Sélection d'attributs multi-espace à partir de motifs binaires locaux pour la classification de textures couleur

Vinh Truong Hoang

▶ **To cite this version:**

Vinh Truong Hoang. Sélection d'attributs multi-espace à partir de motifs binaires locaux pour la classification de textures couleur. Signal and Image Processing. Université du Littoral Côte d'Opale, 2018. English. NNT : 2018DUNK0468 . tel-01756931

HAL Id: tel-01756931

https://theses.hal.science/tel-01756931

Submitted on 3 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'ordre

École doctorale SPI - Université Lille Nord-De-France

THÈSE

pour l'obtention du grade de

Docteur de l'Université du Littoral Côte d'Opale

Discipline : Traitement du Signal et des Images

Vinh TRUONG HOANG

Présentée et soutenue publiquement le 15 Février 2018

Multi color space LBP-based feature selection for texture classification

Devant le jury composé de:

M. **Fadi DORNAIKA**	Professeur à l'Université du Pays Basque	Rapporteur
M. **Jean-Christophe BURIE**	Professeur à l'Université de La Rochelle	Rapporteur
M. **Olivier ALATA**	Professeur à l'Université Jean Monnet de Saint-Etinenne	Président du jury
M. **Denis HAMAD**	Professeur à l'Université du Littoral Côte d'Opale	Directeur de thèse
Mme. **Alice POREBSKI**	Maître de conférences à l'Université du Littoral Côte d'Opale	Co-Encadrante de thèse
M. **Nicolas VANDENBROUKE**	Maître de conférences - HDR à l'Université du Littoral Côte d'Opale	Co-Encadrant de thèse

Laboratoire d'Informatique, Signal et Image de la Côte d'Opale – EA 4491
Maison de la Recherche Blaise Pascal – Centre Universitaire de la Mi-Voix
50 rue Ferdinand Buisson – B.P. 719, 62228 Calais Cedex, France

Contents

Acknowledgment **i**

Notations **iii**

Abbreviations **v**

Introduction **7**

1 Color texture classification **11**

1.1 Color texture representation 12

1.1.1 Color spaces 12

1.1.2 Color texture 16

1.1.3 Color texture features 18

1.1.4 Local binary patterns 24

1.2 Supervised texture classification 32

1.2.1 Context 32

1.2.2 Classifiers 35

1.2.3 Evaluation methods 38

1.2.4 Benchmark color texture image databases 39

1.2.5 Review of the considered databases 44

1.3 Conclusion 47

2 Feature selection **49**

2.1 Taxonomy of feature selection methods 50

2.1.1 Context of feature selection 53

2.1.2 Feature selection evaluation 55

2.2 Data and knowledge representation 57

2.2.1 Graph data representation 58

2.2.2 Sparse graph construction . . . 59
2.3 Ranking-based approaches . . . 61
2.3.1 Ranking based on scores . . . 61
2.3.2 Ranking based on weighting algorithms . . . 66
2.4 Feature selection applied to LBP . . . 71
2.4.1 LBP bin selection . . . 71
2.4.2 LBP histogram selection . . . 73
2.5 Conclusion . . . 79

3 Multi color space LBP-based features selection 81
3.1 Introduction . . . 82
3.2 Color space combination . . . 83
3.2.1 Single color space approach . . . 83
3.2.2 Multi color space approach . . . 87
3.3 Multi color space LBP selection . . . 92
3.3.1 Considered color spaces . . . 92
3.3.2 Candidate color texture descriptors . . . 92
3.3.3 Dimensionality reduction . . . 95
3.4 Multi color space histogram selection . . . 95
3.5 Multi color space bin selection . . . 98
3.5.1 Occurrence based ranking . . . 100
3.5.2 Sparsity based ranking . . . 101
3.5.3 Bin selection procedure . . . 103
3.5.4 Results . . . 103
3.6 Combination of histogram ranking and bin selection . . . 105
3.7 Conclusion . . . 110

4 Experimental evaluation 111
4.1 Impact of similarity measure . . . 115
4.2 Impact of the histogram score . . . 116
4.3 Impact of the LBP-based feature selection strategy . . . 117
4.3.1 Classification results detailed on the New BarkTex . . . 117
4.3.2 Validation on STex, OuTex-TC-00013 and USPTex sets . . . 121
4.3.3 Processing times . . . 125
4.4 Conclusion . . . 127

Conclusion and perspectives 131

Appendices 135

A A summary of studies on color texture classification **137**

B Application of LBP-based features for lace images classification **145**

C Supplementary results **151**

Bibliography **155**

Publications **179**

List of Tables **180**

List of Figures **181**

Abstract **184**

Résumé **185**

Acknowledgment

I would like to express my gratitude here to many people who have been helping me during my thesis work since 2014. This thesis could not be accomplished without their help.

First of all, I would like to greatly thank my director of the thesis, Prof. Denis HAMAD, for offering me the opportunity to work in his research team and supporting me with his instructive guidance during the whole thesis work. His guidance inspires me to see problems from different perspectives.

I am also greatly thankful to my co-supervisors of the thesis, Dr. Nicolas VANDENBROUKE, and Dr. Alice POREBSKI for their valuable advice during my thesis work. It has been an honor working with them.

I would like to express my special thanks to Prof. Fadi DORNAIKA and Prof. Jean-Christophe BURIE for their precious time and hard work to review my thesis, and giving me valuable remarks to improve it. Special thanks to Prof. Olivier ALATA as well for examining my thesis.

I would also like to thanks all my lab colleagues at LISIC and friends for their interaction and friendly support during those years: Christopher Jankee, Hayder Hendi, Ahmad Mazyad, Ali Darwich, Cyril Briquet, Hong Phan, Samah Hijazi, Hiba Toufic, Tse Muugii, Yves Dugény, Dominique Walz, Tam Nguyen, Thong Vu, Tuong Huynh, Chau Nguyen, Sarah Bertrand, Rebhi Ali and Faten Sandid.

Most of all, I am so grateful to my family, for their love and support. They accompany and encourage me no matter what happens.

The last words go to Bach Tuyet. She supports me already, for many years, and I hope she will do that forever.

Notations

ASL^r	ASL-score of rth histogram
C	Number of classes
C_1	Color component C_1
C_2	Color component C_2
C_3	Color component C_3
$\mathcal{C}$	Set of cannot-link constraints
$\mathcal{D}$	Number of candidates features
$\mathbf{D}$	Diagonal matrix
D_{int}	Histogram instersection distance
D_{Jef}	Jeffrey distance
D_{χ^2}	χ^2 distance
δ	Number of candidates histograms
$Fisher^r$	Fisher score of rth feature
$\mathbf{f}^r$	r^{th} feature vector
$\mathbf{f}^{rc}$	r^{th} feature vector of the class c
G	Undirected graph
$\mathcal{H}_i$	Concatenated histogram to represent image I_i
$\mathbf{H}_i^r$	r^{th} histogram of color image I_i
$H_i^r(k)$	k^{th} bin of histogram $\mathbf{H}_i^r$
I_i	i^{th} of color image I_i
ICS^r	ICS-score of rth histogram
J_i	Domimant bins set
J_{global}	Domimant bins set for the whole dataset
N	Number of images
N_c	Number of images in the class c
N_S	Number of color spaces c

$\mathbf{L}$	Laplacian matrix with $\mathbf{L} = \mathbf{D} - \mathbf{S}$
$Laplacian^r$	Laplacian score of rth feature
$NM(\mathbf{x}_i)$	nearmiss of $\mathbf{x}_i$
$NH(\mathbf{x}_i)$	nearhit of $\mathbf{x}_i$
μ_r	Mean of rth feature
μ_r^c	Mean of rth feature within the class c
$\mathcal{M}$	Set of must-link constraints
$\mathcal{P}$	Number of LBP neighbor pixels
Q	Quantization level
$\mathcal{R}$	Radius size of LBP
ρ	Hypothesis margin
$\mathbf{S}$	Similarity matrix
$\mathbf{S}^c$	Similarity matrix of data within class c
σ	Gaussian variance values
$SpASL^r$	SpASL-score of rth histogram
$SupLaplacian^r$	Supervised Laplacian score of rth feature
$Variance^r$	Variance score of rth feature
$UnsupSparse^r$	Unsupervised sparsity score of rth feature
$SupSparse^r$	Supervised sparsity score of rth feature
$\mathbf{X}$	Data matrix, $\mathbf{X} = \{x_i^r\}_{i=1...N}^{r=1...\mathcal{D}}$
x_i^r	r^{th} feature value of the ith color image I_i
$\mathbf{Y}$	Labels of the data
y_i	Label of x_i

Abbreviations

ASL	Adapted Supervised Laplacian
BS	Bin Selection
DLBP	Dominant Local Binary Pattern
DFT	Discrete Fourier Transform
disLBP	discriminant Local Binary Pattern
EOCLBP	Extend Opponent Color Local Binary Pattern
FCBF	Fast Correlation-Based Filtering
FSC	Fisher Separation Criterion
FS	Feature Selection
HS	Histogram Selection
HOG	Histogram of Oriented Gradients
ICS	Intra-Class Similarity
K-NN	*K*-Nearest Neighbor
LTP	Local Ternary Pattern
LBP	Local Binary Pattern
LBP-HF	Local Binary Pattern Histogram Fourier
LPP	Locality Preserving Projections
LBP^{u2}	LBP uniform pattern
LBP^{ri}	LBP rotation invariant pattern
LBP^{riu2}	LBP rotation invariant uniform pattern
LDA	Linear Discriminant Analysis
MCOLBP	Mixed Color Order Local Binary Pattern

MCSBS	Multi Color Space Bin Selection
MCSHS	Multi Color Space Histogram Selection
MCWS	Multi Color Space Without Selection
OCLBP	Opponent Color Local Binary Pattern
PCA	Principal Component Analysis
QLBP	Quaternionic Local Binary Pattern
RSCCM	Reduced Size Chromatic Co-occurrence Matrices
SIFT	Scale Invariant Feature Transform
SpASL	Sparse Adapted Supervised Laplacian
SR	Sparse Representation
SFFS	Sequential Forward Floating Selection
SVM	Support Vector Machines
SFS	Sequential Forward Selection
SBS	Sequential Backward Selection
3D-ASDH	Three-Dimensional Adaptive Sum and Difference Histograms

Introduction

Texture analysis is one of the major topics in the field of computer vision and has many important applications including face recognition, object detection, image filtering, segmentation and content-based access to image databases [1]. Texture classification can be defined as a step to assign a texture image into one of a set of predefined categories. This step requires to define a efficient descriptors in order to represent and discriminate the different texture classes.

In the past decades, texture analysis has been extensively studied and a wide variety of description approaches have been proposed. Among these approaches, Local Binary Pattern (LBP) proposed by Ojala et al. is known as one of the most successful statistical approaches due to its efficacy, robustness against illumination intensity changes and relative fast calculation [2]. It has been successfully applied to the applications as diverse as texture classification. In order to encode LBP, the gray level of each pixel is compared with those of its neighbors and the results of these comparisons are weighted and summed in order to give a binary number. The obtained texture features is the LBP histogram whose bin count depends on the number of neighbors. However, when the number of considered neighboring pixels increases, the feature dimensionality will increase exponentially.

Texture analysis methods and descriptor like LBP were firstly designed for dealing with gray-scale images. Otherwise, it has been demonstrated that color information is very important to represent the texture, especially natural textures [3]. Several extensions of LBP to color have been proposed and this leads to consider several LBP histograms to represent a texture. The number of feature is so increase but only some of which are relevant for texture classification [4]. That is the reason why many approaches have been proposed to reduce the dimension of the feature space based on the LBP histogram in order to improve the classification performances. The following section presents the problem and goals of the thesis.

Problems and goals

Usually, color images are acquired by devices that code the colors in the *RGB* color space. However, the color of pixels can be represented in different color spaces which respect different properties. Many authors have compared the classification performances reached by these color spaces in order to determine the "best" color space for texture analysis [5]. Nevertheless, the synthesis of these works illustrates that the choice of the color space depends on the considered texture images. Moreover, the prior determination of a color space which is well suited to the considered class discrimination is still an open question.

We propose to consider a multi color space approach designed for color texture classification. Instead of searching the best color space for color image analysis, this approach propose to combine different color spaces in order to improve the performances reached by classification schemes. It consists in selecting, among a set of color texture features extracted from images coded in different color spaces, those which are the most discriminating for the considered color textures.

Feature selection is a procedure of finding a set of most compact and informative original features [6]. It is known that some features describing a dataset can be either redundant or irrelevant. They can decrease the performances of the classification. Moreover, removing such useless features generally allow to decrease the processing time of the classifiers.

Thus, applying color LBP in multiple color spaces will tend to produce high-dimensional feature vectors with irrelevant features. Thus, a dimensionality reduction method is needed to address this problem. Many authors have attempted to obtain more discriminative, robust and compact LBP-based features in order to reduce feature space dimensionality. The first strategy consists in identifying the most informative pattern groups based on some rules or the predefinition of patterns of interest. The second strategy consists in applying feature selection methods in order to find the most discriminative patterns [7].

Porebski et al. firstly proposed a different approach which selects the most discriminant whole LBP histograms [8]. In this approach, the most discriminant LBP histograms are selected in their entirety, out of the different LBP histograms extracted from a color texture. Recently, Kalakech et al. propose to adapt the supervised Laplacian score used in the literature for feature ranking and selection, to select and rank histograms in the supervised context, namely "Adapted Supervised Laplacian" (ASL-score) [9].

Contributions

In this thesis, we mainly focus on the LBP histogram and bin selection approaches in a multi color space framework. Our contributions are summarized as follows.

- The first contribution is the proposed novel histogram score, namely “Sparse Adapted Supervised Laplacian” (SpASL-score). We propose to extend the ASL-score to SpASL-score by using a sparse representation.

- Our second contribution is the extension of the LBP histogram selection proposed by Porebski et al. for a single color space to multiple color space. Applying a multi color space strategy avoids the difficulty of choosing a relevant color space. However, the LBP descriptor produce high dimensional feature vectors, especially when several color spaces is combined to represent the texture. A selection method helps to reduce the number of LBP histograms in order to reduce the processing times and improve the classification.

- The third contribution is the extension to color of the LBP histogram bin selection proposed by Guo et al. for a gray level analysis to color by applying a multi color space strategy [10]. Instead of selecting the most discriminating histograms, this approach selects the most discriminate LBP histograms bin.

- The fourth contribution is the introduction of the combination of bin and histogram selection in the multi color space framework. The histogram selection step selects the whole relevant histograms which are used for the classification step. We observe that, the selected histogram might contains some redundant or irrelevant LBP histogram bins and the filtered histogram might contains some informative bins. This approach is achieved by firstly applying a ranking histogram method and then by applying a bin selection procedure.

Structure of the thesis

The remainder of this thesis is organized as follows:

The first chapter introduces several notions related to the color texture classification. The principal families of color spaces are presented as well as the main color descriptors. The LBP descriptors and its extensions to color are briefly reviewed. We introduced the general framework of texture classification in different learning context. We focus on the supervised color texture classification by presenting the most commonly classifier used in this context. Next, several popular benchmarks datasets are introduced for color texture classification tasks that will be used to carry out experiments in the following chapters.

The second chapter summarizes basic principles of the feature selection by introducing its general framework. The categorization of feature selection methods which are based on the

evaluation strategies and the learning context are then discussed. Several notations and terminologies used over the thesis are provided in this chapter. Two main ranking-based approaches are briefly reviewed. These approaches can be achieved by associating a score for each feature or by applying an algorithm which gives weights for a feature subset. Then, two feature selection approaches applied to LBP are discussed: LBP bin selection and LBP histogram selection. This chapter also presents our first contribution by introducing the SpASL-score for histogram selection.

The third chapter is devoted to our main contribution of this thesis. The contribution of the multi color space strategies for texture classification are firstly presented. The extensions of histogram selection and bin selection to a multi color space framework is then introduced. Next, two novel strategies of combination of histogram ranking and bin selection are proposed. The results of these approaches are first illustrated on a benchmark color texture database in a single and multiple color spaces.

The comparison of the color texture classification results reached by different proposed approaches of bin selection and histogram selection are presented in chapter 4. The experimental evaluation is carried out on four benchmark texture databases in the multi color space framework. We first study the impact of the distance used for measuring the similarity between two histograms during the computation of the proposed SpASL-score. The results obtained by this score is then compared with other scores. Next, the results of different proposed approaches are compared with the results obtained in the state-of-the-art.

Finally, we conclude this thesis by highlighting our contributions and discussing possible future work.

Chapter 1

Color texture classification

Contents

1.1 Color texture representation 12

1.1.1 Color spaces 12

1.1.2 Color texture 16

1.1.3 Color texture features 18

1.1.4 Local binary patterns 24

1.2 Supervised texture classification 32

1.2.1 Context 32

1.2.2 Classifiers 35

1.2.3 Evaluation methods 38

1.2.4 Benchmark color texture image databases 39

1.2.5 Review of the considered databases 44

1.3 Conclusion 47

This chapter provides some background concepts for color texture classification. Section 1.1 outlines the color texture representation with the introduction of color spaces, the definition of color texture and the way to characterize it thanks to color texture features. Several notions of classification are then introduced in section 1.2: a brief review of the most commonly used classifiers and evaluation methods is provided in this section. Four benchmark texture databases usually used in the framework of color texture classification are also presented. Finally, section 1.3 contains a summary of the key topics presented in this chapter.

1.1 Color texture representation

Color is an important element in human vision and so in digital image processing. Color images contain more discriminative informations than grayscale images and it has been demonstrated that considering the color in the images significantly improves the performance for many tasks in pattern recognition [3, 11, 12, 13, 14, 15, 16]. To characterize the color texture, it is necessary to take an interest in the representation of color and texture in the images. This section thus firstly presents the different classical color spaces used in image analysis, the definition of color texture and reviews the main color texture features used in color texture analysis.

1.1.1 Color spaces

Color is the perceptual result of light in the visible region of the electromagnetic spectrum. The human retina has three types of color photoreceptor cells, which respond to incident radiation with somewhat different spectral response curves [17]. Because there are exactly three types of color photoreceptor, three numerical components are necessary and theoretically sufficient to represent a color. A digital color image is thus composed of three component images and the color of each pixel is specified in a three dimensional color space, which defines a color coordinate system. That is the reason why most of color images are acquired by devices that code the colors in the *RGB* space. A point in this space is characterized by the three components of the corresponding pixel which are the red (R), the green (G) and the blue (B). Figure 1.1 illustrates an example of a color image (from the USPTex database that will be introduced in section 1.2.4) with its three corresponding component images. Other color spaces can be computed from the *RGB* space by means of either linear or nonlinear transformations and in the last few years, numerous color spaces, which take into account different physical, physiologic and psycho-visual properties have been defined. They can be grouped into four families [18]:

1. The **primary color spaces** are based on the trichromatic theory, which assumes that it is possible to match any color by mixing appropriate amounts of three primary colors. They can be divided into:

   - the *real primary spaces*, for which the primary colors can be physically reproduced. The *RGB* spaces use the red, the green and the blue as primary colors and are device dependent.
   - the *imaginary primary spaces*, whose primaries do not physically exist. Each of the *RGB* spaces can be transformed into the single CIE *XYZ* virtual primary space which is device independent [19].

   Primary spaces can be normalized by dividing each color component value by the sum

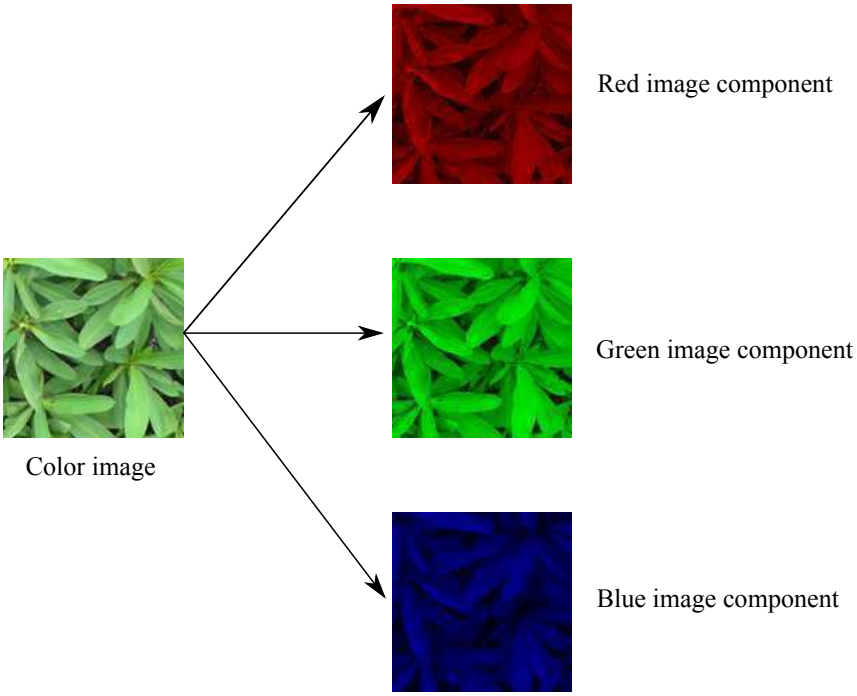


Figure 1.1: An example of a color image coded in the *RGB* color space and its three corresponding component images.

of the three ones in order to obtain normalized coordinate spaces, like the *rgb* [20] and *xyz* [19] color spaces.

2. The **luminance-chrominance color spaces** are composed of one component that represents an achromatic information (here called "luminance") and two components, that quantify a chromatic information (here called "chrominance"). They can be grouped into the following subfamilies:

   - the *antagonist (or opponent color) spaces*, which aim at reproducing the model of the opponent color theory proposed by Hering, like AC_1C_2 [21] and $b_w r_g b_y$ [22].
   - the *television color spaces*, which separate the achromatic signal and the chromatic signals for the television signal transmission, like YIQ, YUV and YC_bC_r [22].
   - the *perceptually uniform color spaces*, which propose a metric to establish a correspondence between a color difference perceived by a human observer and a distance measured in the color space, like $L^*a^*b^*$, $L^*u^*v^*$ and UVW [19].
   - *others luminance-chrominance color spaces*, like *Irg* [23], CIE *Yxy* [19] and the Carron's LCh_1Ch_2 color space [24].

3. The **independent axis color spaces** result from different statistical methods (like Karhunen-Loeve Transform (KLT) or Principal Component Analysis (PCA), which provide the less

correlated components as possible, like the well-known $I_1I_2I_3$ color space proposed by Ohta [23].

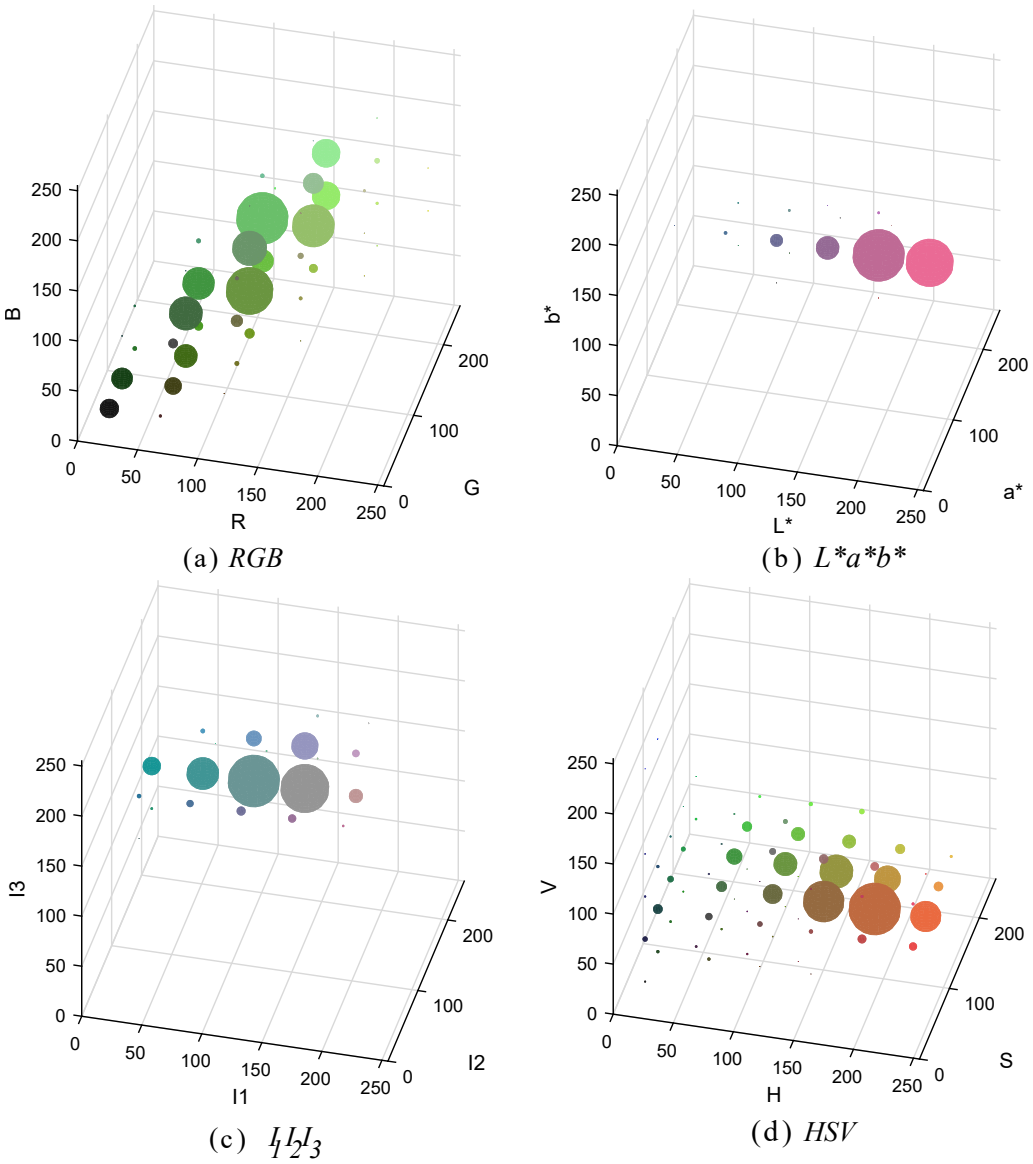
4. The **perceptual color spaces** attempt to quantify the subjective human color perception by using the intensity, the hue and the saturation components. They can be categorized into:

   - the *polar (or cylindrical) coordinate spaces* that correspond to expressions in polar coordinates of the luminance-chrominance components, like $AC_{C_1C_2}h_{C_1C_2}$, $L^*C^*_{ab}h_{ab}$, $b_wC_{r_gb_y}h_{r_gb_y}$, $YC_{IQ}h_{IQ}$, $YC_{UV}h_{UV}$ and $LC_{Ch_1Ch_2}h_{Ch_1Ch_2}$.
   - the *perceptual coordinate spaces* that are directly evaluated from primary color spaces, like the *HSI* triangle model (or *M-HSI* modified triangle model), the *HSV* hexcone model, the *HLS* double hexcone model [25], the *I-HLS* improved *HLS* model [26] and the CIE $L^*S^*_{uv}h_{uv}$ space [19].

