT	384	97.72 (0.08)	SFTA	45	97.18 (0.14)	ALBP	10	94.13 (0.06)
25	SFTA	45	97.35 (0.08)	UDTCWT	384	97.15 (0.15)	Gabor	30	93.91 (0.09)
26	GLCM	32	96.81 (0.12)	glcm	32	95.69 (0.17)	LBPV	10	93.44 (0.09)
27	LBP ^{riu2}	10	94.93 (0.12)	LBP ^{riu2}	10	95.48 (0.14)	LBP ^{riu2}	10	91.51 (0.07)
28	MFS	26	92.08 (0.21)	MFS	26	94.90 (0.10)	Minkowski	142	90.82 (0.09)
29	Minkowski	42	85.96 (0.14)	Minkowski	69	86.69 (0.20)	MFS	26	82.36 (0.11)

Source: Research data.

Random Forest classifiers with 10-fold cross-validation. Most of these descriptors work in the frequency domain getting results that effectively classify the Outex-10 dataset, and most of these descriptors are derived from LBP. It is noted that the DTCWT descriptor can discriminate the images using only 38 attributes, being the lowest value with respect to the other descriptors.

Comparing results from descriptors in the literature not present in the platform, the work of Roy *et al.* (2018) proposed the descriptor Local ZigZag Pattern, they achieved an accuracy of 99.95% (0.12) using the kNN classifier with 10-fold cross validation. The work of Salwa and Mohammed (2017) proposed a rotation invariant and noise texture descriptor, and they got an accuracy of 99.14% using the Support Vector Machine (SVM) classifier with 10-fold cross-validation. It can be seen that the BSIF and LFD descriptors produced a slightly better result in Outex-10 dataset with an accuracy of 99.98% (0.02). In this case the BSIF and LFD descriptors work effectively with rotated images of Outex-10.

Figure 35 shows the confusion matrices for Outex-10 dataset. The classification result in both LFD and BSIF descriptors got 99.98% (0.02), this means that only one image was misclassified. In the confusion matrix coming from LFD the image from class *carpet009* was

Table 19 – The best 10 classification results in Outex-10 dataset comes from using different descriptors and parameters and kNN and Random Forest classifiers.

Descriptor	Length	Classifier	Accuracy	Misses
LFD(radius:3, points:24, channels:12)	276	kNN	99.98 (0.02)	1
BSIF(filter:3x3_8bit)	256	kNN	99.98 (0.02)	1
BSIF(filter:3x3_7bit)	128	kNN	99.98 (0.01)	1
LPQ(winSize:5, decorr:1, freqestim:1)	256	kNN	99.96 (0.02)	2
FLBP(fuzziness:6)	256	kNN	99.94 (0.02)	3
LPQ(winSize:3, decorr:1, freqestim:1)	256	kNN	99.92 (0.02)	3
DTCWT(levels:3)	38	kNN	99.92 (0.02)	4
LPQ(winSize:5, decorr:1, freqestim:1)	256	RF	99.91 (0.03)	4
LFD(radius:4, points:24, channels:12)	276	kNN	99.91 (0.02)	4
LFD(radius:3, points:24, channels:12)	276	RF	99.90 (0.03)	4

Source: Research data.

misclassified as *carpet005*, instead in the confusion matrix coming from BSIF the image from class *tile006* was misclassified as *tile005*. The set of images in Outex-10 dataset were effectively classified, reaching almost 100% of classification accuracy, thus Outex-10 constitutes another good benchmark for texture descriptors.

Figure 35 – Confusion matrices for the Outex-10 dataset. The confusion matrices correspond to the classification result using the LFD and BSIF descriptors with the kNN classifier and 10-fold cross-validation scheme.

Source: Elaborated by the author.

6.1.8 STex dataset

The Salzburg Texture Image Database (STex) is a collection of 476 texture images, each image of 512×512 is split into 16 non-overlapping tiles making a total of 7616 images of size 128×128 . This dataset has the highest number of classes with respect to the other datasets included in the experiments. Table 20 shows the classification results in the STex dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best classification result for kNN got 86.81% (0.14) of accuracy using the CLBP descriptor, for Random Forest got 84.50% (0.21) using the DTCWT descriptor, and for LDA got 87.46% (0.08) using the LPQ descriptor. The best result corresponds to the LPQ descriptor with the LDA classifier, in terms of number of images classified in STex out of 7616 images there are 955 misclassified images. The second best classification result is 87.40% (0.11) and it comes from using the BSIF descriptor with the LDA classifier, this time there were 960 misclassified images. In this dataset the descriptors MFS, Minkowski and UDTCTWT got the lowest classification results, the other descriptors got an accuracy over 50%, in the same way they do not manage to extract texture features that can separate the classes of STex.

Comparing results from descriptors in the literature not present in the platform we have the work by Kim and So (2018) who proposed a descriptor based on multi resolution Gabor filters and they achieved 89.57% (1.08) on STex using kNN classifier, this means 794 misclassified elements. The classification process used 8 images for training and 8 images for testing per class and this process was repeated 100 times, for validation purposes uses a hold-out strategy. In the platform, the LPQ descriptor got 87.46% (0.08) of accuracy, and 955 misclassified elements, this result is lower than that found in literature. However, the setup of the experiment used is different and the LPQ descriptor only used intensity information.

These results indicate that there are various classes that have few well-classified elements, one reason for this is because of the nature of the images itself, since they come from a split process that results in non-overlapping images. Figure 36 shows a sample of texture images from STex in which they present the most misclassified elements, *Flower0000* with 15 misses, *Flower0003* with 15 misses, *Leather0003* with 13 misses, and *Misc0000* with 12 misses. For example, visually inspecting the *Flower0003* class, it has more than one type of texture, the samples from the central part are different from the surrounding samples.

6.1.9 UIUC dataset

The UIUC texture dataset contains images under rotation and viewpoint variation. UIUC is composed of 25 classes with 40 samples per class, the images are gray scale and of size 640×480 . Table 21 shows the classification results in UIUC dataset, it is presented the best results for the 29 different texture descriptors using kNN, Random Forest and LDA classifiers with 10-fold cross validation scheme. The best results correspond to the LETRIST descriptor across the

Table 20 – Classification results in STex dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	CLBP	128	86.81 (0.14)	DTCWT	62	84.50 (0.21)	LPQ	256	87.46 (0.08)
2	BSIF	256	86.62 (0.14)	CLBP	128	84.30 (0.18)	BSIF	256	87.40 (0.11)
3	AHP	252	86.11 (0.14)	AHP	252	80.69 (0.30)	CLBP	128	86.09 (0.11)
4	LPQ	256	86.06 (0.12)	BSIF	128	80.56 (0.26)	AHP	120	86.09 (0.10)
5	DTCWT	38	82.85 (0.12)	CLBP-HFSM	76	80.51 (0.27)	CLBP-HFSM	76	83.33 (0.15)
6	LETRIST	352	78.94 (0.13)	LETRIST	315	80.48 (0.32)	DTCWT	50	82.89 (0.14)
7	FLBP	256	78.84 (0.15)	SLBP	256	79.22 (0.26)	LETRIST	315	82.50 (0.15)
8	SLBP	64	78.79 (0.13)	LPQ	256	79.12 (0.26)	SLBP	256	82.33 (0.16)
9	LBP ^{u2}	59	77.69 (0.11)	FLBP	256	79.09 (0.27)	LCP	81	82.23 (0.15)
10	LFD	276	76.68 (0.11)	LBP ^{u2}	59	78.28 (0.21)	FLBP	256	81.84 (0.13)
11	LBP	256	76.29 (0.11)	LCP	81	77.29 (0.28)	LBP ^{u2}	59	80.73 (0.12)
12	URLBP	33	76.25 (0.16)	LFD	276	75.47 (0.31)	LBP	128	79.78 (0.19)
13	Gabor	49	74.24 (0.12)	LBP	256	75.45 (0.22)	GLCM	48	78.20 (0.14)
14	LCP ^{re}	36	72.31 (0.15)	Gabor	35	75.37 (0.30)	CN	108	76.77 (0.17)
15	CLBP-HFSM	76	71.18 (0.14)	URLBP	33	75.05 (0.29)	URLBP	33	76.73 (0.17)
16	GLCM	48	70.08 (0.11)	LCP ^{re}	36	73.80 (0.26)	LBP-HF	27	75.76 (0.15)
17	LTP ^{riu2}	20	69.07 (0.15)	LBP-HF	38	71.71 (0.25)	LFD	276	72.66 (0.20)
18	LCP	64	68.67 (0.18)	LTP ^{riu2}	16	70.25 (0.25)	Fourier	80	71.87 (0.17)
19	Fourier	80	67.63 (0.13)	GLCM	48	69.82 (0.26)	LTP ^{riu2}	20	71.12 (0.11)
20	LBP-HF	27	66.24 (0.14)	CN	108	67.06 (0.18)	LCP ^{re}	36	68.92 (0.15)
21	CN	108	65.41 (0.16)	Fourier	80	66.64 (0.26)	LBP ^{ri}	36	66.62 (0.14)
22	SFTA	45	60.84 (0.18)	SFTA	45	62.50 (0.22)	Gabor	49	66.53 (0.20)
23	ALBP	10	59.35 (0.17)	LBP ^{ri}	36	61.90 (0.26)	SFTA	45	63.48 (0.13)
24	LBP ^{ri}	36	56.25 (0.19)	ALBP	10	58.23 (0.26)	ALBP	10	60.05 (0.19)
25	LBPV	10	55.76 (0.15)	LBPV	10	56.72 (0.26)	LBPV	10	59.72 (0.19)
26	LBP ^{riu2}	8	50.42 (0.16)	LBP ^{riu2}	8	51.18 (0.32)	UDTCWT	384	56.41 (0.16)
27	UDTCWT	384	41.04 (0.10)	MFS	26	50.92 (0.30)	LBP ^{riu2}	10	54.89 (0.14)
28	MFS	26	37.79 (0.20)	UDTCWT	384	36.68 (0.31)	Minkowski	102	51.13 (0.16)
29	Minkowski	102	25.56 (0.06)	Minkowski	69	26.32 (0.19)	MFS	26	21.60 (0.15)

Source: Research data.