(a) *RGB* (b) *L*a*b** (c) $I_1I_2I_3$ (d) *HSV*

Figure 1.2: An example of an image coded in different color spaces.

A color image can thus be represented in these different color spaces. Figure 1.2 illustrates the image of leaves of the figure 1.1 that has been coded in the *RGB* (a), *L*a*b** (b), $I_1I_2I_3$ (c) and *HSV* (d) spaces. Figure 1.3 illustrates the 3D distribution of this image in these color spaces. Under the human vision system, the *RGB* space reflects the leaves in the real world better than the other. In these 3D distributions, each axe represents one of the three color components, coded in this example from 0 to 255. Each pixel of the image is coded by three color component values according to the considered color space. The pixels with same coordinates are grouped thanks to a quantization scheme and represented thanks a ball. The number of pixels belonging to each group is visualized by the size of ball and the color of the ball represents the mean color of the pixels of the group. Visually and graphically, we can see that the different color spaces give different representations of the same image. These differences could be interesting since they allow to analyze a same texture under different viewpoints, that increases the possibilities to accurately represent the color textures.



(a) *RGB* (b) *L*a*b** (c) $I_1I_2I_3$ (d) *HSV*

Figure 1.3: Color distributions of the image of leaves coded in four color spaces.

As, there is a wide range of color spaces with different properties, we have proposed an approach detailed in chapter 3 which uses the properties of several color spaces in the following of this thesis. In this approach, images are first coded in different color spaces, then color texture features are extracted from these so coded images to characterize the texture.

Before representing the features which allow to characterize the color textures in the subsection 1.1.3, we propose to briefly introduce this and its principal properties.

1.1.2 Color texture

1.1.2.1 Definition

Texture image usually contains a material or an object which presents a texture information. Figure 1.4 illustrates an example of four color texture images from the USPTex database that represent textures of stone, brick, sand and granit. So, texture is an important element of human vision.

(a) Texture of stone (b) Texture of brick (c) Texture of sand (d) Texture of granit

Figure 1.4: A variety of textures

The human visual system is extremely effective at characterizing textures using adjectives such as smooth or rough, fine or coarse, granular or fibrous, isotropic or with preferred directions, regular or irregular, contrasting and so on. However, it is difficult to define what is texture while it is easy to identify by a human. There is no definition of a texture that would universally be recognized and accepted, and many definitions have been proposed in the computer vision literature:

- “Texture is a fundamental characteristic of the appearance of virtually all natural surfaces and plays a major role in computer vision systems. Image texture may provide information about the physical properties of objects, such as smoothness or roughness, or differences in surface reflectance, such as color” [27].

- “Texture can be viewed as a global pattern arising from the repetition of local subpatterns” [28].

- “A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic” [29].

- “Texture can be defined as the set of local neighborhood properties of the gray levels of an image region” [30].

- “A color texture can be regarded as a pattern described by the relationship between its chromatic and spatial distribution” [11].

The various definitions of texture leads to a variety of different ways to analyse texture. How to efficiently represent the texture patterns is a fundamental problem in computer vision and this problem depends on the observation conditions of the considered texture.

1.1.2.2 Influence of the observation conditions

The characterization of a texture depends on several parameters including the perception level and the acquisition conditions.

Concerning the perception level, there are in practice two main observational scales on which a texture can be defined [31]:

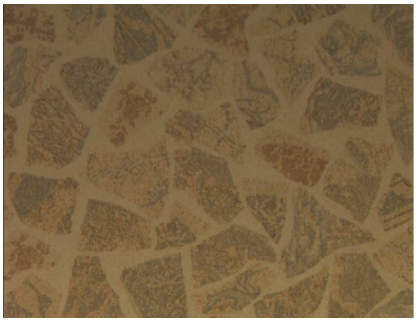
- **Microscopic observation** reveals the irregular or disordered structure of pixel colors within the image. The texture is considered on the reduced neighborhood of a given pixel. It is then defined using a probabilistic or statistical approach. The fine structure of plastic illustrated in Figure 1.5a corresponds to a microscopic texture.

- **Macroscopic observation** involves the concept of either an elementary pattern or a privileged direction. In the presence of an elementary pattern, also known as texton [32], the macroscopic texture is viewed as a repetitive or periodic spatial distribution of this pattern. The macroscopic texture is then defined using a deterministic or structural approach. This is illustrated by the image in figure 1.5b, which shows the plastic image in which the elementary pattern is a block that is repeated in a structured manner.

Besides the perception level influences the characterization of texture, texture information can also depend on the acquisition conditions. Indeed, when the conditions differ from an acquisition to another, the resulting images may reveal different textures. The image in figure 1.5c represents a change in orientation of the texture of the figure 1.5b with a 90-degree rotation. The image in figure 1.5d illustrates an illumination change by using the simulated illumination source 4000K fluorescent TL84, instead of the horizon light source.

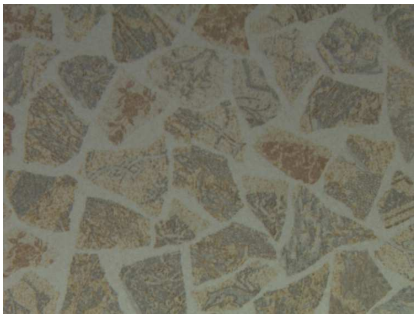
Acquisition conditions and levels of perception may thus influence the choice of the features to be used to describe the texture information. For example, an irregular texture will be better characterized by statistical features, whereas a regular texture will be well described in terms of frequency-based and geometric features. Feature that are invariant to rotation or illumination change can also be relevant in certain applications.

(a) Microscopic observation of plastic



(b) Macroscopic observation of plastic

(c) Change in orientation of plastic



(d) Change in illumination of plastic

Figure 1.5: Different observation conditions of a texture from the OuTex database (that will be introduced in section 1.2.4).

In the next subsection, we propose to review the main color texture features which have been used in the framework of color texture analysis.

1.1.3 Color texture features

Color and texture are two naturally related characteristics of the image, but these characteristics are often analyzed separately. Many authors demonstrate that texture features incorporating color information can improve the discrimination (especially when dealing with natural textures observed under fixed illumination conditions) [11, 12, 33]. According to Mäenpää and Pietikäinen, the color texture analysis can be roughly divided into two categories as seen in the next subsection [13].

1.1.3.1 Color and texture combination

Two main categories of approaches can be considered to combine the color and the texture information: methods that process color and texture information separately and those that consider color and texture jointly [13, 34].

In the first approach, texture features representing the spatial distribution of the luminance

image are used in conjunction with other features describing the color distribution in a given color space [35, 36, 37]. Let us consider the general notation of a given 3D color space (C_1, C_2, C_3). Figure 1.6 illustrates this first approach where color and texture are separately processed.

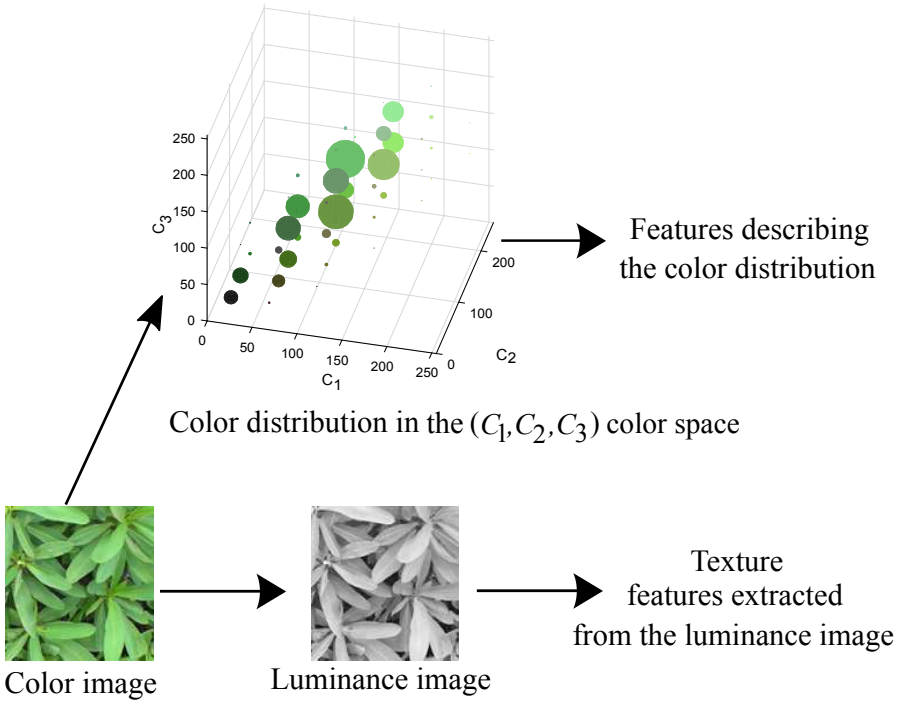


Figure 1.6: Representation of color texture by extracting texture features from the luminance image and by associated them with features describing the color distribution.

There exists other strategies for color and texture combination by incorporating the classifier to compute the similarity between the texture feature vector and the color feature [35].

In the second approach, this is the distribution of the color components of the pixels that is analyzed. It allows to jointly characterized the spatial and the color distribution. In the past few years, several studies have been directed to the problem of joint representation of texture and color and three different strategies have been proposed:

- The first one consists in evaluating the texture features within each color component of an image (within-component relationship) independently, without considering the spatial interactions between the levels of two distinct color components. In this case, the texture features defined for luminance images are applied to each of the three color components independently [38, 39, 35, 40]. Figure 1.7 illustrates this approach.

- The second strategy consists in considering the spatial distribution both within each color component of a given color space and also between these different color components

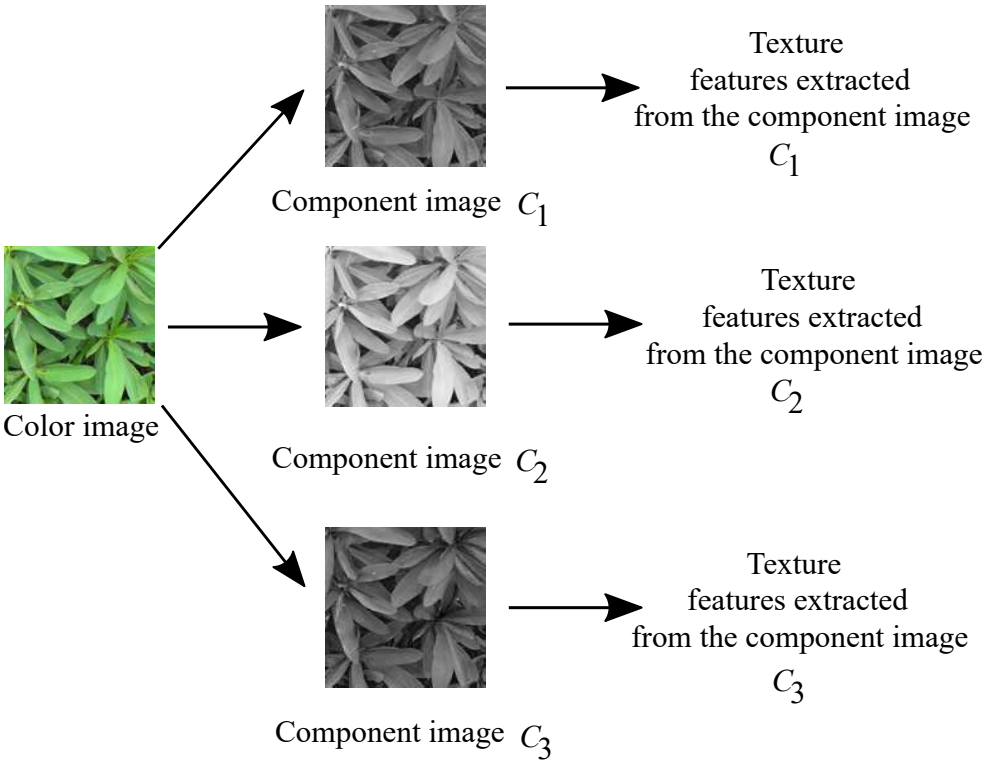


Figure 1.7: Representation of color texture by extracting features that are evaluated within each component of the (C_1, C_2, C_3) space.

(within and between-component relationship) [41, 42, 43, 44, 45]. Figure 1.8 illustrates this strategy. Note that the texture features extracted from the component image C_1 and C_2 are different with those extracted from the component image C_2 and C_1 by a several descriptors [13].

- Finally, the third strategy consists in analyzing the spatial interactions between the colors of pixels, that relies on considering a color order relation that defines the inferiority/superiority between colors [46, 47].

The methods developed in the context of our work are based on the second strategy (figure 1.8). However, this strategy increases the number of attributes, especially when within and between-component relationships are considered with one or more color spaces. The methods that we propose are therefore for the purpose of reducing the dimension of attributes and overcome this disadvantage.

Many features, initially defined for analyzing gray level images, have been extended to one or several strategies of color and texture combination. We propose in the next section to describe the main color texture descriptors, in the general framework of color texture classification, independently of the color and texture combination strategy.

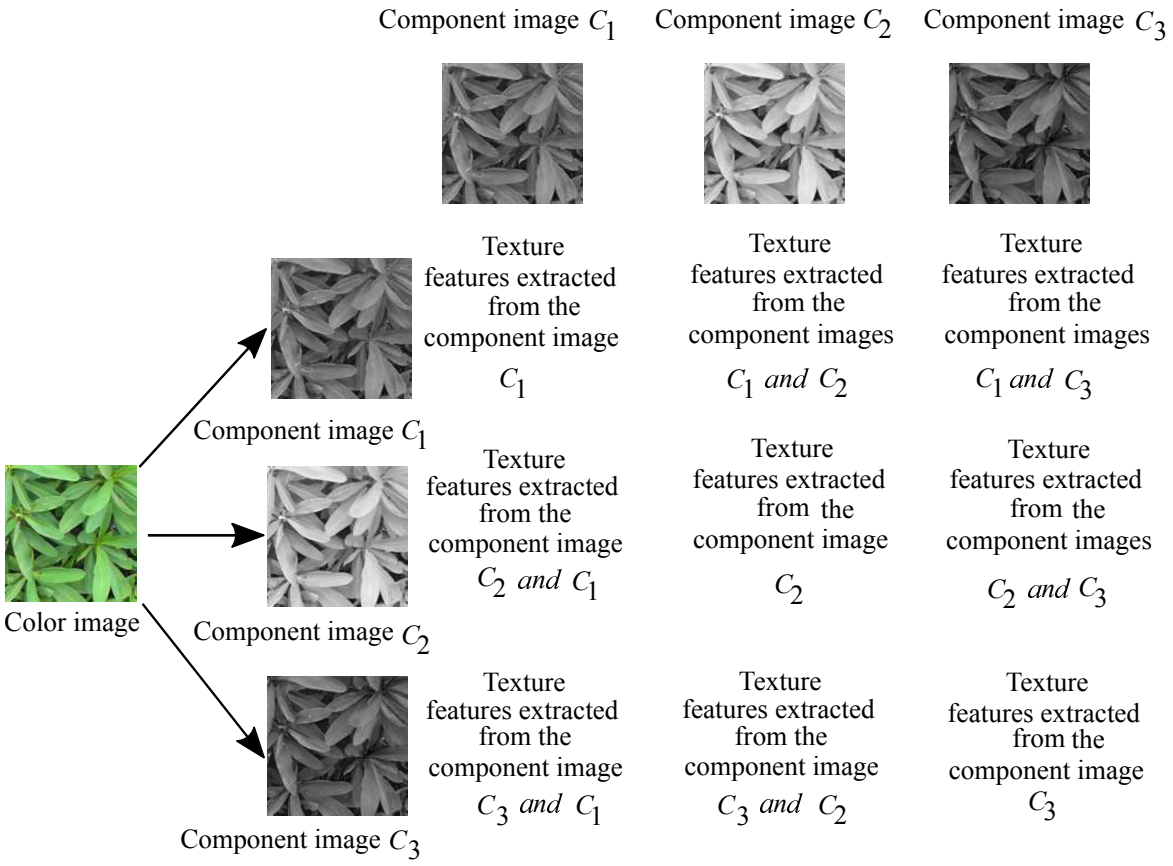


Figure 1.8: Representation of color texture by extracting features that are evaluated within and between the components of the (C_1, C_2, C_3) space.

1.1.3.2 Color texture attributes

Texture analysis has been a topic of intensive research and, over the years, a wide variety of color description approaches for discriminating textures have been proposed. A comprehensive evaluation of color texture attributes is presented in [48, 49]. Typically, texture attributes can be divided into three categories defined in terms of geometrical, spatio-frequential and statistical features [27].

- **Geometrical features**: These features take into account the structural and contextual of the image. They are well adapted to describe the texture observed at the macroscopic level. The geometrical description of a texture includes the extraction of several primitives features (corners, edges, points, lines, curves or surfaces) and the placement rules for those primitives. Zheng et al. proposed a set of specific geometrical features for color textures that are sensitive to the regular texture of meat [50]. However, this type of features does not allow to characterize the irregular textures which are usually found in the natural images.

- **Spatio-frequential features**: The spatio-frequential features can be divided into three

domains: spatial domain, frequency domain and spatio-frequential domain.

1. Spatial domain involves features which aim to characterize the texture in terms of the number of transitions per unit area, in contrast to coarser textures. Cumani, Laplace and Sobel filters are all examples of filters widely used for the detection of color edges [51]. However, this kind of feature has never yet been applied to the classification of color texture images.
2. Fourier transform or discrete cosine transform gives an alternative texture representation that is purely based in the frequency domain. These attributes are particularly well suited to the case of images containing coarse textures, where there is considerable continuity between the levels of each pixel color component. Since high frequencies are restricted to local changes in component levels, it is possible to express all the information present in the image using just a small number of coefficients, corresponding to low frequencies. Drimbarean et al. used the gray level discrete cosine transform and its color extension to characterize textures in their experiments [11].
3. Spatio-frequential domain involves features that combine the two different representations previously discussed. The Gabor transform and the wavelet transform are the most widely used in color texture classification, since they are effective for the analysis of both macrotextures and microtextures. Several authors have investigated the use of Gabor filters on color images and showed that the consideration of color texture features could improve classification results over those obtained through the use of gray level features [11, 12, 52]. However, this method presents two disadvantages: the need to set parameters for the filters and the long computation time to characterize certain textures. Wavelet transform has the advantage of giving a multiscale characterization of a texture by considering both global and local information content within the image. Moreover, wavelet-based features have parallels with the process of human vision, which performs a systematic frequency decomposition of the images falling onto the retina [53]. Several authors have studied problems of color texture classification through the use of the wavelet transform [54, 55]. In spite of the many advantages of this transform, Iakovidis et al. revealed that the features obtained from wavelets are not always the most suitable for texture characterization.

- **Statistical features** can be used to characterize any type of texture. For this kind of feature, a texture is defined in terms of its gray level or color variation in a neighborhood, and it is the relationship between a pixel and its neighbors that is examined. A large range of statistical features are used for color texture classification. These include image

statistics, image histograms, chromatic co-occurrence matrices and sum and difference histograms, local binary patterns. These different descriptors can be grouped into a number of categories based on their order, where the order of the attribute depends on the type of spatial interaction between the pixels in question. Image histograms are an example of first order statistical features. They describe the distribution of color component levels in the input image [56, 13]. In contrast, co-occurrence matrices, for example, consider pairs of pixels, which means that they are second order attributes. This descriptor, introduced by Haralick et al. in 1973, was initially implemented for gray level images [57]. Since the use of color can improve texture classification results, Palm proposed to extend the concept of co-occurrence matrices to color images, starting from the definition of multichannel co-occurrence matrices proposed by Rosenfeld [58]. The Haralick features obtained from the chromatic co-occurrence matrices have been used by different authors in the context of color texture classification [11, 36, 59]. Sum and difference histograms, which also belong to the second order strategy, have an almost identical discrimination ability to the chromatic co-occurrence matrix, with the advantage that their calculation are much less demanding in terms of memory requirements [60, 43]. Another second order descriptor is the color Local Binary Pattern, which has been used repeatedly by many authors in their studies of the joint use of texture and color for image classification purposes [61, 13, 62, 34, 47].

In the recent years, a various discriminative and computationally efficient local and global texture descriptors have been introduced, which has led to significant progress in the analysis of color texture for many computer vision problems. Several of color texture analysis methods based on global feature, include color Gabor filtering [52], Markov random field model [63]. Some of the effective local feature methods are color Scale Invariant Feature Transform (SIFT) [64], color Pyramid of Histograms of Oriented Gradients (PHOG) [65], Discriminative Color Descriptors (DCD) [15], Three-Dimensional Adaptive Sum and Difference Histograms (3D-ASDH) [66], Color Local Binary Pattern [61, 13] and many more.

Among the proposed texture descriptors as shown in the table A.1 of the appendix A where a state-of-the-art of the color texture descriptors used in the framework of color texture classification is presented, the Local Binary Pattern (LBP) operator is one of the most successful descriptor to characterize texture images. The distinctive advantages of LBP are its ease of implementation, its invariance to monotonic illumination changes and its low computational complexity [67]. In the following, we will discuss the definition of LBP, its variants and its extensions to color.

1.1.4 Local binary patterns

1.1.4.1 The original LBP descriptor

In the original definition of LBP, proposed by Ojala et al., the local neighborhood structure used to characterize the texture around each pixel of the image is simply a set of pixels taken from a square neighborhood of 3×3 pixels [2]. LBP features capture microscopic local image texture. Figure 1.9 illustrates an example of the LBP computation for the pixel labeled as gray.

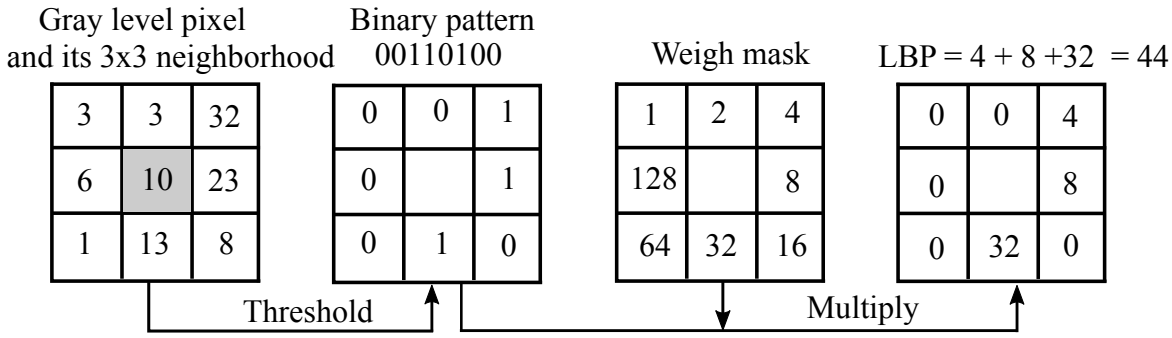


Figure 1.9: An example of the original LBP computation.

The gray level values of each neighboring pixel is first thresholded by the value of the central pixel, that allows to extract a binary vector of 8 bits. This binary vector is called "local binary pattern". For each neighboring pixel, the result of the comparison is set to one if its value is greater or equal than the value of the central pixel, otherwise the result is set to zero. The LBP code of the pixel labeled as gray is then obtained by multiplying the results of the thresholding with weights given by powers of two and summing them up together. The histogram of the binary patterns computed over a region is generally used as LBP feature.

The definition of the original LBP operator has then been generalized to explore intensity values of points on a circular neighborhood. The circular neighborhood is defined by considering the values of radius $\mathcal{R}$ and $\mathcal{P}$ neighbors around the central pixel. The $\mathrm{LBP}_{\mathcal{P},\mathcal{R}}(x_c,y_c)$ code of each pixel (x_c,y_c) is computed by comparing the gray value g_c of the central pixel with the gray values $\{g_i\}_{i=0}^{\mathcal{P}-1}$ of its $\mathcal{P}$ neighbors , as follows:

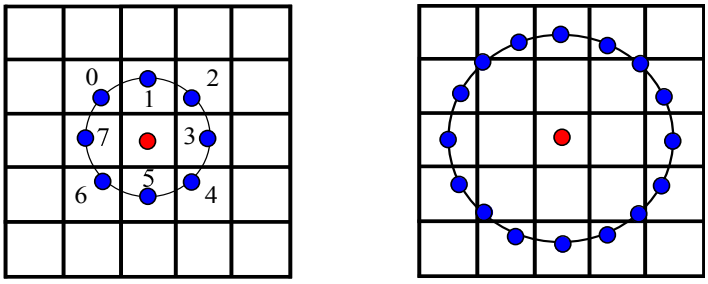
$$\mathrm{LBP}_{\mathcal{P},\mathcal{R}}(x_c,y_c) = \sum_{i=0}^{\mathcal{P}-1} \Phi(g_i - g_c) \times 2^i \tag{1.1}$$

where Φ is the threshold function which is defined as:

$$\Phi(g_i - g_c) = \begin{cases} 1 & \text{if } (g_i - g_c) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

By modifying $\mathcal{R}$ and $\mathcal{P}$, one can compute LBP features for dealing with the texture at different scales. For example, $\mathrm{LBP}_{16,2}$ refers to 16 neighbors in a circular neighborhood of radius

2. The LBP feature produces $2^{\mathcal{P}}$ different output values, and gives rise to a $2^{\mathcal{P}}$-dimensional histogram. Figure 1.10 shows examples of neighborhoods with different radii and numbers of neighbors. The number of bins of the LBP histogram will be 256 or 65536 if 8 or 16 neighboring pixels are considered, respectively.



(a) $LBP_{8,1}$ (b) $LBP_{16,2}$

Figure 1.10: Circular neighborhoods of the center pixel with different neighbors : (a) $LBP_{8,1}$ and (b) $LBP_{16,2}$

Although LBP has several advantages previously cited, the original LBP also has significant disadvantages: it is sensitive to image rotation and noise, it captures only the very local structure of the texture and fails to detect large-scale textural structures [68]. Since Ojala's work, many variants of LBP operator have been proposed in the literature to improve its robustness to noise and increase its discriminative power and applicability to different types of problems including, facial image analysis, biometrics, medical image analysis, motion and activity analysis and content-based retrieval [69, 70, 71, 72, 73, 74, 75, 76, 77, 78].

1.1.4.2 The extensions of LBP

More recently, Liu et al., propose a detailed review of the LBP variant based on a gray scale analysis which can be grouped into several categories [67].

- **Traditional extensions of LBP**: Ojala et al. observed that some LBP patterns occur more frequently in texture images than others. They proposed to define the "LBP uniform pattern" $LBP^{u2}_{\mathcal{P},\mathcal{R}}$ which is a subset of the original LBP [4]. For this, they consider a uniformity measure of a pattern which analyzes the number of bitwise transitions from 0 to 1 or vice versa when a circular bit pattern is considered. A local binary pattern is called uniform if its uniformity measure is at most 2. For example, the patterns 00000000 (0 transitions), 00011110 (2 transitions) and 11100111 (2 transitions) are uniform whereas the patterns 00110010 (4 transitions) and 01010011 (6 transitions) are not. Figure 1.11 illustrates an example of uniform and non uniform LBP patterns where the black points denote 1-bit and white points denote 0-bit. For the computation of the uniform LBP

histogram, the uniform patterns are used such as each uniform pattern has an individual code and the non-uniform patterns are all assigned to a separate code. The $\text{LBP}^{u2}_{\mathcal{P},\mathcal{R}}$ has $(\mathcal{P}(\mathcal{P}-1)+3)$ output values compared with $2^{\mathcal{P}}$ of original LBP.

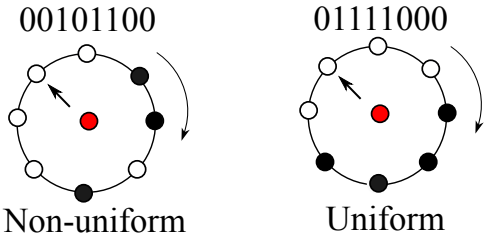


Figure 1.11: An example of non-uniform and uniform patterns LBP

Another traditional extension of LBP was created to respond to the limit that the original LBP descriptor is not rotationally invariant, which is a important requirement for many real-world applications. If the image is rotated, the surrounding pixels in each neighborhood will correspondingly move along the perimeter of the circle, resulting in a different LBP value, except patterns with only 1-bit and 0-bit. Figure 1.12 gives an example of image with two different rotated versions. In order to address this limitation, Pietikäinen et al. proposed a rotation invariant version $\text{LBP}^{ri}_{\mathcal{P},\mathcal{R}}$ of LBP by grouping together the LBP that are actually rotated versions of a same pattern [79]. They also illustrate the 36 unique rotation invariant LBP that can occur in the case of $\mathcal{P}=8$.

The rotation invariant uniform LBP descriptor, $\text{LBP}^{riu2}_{\mathcal{P},\mathcal{R}}$ is then proposed to improve the rotation invariance and reduce the feature dimensionality which is defined as follows [4]:

$$\text{LBP}^{riu2}_{\mathcal{P},\mathcal{R}}(x_c,y_c) = \begin{cases} \sum_{i=0}^{\mathcal{P}-1}\Phi(g_i-g_c) & \text{if} \quad \mathcal{U}(\text{LBP}_{\mathcal{P},\mathcal{R}}) \leq 2, \\ \mathcal{P}+1 & \text{otherwise.} \end{cases} \tag{1.3}$$

where $\mathcal{U}(\text{LBP}_{\mathcal{P},\mathcal{R}})$ is a function to determine the number of bitwise changes in LBP pattern from 0 to 1 (or vice-versa). The $\text{LBP}^{riu2}_{\mathcal{P},\mathcal{R}}$ has $(\mathcal{P}+2)$ output values.

In summary, with $\mathcal{P}=8$ neighboring pixels, the feature dimensionality of the original $\text{LBP}_{8,\mathcal{R}}$, $\text{LBP}^{u2}_{8,\mathcal{R}}$, $\text{LBP}^{ri}_{8,\mathcal{R}}$ and $\text{LBP}^{riu2}_{8,\mathcal{R}}$ are 256, 59, 36, 10, respectively.

- **Neighborhood topology and sampling**: The traditional LBP method identifies a neighborhood as a set of pixels on a circular ring. In order to increase the discriminative power, many neighborhood topologies have been proposed, such as elliptical neighborhoods in Elliptical Binary Patterns [80] and also generalized in [81] to parabolic, hyperbolic and spiral neighborhood topologies.

  The original LBP methods and several variants have also been reviewed to only encode local microtextures and be unable to capture nonlocal macrotextures. Patch-based LBP

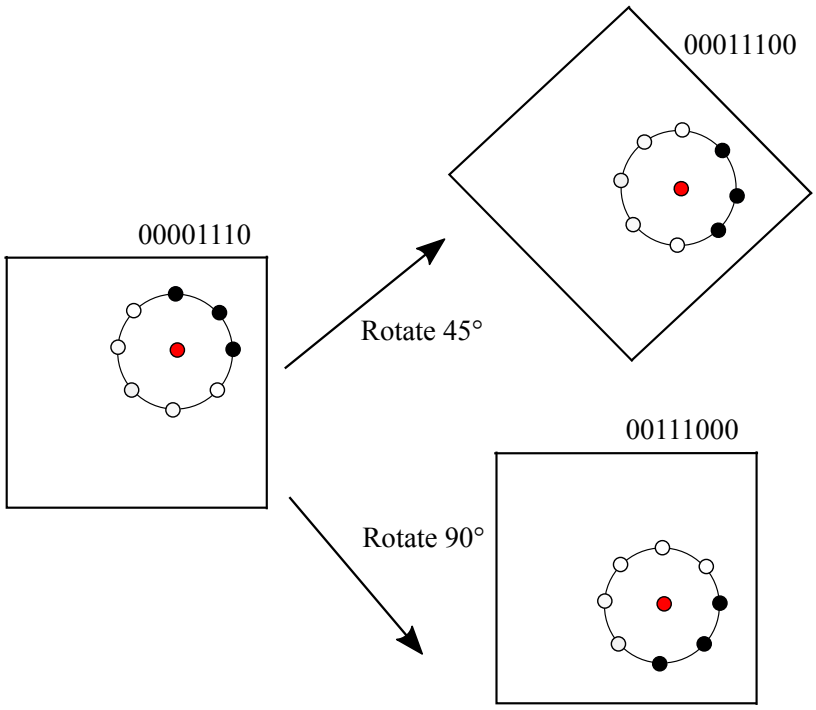


Figure 1.12: An example of the influence of image rotation on points in a circular neighborhood.

variants, including the Local Binary Pattern Filtering [82], Multiscale Block LBP [83], Three Patch LBP [84], Four Patch LBP [84], Pixel to Patch [70] and Median Robust Extended LBP [75], aim to overcome this problem by integrating over larger areas.

- **Thresholding and quantization**: The original LBP operator is sensitive to noise due to the thresholding operation that directly compares pixel values. Many authors have proposed several LBP variants by changing the thresholding scheme or the number of quantization level to gain noise robustness and discrimination power, including Neighborhood Intensity LBP [73], Improved LBP [85], Local Median LBP [86] and Threshold Modified LBP [87]. Among these approaches, the Local Ternary Patterns (LTP), proposed by Tan et al., have achieved a great success [68]. In this approach, an additional parameter is used to define a tolerance for similarity between different gray intensities to be robust to noise and reduce the dimensionality. Each ternary is then split into positive and negative parts, which are subsequently treated as two separate LBP component for which histograms are computed and finally concatenated.

- **Combining with complementary features**: In order to improve the texture characterization, many authors have combined LBP with other complementary features. A rotation

invariant measure of the local variance can be defined as [4]:

$$VAR_{\mathcal{P},\mathcal{R}} = \frac{1}{\mathcal{P}} \sum_{i=0}^{\mathcal{P}-1} (g_i - u) \qquad \text{with:} \quad u = \frac{1}{\mathcal{P}} \sum_{i=0}^{\mathcal{P}-1} g_i \tag{1.4}$$

Ojala et al. propose to use the joint distribution of $VAR_{\mathcal{P},\mathcal{R}}$/$\text{LBP}^{riu2}_{\mathcal{P},\mathcal{R}}$ for a better characterize of the local texture instead of using $\text{LBP}^{riu2}_{\mathcal{P},\mathcal{R}}$ alone. Another combination can be found as the fusion of LBP variants and Gabor features has been explored, with applications in texture classification [88] and face recognition [89, 68]. Wang et al. combine Histogram of Gradients (HOG) with LBP, performing a good result in human detection [90]. Hussain and Triggs combine LTP and LBP [91]. Klare and Jain exploit the combination of LBP and Scale Invariant Feature Transform (SIFT) for heterogeneous face recognition [92]. Roy et al., combine Haar and LBP features for an illumination invariant face detection [93].

Among the local rotation invariant LBP features, LBP Histogram Fourier features (LBP-HF) is proposed by Ahonen et al. to combine LBP and the Discrete Fourier Transform (DFT) [94]. Unlike the existing local rotation invariant LBP features, the LBP-HF descriptor is produced by computing an LBP histogram over the whole region and then constructing rotationally invariant features from the histogram with DFT. In order to apply LBP descriptor for a specific application, we used LBP-HF features for lace texture images classification in the beginning of this work. The LBP tuning, including radius and number of neighbors, have been adjusted to reveal the lace structure in [95]. This work is presented in appendix B.

A comprehensive literature survey introduced by Brahnam et al. has revealed the huge diversity of LBP [96] that is confirmed by [67]. In comparison with other grayscale LBP variants, color variants have received significantly less attention in the literature. The next subsection details the different ways that have been proposed to extend LBP to color.

1.1.4.3 The color LBP

The original LBP computation is based on grayscale images. However, it has been demonstrated that color information is very important to represent the texture, especially in natural textures [3, 11, 12, 97, 98]. In literature, the extension of LBP to color follows the strategies of color and texture combination presented in section 1.1.3.1.

- In the first strategy, the original LBP operator is computed from the luminance image and combined with color features. For example, Mäenpää or Ning proposed to characterize the color texture by concatenating the 3D color histogram of the color image and the LBP histogram of the corresponding luminance image [13, 99]. Cusano et al. propose a texture

descriptor which combines a luminance LBP histogram with color features based on the local color contrast [34]. Recently, Lee et al. propose another color LBP variant for face recognition tasks, the Local Color Vector Binary Pattern [62]. In the proposed approach, each color texture image is characterized by the concatenation of four LBP histograms, namely one LBP extracted from the luminance image and three for the angles between the possible pairs of different color components.

- The second strategy consists in applying the original LBP operator independently on each of the three components of the color image, without considering the spatial interactions between the levels of two different color components. The texture descriptor is obtained by concatenating the three resulting LBP histograms. Figure 1.13 illustrates this strategy of color LBP computation by considering the (C_1,C_2,C_3) color space. This within component strategy has been applied by several authors [56, 100, 101, 14, 102].

- The third strategy consists in taking into account the spatial interactions within and between color components. In order to describe color texture, Opponent Color LBP (OCLBP) was defined [13]. For this purpose, the LBP operator is applied on each pixel and for each pair of components (C_k,C'_k), $k,k' \in \{1,2,3\}$. In this definition, opposing pairs such as (C_1,C_2) and (C_2,C_1) are considering to be highly redundant, and so, one of each pair is used in the analysis. This leads to characterize a texture with only six histograms pairs $((C_1,C_1), (C_2,C_2), (C_3,C_3), (C_1,C_2), (C_1,C_3), (C_2,C_3))$ out of the nine available ones. However, these *a priori* chosen six histograms are not always the most relevant according to the different considered data sets [8] and it is preferable to consider the Extended Opponent Color LBP (EOCLBP). This way to describe the color textures thanks LBP has been proposed by Pietikäinen in 2002 [56]. It consists in taking into account each color component independently and each possible pair of color components, leading to nine different histograms: three within-component $((C_1,C_1),(C_2,C_2),(C_3,C_3))$ and six between-component $((C_1,C_2), (C_2,C_1), (C_1,C_3), (C_3,C_1), (C_2,C_3), (C_3,C_2))$ LBP histograms. These nine histograms are finally concatenated so that a color texture image is represented in a $(9 \times 2^{\mathcal{P}})$-dimensional feature space. Figure 1.14 illustrates the computation steps achieved to obtain the LBP values for the pairs of color components $(C_1,C_1),(C_1,C_2),(C_2,C_1)$ and (C_1,C_3) and shows that the pair (C_1,C_2) is different to (C_2,C_1). The OCLBP and EOCLBP have often been considered to classify color texture images [13, 103, 104, 105, 8, 9].

- The fourth strategy consists in analyzing the spatial interactions between the colors of the neighboring pixels based on the consideration of an order relation between colors. Instead of comparing the color components of pixels, Porebski et al. represent the color of pixels by a vector and compare the color vectors of the neighboring pixels with the

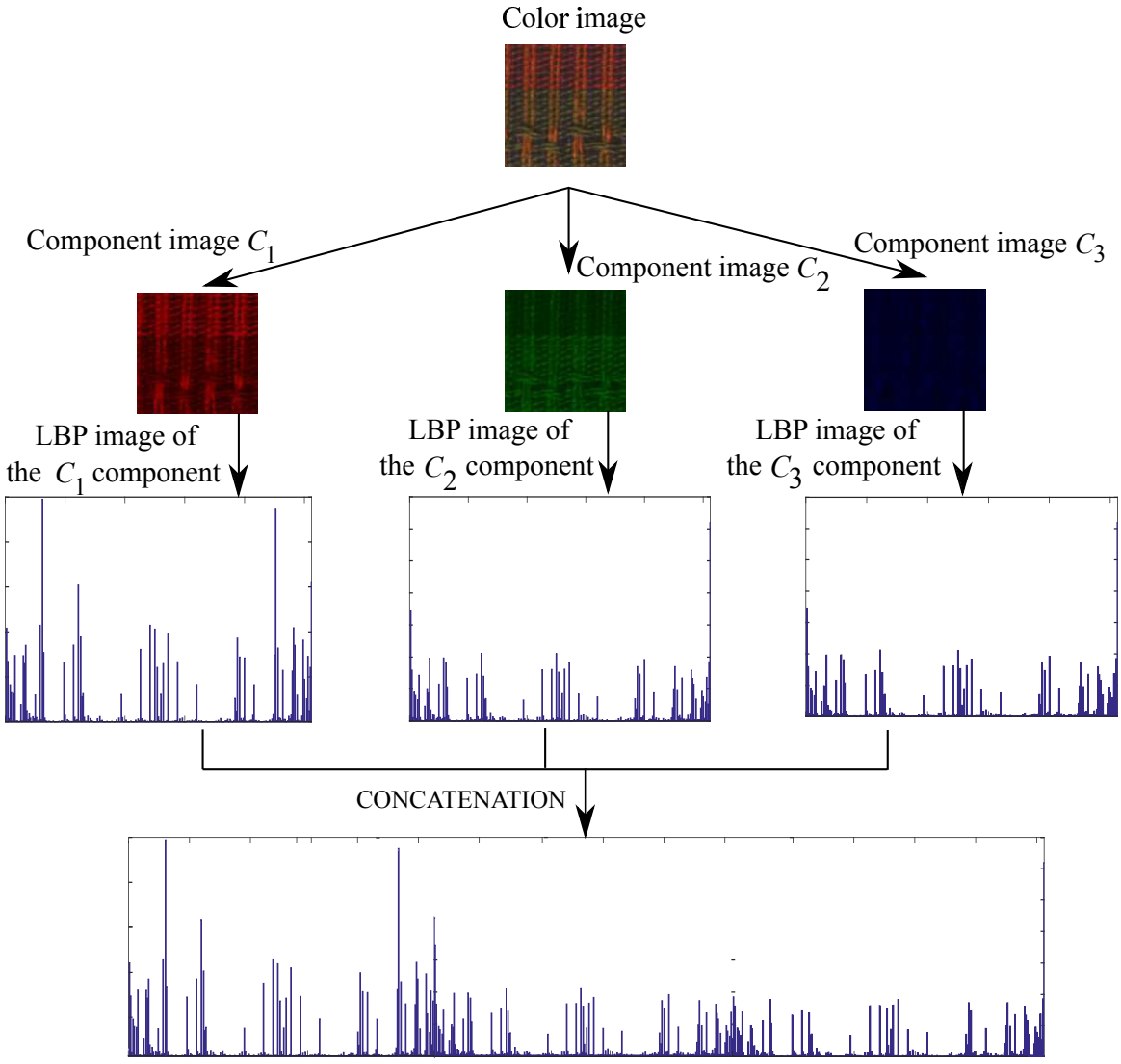


Figure 1.13: Illustration of the within component color LBP computation.

color vector of the central one [46]. They use a partial color order relation based on the Euclidean distance for comparing the rank of color. As a result a single color LBP histogram is obtained instead of the 6 or 9 provided by OCLBP or EOCLBP respectively [8, 9]. Another possible way consists in defining a suitable total ordering in the color space and using it as a replacement for the natural gray level ordering in LBP definitions. This strategy has recently been investigated by Ledoux et al. whose propose the Mixed Color Order LBP (MCOLBP) [47].

In order to give a single code by color LBP, quaternion representation can be used. Quaternion is shown as a efficient mathematical tool for representing color images based on a hypercomplex representation [106]. Lan et al. have thus proposed the Quaternionic Local Binary Pattern (QLBP) that makes use of quaternion to represent each pixel

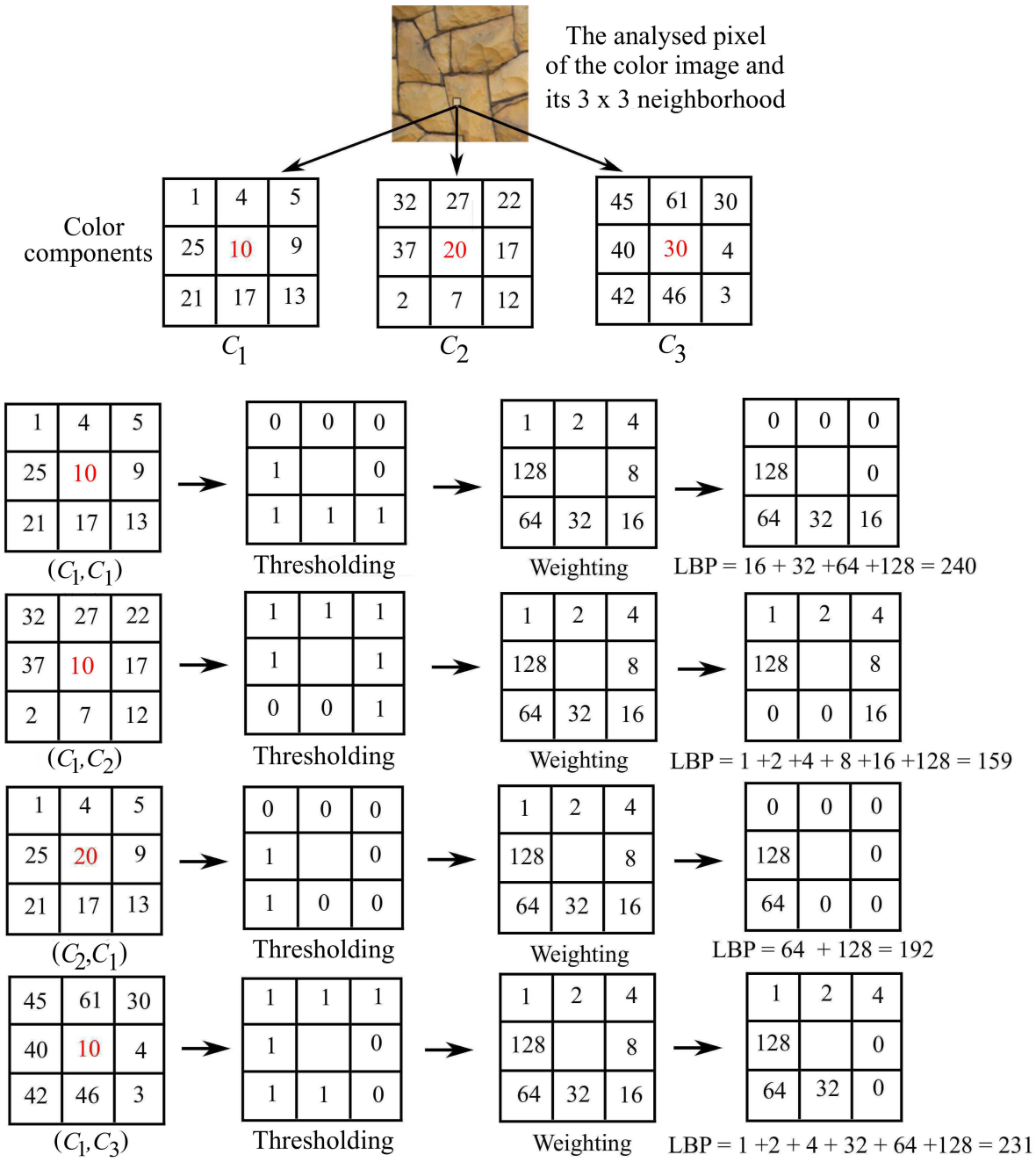


Figure 1.14: The different steps to obtain the LBP values for the pairs of color components $(C_1,C_1),(C_1,C_2),(C_2,C_1)$ and (C_1,C_3) for the analyzed pixel. For example, to compute the LBP value (C_1,C_2) of the analysed pixel, the color component C_2 of each of the 8 neighboring pixels is compared with the color component C_1 of the considered pixel. This step is realized as demonstrated in Figure 1.9.

color by all color components at one time. Under this representation, the dimension of QLBP is equal to the dimension of a grayscale LBP. QLBP has been used for person re-identification problems by Lan and Chahla in [107, 108].

Among the different extension strategies of LBP to color, MCOLBP and QLBP have the advantage of providing a texture descriptor whose dimension is equal to gray level LBP histogram, what allows a low computation time. However, the classification results obtained with these descriptors on two benchmark texture databases are not as good as those obtained thanks to OCLBP [47, 109].

Another possible way to obtain a good compromise between classification results and computation time is to consider a higher dimensional descriptor, such as OCLBP or EOCLBP, and to proceed to a dimensionality reduction. This is the strategy that we propose to explore in our works. Before presenting the subject of LBP selection, we will see how the characterization of the color texture fits in the classification scheme.

1.2 Supervised texture classification

Texture classification is a task which allows to assign a given texture to one of several texture classes. It is a fundamental issue of texture analysis, playing a significant role in many applications such as biomedical image analysis, industrial inspection, analysis of satellite or aerial imagery, document analysis, face analysis, biometrics and many more. Color texture classification has become a challenging topic in computer vision because the real world images often exhibit a high degree of complexity, randomness and irregularity. For example, two images containing the same color with different texture patterns or the same texture pattern but different colors are considered as different color textures [11]. Figure 1.15 (a-b) illustrates an example of two images (from the USPTex database) with the same color but different textures of granite and figure 1.15 (c) and 1.15 (d) show the texture of stones with different colors.

Color texture classification is typically categorized into two subproblems of representation and classification [5, 27] as shown in figure 1.16. The feature generation step allows to characterize the image thanks to a texture feature and the decision step assigns the feature to one of the available texture classes. Our work focuses on the first subproblem.

In this section, we briefly review the three principal categories of classification context, introduce several standard color texture databases available for color texture classification tasks and present the most commonly classifiers used for color texture classification.

1.2.1 Context

According to the prior knowledge of class label, there are three major types of classification contexts: the supervised, unsupervised and semi-supervised classification. A comprehensive literature survey of classification approaches is proposed by Jain et al. [110].

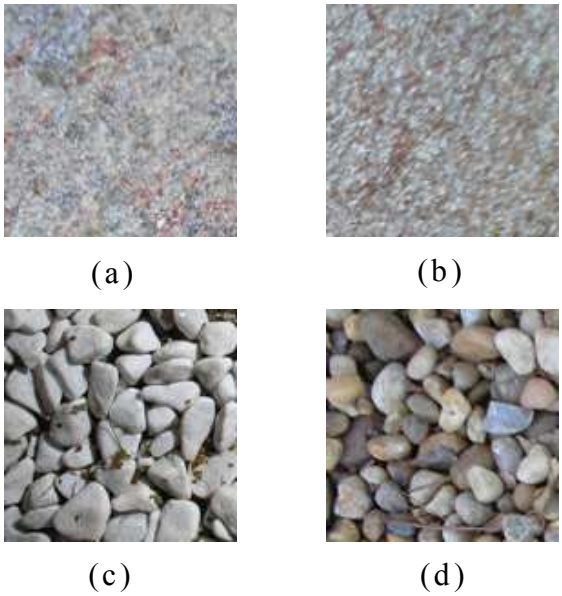


Figure 1.15: Exemple of two different textures with the same color (a-b) and the same texture of stone with different colors (c-d).

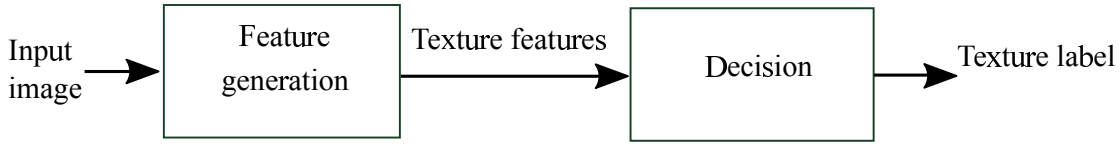


Figure 1.16: A basic scheme of texture classification.

1.2.1.1 Supervised classification

In the supervised classification, we have a class label for each input prototype image and use it to predict the label of a new unseen image. In practice, this process can be evaluated into two steps: training and decision. In the training step, feature generation is applied on a certain number of prototype color texture images with known class labels. In the decision step, feature generation is applied on unknown input image (as in the training step). Then, a prediction on the class label assignment for this image, based on a similarity measure computed between the training feature vector and the feature vector of the input image. Figure 1.17 illustrates a general framework of supervised classification.