Figure 36 – Sample of texture images from STex dataset with most misclassified elements.

Source: Elaborated by the author.

three classifiers. The LETRIST descriptor was designed to deal with rotation, illumination, scale and viewpoint changes. The best classification result for kNN got 98.02% (0.18) of accuracy, for Random Forest got 97.52% (0.27), and for LDA got 98.25% (0.22). The best result corresponds to the LETRIST descriptor with the LDA classifier, in terms of number of images classifieds in UIUC, out of 1000 images there are 18 misclassified images. The CLBP descriptor appears in second place across the three classifiers too, using LDA got 97.99% (0.25), this means 20 misclassified images. The Minkowski, CN, and Gabor descriptors did not manage to extract features that could discriminate the classes of UIUC and they got the lowest classification results.

Table 21 – Classification results in UIUC dataset using kNN, Random Forest and LDA with 10 fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	LETRIST	413	98.02 (0.18)	LETRIST	413	97.52 (0.27)	LETRIST	413	98.25 (0.22)
2	CLBP	392	97.57 (0.18)	CLBP	392	96.93 (0.26)	CLBP	392	97.99 (0.25)
3	AHP	432	94.96 (0.31)	LCP ^{re}	36	94.52 (0.44)	AHP	624	97.56 (0.17)
4	LCP ^{re}	36	94.10 (0.24)	AHP	432	94.16 (0.41)	CLBP-HFSM	76	95.57 (0.29)
5	URLBP	135	94.01 (0.25)	URLBP	135	92.98 (0.44)	LCP	81	95.27 (0.24)
6	CLBP-HFSM	76	88.62 (0.45)	LFD	276	90.69 (0.50)	URLBP	243	94.19 (0.33)
7	LFD	276	87.71 (0.28)	CLBP-HFSM	76	90.43 (0.53)	LCP ^{re}	36	92.22 (0.34)
8	LCP	81	87.06 (0.44)	LCP	81	88.81 (0.60)	SFTA	45	90.27 (0.28)
9	SFTA	45	86.21 (0.38)	DTCWT	62	87.74 (0.64)	LBP-HF	38	88.47 (0.30)
10	UDTCWT	1536	86.03 (0.35)	SFTA	45	87.46 (0.45)	LTP ^{riu2}	20	86.83 (0.27)
11	LTP ^{riu2}	20	81.95 (0.28)	UDTCWT	384	86.44 (0.65)	UDTCWT	1536	85.92 (0.39)
12	DTCWT	62	81.92 (0.54)	MFS	26	83.12 (0.55)	LFD	276	85.55 (0.38)
13	ALBP	10	81.46 (0.36)	LTP ^{riu2}	20	81.52 (0.60)	CN	108	85.07 (0.51)
14	MFS	26	80.12 (0.46)	LPQ	256	80.74 (0.55)	LBP ^{ri}	36	83.74 (0.42)
15	LPQ	256	79.23 (0.62)	LBP-HF	38	79.88 (0.49)	SLBP	256	82.86 (0.49)
16	LBP-HF	38	78.88 (0.42)	Fourier	80	79.77 (0.61)	LBP	256	81.86 (0.52)
17	LBP ^{ri}	36	77.06 (0.43)	ALBP	10	77.72 (0.50)	LPQ	256	81.33 (0.48)
18	Fourier	80	76.39 (0.40)	LBP ^{ri}	36	76.30 (0.59)	FLBP	256	80.66 (0.59)
19	LBP	256	71.58 (0.47)	LBP ^{u2}	59	72.66 (0.83)	BSIF	256	80.48 (0.33)
20	LBP ^{u2}	59	70.33 (0.52)	LBP	256	72.59 (0.73)	DTCWT	62	80.44 (0.37)
21	LBPV	10	70.16 (0.41)	BSIF	256	72.32 (0.84)	LBP ^{u2}	59	79.49 (0.44)
22	FLBP	256	69.76 (0.60)	Gabor	30	71.70 (0.68)	ALBP	10	76.10 (0.47)
23	LBP ^{riu2}	10	69.36 (0.47)	SLBP	256	71.26 (0.60)	LBP ^{riu2}	10	74.20 (0.37)
24	SLBP	256	69.36 (0.56)	FLBP	256	70.34 (0.70)	LBPV	10	73.80 (0.43)
25	BSIF	256	67.81 (0.59)	LBPV	10	68.79 (0.58)	Minkowski	102	72.22 (0.27)
26	GLCM	48	66.71 (0.49)	GLCM	48	67.74 (0.74)	glcm	48	71.55 (0.69)
27	Gabor	30	65.03 (0.67)	LBP ^{riu2}	10	66.76 (0.69)	MFS	26	66.78 (0.56)
28	CN	108	62.57 (0.49)	CN	108	63.45 (0.75)	Fourier	80	64.52 (0.67)
29	Minkowski	142	48.62 (0.46)	Minkowski	142	48.79 (0.98)	Gabor	36	53.16 (0.51)

Source: Elaborated by the author.

Comparing results from descriptors in the literature not present in the platform we have the work by [Pan et al. \(2017\)](#) who proposed a descriptor based on LBP that is rotation and illumination invariant and they achieved 95.23% on UIUC using the descriptor in a multi-scale approach. The feature vector has dimension 1040 and the classifier used is kNN. The classification process uses 20 images for training and 20 images for testing per class and this process was repeated 100 times. [Ji et al. \(2018\)](#) proposed a noise robust, rotation invariant descriptor based on an improved Local Ternary Pattern (LTP). They got 96.97% (1.28) using

the descriptor in a multi-scale approach and Support Vector Machine (SVM) as classifier, and 96.89% (1.51) with kNN as classifier. The classification process is the same as the previous method, they differ in the number of repetitions, it was repeated 10 times. Both descriptors employ a multi-scale approach to get better classification results, unlike the simple approach used in the platform. It can be seen that the LETRIST descriptor produced a better result than those found in literature in UIUC dataset with an accuracy of 98.25% (0.22) using the LDA classifier. The UIUC dataset presents rotation and viewpoint variations, and the LETRIST descriptor deals with them effectively.

Table 22 shows metrics of the classification results of the best two descriptors in UIUC dataset. There are 18 misclassified elements using LETRIST with LDA and 20 misclassified elements using LETRIST with kNN. The *T20_upholstery* class has the most misclassified with 5 elements using LETRIST with LDA, instead this same class has only 1 misclassified element using LETRIST with kNN. The *T03_bark3* class has 4 misclassified elements using LETRIST with kNN, and this same class has 1 misclassified element using LETRIST with LDA. In the rest of the classes the elements have 3 or less misclassified elements. This is the first dataset in which the best two results come from the same descriptor using different classifiers. It can be seen that the same texture characteristics in general produce a similar result, since the difference in the number of misclassified images between them is 2 images. By previous results found in other datasets, the classifiers used influence the accuracy in the discrimination of the classes, besides that the misclassified images are similar when comparing the number of well-classified elements between the two metrics.

Figure 37 shows the confusion matrices for the best classification results for the UIUC dataset using the LETRIST descriptor with the LDA and kNN classifiers and 10-fold cross-validation scheme. The class *T20Upholstery* has 35 images well-classified, the other classes have above 38 images well-classified, as seen in the main diagonal of the confusion matrix that comes from LETRIST and LDA. One thing to notice among the two confusion matrices is that the misclassified images differ among them, which can be clearly seen in the first rows and columns of the confusion matrix that comes from LETRIST and kNN, and again this indicates that apart from influencing the classification results, the discriminated elements are also different even though it is used the same texture features but with different classifiers.

6.1.10 USPTex dataset

The USPTex texture dataset contains real world images with 191 classes and 12 samples per class, making a total of 2292 images, and they are of size 128×128 . USPTex is considered as a challenging dataset because of the number of classes (FLORINDO; CASANOVA; BRUNO, 2018). Table 23 shows the classification results in the USPTex dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best results in kNN and LDA correspond to the Adaptive

Figure 37 – Confusion matrices for the UIUC dataset. The confusion matrices correspond to the classification result using the LETRIST descriptor with the LDA and kNN classifiers and 10-fold cross-validation scheme.

Source: Elaborated by the author.

Table 22 – Classification metrics in UIUC dataset using the LETRIST descriptor with LDA and kNN classifier and 10-fold cross validation.

LETRIST LDA						LETRIST kNN				
precision	recall	f-measure	TP	FN	Classes	precision	recall	f-measure	TP	FN
1	1	1	40	0	T01_bark1	0.93	1	0.964	40	0
1	0.975	0.987	39	1	T02_bark2	0.974	0.925	0.949	37	3
0.951	0.975	0.963	39	1	T03_bark3	0.9	0.9	0.9	36	4
0.976	1	0.988	40	0	T04_wood1	0.974	0.95	0.962	38	2
1	0.975	0.987	39	1	T05_wood2	1	0.95	0.974	38	2
1	0.95	0.974	38	2	T06_wood3	0.95	0.95	0.95	38	2
0.976	1	0.988	40	0	T07_water	1	1	1	40	0
0.952	1	0.976	40	0	T08_granite	0.952	1	0.976	40	0
1	1	1	40	0	T09_marble	1	1	1	40	0
0.976	1	0.988	40	0	T10_floor1	1	1	1	40	0
1	0.975	0.987	39	1	T11_floor2	1	1	1	40	0
0.976	1	0.988	40	0	T12_pebbles	1	1	1	40	0
1	0.975	0.987	39	1	T13_wall	0.952	1	0.976	40	0
0.951	0.975	0.963	39	1	T14_brick1	1	0.975	0.987	39	1
1	0.975	0.987	39	1	T15_brick2	0.975	0.975	0.975	39	1
1	1	1	40	0	T16_glass1	1	1	1	40	0
0.976	1	0.988	40	0	T17_glass2	1	1	1	40	0
0.976	1	0.988	40	0	T18_carpet1	1	0.975	0.987	39	1
0.909	1	0.952	40	0	T19_carpet2	0.93	1	0.964	40	0
1	0.875	0.933	35	5	T20_upholstery	1	0.975	0.987	39	1
0.975	0.975	0.975	39	1	T21_wallpaper	1	0.925	0.961	37	3
1	1	1	40	0	T22_fur	1	1	1	40	0
1	0.975	0.987	39	1	T23_knit	1	1	1	40	0
0.976	1	0.988	40	0	T24_corduroy	0.976	1	0.988	40	0
1	0.95	0.974	38	2	T25_plaid	1	1	1	40	0

Hybrid Pattern descriptor getting 92.79% (0.14) with kNN, and 96.32% (0.14) with LDA, being the best overall result. The best result in Random Forest comes from the CLBP descriptor with 91.77% (0.35). And the second best overall result comes from the CLBP descriptor using LDA with 93.67% (0.19).

Comparing classification results from descriptors in the literature not present in the platform we have the work by [Florindo, Casanova and Bruno \(2018\)](#), they used a fractal descriptor on a set of images of different resolution in a Gaussian pyramid approach. In the USPTex, their descriptor achieved an accuracy of 92.54% $\pm$ 0.02, using LDA classifier with 10-fold cross-validation. The work by [Liu et al. \(2017\)](#) combine color features and LBP based features in the processes of image retrieval and classification. They preprocessed the images to extract codebooks of color characteristics. The accuracy they achieved in USPTex is 95.94% using color information feature and LBP based features with a double codebook, using the kNN classifier, and they do not indicate if it were employed a cross-validation scheme. It is interesting to note that working only with gray scale information the Adaptive Hybrid Pattern got a result that is better than the descriptor that uses color information. This indicates that this LBP based descriptor manages to extract texture features that are useful to discriminate among classes.

[Table 24](#) shows the 10 best classification results in USPTex dataset. All the results correspond to the LDA classifier. It is noted that most of the results come from the Adaptive

Table 23 – Classification results in USPTex dataset using kNN, Random Forest and LDA with 10-fold cross validation.

	kNN			RF			LDA		
N	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	AHP	336	92.79 (0.14)	CLBP	128	91.77 (0.35)	AHP	252	96.32 (0.14)
2	CLBP	128	91.17 (0.20)	LETRIST	315	91.15 (0.43)	CLBP	128	93.67 (0.19)
3	DTCWT	50	87.42 (0.20)	DTCWT	86	90.86 (0.33)	DTCWT	62	93.52 (0.17)
4	BSIF	256	85.92 (0.18)	AHP	336	90.84 (0.23)	BSIF	256	92.37 (0.14)
5	LETRIST	413	85.36 (0.17)	LCP ^{re}	36	86.17 (0.25)	LCP	64	92.22 (0.16)
6	LCP ^{re}	36	82.89 (0.19)	CLBP-HFSM	76	85.91 (0.39)	LPQ	256	90.76 (0.28)
7	LPQ	256	80.26 (0.21)	LCP	81	85.57 (0.32)	CLBP-HFSM	76	90.74 (0.19)
8	FLBP	256	78.89 (0.17)	BSIF	128	84.35 (0.49)	LETRIST	315	89.75 (0.27)
9	LFD	276	78.85 (0.21)	SLBP	64	82.75 (0.37)	GLCM	48	89.52 (0.19)
10	URLBP	33	78.83 (0.34)	LFD	276	82.46 (0.42)	SLBP	64	89.00 (0.30)
11	LBP ^{u2}	59	78.34 (0.22)	FLBP	256	81.99 (0.52)	LBP ^{u2}	59	88.74 (0.22)
12	LTP ^{riu2}	20	77.98 (0.25)	Gabor	30	81.91 (0.29)	CN	108	87.68 (0.23)
13	GLCM	48	77.71 (0.31)	LBP ^{u2}	59	81.69 (0.44)	LBP	128	86.75 (0.23)
14	LCP	64	77.17 (0.23)	LBP	64	81.09 (0.42)	LCP ^{re}	36	85.93 (0.27)
15	SLBP	64	75.89 (0.22)	URLBP	33	80.55 (0.41)	URLBP	33	85.79 (0.18)
16	SFTA	21	75.16 (0.27)	LTP ^{riu2}	20	78.89 (0.31)	LTP ^{riu2}	20	85.45 (0.19)
17	Gabor	49	74.67 (0.23)	GLCM	48	78.46 (0.35)	FLBP	256	85.28 (0.30)
18	Fourier	80	74.38 (0.24)	LBP-HF	38	78.00 (0.55)	LBP-HF	38	84.40 (0.25)
19	ALBP	10	74.11 (0.24)	SFTA	21	76.52 (0.44)	SFTA	21	83.98 (0.27)
20	CN	108	73.46 (0.15)	Fourier	80	76.22 (0.33)	Fourier	80	81.86 (0.28)
21	LBP	256	73.08 (0.29)	LPQ	256	75.55 (0.44)	LBPV	10	80.63 (0.29)
22	LBPV	10	71.23 (0.25)	CN	108	74.74 (0.47)	LFD	276	80.20 (0.44)
23	LBP-HF	27	69.00 (0.25)	ALBP	10	74.31 (0.37)	LBP ^{ri}	36	79.16 (0.25)
24	CLBP-HFSM	76	66.46 (0.30)	LBPV	10	69.95 (0.33)	ALBP	10	78.54 (0.26)
25	LBP ^{riu2}	10	60.70 (0.26)	LBP ^{ri}	36	69.63 (0.55)	Gabor	30	73.85 (0.25)
26	LBP ^{ri}	36	57.19 (0.22)	MFS	26	63.74 (0.50)	LBP ^{riu2}	10	71.67 (0.33)
27	MFS	26	49.98 (0.37)	LBP ^{riu2}	10	61.68 (0.42)	UDTCWT	1536	67.23 (0.45)
28	UDTCWT	384	45.03 (0.21)	UDTCWT	384	44.33 (0.59)	Minkowski	142	65.70 (0.31)
29	Minkowski	69	40.85 (0.28)	Minkowski	142	42.43 (0.37)	MFS	26	44.90 (0.50)

Source: Research data.

Hybrid Pattern (AHP) descriptor using a radius of 2 and 3 with 12 points, the first two results have 84 and 89 misclassified elements. The best result with another descriptor is CLBP with 93.52% (0.17) and it has 146 misclassified elements, so there is an important difference in the descriptor used and the number of misclassified elements, because of the nature of this dataset that has a great number of classes and a small number of samples per class.

6.1.11 VisTex dataset

VisTex dataset contains color texture images captured in real world conditions, it is composed of 61 classes with 64 samples per class, the images are of size 64×64 Table 25 shows the classification results in the VisTex dataset, it is presented the best results for the 29 different texture descriptors using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. The best classification results correspond to the DTCWT descriptor across the three classifiers. The best overall result is for LDA with 92.46% (0.14), out of 3904 images DTCWT correctly classified 3610 images and incorrectly classified 294 images. The second overall best result is for Random Forest with 92.39% (0.21), this time it correctly classified 3607

Table 24 – The best 10 classification results in USPTex dataset are obtained using LDA and 10-fold cross validation.

Descriptor	Attributes	Length	Misses
AHP(radius:2, points:12, quantization:7)	252	96.32 (0.14)	84
AHP(radius:2, points:12, quantization:9)	336	96.12 (0.16)	89
AHP(radius:3, points:12, quantization:9)	336	95.28 (0.17)	108
AHP(radius:2, points:8, quantization:5)	120	95.17 (0.13)	111
AHP(radius:2, points:6, quantization:5)	96	95.00 (0.16)	115
AHP(radius:3, points:12, quantization:7)	252	94.86 (0.19)	118
AHP(radius:1, points:8, quantization:5)	120	94.80 (0.10)	119
AHP(radius:1, points:6, quantization:5)	96	94.73 (0.13)	121
CLBP(radius:1, points:6)	128	93.67 (0.19)	146
DTCWT(levels:5)	62	93.52 (0.17)	148

Source: Research data.

images and incorrectly classified 297 images. The best result for kNN is 87.91% (0.19), the features comes from DTCWT with a different number of features than in Random Forest and LDA. Only the DTCWT and GLCM descriptors got an accuracy over 90% using Random Forest and LDA, the rest of the descriptors got lower values.

Looking at the number of attributes, it can be noted that they vary from 8 (LBPV and ALBP in LDA) to 1536 (UDTCWT in LDA), this indicates that most of the descriptors are working in local regions, which in the case of LBP-based descriptors means working with neighboring points around one pixel. The Fourier descriptor does not present a value for the LDA classifier because, as in the Midrib dataset, the texture features present missing values and LDA throws an exception in its execution.

In the literature most of the works used different versions of VisTex dataset, for example a version of VisTex with 25 classes and 16 samples of 128×128 making a total of 400 samples got an accuracy of 84.6% using kNN (DASH; JENA, 2017), another version of VisTex uses 54 classes with 16 samples of 128×128 making a total of 864 samples got an accuracy of 98.14% using LDA (JUNIOR; BACKES, 2016).