1.2.1.2 Unsupervised classification

On the other hand, unsupervised classification (or clustering) does not provide any prior knowledge about the class labels. The classification automatically discovers the different classes (or clusters) from input textures, based on various types of criteria of feature descriptors such as distance, information or correlation. Figure 1.18 illustrates a general framework of unsuper-

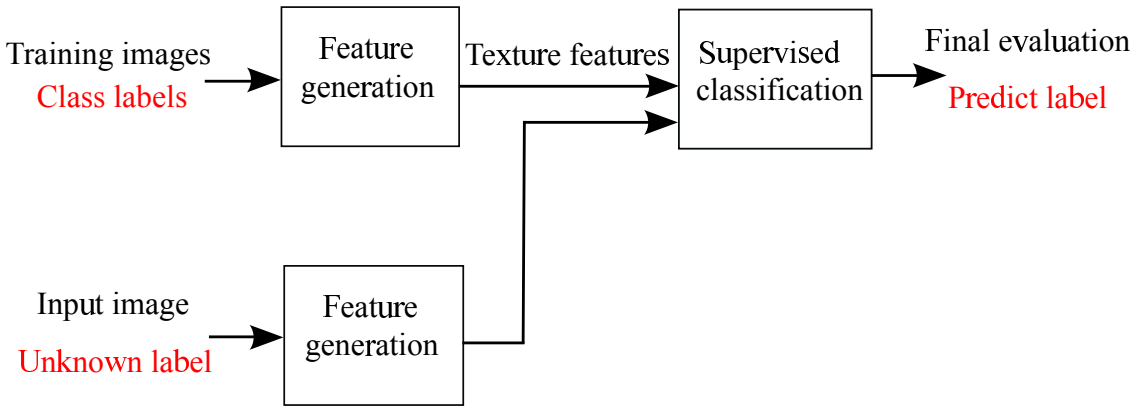


Figure 1.17: A general framework of a supervised classification.

vised classification.

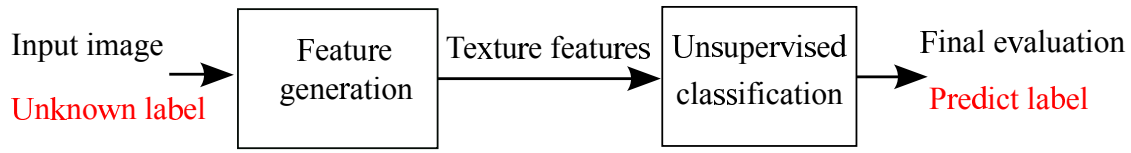


Figure 1.18: A general framework of an unsupervised classification.

1.2.1.3 Semi-supervised classification

Semi-supervised classification falls between unsupervised and supervised classification. The label training images is often limited or expensive to be obtained. When a small portion of data is labeled, the classification in this context can take advantages of both labeled data and unlabeled data. Figure 1.19 illustrates a general framework of supervised classification.

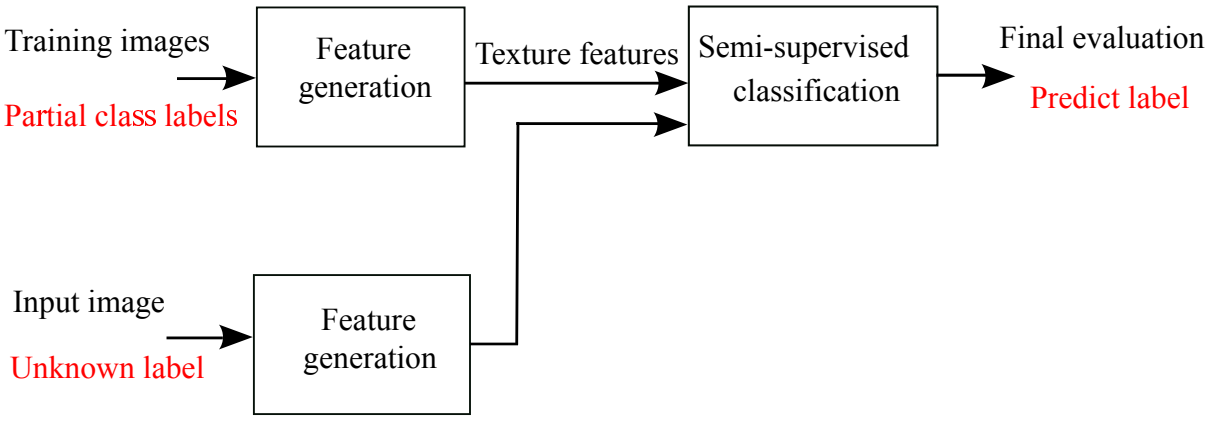


Figure 1.19: A general framework of a semi-supervised classification.

1.2.1.4 Dimensionality reduction

We briefly presented three classification contexts. Before the decision step (figure 1.16), the set of generated features can be modified by a dimensionality reduction technique in order to have a better representation of data (as illustrated in figure 1.20). For example, in the color texture classification framework, each texture image is characterized by nine LBP histograms which lead to a concatenated feature vector of $256 \times 9 = 2304$ bins (features) for a single color space when 8 neighbors are used. It is clear that all the features contribute unequally in the classification task that leads to decrease the classification performance. So, the dimensionality reduction is needed to address this problem. Many authors have developed LBP-based features by many approaches in order to reduce the feature dimensionality.

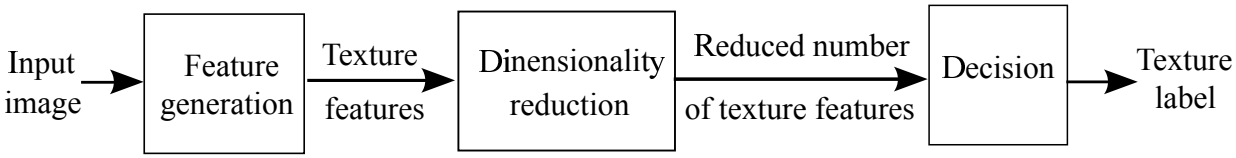


Figure 1.20: A scheme of texture classification with dimensionality reduction step.

According to whether the original features space information is changed or not, dimensionality reduction methods can be categorized into feature extraction and feature selection. The following of this work is conducted by the dimensionality reduction methods for color texture classification in a supervised context. These methods are detailed in the chapter 2. The next subsection thus presents the most commonly used classifiers for color texture classification problems in the supervised context.

1.2.2 Classifiers

A classifier is a function which takes the features as inputs and gives the texture classes as outputs. There is a large number of methods for constructing a classifier [111]. Here, we briefly introduce the three classifiers widely used in the supervised color texture classification context:

1. Linear Disciminant Analysis
2. Support Vector Machines
3. *K*-Nearest Neighbors

1.2.2.1 Linear disciminant analysis

Linear Discriminant Analysis (LDA) is a supervised statistical method that allows to classify an image between two or more classes [112, 113]. Given the training images, the LDA tries

to find a new feature space where the distances between the class centers are maximized and the distances between the images of a same class are simultaneously minimized. This new feature space is obtained thanks to linear combinations of the original features. This can be achieved through the analysis of the within-class and between-class scatter matrices. The new features are found by solving the generalized eigenvalue problem. Figure 1.21 illustrates the separation of two classes before LDA (a) and after LDA (b). After the projection, the distances between the images of a same class is minimized ($V'_1 < V_1$ and $V'_2 < V_2$) whereas the distance between the class centers is maximized ($D_2 > D_1$). To classify a testing image, LDA estimates the probability that the considered image belongs to each class. The class that gets the highest probability is the output class.

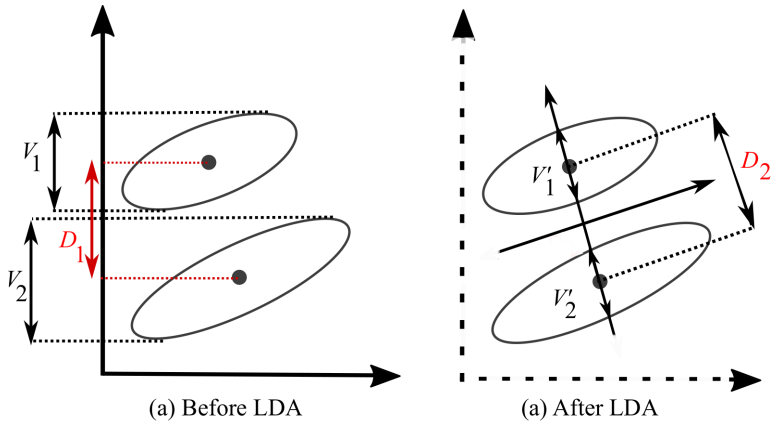


Figure 1.21: An illustration of LDA.

The LDA classifier has been used by several authors in order to validate the performance of their color texture classification approach [114, 115, 116, 117].

1.2.2.2 Support vector machines

The standard Support Vector Machines (SVM) proposed by Cortes et al. [118] are a type of linear discriminant binary classifier. A linear boundary between two classes is represented by a hyperplane. The optimal hyperplane is the boundary that maximizes the margin of separation between the classes (i.e., maximizes the distance between the boundary and the images that are close to the boundary). An example of separation hyperplanes is illustrated in Figure 1.22. From this figure, we see that H_2 is the separating hyperplane with the maximum margin contrary to H_1. The SVM classification is performed by determining on which side of the decision boundary a given testing image falls into and then by assigning the corresponding class label.

In addition to performing linear classification, SVM can efficiently perform non-linear classification using different kernel functions by performing a non-linear mapping from the input

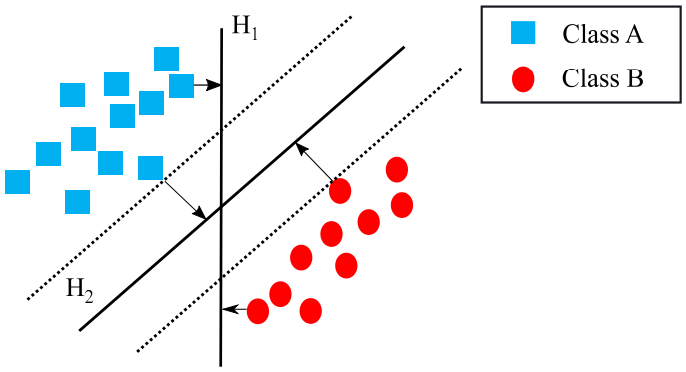


Figure 1.22: Illustration of different hyperplanes. H_1 separates the two classes with a small margin, whereas H_2 separates them with the maximum margin.

space to the transformed space, a straight separating line in the transformed space may correspond to non-linear decision boundary in the original space [119]. The choice of this kernel function and the tuning of its parameters will directly impact the final result.

The standard SVM is a two-class classifier, whereas many real world classification problems involve several classes. There are two strategies to extend SVM for dealing with multi-class problems: one-versus-all and one-versus-one strategies. The first strategy constructs one two-class SVM classifier for each class. An image would be classified under a certain class if and only if that class's SVM accepts it and all other classes's SVMs reject it. The second strategy constructs one SVM binary classifier for each pair of classes and the classification phase is realized by a voting way: the image is assigned to the class that is selected by the majority of the classifiers. An extensive comparison of multi-class SVM approach is discussed in [120, 121].

SVM classification approach has been used by Iakovidis and Sandid [122, 66] to classify color texture images.

1.2.2.3 *K*-nearest neighbors

The *K*-Nearest Neighbors (*K*-NN) classifier is among the simplest classifiers of all machine learning algorithms and it is frequently used in pattern recognition. In the *K*-NN approach, the testing images are classified based on the closest training images in the feature space [123, 124]. The distance between each testing images and each training image is first computed. The testing image is then assigned to the class that is most common among its *K*-nearest neighbors. *K* is a user-defined constant. In the case of $K = 1$, each testing image is assigned to the class of its nearest neighbor. The optimum value of *K* depends upon the data. Generally, larger values of *K* decrease the effect of noise on the classification. An example of *K*-NN classification is illustrated in Figure 1.23. Based on the value of *K*, the testing image (represented by a green star) will be classified into the class A or the class B. If $K = 3$ (dashed line circle), this image is

assigned to the class B, whereas if $K = 5$ (solid line circle), this image is assigned to the class A.

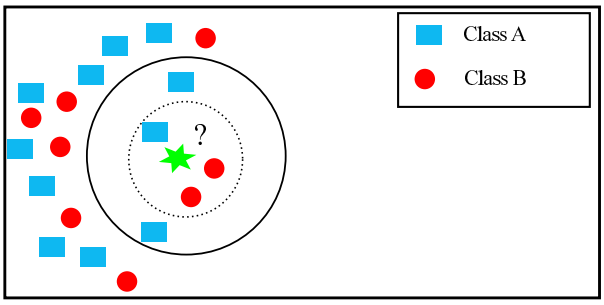


Figure 1.23: Example of 3-NN and 5-NN classification.

There are various authors that have used K-NN to validate the performance of their proposed method in the framework of color texture classification [12, 56, 60, 13, 36, 41, 54, 125, 126, 127, 117, 128, 129, 130, 47, 131, 132, 109]. The table A.1 in the appendix A shows that there is about 80% of studies in color texture classification that use the K-NN in their experiments. The value of K that is commonly used is 1 while the typically considered distance metrics are $L1$ [56, 13, 47, 130, 109], Euclidean [60, 54, 131], Mahalanobis [12, 129], and χ^2 [127, 128, 132].

1.2.3 Evaluation methods

In order to validate the performance of a classification scheme, there are several evaluation methods that divide an available dataset into training and testing sets: bootstrap, resubstitution and cross validation methods as presented in [110]. Generally, there are three kinds of validation techniques which are widely used in the framework of color texture classification [133]:

- **Holdout** method is the simplest cross validation approach. The dataset is here divided into two exclusive sets. The proportion of the training and testing sets is usually equal to 1/2. Many authors have used this method to split the dataset in order to validate the proposed approach in color texture images classification [134, 135, 105, 136, 8, 137, 138, 139, 66, 140, 9, 47, 130, 141, 115, 56, 13, 41, 122, 126, 127, 128, 142, 114].

- $\mathcal{K}$**-fold cross validation** is an extension of the holdout method. The dataset is spitted into $\mathcal{K}$ subsets and the holdout method is repeated $\mathcal{K}$ times. Each time, one of the $\mathcal{K}$ subsets is used as the testing set and the other $\mathcal{K} - 1$ subsets are put together to form the training set. In order to evaluate the performance, several authors used this technique to split the color texture image database into training and testing sets [131, 143, 144, 145, 146].

- **Leaving-one-out cross validation** is a special case of $\mathcal{K}$-fold cross validation, with $\mathcal{K}$ equal to N, the number of images in the dataset. This technique is used by several authors [60, 36, 147, 132, 54].

After creating the partition of the dataset, the classifier uses the training set to predict the output labels for the images of the testing set. The classification performance is calculated by comparing the predicted class labels obtained by the classifier with the true class labels. *Accuracy* is estimated as the sum of correct classified images divided by the total number of testing images. Most of the authors used *Accuracy* to measure the performance of the classifier in the color texture image classification applications. Other evaluation criteria such as *Precision* and *F-measure* are used in[134, 148].

1.2.4 Benchmark color texture image databases

In order to evaluate the performance of different texture analysis methods and to compare their performance, various image databases have been proposed in the literature [149, 33]. In the framework of color texture classification, several datasets of color textures have been categorized and critically surveyed by Porebski et al. [138]. We introduce here, four benchmark color texture databases that are used in our experimentations in chapter 4: OuTex-TC-00013, USPTex, STex and BarkTex.

- **OuTex-TC-00013**: The test suite OuTex-TC-00013 is provided by the OuTex texture database [150]. The images of this database are acquired with a three-CCD color camera under the same illumination conditions. This database is a collection of heterogeneous materials such as cardboard, fabric, paper, wool, etc. It contains 68 texture images of 746 $\times$ 538 pixels. The test suite is constructed by splitting each one of the original texture image into 20 sub-images (128 $\times$ 128 pixels) without overlapping, thus resulting in a dataset containing 1360 images. Figure. 1.24 illustrates the images of this dataset where each image represent each class of texture. The specificity of this database is that it contains several categories with similar colors and textures, resulting in a high inter-class similarity. This database is publicly available at http://www.outex.oulu.fi.

- **USPTex**: The USPTex database consists of a set of 191 color texture images acquired using a digital camera under an unknown but fixed light source. The considered 191 texture classes are typically daily found, such as beans, rice, tissues, road scenes, various types of vegetation, walls, clouds and soils [117]. Each image has a size of 512$\times$384 pixels from which 12 sub-images with a size of 128$\times$128 pixels are extracted without overlapping, so that a total of 2292 images is obtained. Figure. 1.25 shows some examples of texture of the USPTex database.

Figure 1.24: The OuTex-TC-00013 dataset includes 68 different texture classes.

This database is publicly available at http://fractal.ifsc.usp.br/dataset/USPtex.php

- **STex**: The Salzburg Texture Image Database (STex) is a large collection of color textures image. It is publicly available at http://wavelab.at/sources/Stex. It is more homogeneous than the other databases proposed for texture classification purposes. There is unavailable information about the type of acquisition device used and the lighting conditions. STex contained a total of 476 texture images of 512×512 pixels. For experimental purpose, each texture is subdivided into 16 non overlapping sub-images of 128×128 pixels. The database thus consists of 7616 color texture images belonging to 476 different classes. Figure. 1.26 illustrates some examples of texture of this database.

- **BarkTex** : The BarkTex database has been proposed by Lakmann at the University of Koblenz-Landau, Germany [151]. The images of the BarkTex database represent natural color

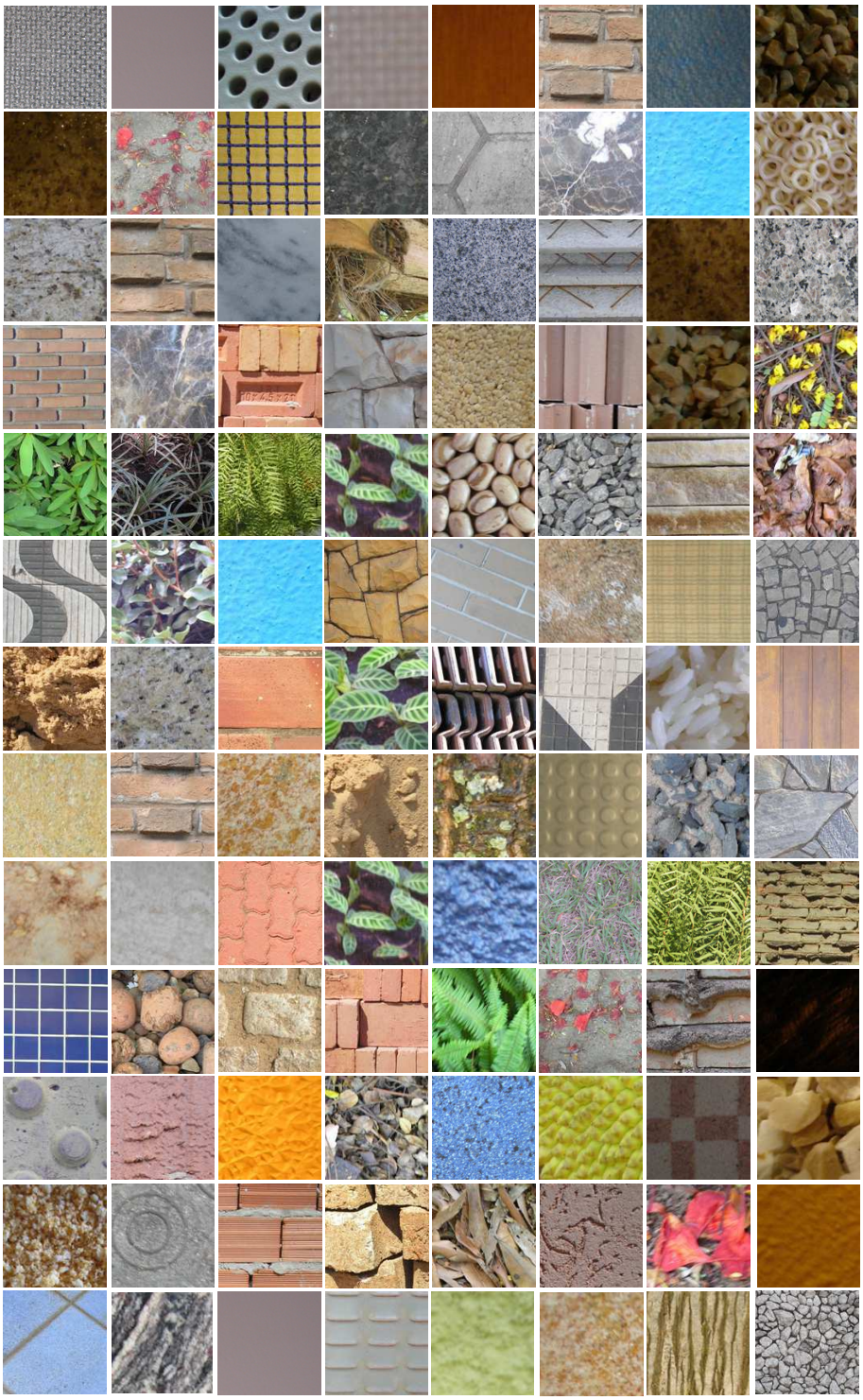


Figure 1.25: Selected textures among 191 classes from the USPTex database (one image per class).

textures which have been acquired under non-controlled illumination conditions. Each image of size 256 $\times$ 384 displays the bark of a certain tree. The BarkTex database is composed of six different kinds of trees, with 68 images per class. Figure 1.27 illustrates an example of

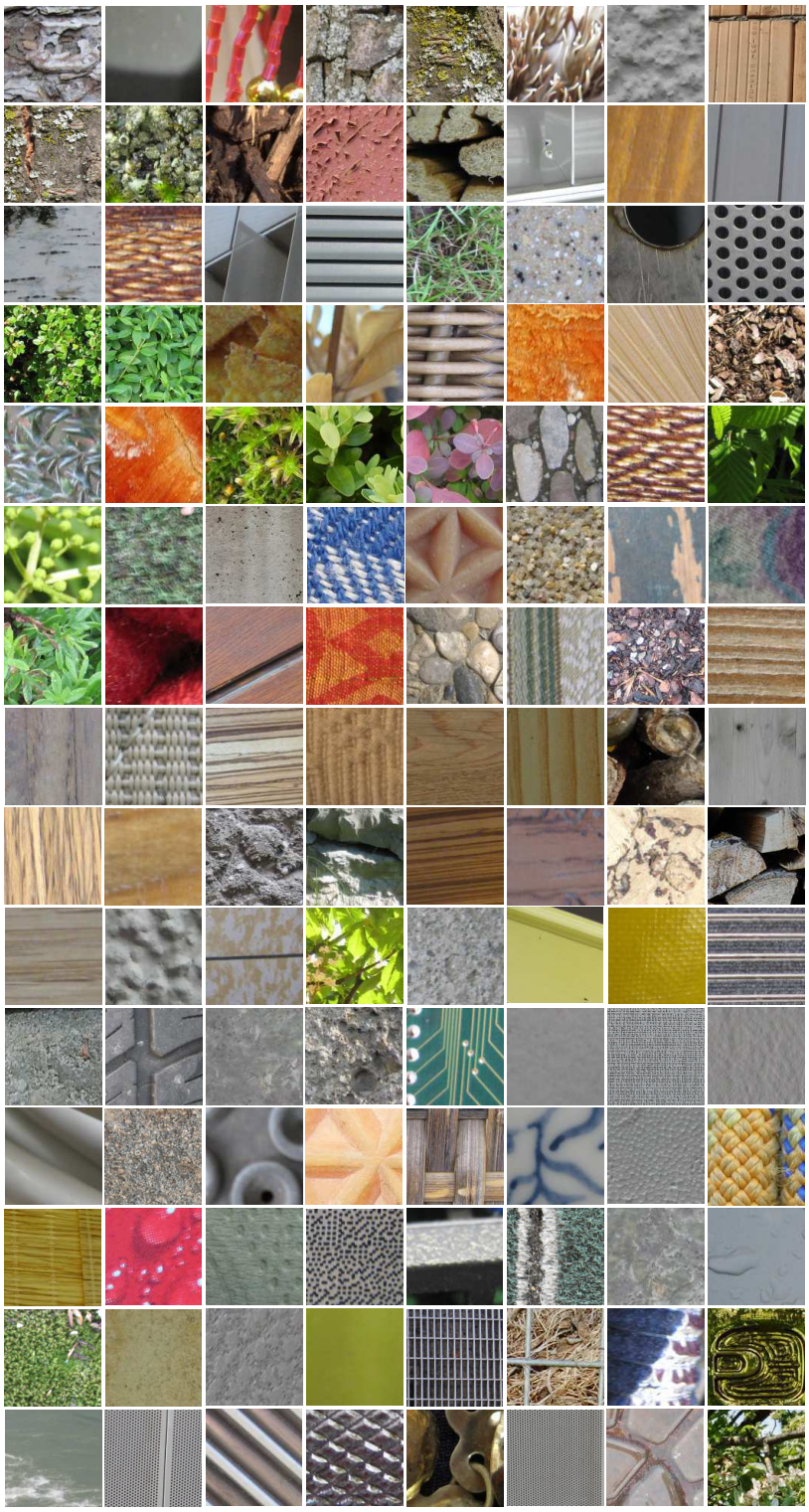


Figure 1.26: Selected textures among 476 classes from STex database (one image per class).

each of the six tree bark classes of this database. Since the images are not restricted to the bark texture and also show background structures, the image border is excluded defining a Region-of-Interest of fixed size 300×200 located at the image center. This database is publicly

Figure 1.27: Examples of images of the BarkTex database.

available at ftp://ftphost.uni-koblenz.de/outgoing/vision/Lakmann/BarkTex

The OuTex-TC-00013, USPTex and STex databases have the common limitation that the sub-images of a same class are extracted from the same acquired original image. Porebski et al. have shown that the partitioning which builds a training and a testing subset from an initial image set can lead to biased classification results. This partitioning could provide high classification accuracy whatever the considered features when it is combined with a classifier such as the nearest neighbor classifier. In order to overcome this drawback, Porebski et al. propose a modified version of Barktex, namely New Barktex [138].