6.2 Global Comparison

Table 26 shows the classification results in every texture dataset studied in this work, for each dataset it is presented 5 texture descriptors that produced the best classification results using the classifiers kNN, Random Forest and LDA with 10-fold cross validation. From the results obtained, it can be noted that most of the texture descriptors are developed to get good results in benchmark datasets .

The descriptors CLBP and AHP are present in most of the results in each dataset, this indicates that with an appropriate setting of parameters, the discriminative power of these

Table 25 – Classification results in VisTex dataset using kNN, Random Forest and LDA with 10-fold cross validation.

N	kNN			RF			LDA		
	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy	Descriptor	Length	Accuracy
1	DTCWT	38	87.91 (0.19)	DTCWT	62	92.39 (0.21)	DTCWT	62	92.46 (0.14)
2	GLCM	48	82.98 (0.17)	FLBP	256	87.16 (0.34)	GLCM	48	90.20 (0.11)
3	AHP	96	82.21 (0.29)	CLBP	128	86.87 (0.29)	FLBP	256	89.73 (0.19)
4	Fourier	80	81.44 (0.25)	AHP	96	86.64 (0.24)	LPQ	256	89.06 (0.20)
5	CLBP	128	81.00 (0.31)	LBP ^{u2}	59	85.97 (0.41)	SLBP	256	88.27 (0.20)
6	Gabor	35	79.16 (0.21)	SLBP	256	85.90 (0.29)	LBP	256	87.93 (0.23)
7	LPQ	256	77.83 (0.27)	CLBP-HFSM	46	85.66 (0.32)	AHP	96	87.67 (0.14)
8	FLBP	256	77.47 (0.26)	Gabor	25	85.54 (0.23)	CLBP	128	87.34 (0.15)
9	LTP ^{riu2}	16	76.94 (0.24)	GLCM	48	85.15 (0.24)	LBP ^{u2}	59	86.67 (0.14)
10	LBP ^{u2}	59	76.44 (0.22)	LBP	128	84.89 (0.32)	LCP	64	85.11 (0.16)
11	SLBP	64	76.38 (0.20)	LCP	49	84.44 (0.28)	CLBP-HFSM	46	84.88 (0.18)
12	CN	108	75.91 (0.23)	Fourier	80	82.72 (0.24)	CN	108	82.41 (0.21)
13	LBPV	8	73.34 (0.27)	URLBP	33	81.15 (0.24)	URLBP	33	82.17 (0.16)
14	BSIF	32	72.07 (0.21)	LTP ^{riu2}	16	81.04 (0.37)	Gabor	35	81.66 (0.25)
15	LBP	64	71.79 (0.23)	CN	108	79.07 (0.40)	BSIF	32	81.21 (0.15)
16	URLBP	33	71.33 (0.31)	BSIF	32	78.91 (0.35)	LBP-HF	27	78.81 (0.25)
17	SFTA	45	68.16 (0.26)	LBP-HF	27	78.51 (0.26)	LTP ^{riu2}	16	78.10 (0.15)
18	ALBP	8	67.66 (0.33)	LBPV	8	78.49 (0.24)	SFTA	45	75.97 (0.23)
19	LBP-HF	27	66.27 (0.27)	SFTA	45	75.88 (0.34)	LBPV	8	73.35 (0.18)
20	LCP	64	64.80 (0.20)	LPQ	256	74.23 (0.60)	ALBP	8	73.05 (0.23)
21	LBP ^{riu2}	8	62.99 (0.32)	ALBP	8	73.84 (0.31)	LETRIST	315	72.68 (0.31)
22	LFD	46	62.09 (0.21)	LBP ^{ri}	14	72.16 (0.35)	LBP ^{ri}	14	70.23 (0.23)
23	LBP ^{ri}	14	61.30 (0.39)	LCP ^{re}	36	71.35 (0.35)	LCP ^{re}	36	69.96 (0.29)
24	CLBP-HFSM	76	60.15 (0.32)	LETRIST	315	71.22 (0.47)	LBP ^{riu2}	8	68.27 (0.22)
25	LETRIST	352	57.48 (0.29)	LFD	46	70.83 (0.38)	LFD	46	67.98 (0.24)
26	MFS	26	56.57 (0.21)	LBP ^{riu2}	8	69.11 (0.32)	UDTCWT	1536	66.93 (0.41)
27	LCP ^{re}	36	55.36 (0.33)	MFS	26	66.40 (0.38)	Minkowski	142	53.49 (0.26)
28	Minkowski	142	42.58 (0.27)	Minkowski	69	46.61 (0.44)	MFS	26	51.38 (0.30)
29	UDTCWT	384	41.74 (0.31)	UDTCWT	384	44.61 (0.55)	Fourier	-	-

Source: Research data.

descriptors improves, and so they manage to extract texture features to discriminate better the classes of a dataset. With respect to the classifiers, it is noted that the LDA and kNN classifier present better results in most texture datasets. For the previous tables, it is noticed that the classifiers influences the accuracy of the classification.

From the number of misses or misclassified images in each dataset, most of the descriptors got good results to classify the Brodatz, Kylberg, Outex-10 and UIUC benchmark datasets, these datasets are gray scale. Instead, the datasets 1200Tex, CURET, KTH-TIPS2b, STex and USPTex have a great number of misclassified images, and they are color datasets. The color information could improve the classification results, although as seen in the previous results of each dataset, the results using texture in gray scale are close to or better than those using texture and color.

It can be noted that there is a relationship between the size of the images in the datasets and the number of attributes of the feature vector. This indicates that according to the size of the images it is necessary to cover a larger or smaller area to extract texture characteristics. For example, in datasets such as Brodatz, KTH-TIPS2b, Outex-10, USPTex and VisTex their number of features in most cases is lower than in datasets such as UIUC and Midrib.

Table 26 – Comparison of Texture Descriptors and Texture Datasets

N	Dataset	Descriptor	Accuracy	Length	Classifier	Images	Classes	Misses
1	1200Tex	AHP	87.45 (0.36)	432	LDA	1200	20	151
		CN	84.90 (0.33)	108	LDA			181
		DTCWT	83.51 (0.32)	86	LDA			198
		CLBP	83.04 (0.42)	128	LDA			204
		AHP	81.66 (0.44)	324	RF			220
2	Brodatz	CLBP	98.76 (0.09)	128	LDA	1110	111	14
		AHP	98.46 (0.11)	252	LDA			17
		FLBP	98.36 (0.13)	256	LDA			18
		SLBP	98.32 (0.19)	256	LDA			19
		LETRIST	98.28 (0.16)	352	LDA			19
3	CURET	AHP	98.06 (0.07)	324	kNN	5612	61	109
		DTCWT	97.73 (0.05)	62	kNN			127
		AHP	97.26 (0.13)	432	RF			154
		CLBP	96.99 (0.13)	200	kNN			169
		BSIF	96.63 (0.13)	256	kNN			189
4	Kylberg	CLBP	100.00 (0.01)	200	kNN	4480	28	0
		CLBP	99.97 (0.02)	200	RF			1
		LCP ^{re}	99.96 (0.01)	36	kNN			2
		UDTCWT	99.95 (0.02)	384	RF			2
		UDTCWT	99.94 (0.02)	384	kNN			3
5	KTH-TIPS2b	BSIF	98.12 (0.08)	256	kNN	4752	11	89
		FLBP	98.02 (0.09)	256	kNN			94
		SLBP	97.83 (0.12)	256	kNN			103
		DTCWT	97.82 (0.09)	62	kNN			104
		LBP	97.82 (0.09)	256	kNN			104
6	Midrib	CLBP	92.62 (0.35)	648	kNN	606	50	45
		CLBP	88.44 (0.65)	200	LDA			70
		CLBP	87.19 (0.64)	648	RF			78
		URLBP	86.86 (0.50)	243	kNN			80
		LBP ^{u2}	86.43 (0.58)	243	kNN			82
7	Outex-10	LFD	99.98 (0.02)	276	kNN	4320	24	1
		BSIF	99.98 (0.02)	256	kNN			1
		LPQ	99.96 (0.02)	256	kNN			2
		FLBP	99.94 (0.02)	256	kNN			3
		DTCWT	99.92 (0.02)	38	kNN			3
8	STex	LPQ	87.46 (0.08)	256	LDA	7616	476	955
		BSIF	87.40 (0.11)	256	LDA			960
		CLBP	86.81 (0.14)	128	kNN			1005
		BSIF	86.62 (0.14)	256	kNN			1019
		AHP	86.11 (0.14)	252	kNN			1058
9	UIUC	LETRIST	98.25 (0.22)	413	LDA	1000	25	17
		LETRIST	98.02 (0.18)	413	kNN			20
		CLBP	97.99 (0.25)	392	LDA			20
		CLBP	97.57 (0.18)	392	kNN			24
		AHP	97.56 (0.17)	624	LDA			24
10	USPTex	AHP	96.32 (0.14)	252	LDA	2292	191	84
		CLBP	93.67 (0.19)	128	LDA			145
		DTCWT	93.52 (0.17)	62	LDA			149
		AHP	92.79 (0.14)	336	kNN			165
		BSIF	92.37 (0.14)	256	LDA			175
11	VisTex	DTCWT	92.46 (0.14)	62	LDA	3904	61	294
		DTCWT	92.39 (0.21)	62	RF			297
		GLCM	90.20 (0.11)	48	LDA			383
		FLBP	89.73 (0.19)	256	LDA			401
		LPQ	89.06 (0.20)	256	LDA			427

6.3 Final Considerations

In this work, the features obtained from each texture descriptor influence the classification accuracy.

According to the size of the images, for texture images of small size using a local neighborhood it is enough to get good classification results, as is the case of CLBP using a radius of 1 and 6 points got its best result in USPTex. In datasets with large images, like those found in the UIUC dataset, using a large neighborhood we were able to get better classification results, as happened with AHP that got its best result using a radius of 4 and 24 points in UIUC.

The size of the samples in a dataset affect the execution of the descriptors, many texture descriptor failed to extract features of the Midrib dataset, but work without problems in benchmark datasets whose image sizes are smaller, typically around of 200×200 . This is because of the algorithms did not were designed to work with large images.

The time taken to extract features influences the upcoming processes of practical applications of image processing or computer vision, thus, there is a growing interest to get efficient implementations of texture descriptors, such as parallel versions.

Making use of cross-validation implies an increase in the computational cost of the classification process, however, the k Nearest Neighbors and LDA classifiers provide results in a less time than Random Forest and they are used with their default parameters, and they proved their discriminative power in the experiments.

The LDA classifier with a large number of features, for example a feature vector of length 4096, fails to perform the classification with the default setup of available memory in Weka (approximately 3.5GB) throwing an exception of Java Heap Space. One way to solve this is to increase the maximum amount of the memory heap. In the application the memory heap for Java was set to 12GB, thus ensuring that all classifiers work without memory problems when dealing with feature vectors of large size.

The next chapter presents the conclusions gained from this work, and ideas for future improvements to this work.

CONCLUSIONS

Texture analysis is an area of continuous research and study in computer vision. This research presented how a web based platform can help to find texture descriptors that provide the best discriminative power when extracting features over a texture dataset.

The result of this research is a web based tool focused on texture analysis to evaluate texture descriptors and texture datasets. A web based application is a sensible approach to gather and share data and information about texture analysis, and it can greatly contribute to the scientific community.

The platform allows the researcher to perform experiments in texture classification, the platform records the classification results for a posterior review of the classification metrics as a result of the interaction of texture descriptors, texture datasets and classifiers.

In addition to having the classification results we have the configuration of the parameters that were made for each experiment, in this way the researcher can review previous configurations and adapt these configurations for new experiments and thus analyze new texture descriptors or new image datasets.

Every time a researcher devises and develops a texture descriptor, the researcher can use this tool to compare the performance of its descriptor with other descriptors present in the platform and evaluate if it is better or worse than others.

Among the benefits produced by the platform we have the data stored about the interrelationship of texture descriptors, texture datasets and classifiers. All of this data can be applied in subsequent research in texture analysis, focusing the work in characteristics of those texture descriptors that better discriminate the sample images of a class.

Among the classifiers used, the LDA and kNN classifiers shows better results in most texture datasets, using them with their standard values and without parameter customization, even the time taken to classify is lesser than employing a Random Forest classifier.

7.1 Contributions

The main contribution of this work was the development and implementation of the web-based application platform for the analysis and comparison of texture descriptors and texture datasets. The contributions reached throughout the development of this work are the following:

- A collaborative tool, that allows the uploading of source code and datasets, in which the source code and the datasets are available to use, but not to download.
- A web application that simplifies the work in texture analysis, giving the researcher tools to study, examine and compare texture descriptors.
- The ability to compare the discriminative power of texture descriptors over texture datasets using different classifiers.
- It allows the optimization of texture methods exploring the best parameters and verifying its behavior on the dataset.
- A general contribution for the area it is to provide a tool that presents which is the best texture descriptor for a specific dataset.
- An interactive confusion matrix that complements the set of metrics in which the researcher can visually investigate the images that were correctly and incorrectly classified.
- The creation of a database that contains data related to texture classification that can be used in future studies of texture analysis.
- In the future, we intend to make the platform available for use in the community

7.2 Ideas for future works

This platform has the potential to grow in different directions, some interesting proposals for the continuity of this research are the following:

- The inclusion of image processing operations over the texture dataset before the feature extraction process, operations like the use of filters, noise removal, or resize of images.
- The preprocessing of the texture dataset in order to extract some values that influence the following feature extraction process.
- The addition of post processing tasks over the set of texture features before the classification process, tasks such as those to reduce the dimensionality of the texture feature, or to normalize the values of the features.

- In some image datasets is not needed to process the whole image, so it would be useful the integration of processes to compute automatically or select manually a region of interest to work on specific regions of the image.
- Combine different texture descriptors or use a descriptor in a multi-scale approach and test how it influences the classification results.
- In an educational environment, a similar approach can be taken to create a pipeline to make image processing, where all the components are visually created and the results are shown in real time.
- Create an API (Application Programming Interface) for the platform, so a researcher can access the data of the application through code.

BIBLIOGRAPHY

AHONEN, T.; MATAS, J.; HE, C.; PIETIKÄINEN, M. Rotation invariant image description with local binary pattern histogram fourier features. In: SPRINGER. **Scandinavian Conference on Image Analysis**. [S.l.], 2009. p. 61–70. Citation on page 74.

AHONEN, T.; RAHTU, E.; OJANSIVU, V.; HEIKKILA, J. Recognition of blurred faces using local phase quantization. In: IEEE. **Pattern Recognition, 2008. ICPR 2008. 19th International Conference on**. [S.l.], 2008. p. 1–4. Citation on page 74.

ANANTRASIRICHA, N.; BURN, J.; BULL, D. R. Robust texture features for blurred images using undecimated dual-tree complex wavelets. In: **2014 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2014. p. 5696–5700. ISSN 1522-4880. Citations on pages 42 and 74.

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Color texture analysis based on fractal descriptors. **Pattern Recognition**, v. 45, n. 5, p. 1984 – 1992, 2012. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320311004614>>. Citations on pages 34 and 52.

_____. Texture analysis and classification: A complex network-based approach. **Information Sciences**, v. 219, p. 168 – 180, 2013. ISSN 0020-0255. Available: <<http://www.sciencedirect.com/science/article/pii/S0020025512004677>>. Citations on pages 42, 74, and 76.

BACKES, A. R.; JUNIOR, J. J. de M. S. Lbp maps for improving fractal based texture classification. **Neurocomputing**, v. 266, p. 1 – 7, 2017. ISSN 0925-2312. Available: <<http://www.sciencedirect.com/science/article/pii/S0925231217308366>>. Citation on page 80.

BARLEY, A.; TOWN, C. Combinations of feature descriptors for texture image classification. **Journal of Data Analysis and Information Processing**, Scientific Research Publishing, v. 2014, 2014. Citations on pages 34 and 35.

BARNATHAN, M.; ZHANG, J.; MEGALOOIKONOMOU, V. A web-accessible framework for the automated storage and texture analysis of biomedical images. In: IEEE. **Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on**. [S.l.], 2008. p. 257–259. Citation on page 26.

BERTOLINI, D.; OLIVEIRA, L.; JUSTINO, E.; SABOURIN, R. Texture-based descriptors for writer identification and verification. **Expert Systems with Applications**, v. 40, n. 6, p. 2069 – 2080, 2013. ISSN 0957-4174. Available: <<http://www.sciencedirect.com/science/article/pii/S095741741201130X>>. Citation on page 25.

BHARATI, M. H.; LIU, J.; MACGREGOR, J. F. Image texture analysis: methods and comparisons. **Chemometrics and Intelligent Laboratory Systems**, v. 72, n. 1, p. 57 – 71, 2004. ISSN 0169-7439. Available: <<http://www.sciencedirect.com/science/article/pii/S0169743904000528>>. Citation on page 41.

BIANCONI, F.; BELLO-CEREZO, R.; NAPOLETANO, P.; MARIA, F. D. Improved opponent colour local binary patterns for colour texture classification. In: BIANCO, S.; SCHETTINI, R.; TRÉMEAU, A.; TOMINAGA, S. (Ed.). **Computational Color Imaging**. Cham: Springer International Publishing, 2017. p. 272–281. ISBN 978-3-319-56010-6. Citation on page 87.

BIANCONI, F.; GONZÁLEZ, E.; FERNÁNDEZ, A.; SAETTA, S. A. Automatic classification of granite tiles through colour and texture features. **Expert Systems with Applications**, v. 39, n. 12, p. 11212 – 11218, 2012. ISSN 0957-4174. Available: <http://www.sciencedirect.com/science/article/pii/S0957417412005751>. Citation on page 25.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Available: <https://doi.org/10.1023/A:1010933404324>. Citation on page 43.

BRODATZ, P. **Textures: a photographic album for artists and designers**. [S.l.]: Dover Publications Inc., 1966. Citation on page 47.

CAPUTO, B.; HAYMAN, E.; MALLIKARJUNA, P. Class-specific material categorisation. In: IEEE. **Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on**. [S.l.], 2005. v. 2, p. 1597–1604. Citation on page 49.

CASANOVA, D.; FLORINDO, J.; FALVO, M.; BRUNO, O. Texture analysis using fractal descriptors estimated by the mutual interference of color channels. **Information Sciences**, v. 346-347, p. 58 – 72, 2016. ISSN 0020-0255. Available: <http://www.sciencedirect.com/science/article/pii/S0020025516300135>. Citation on page 34.

CASTAÑÓN, C. A.; FRAGA, J. S.; FERNANDEZ, S.; GRUBER, A.; COSTA, L. d. F. Biological shape characterization for automatic image recognition and diagnosis of protozoan parasites of the genus eimeria. **Pattern Recognition**, ELSEVIER, v. 40, n. 7, p. 1899–1910, 2007. Citation on page 26.