To build the New-Barktex set, a region of interest, centered on the bark and whose size is 128×128 pixels, is first defined. Then, four sub-images whose size is 64×64 pixels are extracted from each region. We thus obtain a set of $68 \times 4 = 272$ sub-images per class. To ensure that color texture images used for the training and the testing images are less correlated as possible, the four sub-images extracted from a same original image all belong either to the training subset or to the testing one: 816 images are thus used as training images and the remaining 816 as testing images. Figure. 1.28 illustrates the images of New BarkTex test suite where each row represents each kinds of trees.

Figure 1.28: Example of New BarkTex color test suite: each row represents a class of texture.

The evaluation methods and four benchmark databases have been introduced[1]. All the recent studies related to those databases are now rewiewed in the following section.

1.2.5 Review of the considered databases

In this section, we synthesize the recent works on color texture classification which are carried out in the literature on the four databases previously presented. The goal of this review is to highlight the test suits that are most frequently used, in order to compare our work with the maximum of previous studies. Indeed, each database can be partitioned to different test suites according to the size of images, the number of training or testing images and evaluation method. This synthesis will allow us to follow the most used partition for each database.

Table 1.1, 1.2, 1.3 and 1.4 summary respectively the characteristics of OuTex-TC-00013, New BarkTex, USPTex and STex databases, when they have been used in the framework of color texture classification. The first column of those tables gives the original name of the image database. The second column shows the name of the test suite. The third column gives the number of classes of the test suite. The fourth column presents the number of images per class. The fifth column indicates the size of the images. The sixth column mentions the cross validation method used to split the image databases into training and testing sets. The last column presents the name of the authors, the year of publication and their references.

In each table, the most frequently used test suite is highlighted. In the following, we will use these test suites to evaluate the performance of our approach and compare our results with those of other works. For example, in the Table 1.2, there are 4 different test suites used for the BarkTex dataset, according to the number of images per class, the size of images and the cross

[1] All the image test suites can be downloaded at https://www-lisic.univ-littoral.fr/~porebski/Recherche

validation method and the most used test suite is New BarkTex.

Table 1.1: Characteristics of OuTex-TC-00013 dataset used in texture classification.

Database	Name	Number of classes	Number of images per class	Size of images	Validation method	Reference
OuTex	OuTex-TC-00013	68	20	128×128	Holdout (1/2 - 1/2)	Pietikäinen, 2002 [56] Mäenpää, 2004 [13] Arvis, 2004 [41] Iakovidis, 2005 [122] Xu, 2005 [54] Alvarez, 2012 [127] Cusano, 2013 [152] Qazi, 2013 [126] El Maliani, 2014 [134] Porebski, 2014 [138] Cusano, 2014 [34] Kalakech, 2015 [9] Martinez, 2015 [135] Hammouche, 2015 [128] Guo, 2016 [142] Casanova, 2016 [114] Ledoux, 2016 [47] Sandid, 2016 [66] Naresh, [148]
		68	20	128×128	$\mathcal{K}$-fold	Paci, 2013 [143] Fernandez, 2013 [144] Sá Junior, 2016 [145] Bello-Cerezo, 2016 [131] Bianconi, 2017a [146] Bianconi, 2017b [153]
		68	20	128×128	Holdout (2/3 - 1/3)	Lan, 2016 [109]
		68	20	128×128	Leaving-one-out	Cernadas, 2017 [132]

Table 1.2: Characteristics of BarkTex dataset used in texture classification.

Database	Name	Number of classes	Number of images per class	Size of images	Validation method	Reference
BarkTex	New BarkTex	6	136	64×64	Holdout (1/2 - 1/2)	Porebski, 2013 [8] Kalakech, 2015 [9] Ledoux, 2016 [47] Sandid, 2016 [66] Wang, 2017 [154]
	BarkTex	6	68	300×200	Leaving-one-out	Palm, 2004 [36]
		6	272	64×64	Leaving-one-out	Münzenmayer, 2002 [60]
		6	68	64×64	Holdout (0.47–0.53)	Porebski, 2007 [136]

Table 1.3: Characteristics of USPTex dataset used in texture classification.

Database	Name	Number of classes	Number of images per class	Size of images	Validation method	Reference
USPTex	USPTex	180	12	128×128	Leaving-one-out	Backes, 2012 [117]
		191	12	128×128	Holdout (1/2 - 1/2)	Oliveira, 2015 [155] Guo, 2016 [142] Ledoux, 2016 [47] Florindo, 2016 [115]
		332	12	312×384	Holdout (1/2 - 1/2)	Casanova, 2016 [114]
		191	12	128×128	$\mathcal{K}$-fold	Bianconi, 2017 [146] Bello-Cerezo, 2016 [131] Gonçalves, 2016 [156]
		191	12	128×128	Holdout (2/3 - 1/3)	Lan, 2016 [109]
		191	12	128×128	Leaving-one-out	Chen, 2016 [147] Cernadas, 2017 [132]

Table 1.4: Characteristics of STex dataset used in texture classification.

Database	Name	Number of classes	Number of images per class	Size of images	Validation method	Reference
STex	STex	476	16	128×128	Holdout (1/2 - 1/2)	El Maliani, 2014 [134] Martinez, 2015 [135]
		476	16	128×128	$\mathcal{K}$-fold	Bello-Cerezo, 2016 [131]

1.3 Conclusion

Texture classification is a fundamental topic in computer vision, playing a significant role in various applications. Texture descriptors have early been applied on grayscale images and have thereby ignored the color information. Many authors demonstrate that color texture features could enhance the performance of texture classification. In this chapter, we have described the key concept of color texture classification. The principal families of color spaces have been presented as well as the main color texture descriptors. The most commonly used classifiers (LDA, SVM, K-NN) in the supervised context have been presented. Four color texture databases have been introduced: OuTex-TC-00013, USPTex, STex and BarkTex and we have reviewed recent works related on those databases to show the most frequently test suites used in the state-of-the-art.

The LBP operator is one of most popular descriptors in texture classification due to its simplicity and good performances. With the objective of improving the performance of texture classification, an extension of LBP to color is proposed. However, color LBP still has some limitations because it lies in high-dimensional feature space. In order to overcome this drawback, many dimensionality reduction approaches have been proposed to reduce the LBP dimension space. The next chapter introduce the feature selection which is an important and frequently used technique for dimension reduction.

Chapter 2

Feature selection

Contents

2.1 Taxonomy of feature selection methods 50

2.1.1 Context of feature selection 53

2.1.2 Feature selection evaluation 55

2.2 Data and knowledge representation 57

2.2.1 Graph data representation 58

2.2.2 Sparse graph construction 59

2.3 Ranking-based approaches 61

2.3.1 Ranking based on scores 61

2.3.2 Ranking based on weighting algorithms 66

2.4 Feature selection applied to LBP 71

2.4.1 LBP bin selection 71

2.4.2 LBP histogram selection 73

2.5 Conclusion 79

As described in chapter 1, the LBP operators is a good candidate for local image texture descriptor. However, this operator tends to produce high dimensional feature vectors, especially when the number of considered neighboring pixels increases or when it is applied to color images. Thus, a dimensionality reduction method for LBP is needed to address this problem. Various approaches are proposed to obtain more discriminative, robust LBP-features with reduced feature dimensionality.

Indeed, many machine learning problems in computer vision and several related domains need to deal with very high dimensional data. Many of these features may not be relevant

for the final prediction task and degrade the classification performance. Multiple studies have shown that the classification performance can be improved by eliminating these features. These issues can be solved by the method of the dimensionality reduction. For this purpose, the dimensionality reduction can be achieved either by feature extraction or feature selection to a low dimensional space. Feature extraction refers to the methods that create a set of new features based on the linear or non-linear combinations of the original features. Further analysis is problematic since we cannot get the physical meanings of these features in the transformed space. Examples of feature extraction methods include Principal Component Analysis (PCA) [112], Locality Preserving Projections (LPP) [157]...

In contrast, the feature selection methods aims at finding adequate subsets of features by keeping some original features and therefore maintains the physical meanings of the features. The use of both methods have the advantage of improving performance of classification and increasing computational efficiency. Recently, feature selection has gained increasing interest in the field of machine learning [158, 159, 160, 161], data analysis [162, 163, 164], and successfully applied in computer vision such as information retrieval [165, 166, 167] or visual object tracking [168, 169, 170]. In this work, we focus on the application of feature selection methods to LBP-based features in the framework of color texture classification.

This chapter is organized as follow. We first present background information on the concept of feature selection and review works related to our research. We introduce the taxonomy of feature selection methods in section 2.1 by summarizing basic principles applied in feature selection and the context of feature selection. We present the data and knowledge representation by presenting the definitions and notations related to the feature selection methods in section 2.2. We then, review the related literature of the ranking-based approaches, including the ranking-based scores and ranking-based algorithms in section 2.3. Next, the feature selection applied to the local binary pattern is briefly reviewed in section 2.4. The bin selection and histogram selection approaches are discussed, including our first contribution for histogram selection. Finally, in section 2.5, we summarize the principal topics presented in this chapter.

2.1 Taxonomy of feature selection methods

Feature selection is defined as a process of selecting the features that best describe a dataset out of a larger set of candidate features. Typically, there are two types of features: relevant and irrelevant features [171]. In the framework of classification, relevant features are the features that contain discriminative information about the classes (supervised context) or clusters (unsupervised context). In contrast, irrelevant features are noisy and redundant features that cannot discriminate samples from different classes. Thus, removing irrelevant features reduces computational cost and improves the classification performance. For the classification prob-

lems, feature selection aims to select subsets of highly discriminant features while retaining a suitably high accuracy in representing the original features. In other words, it selects relevant features that are capable of discriminating data that belong to different classes. For example, in figure 2.1 (a), f_2 is a relevant feature because it can discriminate class A and class B while f_1 is irrelevant feature. In figure 2.1 (b) f_3 and f_4 are noisy features because they do not allow to discriminate the classes. In figure 2.1 (c), f_5 and f_6 are redundant features because they are highly correlated, we only need one of them to discriminate the class A and B. The terms of "feature selection" can be replaced by different synonyms in the literature: "variable selection", "attribute selection" and "feature ranking".

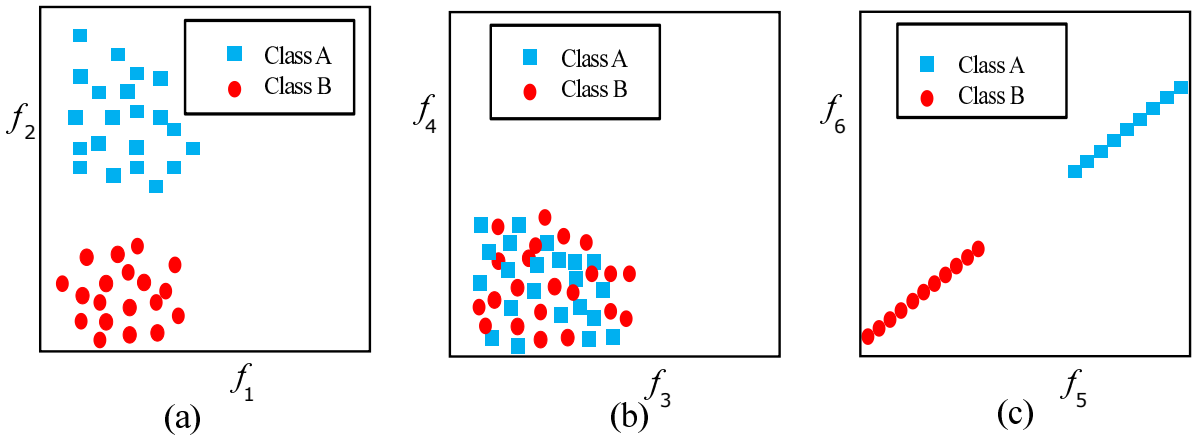


Figure 2.1: Examples to illustrate the concept of relevant, noisy and redundant features. (a) f_2 is a relevant feature which can discriminate the two classes A and B while f_1 is an irrelevant feature. (b) f_3 and f_4 are noisy features. (c) f_5 and f_6 are redundant features.

There are many feature selection approaches proposed in the literature. According to Dash and Liu, they generally involve four steps (as shown in figure 2.2) [162]:

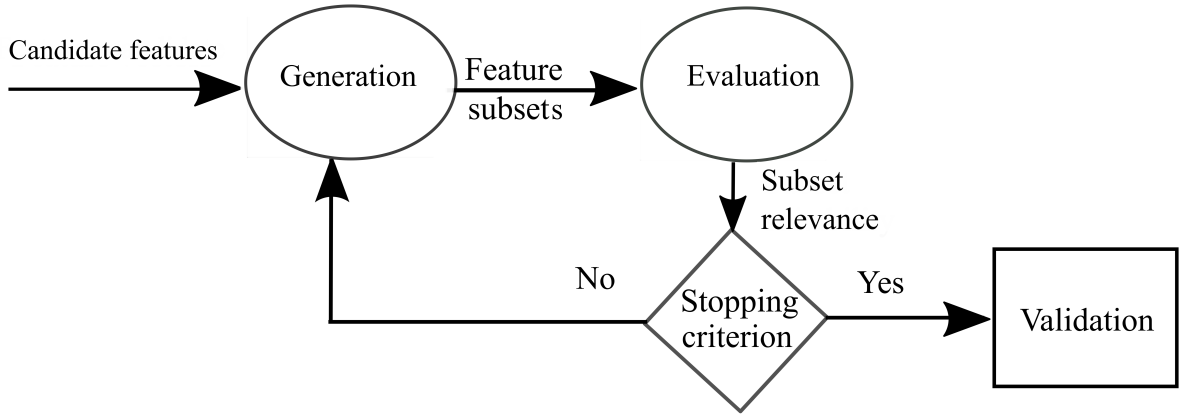


Figure 2.2: The different steps of feature selection [162].

1. **A generation step** which is based on a search method generates subsets of features to be evaluated. A subset search strategy generates candidate feature subsets in order to find the optimal subset. Usually, search strategies are usually categorized into complete, sequential, and random models [172]:

    - **Complete** search among all possible feature subsets. If the input data have $\mathcal{D}$ features, the most direct search strategy is the exhaustive search, i.e., search among all possible feature subsets ($2^{\mathcal{D}}$ in total). However, this approach might become computationally very expensive for high dimensional problems, because the size of the explored space corresponds to the number of all possible combinations of features.
    - **Sequential**: Starting with an empty set and subsequent addition of features is referred to as a bottom-up approach like the Sequential Forward Selection (SFS) approach. Or using the full set of features at the beginning and subsequent features removal is called a top-down approach like the Sequential Backward Selection (SBS) approach.
    - **Random**: Starting with a randomly selected feature set and adds randomly selected features or removes them from the set.

2. **An evaluation function** then calculates the relevance of the feature subset built during the generation step. It compares this with the previous best candidate subset, and then replacing it if found to be better. It can be either classifier independent (i.e., filter approach) or classifier dependent (i.e., wrapper approach or hybrid method) [173].

3. **A stopping criterion** decides when to stop. This step is executed every iteration to determine whether the feature selection process should continue or not. Without a suitable stopping criterion the feature selection process may run exhaustively through feature subset space. Generation step and evaluation functions can influence the choice of the stopping criterion. The stopping criteria based on a generation step include: (i) whether a predefined number of features is selected, and (ii) whether a predefined number of repetitions is reached. Stopping criteria based on an evaluation function include: (i) whether addition (or removal) of any features does not produce a better subset; and (ii) whether an optimal subset based on evaluation function is obtained.

4. **A validation step** verifies whether the feature subset is valid. Once the stopping criterion has been satisfied, the loop will be stopped and the resulting feature subset may be validated.

Feature selection methods can be categorized into one of two categories according to the context or the evaluation strategy, as shown in figure 2.3a and figure 2.3b, respectively. The subsequent sections will describe each strategy in detail.

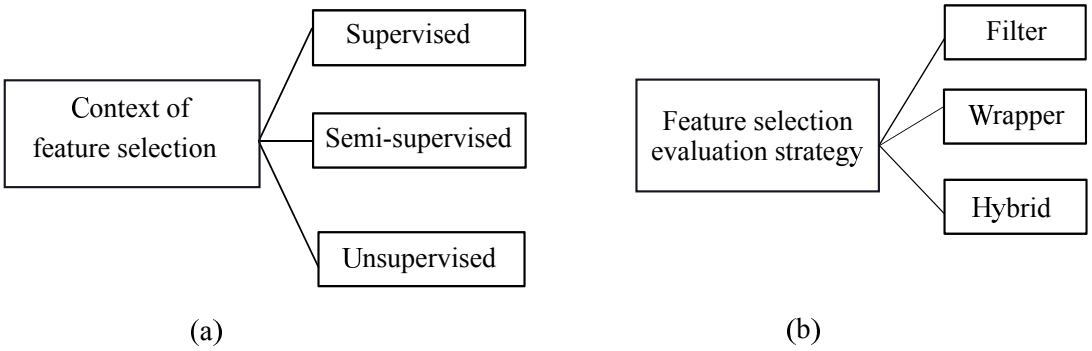


Figure 2.3: Categorization of feature selection (a) context and (b) evaluation strategy.

2.1.1 Context of feature selection

In terms of availability of supervised information, feature selection techniques can be roughly classified into three groups: supervised, unsupervised and semi-supervised methods [174]. Most of supervised and semi-supervised feature selection methods assess the relevances of features by the information of class label.

- **Supervised methods**: The availability of label information allows supervised feature selection algorithms able to discriminate samples from different classes. There are several literature reviews discussed on supervised feature selection [175, 176]. A general framework of supervised feature selection is illustrated in figure 2.4. The training step of the classification depends on the feature selection method. After splitting the data into training and testing sets, classifier is trained. This training step based on a subset of features selected by a feature selection method. It is worth to notice that the feature selection step can either be independent of classifier (filter methods), or it may take into account the performance of a classifier to assess the quality of selected features (wrapper methods). Finally, the classifier predicts class labels of the testing set based on the selected features. One challenge of this approach is the process of labeling the data given by the human user which is expensive and may be unreliable [140].

- **Unsupervised methods**: Unsupervised feature selection is a more challenging problem due to the absence of class label information used for guiding the search of discriminative features. Nevertheless, it has one advantage that it is unbiased by the labeling of data by human experts or data analysts. A general framework of unsupervised feature selection is illustrated in figure 2.5. Different from supervised feature selection, unsupervised feature selection usually uses all available data in the feature selection step. The feature selection step is either independent of the unsupervised learning algorithms (filter methods), or it relies on the learning algorithm to select features (wrapper methods). Unsupervised feature selection methods seek alternative criteria such as data similarity

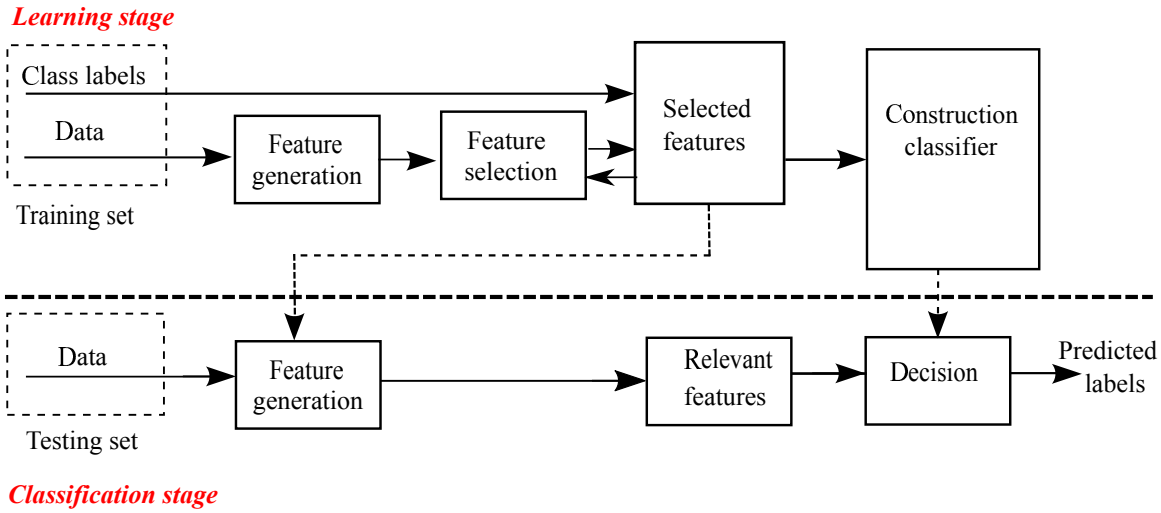


Figure 2.4: A general framework of supervised feature selection

and local discriminative information to define feature relevance [162, 177, 178, 179]. After the feature selection step, a clustering algorithm gives the output of cluster structure. The main drawbacks of the unsupervised approach are it neglects the possible correlation between different features and it relies on some criteria without the guarantee that the principles are universally valid for all types of data.

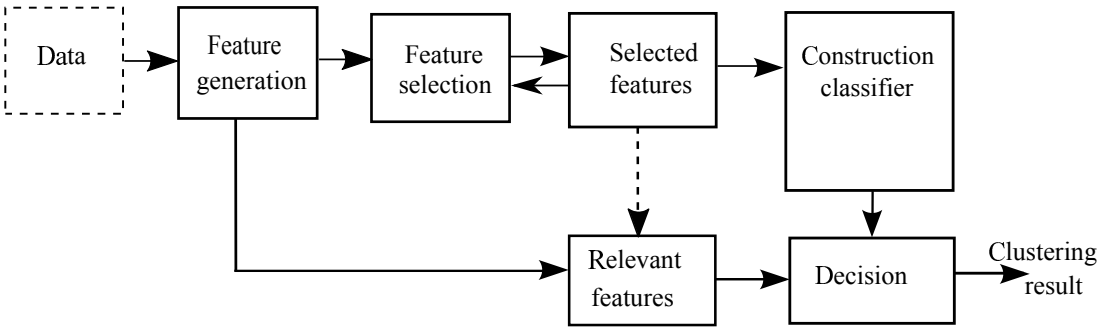


Figure 2.5: A general framework of unsupervised feature selection

- **Semi-supervised methods**: Semi-supervised feature selections are the extensions of supervised and unsupervised feature selections. In reality, the label training data is often limited or expensive to be obtained. When a small portion of data is labeled, we can utilize semi-supervised feature selection which can take advantages of both labeled data and unlabeled data. The general framework of semi-supervised feature selection is illustrated in figure 2.6. The only difference with the supervised feature selection method is the partial label information used as input. Usually, the labeled data is used to maximize the margin between data points of different classes, and the unlabeled data is used to discover the geometrical inherent structure of the data [162, 180, 181, 182, 183].

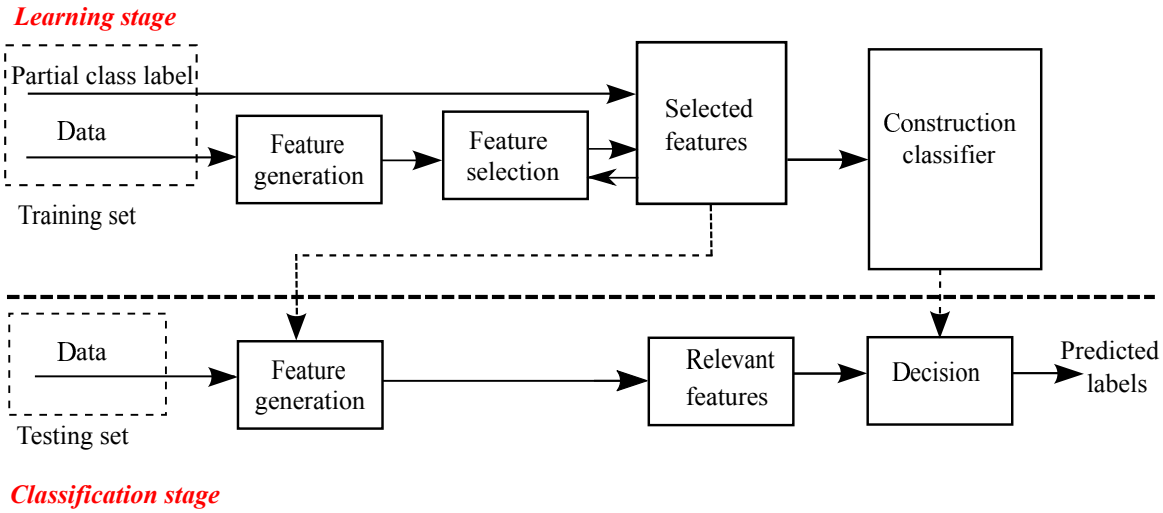


Figure 2.6: A general framework of semi-supervised feature selection

2.1.2 Feature selection evaluation

Based on the different strategies of evaluation, feature selection can be classified into three groups: filter, wrapper and hybrid methods [6].

- **Filter methods** select the subset of features as pre-processing step without involving the classifiers. Typical filter methods consist of two steps. In the first step, feature relevance is ranked by a feature score according to some feature evaluation criteria which can be either univariate or multivariate. In the univariate case, each feature is ranked individually regardless of other features, while the multivariate scheme ranks multiple features simultaneously. The methods rely solely on the inherent characteristics of data such as variance [133], correlation [6], mutual information [184, 177], consistency [162]. In the second step, lowly ranked features are filtered out and the remaining features are kept. Figure 2.7 describes a generalized form of a filter algorithm. Filter methods are fast and easy solutions, since they can be combined with any classifiers after the filtering is complete. However, they may miss features that are relevant for the target classifiers. Famous filter methods are based on Variance [133], Laplacian [179], Fisher scores [185] for univariate scheme and maximum Relevance (mRmR) [186], Inconsistency criterion [187] for multivariate scheme.

- **Wrapper methods** evaluate each candidate feature subset through the classification algorithm and using the estimated accuracy of the classification algorithm as its evaluation metric as shown in figure 2.8. They then select the most discriminative subset of features by minimizing the prediction error rate of a particular classifier. This step is a combinatorial problem, with an objective function that is costly to compute for high dimensional

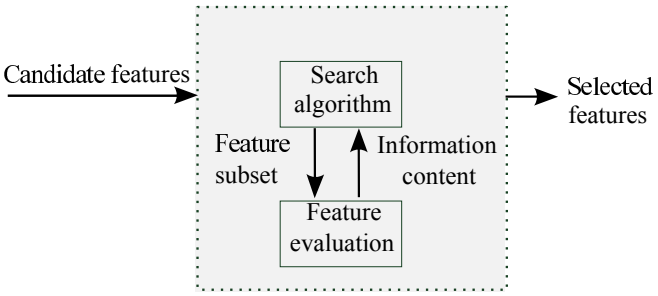


Figure 2.7: An illustration of filter method

features, but often give better results than other methods [188, 189, 190].