CERNADAS, E.; FERNÁNDEZ-DELGADO, M.; GONZÁLEZ-RUFINO, E.; CARRIÓN, P. Influence of normalization and color space to color texture classification. **Pattern Recognition**, v. 61, p. 120 – 138, 2017. ISSN 0031-3203. Available: <http://www.sciencedirect.com/science/article/pii/S0031320316301510>. Citation on page 82.

CHOI, J. Y.; RO, Y. M.; PLATANIOTIS, K. N. Color local texture features for color face recognition. **IEEE transactions on image processing**, IEEE, v. 21, n. 3, p. 1366–1380, 2012. Citation on page 33.

CHOWDHURY, S.; VERMA, B.; STOCKWELL, D. A novel texture feature based multiple classifier technique for roadside vegetation classification. **Expert Systems with Applications**, v. 42, n. 12, p. 5047 – 5055, 2015. ISSN 0957-4174. Available: <http://www.sciencedirect.com/science/article/pii/S0957417415001542>. Citation on page 25.

CIMPOI, M.; MAJI, S.; KOKKINOS, I.; MOHAMED, S.; ; VEDALDI, A. Describing textures in the wild. In: **Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2014. Citation on page 45.

Computer Vision Laboratory. **CUReT, Columbia-Utrecht Reflectance and Texture Database**. 2016. Accessed: 29-October-2017. Available: <http://www.cs.columbia.edu/CAVE/software/curet/>. Citation on page 47.

COSTA, A. F.; HUMPIRE, G.; TRAINA, A. J. M. An efficient algorithm for fractal analysis of textures. In: **2012 25th SIBGRAPI Conference on Graphics, Patterns and Images**. [S.l.: s.n.], 2012. p. 39–46. ISSN 1530-1834. Citation on page 74.

DALCIMAR, C.; JOACI de M. S. J. J.; MARTINEZ, B. O. Plant leaf identification using gabor wavelets. **International Journal of Imaging Systems and Technology**, v. 19, n. 3, p. 236–243, 2009. Available: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.20201>>. Citations on pages 46 and 76.

DANA, K. J.; GINNEKEN, B. V.; NAYAR, S. K.; KOENDERINK, J. J. Reflectance and texture of real-world surfaces. **ACM Transactions On Graphics (TOG)**, ACM, v. 18, n. 1, p. 1–34, 1999. Citation on page 47.

DASH, S.; JENA, U. R. Multi-resolution laws' masks based texture classification. **Journal of Applied Research and Technology**, v. 15, n. 6, p. 571 – 582, 2017. ISSN 1665-6423. Available: <<http://www.sciencedirect.com/science/article/pii/S1665642317301049>>. Citation on page 103.

DONG, W.; WU, F.; KONG, Y.; MEI, X.; LEE, T. Y.; ZHANG, X. Image retargeting by texture-aware synthesis. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 2, p. 1088–1101, Feb 2016. ISSN 1077-2626. Citation on page 26.

DORETTO, G.; CHIUSO, A.; WU, Y. N.; SOATTO, S. Dynamic textures. **International Journal of Computer Vision**, v. 51, n. 2, p. 91–109, Feb 2003. ISSN 1573-1405. Available: <<https://doi.org/10.1023/A:1021669406132>>. Citation on page 33.

FAGETTE, A.; COURTY, N.; RACOCEANU, D.; DUFOUR, J.-Y. Unsupervised dense crowd detection by multiscale texture analysis. **Pattern Recognition Letters**, v. 44, p. 126 – 133, 2014. ISSN 0167-8655. Pattern Recognition and Crowd Analysis. Available: <<http://www.sciencedirect.com/science/article/pii/S0167865513003565>>. Citation on page 29.

FELICIANO, F. F.; LETA, F. R.; MAINIER, F. B. Texture digital analysis for corrosion monitoring. **Corrosion Science**, v. 93, p. 138 – 147, 2015. ISSN 0010-938X. Available: <<http://www.sciencedirect.com/science/article/pii/S0010938X15000256>>. Citations on pages 25 and 29.

FLORINDO, J.; BRUNO, O. Texture analysis by fractal descriptors over the wavelet domain using a best basis decomposition. **Physica A: Statistical Mechanics and its Applications**, v. 444, p. 415 – 427, 2016. ISSN 0378-4371. Available: <<http://www.sciencedirect.com/science/article/pii/S0378437115008778>>. Citation on page 80.

FLORINDO, J. B.; ASSIRATI, L.; BRUNO, O. M. Locally enhancing fractal descriptors by using the non-additive entropy. **Pattern Recognition Letters**, v. 70, p. 32 – 37, 2016. ISSN 0167-8655. Available: <<http://www.sciencedirect.com/science/article/pii/S0167865515003888>>. Citation on page 76.

FLORINDO, J. B.; BRUNO, O. M. Discrete schroedinger transform for texture recognition. **Information Sciences**, v. 415-416, p. 142 – 155, 2017. ISSN 0020-0255. Available: <<http://www.sciencedirect.com/science/article/pii/S0020025516319855>>. Citation on page 76.

_____. Texture classification using non-euclidean minkowski dilation. **Physica A: Statistical Mechanics and its Applications**, v. 493, p. 189 – 202, 2018. ISSN 0378-4371. Available: <<http://www.sciencedirect.com/science/article/pii/S0378437117310221>>. Citation on page 73.

FLORINDO, J. B.; BRUNO, O. M.; LANDINI, G. Morphological classification of odontogenic keratocysts using bouligand–minkowski fractal descriptors. **Computers in Biology and Medicine**, v. 81, p. 1 – 10, 2017. ISSN 0010-4825. Available: <http://www.sciencedirect.com/science/article/pii/S0010482516303171>. Citation on page 29.

FLORINDO, J. B.; CASANOVA, D.; BRUNO, O. M. A gaussian pyramid approach to bouligand–minkowski fractal descriptors. **Information Sciences**, p. –, 2018. ISSN 0020-0255. Available: <https://www.sciencedirect.com/science/article/pii/S0020025518304043>. Citations on pages 99 and 101.

FLORINDO, J. B.; SIKORA, M. S.; PEREIRA, E. C.; BRUNO, O. M. Characterization of nanostructured material images using fractal descriptors. **Physica A: Statistical Mechanics and its Applications**, v. 392, n. 7, p. 1694 – 1701, 2013. ISSN 0378-4371. Available: <http://www.sciencedirect.com/science/article/pii/S0378437112009855>. Citation on page 29.

FLORINDO, J. B.; SILVA, N. R. da; ROMUALDO, L. M.; SILVA, F. de Fátima da; LUZ, P. H. de C.; HERLING, V. R.; BRUNO, O. M. Brachiaria species identification using imaging techniques based on fractal descriptors. **Computers and Electronics in Agriculture**, v. 103, p. 48 – 54, 2014. ISSN 0168-1699. Available: <http://www.sciencedirect.com/science/article/pii/S0168169914000404>. Citation on page 29.

FRANCIS, S. V.; SASIKALA, M.; BHARATHI, G. B.; JAIPURKAR, S. D. Breast cancer detection in rotational thermography images using texture features. **Infrared Physics & Technology**, v. 67, p. 490 – 496, 2014. ISSN 1350-4495. Available: <http://www.sciencedirect.com/science/article/pii/S1350449514001789>. Citations on pages 25 and 29.

GARCIA, I.; GUZMÁN-RAMÍREZ, E.; PACHECO, C. Colfdimap: A web-based tool for teaching of fpga-based digital image processing in undergraduate courses. **Computer Applications in Engineering Education**, v. 23, n. 1, p. 92–108, 2015. ISSN 1099-0542. Available: <http://dx.doi.org/10.1002/cae.21581>. Citation on page 26.

GONÇALVES, W. N. **Análise de texturas estáticas e dinâmicas e suas aplicações em biologia e nanotecnologia**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2013. Citation on page 33.

GUO, W.; YOU, X.; ZHU, Z.; XUE, W.; YU, S.; JIANG, X. Dynamic texture synthesis via image reconstruction. In: **Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on**. [S.l.: s.n.], 2015. p. 1907–1911. Citation on page 26.

GUO, Y.; ZHAO, G.; PIETIKÄINEN, M. Texture classification using a linear configuration model based descriptor. In: CITESEER. **BMVC**. [S.l.], 2011. p. 1–10. Citation on page 74.

GUO, Z.; ZHANG, L.; ZHANG, D. A completed modeling of local binary pattern operator for texture classification. **IEEE Transactions on Image Processing**, IEEE, v. 19, n. 6, p. 1657–1663, 2010. Citations on pages 41 and 74.

_____. Rotation invariant texture classification using lbp variance (lbpv) with global matching. **Pattern Recognition**, v. 43, n. 3, p. 706 – 719, 2010. ISSN 0031-3203. Available: <http://www.sciencedirect.com/science/article/pii/S0031320309003331>. Citation on page 74.

GUO, Z.; ZHANG, L.; ZHANG, D.; ZHANG, S. Rotation invariant texture classification using adaptive lbp with directional statistical features. In: **2010 IEEE International Conference on Image Processing**. [S.l.: s.n.], 2010. p. 285–288. ISSN 1522-4880. Citation on page 74.

HANBAY, K.; ALPASLAN, N.; TALU, M. F.; HANBAY, D. Principal curvatures based rotation invariant algorithms for efficient texture classification. **Neurocomputing**, v. 199, p. 77 – 89, 2016. ISSN 0925-2312. Available: <<http://www.sciencedirect.com/science/article/pii/S0925231216300522>>. Citation on page 84.

HANNAD, Y.; SIDDIQI, I.; KETTANI, M. E. Y. E. Writer identification using texture descriptors of handwritten fragments. **Expert Systems with Applications**, v. 47, p. 14 – 22, 2016. ISSN 0957-4174. Available: <<http://www.sciencedirect.com/science/article/pii/S0957417415007551>>. Citation on page 25.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472. Citation on page 30.

HIEMATH, P.; BHUSNURMATH, R. A. Multiresolution ldbp descriptors for texture classification using anisotropic diffusion with an application to wood texture analysis. **Pattern Recognition Letters**, v. 89, p. 8 – 17, 2017. ISSN 0167-8655. Available: <<http://www.sciencedirect.com/science/article/pii/S0167865517300235>>. Citation on page 84.

HOSSAIN, S.; SERIKAWA, S. Texture databases - a comprehensive survey. **Pattern Recognition Letters**, v. 34, n. 15, p. 2007 – 2022, 2013. ISSN 0167-8655. Smart Approaches for Human Action Recognition. Available: <<http://www.sciencedirect.com/science/article/pii/S0167865513000536>>. Citations on pages 45, 47, 50, and 53.

HU, Y.; LONG, Z.; ALREGIB, G. Scale selective extended local binary pattern for texture classification. In: **The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, USA**. [S.l.: s.n.], 2017. Citation on page 39.

HUMPHREYS, J. Texture analysis: Macrotexture, microtexture and orientation mapping. **Journal of Microscopy**, Wiley Online Library, v. 203, n. 2, p. 231–232, 2001. Citation on page 34.

IAKOVIDIS, D. K.; KERAMIDAS, E. G.; MAROULIS, D. Fuzzy local binary patterns for ultrasound texture characterization. In: SPRINGER. **International conference image analysis and recognition**. [S.l.], 2008. p. 750–759. Citation on page 74.

JL, L.; REN, Y.; PU, X.; LIU, G. Median local ternary patterns optimized with rotation-invariant uniform-three mapping for noisy texture classification. **Pattern Recognition**, v. 79, p. 387 – 401, 2018. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320318300682>>. Citations on pages 87 and 98.

JULESZ, B. Visual pattern discrimination. **IRE transactions on Information Theory**, IEEE, v. 8, n. 2, p. 84–92, 1962. Citation on page 30.

_____. Textons, the elements of texture perception, and their interactions. **Nature**, Nature Publishing Group, v. 290, n. 5802, p. 91, 1981. Citation on page 30.

JUNIOR, J. J. de M. S.; BACKES, A. R. Elm based signature for texture classification. **Pattern Recognition**, v. 51, p. 395 – 401, 2016. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320315003453>>. Citation on page 103.

KANDASWAMY, U.; SCHUCKERS, S. A.; ADJEROH, D. Comparison of texture analysis schemes under nonideal conditions. **IEEE Transactions on Image Processing**, v. 20, n. 8, p. 2260–2275, Aug 2011. ISSN 1057-7149. Citations on pages 34 and 45.

KANNALA, J.; RAHTU, E. Bsif: Binarized statistical image features. In: **Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)**. [S.l.: s.n.], 2012. p. 1363–1366. ISSN 1051-4651. Citation on page 74.

KIM, N. C.; SO, H. J. Directional statistical gabor features for texture classification. **Pattern Recognition Letters**, v. 112, p. 18 – 26, 2018. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865518301855>. Citation on page 96.

KYLBERG, G. **The Kylberg Texture Dataset v. 1.0**. [S.l.], 2011. Available: <http://www.cb.uu.se/~gustaf/texture/>. Citation on page 48.

KYLBERG, G.; SINTORN, I.-M. Evaluation of noise robustness for local binary pattern descriptors in texture classification. **EURASIP Journal on Image and Video Processing**, Springer, v. 2013, n. 1, p. 17, 2013. Citation on page 74.

LANDY, M.; MOVSHON, J. A. The plenoptic function and the elements of early vision. In: _____. **Computational Models of Visual Processing**. MIT Press, 1991. p. 3–20. ISBN 9780262290890. Available: <https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6277403>. Citation on page 30.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 27, n. 8, p. 1265–1278, Aug 2005. ISSN 0162-8828. Citation on page 52.

LI, T.; ZHU, S.; OGIHARA, M. Using discriminant analysis for multi-class classification: an experimental investigation. **Knowledge and information systems**, Springer, v. 10, n. 4, p. 453–472, 2006. Citation on page 43.

LIU, P.; GUO, J.-M.; CHAMNONGTHAI, K.; PRASETYO, H. Fusion of color histogram and lbp-based features for texture image retrieval and classification. **Information Sciences**, v. 390, p. 95 – 111, 2017. ISSN 0020-0255. Available: <http://www.sciencedirect.com/science/article/pii/S0020025517301159>. Citation on page 101.

MAANI, R.; KALRA, S.; YANG, Y.-H. Noise robust rotation invariant features for texture classification. **Pattern Recognition**, v. 46, n. 8, p. 2103 – 2116, 2013. ISSN 0031-3203. Available: <http://www.sciencedirect.com/science/article/pii/S0031320313000344>. Citation on page 74.

Machine Vision and Media Processing. **Outex Texture Database**. 2007. Accessed: 29-October-2017. Available: <http://www.outex.oulu.fi/index.php?page=classification>. Citation on page 51.

Macmillan. **Macmillan Dictionary Texture Definition**. 2018. Accessed: 2-May-2018. Available: <https://www.macmillandictionary.com/dictionary/american/texture>. Citation on page 30.

MARIEB, E. N.; WILHELM, P. B.; MALLATT, J. **Human anatomy**. [S.l.]: Pearson, 2014. Citations on pages 25 and 29.

MARQUES, O. **Practical image and video processing using MATLAB**. [S.l.]: John Wiley & Sons, 2011. Citations on pages 29 and 39.

MARSLAND, S. **Machine learning: an algorithmic perspective**. 2nd. ed. [S.l.]: CRC press, 2015. Citation on page 43.

MATERKA, A.; STRZELECKI, M. *et al.* Texture analysis methods—a review. **Technical university of lodz, institute of electronics, COST B11 report, Brussels**, p. 9–11, 1998. Citation on page 31.

MEHTA, R.; EGIASARIAN, K. O. Rotated local binary pattern (rlbp) - rotation invariant texture descriptor. In: MARSICO, M. D.; FRED, A. L. N. (Ed.). **ICPRAM**. SciTePress, 2013. p. 497–502. ISBN 978-989-8565-41-9. Available: <<http://dblp.uni-trier.de/db/conf/icpram/icpram2013.html#MehtaE13>>. Citation on page 74.

MäENPää, T.; PIETIKÄINEN, M. Classification with color and texture: jointly or separately? **Pattern Recognition**, v. 37, n. 8, p. 1629 – 1640, 2004. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320303004321>>. Citation on page 34.

MIRMEHDI, M.; XIE, X.; SURI, J. **Handbook of Texture Analysis**. [S.l.]: Imperial College Press, 2008. Citation on page 40.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. Sebastopol, CA: O'Reilly Media, 2017. Citations on pages 43 and 44.

Multimedia Signal Processing and Security Lab. **Salzburg Texture Image Database (STex)**. 2008. Accessed: 29-October-2017. Available: <<http://wavelab.at/sources/STex/>>. Citation on page 51.

NIXON, M. **Feature extraction & image processing**. [S.l.]: Academic Press, 2008. Citation on page 39.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. **Pattern Recognition**, v. 29, n. 1, p. 51 – 59, 1996. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320395000674>>. Citations on pages 40 and 74.

OJALA, T.; PIETIKÄINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 7, p. 971–987, Jul 2002. ISSN 0162-8828. Citation on page 74.

Oxford Press. **Oxford Dictionaries Texture Definition**. 2018. Accessed: 24-April-2018. Available: <<https://en.oxforddictionaries.com/definition/texture>>. Citation on page 31.

PAN, Z.; LI, Z.; FAN, H.; WU, X. Feature based local binary pattern for rotation invariant texture classification. **Expert Systems with Applications**, v. 88, p. 238 – 248, 2017. ISSN 0957-4174. Available: <<http://www.sciencedirect.com/science/article/pii/S0957417417304761>>. Citations on pages 82 and 98.

PATHAK, B.; BAROOAH, D. Texture analysis based on the gray-level co-occurrence matrix considering possible orientations. **International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering**, v. 2, n. 9, p. 4206–4212, 2013. Citation on page 74.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. *et al.* Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, n. Oct, p. 2825–2830, 2011. Citation on page 69.

PETROU, M.; SEVILLA, P. G. **Image processing: dealing with texture**. [S.l.: s.n.], 2006. Citations on pages 31 and 32.

PIETIKÄINEN, M.; HADID, A.; ZHAO, G.; AHONEN, T. **Computer vision using local binary patterns**. [S.l.]: Springer Science & Business Media, 2011. Citation on page 39.

PIETIKÄINEN, M.; OJALA, T.; XU, Z. Rotation-invariant texture classification using feature distributions. **Pattern Recognition**, Elsevier, v. 33, n. 1, p. 43–52, 2000. Citation on page 74.

PINTOR, J.; CARRIÓN, P.; CERNADAS, E.; GONZÁLEZ-RUFINO, E.; FORMELLA, A.; FERNÁNDEZ-DELGADO, M.; DOMÍGUEZ-PETIT, R.; RÁBADE-UBEROS, S. Govocitos: A software tool for estimating fish fecundity based on digital analysis of histological images. **Computers and Electronics in Agriculture**, v. 125, p. 89 – 98, 2016. ISSN 0168-1699. Available: <http://www.sciencedirect.com/science/article/pii/S0168169916301387>. Citation on page 25.

Poetry Foundation. **Texture**. 2007. Accessed: 2-May-2018. Available: <https://www.poetryfoundation.org/harriet/2007/08/texture>. Citation on page 30.