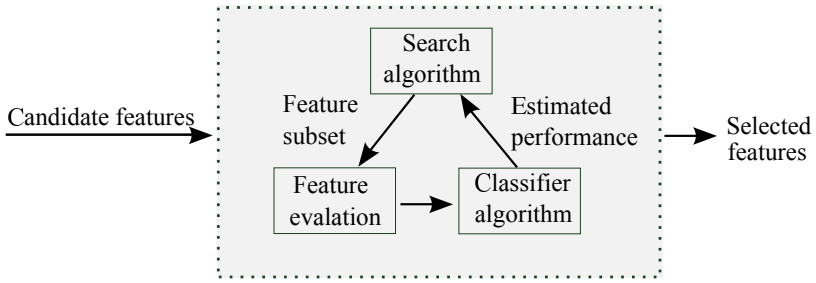


Figure 2.8: An illustration of wrapper method

- **Hybrid methods** combine both filter and wrapper methods into a single framework, in order to provide a more efficient solution to the feature selection problem [191].

  A filter evaluation can be applied to discard some features before going to the wrapper step. The key idea of this combination is to lower the complexity of a wrapper, but keeping its high accuracy. Figure 2.9 shows an illustration of the hybrid method. Dash and Liu proposed a first hybrid method that uses a measure based on the entropy of the similarity of the data (filter step) [187]. Then (wrapper step), they use a clustering algorithm and a scatter separability criterion for evaluating feature subsets. Recently, Solorio-Fernández et al. propose a method based on the Laplacian score ranking jointly with a modification of the Calinski–Harabasz index [192]. Hybrid methods have the disadvantage that they depend on the evaluation of filter methods used for determining the best feature subset.

The feature selection methods and their categorization have been introduced. Among of the feature selection approaches, we are interest in the hybrid methods which takes advantage of both the filters and the wrappers. In the following of this work, we propose a feature selection approach based on this approach for color texture classification in the supervised context.

Before introducing the state-of-the-art of feature selection methods, we will give some definitions and notations used for the representation of data and knowledge in the next section.

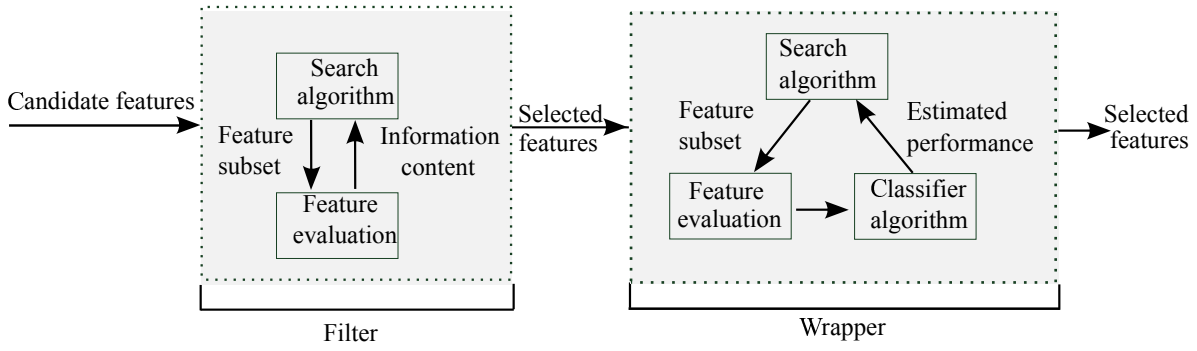


Figure 2.9: An illustration of hybrid method

2.2 Data and knowledge representation

In the feature selection context related to our problem, we dispose a dataset of N color texture images defined in a $\mathcal{D}$-dimensional feature space. Italic letters are used to denote scalars, bold letters to denote vectors or matrices (e.g., x, $\mathbf{x}$, $\mathbf{X}$). We denote $\mathbf{X} = (x_i^r)$, $i \in \{1,...,N\}$; $r \in \{1,...,\mathcal{D}\}$; the associated data matrix represented by equation 2.1, where x_i^r is the rth feature value of the ith color image I_i.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ ... \\ \mathbf{x}_i \\ ... \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_1^1 & ... & x_1^r & ... & x_1^{\mathcal{D}} \\ ... & ... & ... & ... & ... \\ x_i^1 & ... & x_i^r & ... & x_i^{\mathcal{D}} \\ ... & ... & ... & ... & ... \\ x_N^1 & ... & x_N^r & ... & x_N^{\mathcal{D}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^1 & ... & \mathbf{f}^r & ... & \mathbf{f}^{\mathcal{D}} \end{bmatrix} \tag{2.1}$$

Each of the N rows of the matrix $\mathbf{X}$ represents a color texture $\mathbf{x}_i = (x_i^1,...,x_i^r,...,x_i^{\mathcal{D}}) \in \mathbb{R}^{\mathcal{D}}$, while each of the $\mathcal{D}$ columns of $\mathbf{X}$ represents the feature vector $\mathbf{f}^r$, defined as follows:

$$\mathbf{f}^r = \begin{bmatrix} x_1^r \\ ... \\ x_i^r \\ ... \\ x_N^r \end{bmatrix} = \begin{bmatrix} f_1^r \\ ... \\ f_i^r \\ ... \\ f_N^r \end{bmatrix} \tag{2.2}$$

In supervised learning, all the information about the class labels of the training images are available. Let us denote the vector $\mathbf{y}$ the class labels of the different images defined by:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ ... \\ y_i \\ ... \\ y_N \end{bmatrix} \tag{2.3}$$

where $y_i \in \{1,..,c,...,C\}$, C is the number of classes of the data. For each color image I_i, its feature vector $\mathbf{x}_i$ is associated with a class label y_i.

Several studies have shown that graphs constructed in original feature space reflect some intrinsic properties of data, and thus can be used for dimension reduction [193, 194]. Moreover, the spectral graph theory represents a solid theoretical framework which has been the basis of many effective existing feature selection methods. The feature selection methods that we present in this section are based in large part on spectral graph theory [195] including Laplacian score [179, 196], spectral methods [177] and sparsity score [197] and so on. All of these methods used the application of graph matrix in the objective of feature selection. In the next section, we introduce several of graph construction related to those approaches.

2.2.1 Graph data representation

Given a dataset $\mathbf{X}$, let $G = (V,E)$ be the undirected graph constructed from $\mathbf{X}$, where $V = \{v_1,...,v_N\}$ is its vertex set and E is the set of edges. Each vertex v_i in this graph represents an image $\mathbf{x}_i$ and each edge between two vertices v_i and v_j carries a non-negative weight $s_{ij} \geq 0$. The similarity matrix of the graph is the matrix $\mathbf{S} = (s_{ij})_{i,j=1,...,N}$. As G is an undirected graph, $s_{ij} = s_{ji}$. There are many approaches to transform a given dataset with the pairwise similarities s_{ij} into a graph. The most common graph construction methods including ε-neighborhood graph, k-nearest neighbor graph and fully connected graph [198, 199, 200].

- **ε-neighborhood graph**: The neighbors of a given instance $\mathbf{x}_i$ are the instances that belong to a sphere centered at $\mathbf{x}_i$ and having ε as radius. In ε-neighborhood graph the data which have the distance (similarity) less than the threshold ε.

- ***k*-nearest neighbor graph**: An edge between two vertices v_i and v_j is constructed if the corresponding instances $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, i.e. $\mathbf{x}_i$ is among the k-nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the k-nearest neighbors of $\mathbf{x}_i$.

- **Fully connected graph**: We connect all images with positive similarity with each other and we weight all edges by s_{ij}. As the graph should represent the local neighborhood relationships, this construction is only useful if the similarity function itself models local neighborhoods.

  There are several choices for this similarity. Belkin and Niyogi use the heat kernel with different Gaussian variance σ values, as follows [199]:

$$s_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{2.4}$$

where the parameter σ is a constant to be set which controls the width of the neighborhood and $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. In the extreme case where $\sigma \to \infty$ the weights will become 1.

Cortes and Mohri propose the use of inverse of distance as weight [200] as follows:

$$s_{ij} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|}, \quad \mathbf{x}_i \neq \mathbf{x}_j \tag{2.5}$$

The similarity function can also be expressed under cosine function. It is used to measure the similarity between two vectors by computing the cosine of the angle between them as defined as follows:

$$s_{ij} = |cos(\mathbf{x}_i, \mathbf{x}_j)| = \frac{|\mathbf{x}_i^T \mathbf{x}_j|}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|} \tag{2.6}$$

The degree d_i of a vertex $v_i \in V$ is equal to the sum of weights of all edges linked to this vertex. It is defined the diagonal matrix $\mathbf{D} = (d_i)_{i=1...N}$ should be noted as follows:

$$d_i = \sum_{j=1}^{N} s_{ij} \tag{2.7}$$

It is worth to note that the degree d_i of a node i can be considered as a local density measure at $\mathbf{x}_i$.

The Laplacian matrix $\mathbf{L}$ of $\mathbf{X}$ is defined by:

$$\mathbf{L} = \mathbf{D} - \mathbf{S} \tag{2.8}$$

The drawback of all the graphs mentioned above is their dependence on the value of the parameters ε, k, or σ. Without the optimum value of these parameters, the similarity matrix constructed could not reflect the real similarity among data points. For example, the adjustable parameter σ in equation 2.4 plays an important role in the performance of the function, and should be carefully tuned by hand according to the problem. If overestimated, the exponential will behave almost linearly, and the projection will lose its nonlinear power. In contrast, if underestimated, the function will lack the regularization and will be highly sensitive to noise so that it might change the graph structure.

Recently, to address the limitations of classical graph construction methods, sparse representation has been successfully used for graph construction [201, 202]. In the following, we will briefly present the sparse graph construction.

2.2.2 Sparse graph construction

Sparse representation has received a great deal of attention in computer vision, especially in image representation in the recent years. It has many effective applications such as image

compression and coding [203, 204, 205, 206], pattern recognition, image and signal processing [207, 208, 209]. Generally, sparse representation helps to find the most compact representation of the original data.

Recently, Qiao et al. presented a new method to design the similarity matrix based on the modified sparse representation [210]. The graph adjacency structure and corresponding graph weights are built simultaneously by the l_1-norm minimization problem. This is, in fact, a new way that is fundamentally different from the traditional ones (like Euclidean distance, cosine distance, etc.) to measure the similarity between different data points. By introducing sparsity in the linear reconstruction process, it identifies the most relevant data points as well as their estimated similarity to the reconstructed data point. Moreover, many empirical results have shown that a sparse graph is preferred, because sparse graphs have much less spurious connections between dissimilar points and lead to exhibit high quality for data representation [211, 210, 197, 212, 201, 213].

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_i, .., \mathbf{x}_N]^T \in \mathbb{R}^{\mathcal{D}\times N}$ including all the instances in its columns, we want to reconstruct each instance $\mathbf{x}_i$, e.g., a color texture image, using as few entries of $\mathbf{X}$ as possible. This problem can be expressed mathematically as follows:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_0 \quad s.t. \quad \mathbf{x}_i = \mathbf{X}\mathbf{s}_i, \tag{2.9}$$

where $\mathbf{s}_i = [s_{i1}, ..., s_{i(i-1)}, 0, s_{i(i+1)}, ..., s_{iN}]^T$ is an N-dimensional coefficients vector in which the ith element is equal to zero (implying that $\mathbf{x}_i$ is removed from $\mathbf{X}$) and the elements $s_{ij}(i \neq j)$ denotes the contribution of each $\mathbf{x}_j$ to reconstruct $\mathbf{x}_i$, $\|.\|_0$ denotes the l_0-norm, which is equal to the number of non-zero components in $\mathbf{s}_i$.

It is worth to note that the solution of equation 2.9 is NP-hard in general case. A sparse vector $\mathbf{s}_i$ can be approximately solved by the following modified l_1-minimization problem:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1 \quad s.t. \quad \mathbf{x}_i = \mathbf{X}\mathbf{s}_i, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{2.10}$$

where, $\|.\|_1$ denotes the l_1-norm, ; $\mathbf{1} \in \mathbb{R}^N$ is a vector of all ones values.

In reality, the constraint $\mathbf{x}_i = \mathbf{X}\mathbf{s}_i$ in equation 2.10 does not always hold due to the presence of noises. The modified objective function is defined as follows [204] :

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1 \quad s.t. \quad \|\mathbf{x}_i - \mathbf{X}\mathbf{s}_i\|_2 < \xi, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{2.11}$$

where ξ represents a given error tolerance. The sparse vector $\mathbf{s}_i$ is computed for each sample $\mathbf{x}_i$. The optimal solution of equation. 2.11 for each sample $\mathbf{x}_i$ is a sparse vector $\hat{\mathbf{s}}_i$, that allows to build the sparse similarity matrix $\mathbf{S} = (\hat{s}_{i,j})_{N\times N}$, defined by:

$$\mathbf{S} = [\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_i, ..., \hat{\mathbf{s}}_N]^T \tag{2.12}$$

The l_1-minimization problem can be solved in polynomial time by standard linear programming method [214] using publicly available packages such as l_1-toolbox[1]. As the vector $\hat{\mathbf{s}}_i$ is sparse and a lot of its components are zero and few of them have non zero values, the instances in the dataset which are far from the input signal will have very small or zero coefficients. This solution might reflect the intrinsic geometric properties of original data. In cases of the absence of the class label information, the discriminative information can be naturally preserved in the matrix $\mathbf{S}$.

We have presented the definitions and notations related to the following of this chapter. Different strategies have been proposed over the last years for feature selection: filter, wrapper and hybrid methods. Among of them, hybrid methods attempt to have a reasonable compromise between efficiency (computational effort) and effectiveness (by selecting the relevance features). To combine the filter and wrapper methods into a hybrid methods, we are interest the filter-based approach in the univariate scheme by using some criteria to assess each feature sorting them into a list (ranking). The next section is entirely dedicated to ranking-based approaches.

2.3 Ranking-based approaches

The aim of feature ranking is to measure the relevance of features in order to find the most discriminative feature. Among a huge literature on feature ranking methods, we will briefly review several well-known approaches. These approaches can be achieved by associating a score for each feature or by applying an algorithm which gives weights for a feature subset. In the first case, the features are independently evaluated, whereas in the second case, the weight of each feature is determined using all attributes.

2.3.1 Ranking based on scores

The filter approaches selects the relevant features by looking at the inherent properties of the data. In most cases, feature relevance score is individually calculated. In this section, we introduce several score functions based on feature ranking methods according to the learning context in supervised, unsupervised and semi-supervised.

2.3.1.1 Unsupervised feature selection

The unsupervised feature selection methods evaluate the relevance of features based on various types of criteria such as distance, information, correlation, dependency and so on. These

[1]The web site (http://www.ece.ucr.edu/sasif/homotopy/index.html) provides many practical toolboxes and several research works to solve the sparse representation problem. In our experiments, l_1-toolbox is used due to its simplicity.

methods include Variance, Laplacian and Unsupervised sparsity scores.

- **Variance score** uses the variance along a dimension to reflect its representative power and selects the features with the maximum variance. Let f_i^r denote the rth feature of the ith image. The mean of the rth feature $\mathbf{f}^r$ is defined as:

$$\mu^r = \frac{1}{N}\sum_{i=1}^{N} f_i^r \tag{2.13}$$

  The variance of the rth feature denoted as $Variance^r$, which should be maximized, is calculated as follows [133]:

$$Variance^r = \frac{1}{N}\sum_{i=1}^{N}(f_i^r - \mu^r)^2 \tag{2.14}$$

  The features are sorted according to the ascending order of $Variance^r$ to select the most relevant ones.

- **Laplacian score** assumes that instances from the same class are close to each other and the local geometric structure is crucial for discrimination [179]. This score selects features with larger variances which have more representative power and stronger locality preserving ability. The Laplacian score of the rth feature denoted $Laplacian^r$ which should be minimized, is computed as follows:

$$Laplacian^r = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(f_i^r - f_j^r)^2 s_{ij}}{\sum_{i=1}^{N}(f_i^r - \mu^r)^2 d_i} = \frac{\tilde{\mathbf{f}}^{r^T}\mathbf{L}\tilde{\mathbf{f}}^r}{\tilde{\mathbf{f}}^{r^T}\mathbf{D}\tilde{\mathbf{f}}^r} \tag{2.15}$$

  where s_{ij} is defined by the similarity relationship between two images I_i and I_j as defined in section 2.2 and $\tilde{\mathbf{f}}^r$ is defined as:

$$\tilde{\mathbf{f}}^r = \mathbf{f}^r - \frac{\mathbf{f}^{r^T}\mathbf{D1}}{\mathbf{f}^{r^T}\mathbf{D1}}\mathbf{1} \tag{2.16}$$

  where $\mathbf{1} \in \mathbb{R}^N$ is a vector of all ones.

  After calculating the Laplacian score for each feature, they are sorted in the ascending order of $Laplacian^r$ to select the relevant ones.

- **Unsupervised sparsity score** is another feature ranking algorithm proposed by Liu et al. [197]. This approach is based on sparse similarity matrix construction. The proposed unsupervised sparsity score of the rth feature denoted $UnsupSparse^r$, which should be minimized, is defined as follows:

$$UnsupSparse^r = \frac{\sum_{i=1}^{N}(f_i^r - \sum_{j=1}^{N}\hat{s}_{ij}f_j^r)^2}{\frac{1}{N}\sum_{i=1}^{N}(f_i^r - \mu^r)^2} = \frac{\mathbf{f}^{r^T}(\mathbf{I}-\mathbf{S}-\mathbf{S}^T+\mathbf{S}\mathbf{S}^T)\mathbf{f}^r}{\mathbf{f}^{r^T}(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{f}^r} \tag{2.17}$$

where $\mathbf{I}$ is an identity matrix and $\hat{s}_{ij}$ is the entry of the sparse similarity matrix $\mathbf{S}$ constructed using all images which is presented in section 2.2.2.

The sparsity score for each feature is sorted in the ascending order of $UnsupSparse^r$ in order to select the relevant ones.

2.3.1.2 Supervised feature selection

Supervised feature selection evaluates the relationship between the features and their class labels information. Given data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_i, .., \mathbf{x}_N]$, $\mathbf{x}_i \in \mathbb{R}^{\mathcal{D}\times N}$, each image I_i is associated with a class label y_i, $\{\mathbf{x}_i, y_i\}$, $y_i \in \{1, .., c, .., C\}$, where C is the number of classes and N_c denotes the number of instances in the class c. Based on that notation, we introduce several feature selection methods in the supervised context.

- **Fisher score** is one of the most widely used supervised feature selection score. The principal idea of Fisher score is to identify a subset of features so that the distances between samples in different classes are as large as possible, while the distances between samples in the same class are as small as possible.

  Let μ^r denotes the mean of all instances on the rth feature, μ^{rc} and $(\sigma^{rc})^2$ the mean and variance of class c corresponding to the rth feature, respectively. The Fisher score of the rth feature, which should be maximized, is calculated as follows [133]:

$$Fisher^r = \frac{\sum_{c=1}^{C} N_c(\mu^{rc} - \mu^r)^2}{\sum_{c=1}^{C} N_c(\sigma^{rc})^2} \tag{2.18}$$

  where, the numerator is the between-class variance considering the rth feature and the denominator is the within-class variance considering the rth feature.

  After calculating the Fisher score for each feature, they are sorted in the ascending order to select the relevant ones.

- **Supervised Laplacian score**: The Laplacian score which is based on concepts from spectral feature selection, identifies relevant features by measuring their capability of preserving instance similarity. Spectral feature selection also provides a framework for supervised and unsupervised feature selection [177]. When class label information is available, the similarity matrix can be directly formed from label information. The following function is usually used for constructing a similarity matrix $\mathbf{S}$ in a supervised way [196]:

$$s_{ij} = \begin{cases} 1 & \text{if } y_i = y_j = c, \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

We obtain the supervised Laplacian score of the rth feature denoted $SupLaplacian^r$ from equation 2.15 by using the similarity matrix defined by equation 2.19.

Moreover, the relationship between supervised Laplacian and Fisher score can be formulated as follows:

$$SupLaplacian^r = \frac{1}{1 + Fisher^r} \tag{2.20}$$

where $Fisher^r$ is the Fisher score defined by equation 2.18.

Recently, Dornaika and Bosaghzadeh propose another supervised Laplacian score based on locality discrimination [215]. Two undirected weighted graphs G_w and G_b are constructed. The graph G_w reflects the within-class relationship, i.e., it encodes the pairwise similarity and relation associated with samples having the same label. The graph G_b reflects the between-class or global similarity relationship. It encodes the pairwise similarity and relation among heterogeneous samples. The graphs G_w and G_b are characterized by the weight matrices $\mathbf{W}_w$ and $\mathbf{W}_b$, respectively. In this case, the supervised Laplacian score of the rth feature, denoted $SupLaplacian_2^r$, is given by:

$$SupLaplacian_2^r = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(f_i^r - f_j^r)^2 (W_b)_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N}(f_i^r - f_j^r)^2 (W_w)_{ij}} = \frac{\mathbf{f}^{r^T}\mathbf{L}_b\mathbf{f}^r}{\mathbf{f}^{r^T}\mathbf{L}_w\mathbf{f}^r} \tag{2.21}$$

where $\mathbf{L}_b$ and $\mathbf{L}_w$ are the Laplacian matrices of the graphs G_b and G_w, respectively.

- **Supervised sparsity score**: Liu et al. extend the unsupervised sparsity score to supervised context by utilizing the class label information [197]. Let f_i^{rc} denotes the rth feature of ith intance in class c, $\hat{s}_{ij}^c$ is the element of sparse similarity matrix $\mathbf{S}^c$ which is constructed within the class c, $\mathbf{e}^c$ is a N-dimensional vector with $\mathbf{e}^c(i) = 1$, if I_i belongs to the class c and 0 otherwise. The proposed supervised sparsity score of the rth feature, denoted $SupSparse_r$, which should be minimized, is defined as follows:

$$SupSparse^r = \frac{\sum_{c=1}^{C}\sum_{i=1}^{N_c}(f_i^{rc} - \sum_{j=1}^{N_c}\hat{s}_{ij}^c f_j^{rc})^2}{\sum_{c=1}^{C}\sum_{i=1}^{N_c}(f_i^{rc} - \mu^{rc})^2} = \frac{\sum_{c=1}^{C}\mathbf{f}^{rc^T}(\mathbf{I} - \mathbf{S}_c - \mathbf{S}_c^T + \mathbf{S}_c\mathbf{S}_c^T)\mathbf{f}^{rc}}{\mathbf{f}_r^T(\mathbf{I} - \sum_{c=1}^{C}\frac{1}{N_c}\mathbf{e}^c\mathbf{e}^{c^T})\mathbf{f}^r} \tag{2.22}$$

After calculating the score for each feature, they are sorted in the ascending order of $SupSparse_r$ to select the relevant ones. In the classification experiments, Liu et al. have demonstrated that this score outperforms other methods in most cases, especially for multi-class problems [197].

2.3.1.3 Semi-supervised feature selection

In reality, the full class label is often difficult to obtain. In semi-supervised learning, a dataset of N color texture images $\mathbf{X}$ consists of two subsets depending on the label availability: $\mathbf{X}_U = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, ..., \mathbf{x}_{l+u}\}_{u\neq 0}$, which are unlabeled and $\mathbf{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l\}_{l\neq 0}$ for which the labels $\mathbf{y}_L = \{y_1, y_2, ..., y_l\}$ are provided and $N = l + u$. On the other hand, there is another semi-supervised information such as pairwise constraints. The pairwise constraints specify whether a pair of two images belong to the same class (must-link constraints) or different classes (cannot-link constraints). The set of must-link constraints ($\mathcal{M}$) and the set of cannot-link constraints ($\mathcal{C}$) of the dataset $\mathbf{X}$ are defined as follows in [180]:

- $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \quad \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\}$
- $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \quad \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to different classes}\}$

Semi-supervised feature selection methods based on pairwise constraints use both pairwise constraints described in Section 2.2 to evaluate the relevance of features according to their constraint and locality preserving the local data structure. Two graphs $G^{\mathcal{M}}$ and $G^{\mathcal{C}}$ are constructed by using the instances of $\mathcal{M}$ and $\mathcal{C}$ respectively. If two nodes are must-link (or cannot-link), an edge is created in the graph $G^{\mathcal{M}}$ (or $G^{\mathcal{C}}$). The similarity matrix of $G^{\mathcal{M}}$ and $G^{\mathcal{C}}$ is defined as follows:

$$s_{ij}^{\mathcal{M}} = \begin{cases} 1 & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

$$s_{ij}^{\mathcal{C}} = \begin{cases} 1 & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

Two constraint scores are proposed including constraint score-1 (CS_1) and constraint score-2 (CS_2) by Zhang et al [180]. The constraint scores of the rth feature denoted CS_1^r and CS_2^r, which should be minimized, are calculated as follows [180]:

$$CS_1^r = \frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}}(f_i^r - f_j^r)^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}}(f_i^r - f_j^r)^2} = \frac{\mathbf{f}^{r^T}\mathbf{L}^{\mathcal{M}}\mathbf{f}^r}{\mathbf{f}^{r^T}\mathbf{L}^{\mathcal{C}}\mathbf{f}^r} \tag{2.25}$$

$$CS_2^r = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} (f_i^r - f_j^r)^2 - \lambda \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} (f_i^r - f_j^r)^2 = \mathbf{f}^{r^T}\mathbf{L}^{\mathcal{M}}\mathbf{f}^r - \lambda \mathbf{f}^{r^T}\mathbf{L}^{\mathcal{C}}\mathbf{f}^r \tag{2.26}$$

where λ is a parameter to balance the two terms in 2.26.

Kalakech et al. propose another semi-supervised score which uses both pairwise constraints and the local properties of the unlabeled data [181, 140]. The Laplacian and Fisher scores have also been extended in the semi-supervised learning context in [216]. More recently, Liu and

Zhang propose a pairwise constraint-guided sparse learning method for feature selection, where the must-link and the cannot-link constraints are used as discriminative regularization terms that directly concentrate on the local discriminative structure of data [217]. A comprehensive survey of feature selection methods in the semi-supervised context is introduced by Sheikhpour et al [183].

The filter methods based on scores are introduced. The following section presents the two well-known algorithms Relief and Simba for features ranking which are principally based on the largest margin concept.

2.3.2 Ranking based on weighting algorithms

Largest margin concept is very important in the statistical pattern recognition, because it measures confidence of a classifier with respect to its predictions. There are two approaches of describing a margin [218]:

- **The sample-margin** measures the distance between an instance and a decision boundary induced by the classifier. For example Support Vector Machines [118] is a classification algorithm that represents the images as points in space, mapped so that the images of different classes are divided by a clear gap (sample-margin) that is as wide as possible.

- **The hypothesis-margin**: Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_N]$ be a training data set where $\mathbf{x}_i = (x_i^1, ...x_i^2, ..., x_i^{\mathcal{D}})^T$ is the ith representation of an image that contains $\mathcal{D}$ features and y_i its related label.

  The concepts of *nearhit* and *nearmiss* were used in general before the notion of margin. *nearhit* or $NH(\mathbf{x}_i)$ of an image $\mathbf{x}_i$ is the nearest image to $\mathbf{x}_i$ having the same label and the *nearmiss* or $NM(\mathbf{x}_i)$ of an image $\mathbf{x}_i$ is nearest image to $\mathbf{x}_i$ having a different label. Figure 2.10 illustrated the NH and NM concepts.

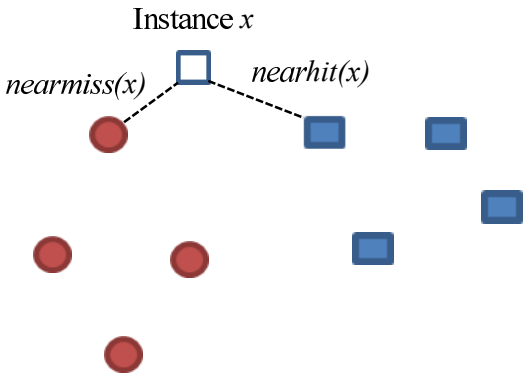


Figure 2.10: Illustration of the *nearhit* and *nearmiss* concepts.

Hypothesis margin of $\mathbf{x}_i$ denoted by ρ_i is calculated as the difference between the distance to its $NM(\mathbf{x}_i)$ and the distance to its $NH(\mathbf{x}_i)$. When multiple images have large margin, they can move considerably in the feature space without altering the labeling structure of the data set. Hence, a large margin insures high confidence when a classifier is making its decision. Moreover, a feature's contribution to the maximization of hypothesis margin reflects its ability to discriminate the data into different classes.

Recently, various feature selection algorithms have been developed under the large margin principles including SVM-based feature selection and Relief family (1-NN based) algorithms. The feature selection methods based on sample-margin need high computational cost for a high-dimensional data sets [219]. So, we focus on feature selection algorithms developed under the hypothesis margin concept. Two most discussed hypothesis margin methods are the Relief algorithm [220] and Simba algorithm [221] that will be presented in the following. Given a distance function $\|.\|$, a margin ρ_i of $\mathbf{x}_i$ is computed as:

$$\rho_i = \|\mathbf{x}_i - NM(\mathbf{x}_i)\| - \|\mathbf{x}_i - NH(\mathbf{x}_i)\| \tag{2.27}$$

Similarly the margin over the dataset $\mathbf{X}$ is computed as:

$$\rho = \sum_{i=1}^{N} \rho_i \tag{2.28}$$

One natural idea is to scale each feature by a non-negative vector $\mathbf{w}$ to obtain a weighted feature space such that a margin-based function in this induced feature space is maximized.

$$\rho_i(\mathbf{w}) = \|\mathbf{x}_i - NM(\mathbf{x}_i)\|_{\mathbf{w}} - \|\mathbf{x}_i - NH(\mathbf{x}_i)\|_{\mathbf{w}} \tag{2.29}$$

Thus the weighted margin over the dataset $\mathbf{X}$ is computed as

$$\rho(\mathbf{w}) = \sum_{i=1}^{N} \rho_i(\mathbf{w}) \tag{2.30}$$

2.3.2.1 Relief algorithm

The Relief algorithm is based on a measure of relevance of each feature by maximizing a margin-based objective function [220]. If we specify the distance function $\|.\|$ by L1-norm (or Manhattan distance), we obtain the well-known Relief algorithm. The hypothesis margin of an instance $\mathbf{x}_i$ is defined as:

$$\rho_i = \|\mathbf{x}_i - NM(\mathbf{x}_i)\|_1 - \|\mathbf{x}_i - NH(\mathbf{x}_i)\|_1 \tag{2.31}$$

where L1-norm is defined as:

$$\|\mathbf{z}\|_1 = \sum_{r=1}^{\mathcal{D}} |z^r| = |z^1| + ... + |z^r| ... + |z^{\mathcal{D}}| \tag{2.32}$$

The hypothesis margin over the whole dataset $\mathbf{X}$ is computed as:

$$\rho = \sum_{i=1}^{N} \rho_i \tag{2.33}$$

The weighted margin of an instance $\mathbf{x}_i$ is defined as:

$$\rho_i(\mathbf{w}) = \|\mathbf{x}_i - NM(\mathbf{x}_i)\|_{\mathbf{w}} - \|\mathbf{x}_i - NH(\mathbf{x}_i)\|_{\mathbf{w}} \tag{2.34}$$

where,

$$\|\mathbf{z}\|_{\mathbf{w}} = \sum_{i=1}^{\mathcal{D}} w^r |z^r| = w^1|z^1| + ... + w^r|z^r| + ... + w^{\mathcal{D}}|z^{\mathcal{D}}| \tag{2.35}$$

The weighted margin over the whole dataset $\mathbf{X}$ is computed as

$$\rho(\mathbf{w}) = \sum_{i=1}^{N} \rho_i(\mathbf{w}) \tag{2.36}$$

The objective is to find a weighted vector $\mathbf{w}$, which maximizes the evaluation function defined as the weighted hypothesis margin. If we consider the following notation:

$$\mathbf{M} = \begin{pmatrix} |x_i^1 - NM(x_i^1)| \\ ... \\ |x_i^{\mathcal{D}} - NM(x_i^{\mathcal{D}})| \end{pmatrix} - \begin{pmatrix} |x_i^1 - NH(x_i^1)| \\ ... \\ |x_i^{\mathcal{D}} - NH(x_i^{\mathcal{D}})| \end{pmatrix} = \begin{pmatrix} M^1 \\ ... \\ M^{\mathcal{D}} \end{pmatrix} \tag{2.37}$$

The evaluation function related to the weighted margin of the image $\mathbf{x}_i$ becomes:

$$\rho_i(\mathbf{w}) = \mathbf{w}^T\mathbf{M} = w^1M^1 + ...w^rM^r + ... + w^{\mathcal{D}}M^{\mathcal{D}} \tag{2.38}$$

The gradient of the evaluation function is given by:

$$\frac{\partial \rho_i(w)}{\partial w^r} = \frac{\partial w^T\mathbf{M}}{\partial w^r} = M^r \tag{2.39}$$

And the updating equation is:

$$w^r_{new} = w^r_{old} + \frac{\partial \rho_i(\mathbf{w})}{\partial w^r} \tag{2.40}$$

The basic Relief algorithm is given by algorithm 1. The method randomly select T images from the training set and updates the relevance of each feature based on the difference between the selected image and the two nearest instances of the same and different classes. The expected weight is large for relevant features and small for irrelevant ones under some assumptions.

The output is a weight vector, with a weight w^r corresponding of the rth feature. This vector is a ranking list of the features. The threshold value τ can be defined by user to select the most discriminant features. Moreover, Kira et al proposed a relevancy threshold τ to get a subset selection algorithm by a a statistical mechanism which ensures the probability that a given irrelevant feature will be chosen is small [220].

Algorithm 1 Relief algorithm

1. Initialize the weight vector to zero $\mathbf{w} = (0, 0, ..., 0)$
2. For $t = 1, ..., T$
   (a) Pick randomly an instance $\mathbf{x}$ from $\mathbf{X}$
   (b) Find $NH(\mathbf{x})$ and $NM(\mathbf{x})$
   (c) For $r = 1, ..., \mathcal{D}$, calculate
   $\Delta^r = |x^r - NM(x^r)| - |x^r - NH(x^r)|$
   $w^r = w^r + \Delta^r$
   End For

   End For
3. The chosen feature set is $\{r|w^r > \tau\}$ where τ is a fixed threshold

2.3.2.2 Simba algorithm

Simba algorithm is an iterative algorithm proposed by Gilad-Bachrach et al. [221]. If we specify the distance function $\|.\|$ by l_2 norm (or Euclidean distance) like the one used in Simba algorithm, we obtain the following margin of an instance $\mathbf{x}_i$

$$\rho_i = \frac{1}{2}(\|\mathbf{x}_i - NM(\mathbf{x})_i\|_2 - \|\mathbf{x}_i - NH(\mathbf{x})_i\|_2) \tag{2.41}$$

where the l_2 norm is defined as

$$\|\mathbf{z}\|_2 = \sqrt{\sum_{r=1}^{\mathcal{D}} z^{r2}} \tag{2.42}$$

The margin based on Euclidean distance over the whole dataset $\mathbf{X}$ is computed as:

$$\rho = \sum_{i=1}^{N} \rho_i = \sum_{i=1}^{N} \frac{1}{2}(\|\mathbf{x}_i - NM(\mathbf{x}_i)\|_2 - \|\mathbf{x}_i - NH(\mathbf{x}_i)\|_2) \tag{2.43}$$

The weighted margin of an image $\mathbf{x}_i$ is computed as:

$$\rho_i(w) = \frac{1}{2}(\|\mathbf{x}_i - NM(\mathbf{x}_i)\|_2 - \|\mathbf{x}_i - NH(\mathbf{x}_i)\|_2) \tag{2.44}$$

where,

$$\|\mathbf{z}\|_{\mathbf{w}} = \sqrt{\sum_{r=1}^{\mathcal{D}} w^{r2} z^{r2}} \tag{2.45}$$

The gradient of the evaluation function is given by:

$$\frac{\partial \rho_i(\mathbf{w})}{\partial w^r} = \frac{1}{2}\left[\frac{(x_i^r - NM(x_i^r))^2}{\|x_i - NM(x_i)\|_{\mathbf{w}}} - \frac{(x_i^r - NH(x_i^r))^2}{\|x_i - NH(x_i)\|_{\mathbf{w}}}\right] w^r \tag{2.46}$$

And the updating equation is:

$$w^r_{new} = w^r_{old} + \frac{1}{2}\left[\frac{(x_i^r - NM(x_i^r))^2}{\|x_i - NM(x_i)\|_{\mathbf{w}}} - \frac{(x_i^r - NH(x_i^r))^2}{\|x_i - NH(x_i)\|_{\mathbf{w}}}\right] w^r \tag{2.47}$$

The difference of distances between samples and their nearest neighbors are weighted by coefficients linked to the quality of features. Those weights are found by maximizing the margin. Simba algorithm embeds stochastic gradient ascent into the Relief algorithm that is slightly modified. The Simba algorithm for feature selection is given by algorithm 2.

Algorithm 2 Simba Algorithm

1. Initialize $\mathbf{w} = (1, 1, ..., 1)$
2. For $t = 1...T$
   (a) Pick randomly an instance $\mathbf{x}$ from $\mathbf{X}$
   (b) Calculate $NM(\mathbf{x})$ and $NH(\mathbf{x})$ with respect to $\mathbf{X}\backslash\{\mathbf{x}\}$and the weight vector $\mathbf{w}$
   (c) For $r = 1, ..., \mathcal{D}$ calculate
   $\Delta^r = \frac{1}{2}\left(\frac{(x^r - NM(x)^r)^2}{\|\mathbf{x} - NM(\mathbf{x})\|_{\mathbf{w}}} - \frac{(x^r - NH(x)^r)^2}{\|\mathbf{x} - NH(\mathbf{x})\|_{\mathbf{w}}}\right) w^r$
   $w^r = w^r + \Delta^r$
   End For
   End For
3. $\mathbf{w} \leftarrow \frac{\mathbf{w}^2}{\|\mathbf{w}^2\|_\infty}$ where, $\mathbf{w}^2 = (w^{1^2}, ..., w^{r2}, ..., w^{\mathcal{D}^2})$ and $\|\mathbf{w}^2\|_\infty = \max(w^{1^2}, ..., w^{r2}, ..., w^{\mathcal{D}^2})$

The major advantage of Simba compared to Relief is that it re-evaluates the margin with respect to the updated weight vector. The computational complexity of Simba is $O(T\mathcal{D}N)$, where T is the number of iterations, $\mathcal{D}$ the number of features and N the size of the dataset $\mathbf{X}$. The numerical experiments show that Simba outperforms Relief [220].

The feature selection based on ranking approaches by the score computation and by the algorithm based on hypothesis margin are introduced. The following section presents the exploitation of feature selection methods applied to LBP-based features.

2.4 Feature selection applied to LBP

We classify the LBP-features dimensionality reduction techniques into two strategies: (1) the first one is to reduce the feature length based on some rules or the predefinition of patterns of interest (like uniform patterns) and (2) the second one exploits feature selection methods to identify the discriminative patterns with similar motivations as the beam search LBP variants [7]. The latter has a better performance but usually requires an off-line training. In this work, we are interesting to the learning discriminative LBP features based on feature selection approaches (see section 2.4.1).

Moreover, we take an interesting in a different approach proposed by Porebski, which selects the most discriminant whole LBP histograms (see section 2.4.2). This section thus briefly reviews several methods related to the LBP bin selection and LBP histogram selection approaches.

2.4.1 LBP bin selection

Smith and Windeatt apply the Fast Correlation-Based Filtering (FCBF) algorithm [184] to select the LBP patterns that are the most correlated to the target class [222]. FCBF is a feature selection method which starts with the full set of features, uses Symmetrical Uncertainty (SU) to calculate dependences of features and finds the best subset using backward selection technique with sequential search strategy. It has an inside stopping criterion that makes it stop when there are no features left to eliminate. In the FCBF method, $\mathbf{Y}$ is the vector of data labels and $\mathbf{f}^r$ is the vector of rth feature value for all data. Let $p(f_i^r)$ be the prior probability for all values of $\mathbf{f}^r$. The entropy of $\mathbf{f}^r$ is:

$$E(\mathbf{f}^r) = -\sum_{i=1}^{N} p(f_i^r) log_2(p(f_i^r)) \tag{2.48}$$

and the entropy of $\mathbf{f}^r$ knowing the class labels $\mathbf{Y}$ is defined as:

$$E(\mathbf{f}^r|\mathbf{Y}) = -\sum_{c=1}^{C} p(y_c) \sum_{i=1}^{N} p(f_i^r|y_c) log_2(p(f_i^r|y_c)) \tag{2.49}$$

where $p(f_i^r|y_c)$ is the posterior probability of $\mathbf{f}^r$ given the class label $\mathbf{Y}$. Symmetrical Uncertainty (SU) is calculated as follows:

$$SU(\mathbf{f}^r|\mathbf{Y}) = 2\frac{E(\mathbf{f}^r) - E(\mathbf{f}^r|\mathbf{Y})}{E(\mathbf{f}^r) + E(\mathbf{Y})} \tag{2.50}$$

Based on the *SU* value and a threshold value defined by user, FCBF operates by repeatedly choosing the feature that is the most correlated with the class, excluding those features already chosen, and rejecting any features that are more correlated with it than with the class. In [222],

they split the image into multi blocks and then extract LBP features with different radii. The final histogram with 107000 features is reduced to 120 by using FCBF.

Lahdenoja et al. define a discrimination concept of symmetry for uniform patterns to reduce the feature dimensionality [223]. The level of symmetry $\mathbb{S}$ of a LBP code is defined as the minimum of the sums of each individual binary element in that pattern and its complement which is defined as:

$$\mathbb{S} = min\{\sum_{i=0}^{\mathcal{P}-1} b_i, \sum_{i=0}^{\mathcal{P}-1} \bar{b}_i\} \tag{2.51}$$

where b_i is the ith bit of the LBP code and $\bar{b}_i$ is its complement. For example, for patterns 00111111 and 00011000, the level of symmetry $\mathbb{S}$ equals to 2. The patterns with a higher level of symmetry are shown to have more discriminative power.

Maturana et al. use heuristic algorithm to select the neighbors used in the computation of LBP [224]. Within a square neighborhood given by a $\mathcal{R}$, there are $(2\mathcal{R}+1)^2-1$ possible neighbors. They thus propose to select among the $(2\mathcal{R}+1)^2-1$ neighbors the subspace of $\mathcal{P}$ neighbors which maximizes the Fisher-like class separability criterion.

Liao et al. introduce Dominant Local Binary patterns (DLBP) which consider the most frequently occurred patterns in a texture image [88]. To compute the DLBP feature vectors from an input image, the pattern histogram which considers all the patterns in the input image is constructed and the histogram bins are sorted in descending order. The occurrence frequencies corresponding to the most frequently occurred patterns in the input image are served as the feature vectors.

Guo et al. propose a Fisher Separation Criterion (FSC) to learn the most reliable and robust patterns by using intra-class and inter-class distances [10]. The most reliable patterns for each class are determined, and then merged to form the global dominant set. This model is generalized and can be integrated with existing LBP variants such as LBP uniform, rotation-invariant patterns or LTP.

It is worth to note that there exist another approach for deriving compact and discriminative LBP-based feature vectors consist of applying subspace methods for learning and projecting the LBP features from the original high-dimensional space into a lower dimensional space. For example, a first approach proposed by Chan et al. uses linear discriminant analysis to project high-dimensional color LBP bins into a discriminant space [103]. Banerji et al. apply PCA to reduce the feature dimensionality of the concatenating LBP features extracted from different color spaces. Zhao et al. compare different dimensionality reduction methods on LBP features, e.g. PCA, kernel PCA and Laplacian PCA [225]. Hussain et al. exploit the complementarity of three sets of features, namely, HOG, LBP, and LTP, and applies partial least squares for improving their visual object detection approach [91]. Nanni and Lumini extract the LBP uniform from the multi blocks of facial image. The concatenated feature vector constructed is adopted by Sequential Forward Floating Selection (SFFS) to select the discriminant LBP

feature [226].

2.4.2 LBP histogram selection

Porebski et al. firstly proposed an approach which selects the most discriminant whole LBP histograms [139]. In this approach, the most discriminant LBP histograms are selected in their entirety, out of the different LBP histograms extracted from a color texture. It fundamentally differs from all the previous approaches which select the bins of the LBP histograms or project them into a discriminant space.

Histogram selection approaches can be grouped in three ways: filters, wrappers and hybrid. The latter combines the reduction of processing time of a filter approach and the high performances of a wrapper approach. Filter approaches consist in computing a score for each histogram in order to measure its efficiency. Then, the histograms are ranked according to the proposed score. In wrapper approaches, histograms are evaluated thanks to a specific classifier and the selected ones are those which maximize the classification rate. The next section reviews three scores proposed in the literature and one is our contribution.

2.4.2.1 Intra-Class Similarity score

We first extend the notation that is introduced in section 2.2 and section 2.4.2 to histogram. In the considered LBP histogram selection context, the database is composed of N color texture images. Each image I_i, $i \in \{1,...,N\}$ is characterized by δ histograms $(\delta = 9)$ in a single 3D color space. Let $\mathbf{H}^r$ is the rth histograms to evaluate. The data is summarized by the matrix $\mathscr{H}^r$ is defined as:

$$\mathscr{H}^r = \left[\mathbf{H}_1^r ... \mathbf{H}_i^r ... \mathbf{H}_N^r\right] = \begin{bmatrix} H_1^r(1) & ... & H_i^r(1) & ... & H_N^r(1) \\ ... & ... & ... & ... & ... \\ H_1^r(k) & ... & H_i^r(k) & ... & H_N^r(k) \\ ... & ... & ... & ... & ... \\ H_1^r(Q) & ... & H_i^r(Q) & ... & H_N^r(Q) \end{bmatrix} \tag{2.52}$$

where, $Q = 2^{\mathfrak{P}}$ being the quantization level. $H_i^r(k)$ represents the values of the kth bin, $k \in \{1,...,Q\}$, of the ith image histogram among N color images.

The Intra-Class Similarity score (ICS-score), proposed by Porebski et al., is based on an intra-class similarity measure. Let $\mathbf{h}^r$ be the corresponding normalized histogram [2] of the rth histogram $\mathbf{H}^r$. $\mathbf{H}_i^r$ and $\mathbf{H}_j^r$ are the rth histograms that characterize respectively two training

[2]To normalize the histogram, the number of counts in each bin is divided by the total count, so that the normalized values sum to 1 across all bins.

images I_i and I_j. In order to evaluate the similarity between two images of a same class, Porebski et al. utilize the histogram intersection:

$$D_{int}(\mathbf{H}_i^r, \mathbf{H}_j^r) = \sum_{k=1}^{Q} min(h_i^r(k), h_j^r(k)) \tag{2.53}$$

The following measure SIM_c is then considered to determine the intra-class similarity of a texture class C:

$$SIM_c = \frac{2}{N_c(N_c-1)} \sum_{i=1}^{N_c-1} \sum_{j=i+1}^{N_c} D_{int}(\mathbf{H}_i^r, \mathbf{H}_j^r), \tag{2.54}$$

where N_c is the number of images belonging to the class c. Porebski et al. suppose that the higher the measure SIM_c of intra-class similarity is, the more relevant the histogram $\mathbf{H}^r$ is.

The score S_{ICS}^r of the histogram $\mathbf{H}^r$, which includes all intra-class similarities, is finally defined as follows:

$$S_{ICS}^r = \frac{1}{C} \sum_{c=1}^{C} SIM_c, \tag{2.55}$$

where C is the number of considered classes. S_{ICS}^r ranges from 0 to 1. The most discriminant histogram maximizes the score S_{ICS}^r.

2.4.2.2 Adapted Supervised Laplacian score

Inspired by the approach proposed by Porebski, Kalakech et al. propose to Adapt the Supervised Laplacian (ASL) score used in the literature for feature ranking and selection, to select and rank histograms in the supervised context [9]. The ASL-score evaluates the relevance of a histogram using the local properties of the image data. The basic idea is to assume that the input histogram pairwise similarity measures in the original histogram space are preserved in the relevant histogram subspace. So, similar images with same class labels have to be close when they are represented by one relevant histogram.

For this score, the considered distance measure between two histograms is the Jeffrey divergence, which is defined as follows:

$$D_{Jef}(\mathbf{H}_i^r, \mathbf{H}_j^r) = \sum_{k=1}^{Q} H_i^r(k) log\left(\frac{H_i^r(k)}{\frac{H_i^r(k)+H_j^r(k)}{2}}\right) + \sum_{k=1}^{Q} H_j^r(k) log\left(\frac{H_j^r(k)}{\frac{H_i^r(k)+H_j^r(k)}{2}}\right) \tag{2.56}$$

The value of the Jeffrey divergence between two histograms is low when their corresponding images are similar to each other.

Using this measure, the ASL-score of the histogram $\mathbf{H}^r$ is then defined as follows:

$$S^r_{ASL} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} D_{Jef}(\mathbf{H}^r_i, \mathbf{H}^r_j) s_{ij}}{\sum_{i=1}^{N} D_{Jef}(\mathbf{H}^r_i, \overline{\mathbf{H}^r}) d_i} \tag{2.57}$$

where:

- N is the total number of images,
- s_{ij} is an element of the similarity matrix $\mathbf{S}$. In a supervised context, for each image I_i, a class label y_i is associated. The similarity between two images I_i and I_j is defined by:

$$s_{ij} = \begin{cases} 1 & \text{if } y_i = y_j, \\ 0 & \text{otherwise} \end{cases} \tag{2.58}$$

- d_i is the degree of the image I_i :

$$d_i = \sum_{j=1}^{N} s_{ij}, \tag{2.59}$$

- $\overline{\mathbf{H}^r}$ is the histogram weighted average:

$$\overline{\mathbf{H}}^r = \frac{\sum_{i=1}^{N} H^r_i d_i}{\sum_{i=1}^{N} d_i} \tag{2.60}$$

The histograms are sorted according to the ascending order of the ASL-score in order to select the most relevant ones.

Given a database of N texture images belonging to C classes. Under this representation, we reformulate the ASL-score in equation 2.57 as follows:

$$S'^r_{ASL} = \frac{\sum_{c=1}^{C}\sum_{i,j=1}^{N_c} D_{Jef}(\mathbf{H}^{rc}_i, \mathbf{H}^{rc}_j) s^c_{ij}}{\sum_{c=1}^{C}\sum_{i=1}^{N_c} D_{Jef}(\mathbf{H}^{rc}_i, \overline{\mathbf{H}}^{rc}) d^c_i} \tag{2.61}$$

where:

- N_c is the number of images of the cth class,
- $\mathbf{S}^c = s^c_{ij}$ is the similarity matrix within the class c, defined by equation 2.58. In this case, $\mathbf{S}^c$ is an all-ones matrix and the matrix diagonal $\mathbf{D}^c = d^c_{ii} = \mathbf{I}^c$ is an identity matrix.
- $\mathbf{H}^{rc}$ is the histogram weighted average of the class c

2.4.2.3 Sparse representation for histogram selection

Kalakech et al. introduced a histogram selection score, named "Adapted Supervised Laplacian score" (ASL-score) based on Jeffrey distance and a similarity matrix [9]. This matrix is deduced from the class labels. It is a hard value which is 0 or 1. In this section, we propose to extend the ASL-score by using sparse representation to build a soft similarity matrix that takes values between 0 and 1. Moreover, a value between 0 and 1 will measure the similarity in a subtle way, instead of being binary with just two values 0 and 1. This may lead to more powerful discriminating information. Instead of using the value 1 or 0, we proposed to construct the sparse similarity matrix based on the sparse representation. This leads to our fist contribution is the proposition the novel histogram score, namely Sparse Adapt the Supervised Laplacian (SpASL).