RANDEN, T. **Brodatz Textures**. 2007. Accessed: 29-October-2017. Available: <http://www.uu.no/~tranden/brodatz.html>. Citation on page 47.

RIBAS, L. C. **Análise de texturas dinâmicas baseada em sistemas complexos**. Phd Thesis (PhD Thesis) — Universidad de São Paulo, 2017. Citations on pages 33 and 34.

RICHARDS, M. **Software architecture patterns**. [S.l.]: O'Reilly Media, Incorporated, 2015. Citation on page 57.

ROMUALDO, L.; LUZ, P.; DEVECHIO, F.; MARIN, M.; ZÚÑIGA, A.; BRUNO, O.; HERLING, V. Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. **Computers and Electronics in Agriculture**, v. 104, p. 63 – 70, 2014. ISSN 0168-1699. Available: <http://www.sciencedirect.com/science/article/pii/S0168169914000696>. Citation on page 29.

ROY, S. K.; CHANDA, B.; CHAUDHURI, B. B.; BANERJEE, S.; GHOSH, D. K.; DUBEY, S. R. Local directional zigzag pattern: A rotation invariant descriptor for texture classification. **Pattern Recognition Letters**, v. 108, p. 23 – 30, 2018. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865518300709>. Citation on page 94.

RUSS, J. C. **The image processing handbook**. 7th. ed. [S.l.]: CRC press, 2016. Citation on page 31.

SALWA, L.; MOHAMMED, R. Novel phase-based descriptor using bispectrum for texture classification. **Pattern Recognition Letters**, v. 100, p. 1 – 5, 2017. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865517303410>. Citation on page 94.

SANDID, F.; DOUIK, A. Robust color texture descriptor for material recognition. **Pattern Recognition Letters**, v. 80, p. 15 – 23, 2016. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865516300885>. Citation on page 33.

SCHMIDT-JONES, C. **The basic elements of music**. [S.l.]: Connexions, 2012. Citation on page 30.

Scientific Computing Group. **1200Tex Dataset**. 2012. Accessed: 29-October-2017. Available: <http://scg.ifsc.usp.br/dataset/1200Tex.php>. Citation on page 46.

_____. **Midrib Dataset**. 2012. Accessed: 29-October-2017. Available: <<http://scg.ifsc.usp.br/dataset/Midrib.php>>. Citation on page 50.

_____. **USPTex Dataset**. 2012. Accessed: 29-October-2017. Available: <<http://scg.ifsc.usp.br/dataset/USPtex.php>>. Citation on page 52.

SELVAN, S.; DEVI, S. S. Automatic seed point selection in ultrasound echography images of breast using texture features. **Biocybernetics and Biomedical Engineering**, v. 35, n. 3, p. 157 – 168, 2015. ISSN 0208-5216. Available: <<http://www.sciencedirect.com/science/article/pii/S0208521614000734>>. Citation on page 25.

SILVA, N. R. D.; OLIVEIRA, M. W. da S.; FILHO, H. A. de A.; PINHEIRO, L. F. S.; ROSSATTO, D. R.; KOLB, R. M.; BRUNO, O. M. Leaf epidermis images for robust identification of plants. **Scientific reports**, Nature Publishing Group, v. 6, p. 25994, 2016. Citation on page 74.

SILVA, N. R. da; FLORINDO, J. ao B.; GÓMEZ, M. C.; ROSSATTO, D. R.; KOLB, R. M.; BRUNO, O. M. Plant identification based on leaf midrib cross-section images using fractal descriptors. **Plos One**, 2015. Citations on pages 42, 50, 74, and 90.

SILVA, N. Rosa da; RIDDER, M. D.; BAETENS, J. M.; BULCKE, J. Van den; ROUSSEAU, M.; BRUNO, O. M.; BEECKMAN, H.; ACKER, J. V.; BAETS, B. D. Automated classification of wood transverse cross-section micro-imagery from 77 commercial central-african timber species. **Annals of Forest Science**, v. 74, n. 2, p. 30, Apr 2017. ISSN 1297-966X. Available: <<https://doi.org/10.1007/s13595-017-0619-0>>. Citation on page 29.

SIMON, P.; UMA, V. Review of texture descriptors for texture classification. In: SATAPATHY, S. C.; BHATEJA, V.; RAJU, K. S.; JANAKIRAMAIAH, B. (Ed.). **Data Engineering and Intelligent Computing**. Singapore: Springer Singapore, 2018. p. 159–176. ISBN 978-981-10-3223-3. Citations on pages 39 and 40.

SOLEM, A. **Celery. Distributed Task Queue**. 2018. Accessed: 11-June-2018. Available: <<http://www.celeryproject.org/>>. Citation on page 69.

SONG, T.; LI, H.; MENG, F.; WU, Q.; LUO, B.; ZENG, B.; GABBOUJ, M. Noise-robust texture description using local contrast patterns via global measures. **IEEE Signal Processing Letters**, IEEE, v. 21, n. 1, p. 93–96, 2014. Citation on page 74.

SONG, T.; LI, H.; MENG, F.; WU, Q.; CAI, J. Letrist: Locally encoded transform feature histogram for rotation-invariant texture classification. **IEEE Transactions on Circuits and Systems for Video Technology**, PP, n. 99, p. 1–1, 2017. ISSN 1051-8215. Citation on page 74.

TAN, X.; TRIGGS, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. **IEEE Transactions on Image Processing**, v. 19, n. 6, p. 1635–1650, June 2010. ISSN 1057-7149. Citation on page 74.

TANG, G.; XIAO, Z.; LIU, Q.; LIU, H. A novel airport detection method via line segment classification and texture classification. **IEEE Geoscience and Remote Sensing Letters**, IEEE, v. 12, n. 12, p. 2408–2412, 2015. Citation on page 25.

TEOREY, T. J.; BUXTON, S.; FRYMAN, L.; GÜTING, R. H.; HALPIN, T.; HARRINGTON, J. L.; INMON, W. H.; LIGHTSTONE, S. S.; MELTON, J.; MORGAN, T. *et al.* **Database design: know it all**. [S.l.]: Morgan Kaufmann, 2008. Citation on page 59.

THAKARE, V. S.; PATIL, N. N.; SONAWANE, J. S. Survey on image texture classification techniques. **International Journal of Advancements in Technology**, v. 4, n. 1, p. 97–104, 2013. Citation on page 35.

The Ponce Group. **Datasets for Computer Vision Research**. 2006. Accessed: 29-October-2017. Available: <http://www-cvr.ai.uiuc.edu/ponce_grp/data/>. Citation on page 52.

TIMO, O.; T., M.; M., P.; J., V.; J., K.; S., H. Outex - new framework for empirical evaluation of texture analysis algorithms. In: . [S.l.: s.n.], 2002. Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, 1:701 - 706. Citation on page 50.

TRAMONTAN, L.; POLETTI, E.; FIORIN, D.; RUGGERI, A. A web-based system for the quantitative and reproducible assessment of clinical indexes from the retinal vasculature. **IEEE Transactions on Biomedical Engineering**, v. 58, n. 3, p. 818–821, March 2011. ISSN 0018-9294. Citation on page 26.

TUCERYAN, M.; JAIN, A. K. **The Handbook of Pattern Recognition and Computer Vision**. 2nd. ed. [S.l.]: World Scientific Publishing Co., 1998. Citation on page 40.

UHL, A.; WIMMER, G. A systematic evaluation of the scale invariance of texture recognition methods. **Pattern Analysis and Applications**, Springer, v. 18, n. 4, p. 945–969, 2015. Citation on page 39.

Unsplash. **Strawberry**. 2018. Accessed: 23-May-2018. Available: <<https://unsplash.com/photos/Fff5ve-rGVA>>. Citation on page 33.

Vision and Modeling Group. **Vision Texture**. 1995. Accessed: 29-October-2017. Available: <<http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>>. Citation on page 53.

WANG, X.; DING, X.; LIU, C. Gabor filters-based feature extraction for character recognition. **Pattern Recognition**, v. 38, n. 3, p. 369 – 379, 2005. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320304003012>>. Citations on pages 41 and 74.

WEI, L.; HONG-YING, D. Real-time road congestion detection based on image texture analysis. **Procedia Engineering**, v. 137, p. 196 – 201, 2016. ISSN 1877-7058. Green Intelligent Transportation System and Safety. Available: <<http://www.sciencedirect.com/science/article/pii/S1877705816002630>>. Citation on page 25.

WEI, L.-Y.; LEVOY, M. Fast texture synthesis using tree-structured vector quantization. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. **Proceedings of the 27th annual conference on Computer graphics and interactive techniques**. [S.l.], 2000. p. 479–488. Citation on page 35.

XU, Y.; JI, H.; FERMÜLLER, C. Viewpoint invariant texture description using fractal analysis. **International Journal of Computer Vision**, v. 83, n. 1, p. 85–100, Jun 2009. ISSN 1573-1405. Available: <<https://doi.org/10.1007/s11263-009-0220-6>>. Citation on page 74.

ZHAO, G.; AHONEN, T.; MATAS, J.; PIETIKAINEN, M. Rotation-invariant image and video description with local binary pattern features. **IEEE Transactions on Image Processing**, v. 21, n. 4, p. 1465–1477, April 2012. ISSN 1057-7149. Citation on page 74.

ZHU, Z.; YOU, X.; CHEN, C. P.; TAO, D.; OU, W.; JIANG, X.; ZOU, J. An adaptive hybrid pattern for noise-robust texture analysis. **Pattern Recognition**, v. 48, n. 8, p. 2592 – 2608, 2015. ISSN 0031-3203. Available: <<http://www.sciencedirect.com/science/article/pii/S0031320315000023>>. Citations on pages 41, 74, and 76.