The sparse representation of $\mathbf{H}_i^r$ is constructed by using a few entries of $\mathscr{H}^r$ as possible. It is defined as follows:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \quad \text{s.t.} \quad \|\mathbf{H}_i^r - \mathscr{H}^r \mathbf{s}_i\|_2 < \xi, \quad 1 = \mathbf{1}^T \mathbf{s}_i, \tag{2.62}$$

where:

- $\|.\|_1$ is the l_1-norm of a vector
- $\|.\|_2$ denotes l_2-norm of a vector.
- $\mathbf{s}_i$ is an N-dimensional vector in which the ith element is equal to zero implying that $\mathbf{H}_i^r$ is removed from $\mathscr{H}^r$. It is defined as:

$$\mathbf{s}_i = [s_{i1}, ..., s_{i(i-1)}, 0, s_{i(i+1)}, ..., s_{iN}]^T \tag{2.63}$$

- $\mathbf{1} \in \mathbb{R}^N$ is a vector of all ones.
- ξ represents the error tolerance

For each histogram $\mathbf{H}_i^r$, we can compute the similarity vector $\hat{\mathbf{s}}_i$, and then get the sparse similarity matrix:

$$\mathbf{S} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, ..., \hat{\mathbf{s}}_N]^T, \tag{2.64}$$

where $\hat{\mathbf{s}}_i$ is the optimal solution of equation (2.62). The matrix $\mathbf{S}$ determines both graph adjacency structure and sparse similarity matrix simultaneously. Note that, the sparse similarity matrix is generally asymmetric.

We propose to integrate the sparse similarity matrix obtained by equation 2.64 into the equation 2.61. For each class, we construct the sparse similarity matrix $\mathbf{S}^c$ using images within the class c by equation (2.64), the SpASL-score is defined as follows:

$$S^r_{SpASL} = \frac{\sum_{c=1}^{C}\sum_{i,j=1}^{N_c} D_{Jef}(\mathbf{H}_i^{rc},\mathbf{H}_j^{rc})\hat{s}_{ij}^c}{\sum_{c=1}^{C}\sum_{c=1}^{N_c} D_{Jef}(\mathbf{H}_i^{rc},\overline{\mathbf{H}}^{rc})d_i^c} \tag{2.65}$$

The histogram selection consists to compute for each histogram $\mathbf{H}^r$ an associated S^r_{SpASL} score and rank these scores in ascending order.

It is interesting to note that the sparse similarity matrix can be constructed by using all histogram globally. In this case, the class label does not incorporate to the construction and we are in the case of unsupervised learning.

2.4.2.4 Adapted version of the margin-based iterative search algorithm

More recently, Moujahid et al. propose an adapted version of the margin-based iterative search algorithm (that so called Simba-2) where the resulting weight vector is used for a selection of histograms in the application of face recognition [227]. The resulting weight vector of Simba-2 is used for a selection of histograms in the supervised context.

Instead of using the weighted Euclidean distance to compute the weight distance, the Simba-2 algorithm use χ^2 distance to identify dissimilarities between histograms. This distance is also used for calculating $nearmiss(\mathbf{x})$, $nearhit(\mathbf{x})$.

Given two images I_i and I_j which are characterized by two histograms $\mathbf{H}_i^r$ and $\mathbf{H}_j^r$, respectively and each vector is composed by Q bins. The histogram-weighted χ^2 distance between these two vectors defined as:

$$D_{\chi^2_{w^r}}(\mathbf{H}_i^r,\mathbf{H}_j^r) = \sum_{k=1}^{Q} w^r \frac{(H_i^r(k)-H_j^r(k))^2}{H_i^r(k)+H_j^r(k)} \tag{2.66}$$

The adapted version of the Simba algorithm based on χ^2 distance is given by algorithm 3

Algorithm 3 The adapted version of Simba χ^2 distance (Simba-2)

1. initialize $\mathbf{w} = (1,1,...,1)$
2. For $t = 1,...,T$ do
   (a) Pick randomly an image I characterized by a vector $\mathbf{x}$ from $\mathbf{X}$
   (b) Calculate $nearmiss(\mathbf{x})$ and $nearhit(\mathbf{x})$ with respect to $\mathbf{X}\backslash\{\mathbf{x}\}$and the weight vector $\mathbf{w}$
   (c) For $r = 1,...,\delta$ calculate

$$\Delta^r = \frac{1}{2}\left[D_{\chi^2}(\mathbf{H}^r, NM(\mathbf{H}^r)) - D_{\chi^2}(\mathbf{H}^r, NH(\mathbf{H}^r))\right]$$

   End for
   (d) $\mathbf{w} = \mathbf{w} + \mathbf{\Delta}$

   End for
3. $\mathbf{w} \leftarrow \frac{\mathbf{w}^2}{\|\mathbf{w}^2\|_\infty}$ where, $\mathbf{w}^2 = ({w^1}^2,...,{w^r}^2,...,{w^\delta}^2)$ and $\left\|\mathbf{w}^2\right\|_\infty = \max({w^1}^2,...,{w^r}^2,...,{w^\delta}^2)$

The increment Δ^r inside the Simba algorithm also changes, since it is based on the hypothesis margin which depends on the distance. The resulting Simba weight vector has a size equal with δ histogram. The histograms are sorted according to the descending order of the weight elements in order to select the most relevant ones.

2.4.2.5 Histogram selection procedure

According to the feature evaluation, the histogram selection can be achieved by filter, wrapper or hybrid methods (see section 2.1.2). An hybrid histogram selection approach which requires the learning stage. During this stage, candidate histograms are generated from training images and ranked thanks to a score which measures the efficiency of each candidate histogram independently. Then, the performance of the different $\mathcal{D}$-dimensional histogram subspace are measured by the classification accuracy reached by the chosen classifier in order to select the most discriminating histogram subspace. The selected color texture subspace is the one which maximizes the rate of well-classified testing images.

2.5 Conclusion

In this chapter we reviewed the literature of feature selection as a dimensionality reduction tool. A brief introduction of the feature selection taxonomy is presented. We first introduce the different step of feature selection. Then, we discussed two feature selection strategies which are based on the evaluation (i.e., filters, wrappers and hybrid methods) and based on the availability of class label lead to consider different learning context (i.e., supervised, unsupervised and semi-supervised).

The data and knowledge representation have been presented and graph data representation, especially the sparse similarity graph as well. Next, we provided several feature ranking methods in different learning contexts which are focus on feature selection based on the score computing and based on algorithms. We organized the compact overview for of the feature selection methods applied in LBP-based features by presenting briefly the LBP histogram bin selection and LBP histogram selection. We also introduce our proposed SpASL-score for histogram ranking.

Furthermore, there exists various color spaces and it is difficult to determine which one give the best performance for color texture classification. We propose a method to extract the color LBP-features on different color spaces in the following. The next chapter is mainly based on the LBP-based feature selection methods in the framework of multi color space.

Chapter 3

Multi color space LBP-based features selection

Contents

3.1 Introduction 82

3.2 Color space combination 83

3.2.1 Single color space approach 83

3.2.2 Multi color space approach 87

3.3 Multi color space LBP selection 92

3.3.1 Considered color spaces 92

3.3.2 Candidate color texture descriptors 92

3.3.3 Dimensionality reduction 95

3.4 Multi color space histogram selection 95

3.5 Multi color space bin selection 98

3.5.1 Occurrence based ranking 100

3.5.2 Sparsity based ranking 101

3.5.3 Bin selection procedure 103

3.5.4 Results 103

3.6 Combination of histogram ranking and bin selection 105

3.7 Conclusion 110

3.1 Introduction

Several extensions of LBP to color have been proposed since 2002 [228]. In order to take advantage of all the color texture information contained in the image, the LBP descriptor is applied on each color component independently and sometimes, on pairs of color components conjointly. It leads to a high dimensional feature vector is required to represent a texture. Moreover, we know that there exist numerous color spaces and that the color space choice impacts the classification accuracy [138]. Each color space takes into account specific physical, physiologic and psycho-visual properties but none is well-suited to the discrimination of all texture databases [139]. The selection of the most discriminating color space has been an open question in recent years [126, 137]. Instead of using one color space, the multi color space approaches have emerged in the last few years [229, 230, 231, 18, 14, 102]. These approaches allow, on the one hand, to avoid the difficulty of choosing a relevant color space and, on the other hand, to take advantage of the specific properties of several color spaces by combining them.

In this work, we propose to study the advantages and the drawbacks of the LBP histogram selection and the LBP bin selection presented in chapter 2. Our first contribution consists in extending to color the bin selection approach proposed by Guo et al. [10] and the bin ranking by sparsity score [197]. These approaches are then compared with the whole LBP histogram selection approach proposed by Porebski [8]. The second originality is founded on the current development of multi color space approaches. As there is a wide range of color spaces with different properties, we have proposed an approach which uses the properties of several color spaces simultaneously. In this approach, images are first coded in different color spaces, then color texture features are extracted from these so coded images to represent the texture. It actually seems interesting to compare the strategies of LBP histogram selection and LBP bin selection in a multi color space framework. Two first approaches are thus proposed and compared in this work, a Multi Color Space Histogram Selection (MCSHS) approach and a Multi Color Space Bin Selection (MCSBS) approach.

In addition, we propose to improve the histogram selection method that selects a whole LBP histogram. Indeed it is clear that not all bins of the selected histograms are meaningful for modeling the characteristics of textures. As it selects the most discriminating histogram and filter out the rest, we think that it might have some redundant bins in the selected histograms and a loss of some meaningful bins of the discarded histograms. This leads to our third contribution that performs a combination of bin and histogram selection.

The rest of this chapter is organized as follows. In section 3.2, we present the color space combination approach. Then, we present the multi color space LBP selection in section 3.3. Section 3.4 presents the proposed MCSHS approach. We introduce the MCSBS approach in section 3.5. Next, we present novel strategies that combine bin and histogram selection in

section 3.6. At the end of each proposition, we present the first results obtained on the New BarkTex benchmark database. Finally, we summarize our work.

3.2 Color space combination

Usually, color images are acquired by devices that code the colors in the *RGB* color space. However, there are many other color spaces with different specific properties presented in section 1.1.1 and it is known that the classification performances depend on the choice of the color spaces in which a classifier operates [139]. That is why numerous authors propose to use other color spaces to discriminate the textures as better as possible. Table A.1 in the appendix A presents the different color texture classification approaches that have been proposed in the literature. The analysis of this table confirms the relevance of considering other color spaces than the acquisition space since many other color spaces have been used in these studies to improve the classification results. However, the prior determination of a color space which is well-suited to a specific classification problem is not easy. In order to determine this space, many authors propose to apply their classification approach in different color spaces singly considered and compare the performances reached in each of these spaces, following a single color space approach described in section 3.2.1. In color image analysis, another strategy has emerged: it consists in simultaneously exploiting the properties of several color spaces. This multi color space approach is presented in section 3.2.2.

3.2.1 Single color space approach

Many authors have lead studies about color space for different applications: machine vision, face recognition, texture analysis, etc. In the framework of color texture classification, they try to determine the “best” color space in order to improve the performances of the proposed classification approach. For this purpose, the classifier is applied to images whose colors are coded in different color spaces which are singly considered and the performances reached with each of them are compared. This single color space approach selects the color space that provides the best classification accuracy.

Table 3.1 focuses on some studies that appear in table A.1 and carried out on the BarkTex and OuTex-13 databases. Since the OuTex-13 database has been used repeatedly for color texture classification in the literature while the New BarkTex is recently proposed to overcome the limit of OuTex-13 and BarkTex (more details are presented in section 1.2.4), we select these databases for this study. This table shows the most suitable color space among several different color spaces compared for the classification of the texture on these two databases. The first column of this table indicates the reference of the study with the color texture descriptor used by the authors. The second column states which image database was used. The third

column enumerates the compared color spaces whereas the last one gives the color space that provides the best classification accuracy. The synthesis of these experimental comparisons does not allow to conclude on the definition of a single color space which is well-suited to the discrimination of all texture databases, whatever the considered texture features. For example, the work of Sandid et al. reveals that the choice of the best color space depends on the used image database, and hence the considered application [66]. They used the same method and the same features to classify the color textures of the BarkTex and OuTex-13 databases, and showed that the best results were obtained using different color spaces for each of these two databases. Similarly, the synthesis of the works of Cusano, Casanova and Cernadas show that the color space that yields the best results may be different depending on which features are used [152, 114, 132]. This confirms that the best color space depends on the considered application and approach [139]

Table 3.1: Studies about color space comparison for classification.

Color Descriptor	Database	Color Space used	Best space
3D histogram [56]	OuTex-13	*RGB*, $I_1I_2I_3$	*RGB*
3D histogram [13]	OuTex-13	*RGB*, *HSV*, $I_1I_2I_3$, $L^*a^*b^*$	*HSV*
Cooccurrence matrix and Haralick features [41]	OuTex-13	*RGB*, *HSV*, YC_bC_r	*HSV*
Wavelet features [54]	OuTex-13	*HSV*, $I_1I_2I_3$	*HSV*
Morphological covariance [232]	OuTex-13	*RGB*, *YUV*, *L*a*b**	*L*a*b**
Local triplet patterns [125]	OuTex-13	*RGB*, YC_bC_r	YC_bC_r
Textons feature [127]	OuTex-13	*HSI*, *HSV*	*HSV*
Intensity-Color Contrast [152]	OuTex-13	*RGB*, *HSV*, *L*a*b**, $I_1I_2I_3$	*HSV*
Parametric stochastic models [126]	OuTex-13	*RGB*, *I-HLS*, *L*a*b**, $I_1I_2I_3$	*L*a*b**

Continued on next page . . .

Table 3.1: Studies about color space comparison for classification continued...

Color Descriptor	Database	Color Space used	Best space
Reduced Size Chromatic Co-occurrence Matrices [139]	OuTex-13	*RGB, XYZ, UVW, HSV, LUV* AC_1C_2, $b_w r_g b_y$, YC_bC_r, $L^*C_{uv}h_{uv}$,*Y'I'Q'*, $L^*a^*b^*$ *Y'U'V',HSI, HLS, L*u*v**, I_1rg, *Yxy*, LCh_1Ch_2, $I_1S_2H_1$, $AC_{C1C2}h_{C1C2}$, $b_wC_{rgby}h_{rgby}$, $LC_{Ch1Ch2}h_{Ch1Ch2}$, $I_1S_1H_3$, $L^*C_{ab}h_{ab}$, $L^*S_{uv}h_{uv}$, $Y'C'_{IQ}h'_{IQ}$, $Y'C'_{UV}h'_{UV}$, $I_1C_{I2I3}h_{I2I3}$	*HLS*
EOCLBP [8, 9]	OuTex-13	*RGB, HSV, YUV*, $I_1I_2I_3$	*RGB*
	New BarkTex		*RGB*
Multi-model distance [134]	OuTex-13	*RGB, HSV*, $L^*a^*b^*$	$L^*a^*b^*$
Soft color descriptors [131]	OuTex-13	*RGB, HSV, YUV*, $I_1I_2I_3$, *YCbCr*, $L^*a^*b^*$, *YIQ*, *L*u*v*, XYZ*	*HSV*
Fractal [114]	OuTex-13	*RGB, HSV, I-HLS*, $L^*a^*b^*$, $I_1I_2I_3$	*RGB*
Intensity texture [132]	OuTex-13	*RGB*, $L^*a^*b^*$, *HSV*, $I_1I_2I_3$	$L^*a^*b^*$
Gabor features [12]	BarkTex	*RGB, HSL*	*HSL*
3D histogram [36]	BarkTex	*RGB, L*u*v**	*L*u*v**

Continued on next page . . .

Table 3.1: Studies about color space comparison for classification continued...

Color Descriptor	Database	Color Space used	Best space
Haralick from color co-occurrence matrices [136]	BarkTex	*RGB, XYZ, xyz, Irg, Y'I'Q', Y'U'V', r*b*b* , $L^*a^*b^*$, $L^*u^*v^*$, $I_1I_2I_3$, $R_FG_FB_F$, AC_1C_2, $R_EG_EB_E$, $R_CG_CB_C$, $b_wr_gb_y$*	$L^*a^*b^*$
Three-dimensional adaptive sum and difference histograms [66]	OuTex-13	*RGB, XYZ, UVW, HSV, LUV, AC_1C_2, $b_wr_gb_y$, YC_bC_r, $L^*C_{uv}h_{uv}$,Y'I'Q', Y'U'V', I_1rg, HSI, HLS, L*u*v*, Yxy, $I_1S_2H_1$, $AC_{C1C2}h_{C1C2}$, $LC_{Ch1Ch2}h_{Ch1Ch2}$, $I_1S_1H_3$, $L^*C_{ab}h_{ab}$, $L^*S_{uv}h_{uv}$, LCh_1Ch_2*	*HSV*
	New BarkTex		*RGB*

In addition, the single color space approach has also been applied by various authors on specific industrial applications in order to find out the best color space. For example, the control and the classification of the wood surface is improved by applying the defect detection and classification approaches in different color spaces [233, 234, 235]. On the other hand, Bianconi et al. compare of the performances of automated classification of natural stone in different color spaces [236, 237].

A wide range of color spaces exists, all with different properties and it is difficult to determine *a priori* the best color space in developing a successful application of color texture classification. For this reason, an alternative approach uses the properties of several different color spaces simultaneously. This multi color space approach is presented in the next section.

3.2.2 Multi color space approach

Instead of searching the best color space for color image analysis, recent works propose to combine different color spaces in order to improve the performances reached by classification schemes. These works can be categorized into three main strategies:

- **Color space fusion**: this strategy involves fusing the results from several classifiers, each one operating in a different color space;
- **Color space selection**: this strategy involves selecting the most well suited color spaces which are based on some specific criterion;
- **Color texture feature selection**: this strategy involves evaluating the texture features over different color spaces and selecting the features that provide the best discrimination between the different textures classed by using a supervised feature selection approach;

The following explains in detail theses strategies.

3.2.2.1 Color space fusion

In these approaches, the color texture features are evaluated in several color spaces and a classification scheme is performed in each of these different spaces independently such as each classifier operates in a specific color space. Thus, for each texture to be classified, several decisions coming from different classifiers are available. The final labeling is obtained thanks to a fusion rule of these decisions.

For texture classification, Chindaro et al. propose a color space fusion scheme by considering six color spaces [230, 238]. Each color space is used to independently design a *k*-nearest neighbor (*k*-NN) classier during a learning scheme. The output of each of the six *k*-NN classifiers are combined thanks to a fusion rule in order to make the decision. A similar approach is proposed by Charrier et al. for microscopic color image segmentation by pixel classification [239]. The first step of the proposed method is to classify pixels with five independent Support Vector Machine (SVM) classifiers, each of them operating in different color spaces. The second step is to categorize pixels in coherent pixels when all the classifiers select the same class or incoherent pixels when at least one classifier output differs from the others. Only incoherent pixels are processed through a fusion method to select their final class. The final segmentation result is obtained from the union of the two pixel sets. Mignotte proposes a segmentation approach based on a fusion procedure which aims at combining several segmentation maps [240]. The segmentation maps to be fused are given by the *k*-means clustering technique applied to an input image coded in six different color spaces.

3.2.2.2 Color space selection

Another strategy consists in automatically selecting color spaces among a set of available ones.

Busin et al. propose an approach that iteratively selects, among 11 color spaces, those which are well suited to automatically segment a color image by pixel classification [241]. This approach is extended by Vandenbroucke et al. with 28 color spaces in order to iteratively identify pixel classes by taking into account both the pixel color distributions in several color spaces and their spatial arrangement in the image [18]. In order to overcome the difficult problem of the color space choice, the algorithm selects the color space that is well suited to construct the class, at each iteration step. An adaptive color space switching strategy has been developed by Stern et al. in order to perform face skin tracking from video under varying illumination [242]. For a given task, the idea is to dynamically select, among all conventional color spaces, the best color space depending on the illumination conditions. The authors apply their adaptive color space switching algorithm to a human face detection and tracking system based on the skin color and show that the performance of the tracking is increased. A similar automatic color space selection and switching approach is also proposed by Laguzet et al. in order to improve the performances of pedestrian visual tracking [243, 244]. The automatically and continuously selection of the color space is based on the good separability between the target and its close background.

3.2.2.3 Color texture feature selection

Rather than selecting color spaces, another strategy is to automatically select color texture features computed in several color spaces. In these approaches, the pixel colors are transformed into different color spaces and texture features are computed from the so converted images. A feature selection procedure selects the most discriminating color texture features for the texture classification.

Vandenbroucke et al. propose a pixel classification algorithm that analyzes the texture in the neighborhood of a pixel, in different color spaces [245]. Among a multidimensional set of first order statistic features evaluated for each color component, the most discriminating ones are selected by means of an iterative feature selection procedure. Pixels are then classified in the so-selected texture feature space for soccer image segmentation purposes. For texture classification, Porebski et al. propose an approach that selects the most discriminating Haralick features extracted from chromatic co-occurrence matrices of color images coded in 28 different color spaces thanks to a sequential forward selection (SFS) scheme [46, 139]. A similar approach is used by Cointault et al. with 23 color components for a wheat ear counting system based on color image segmentation [229]. Nanni et al. use 13 different color spaces from which a set of Gabor features is extracted [231]. For each color component, a Gabor feature vector is first defined. The most relevant feature vectors are then selected thanks to a sequential forward

floating selection (SFFS) scheme. Several nearest neighbor (1-NN) classifiers constructed with each selected feature vector fusion are finally combined in order to authenticate ears for biometric applications. Banerji et al. propose the Color LBP Fusion descriptor for color texture classification [14]. This descriptor is firstly constructed by concatenating LBP descriptors extracted from six color spaces and then applied PCA to reduce the feature dimensionality.

Table 3.2 synthesizes the studies about multi color space approaches which are grouped into the three strategies previously presented with the different used color spaces and descriptors. It seems interesting to wonder which one could be the most relevant. However, there exists no study that has compared the performances of these color space combination strategies and it could be a great prospect to compare these strategies. Our work proposes here to use the color texture feature selection to compare the approaches of LBP histogram selection and LBP bin selection in a multi color space framework.

Table 3.2: Studies about multi color space approach.

Multi color space approach	Descriptors	Color Space used
Color space fusion	Markov random fields [230], Independent component analysis [238]	*RGB, rgb, HSV, YIQ, YUV*, $L^*a^*b^*$
	Denoeux's model [239]	*RGB, XYZ, HSL, YUV, HSI*, $L^*a^*b^*$, YC_bC_r
	Markov random field [246, 240]	*RGB, XYZ, HSI, YIQ, TSL*, $L^*a^*b^*$, $L^*u^*v^*$, $I_1I_2I_3$, $H_1H_2H_3$, YC_bC_r
	Color LBP Fusion [247], Color Grayscale LBP Fusion, Pyramid of Histograms of Orientation Gradients [14]	*RGB, rgb, oRGB, HSV* , YC_bC_r

Continued on next page . . .

Table 3.2: Studies about multi color space approach continued...

Multi color space approach	Descriptors	Color Space used
	H-descriptor [248]	*RGB, oRGB, HSV, YIQ, DCS,* YC_bC_r, $I_1I_2I_3$
	Color Gabor-LBP [52]	*RGB, oRGB, HSV, YIQ, DCS,* YC_bC_r
	Histograms of oriented gradients [249]	*RGB, oRGB, HSV ,* YC_bC_r
	Markowitz model [250]	*RGB, rgb, HSV, iHLS,* $L^*a^*b^*$, YC_bC_r
Color space selection	Histogram multi thresholding [251]	*RGB, rgb, XYZ, xyz, YIQ, YUV,* $w_br_gb_y$, YC_1C2, $L^*a^*b^*$, $L^*u^*v^*$, $I_1I_2I_3$
	Multi Color Space Segmentation [18]	*RGB, rgb, XYZ, xyz, YIQ, YUV, Yxy,* $w_br_gb_y$, AC_1C2, $L^*a^*b^*$, $L^*u^*v^*$, $I_1I_2I_3$, YCh_1Ch_2, I_1rg, $I_1S_1H_1$, $I_1S_2H_2$, $I_4S_3H_2$, $I_5S_4H_2$, $I_6S_5H_1$, $L^*S^*_{uv}h_{uv}$ $AC_{C1C2}h_{C1C2}$, $b_wC_{rgby}h_{rgby}$, $YC_{IQ}hIQ$, $YC^*_{ab}h_{ab}$, $YC_{UV}h_{UV}$, $L^*C^*_{ab}h_{ab}$,$L^*C^*_{uv}h_{uv}$, $LC_{Ch1Ch2}h_{Ch1Ch2}$, $I_1C_{I2I3}h_{I2I3}$
	Switching model [252]	*RGB, HSV,* YC_bC_r
	Switching color space models [242]	*RGB, rgb, HSI, YIQ, XYZs,* YC_bC_r, $L^*a^*b^*$, $L^*u^*v^*$, $I_1I_2I_3$
	Markov random field [253]	*RGB, HSI, rg,* $L^*a^*b^*$, $L^*u^*v^*$

Continued on next page . . .

Table 3.2: Studies about multi color space approach continued...

Multi color space approach	Descriptors	Color Space used
	Color space models based on Mean-Shift [243, 244]	*RGB, rgb, XYZ, HSI, YUV,* $L^*a^*b^*$, YC_bC_r, $I_1I_2I_3$
	Shadow eliminating operator [254]	*RGB, HSV, rgb XYZ,* YC_bC_r, $L^*a^*b^*$, $c_1c_2c_3$, $l_1l_2l_3$
	Gaussian low-pass filter [255]	*RGB, HSV, HSI,* $L^*a^*b^*$, YC_bC_r
Color texture feature selection	Color pixels classification [256]	*RGB, rgb, ISH, XYZ, xyz, YIQ, YUV,* AC_1C_2, $I_1I_2I_3$, $L^*a^*b^*$, $L^*u^*v^*$, $L^*_{ab}C^*_{ab}h^o_{ab}$, $L^*_{uv}C^*_{uv}h^o_{uv}$, $L^*_{uv}S^*_{uv}h^o_{uv}$
	Haralick features [46], Reduced Size Chromatic Co-occurrence Matrices [139]	*RGB, YUV, YIQ, XYZ, xyz, Yxy, rgb, LUV ,* $L^*a^*b^*$, $L^*u^*v^*$, AC_1C_2, I_1rg $I_1I_2I_3$, $b_wr_gb_y$, $I_1S_1H_1$, $I_1S_2H_1$, $I_4S_3H_2$, $I_5S_4H_2$, $I_1S_1H_3$, $L^*S^*_{uv}h_{uv}$, $AC_{C1C2}h_{C1C2}$, $b_wC_{rgby}h_{rgby}$, $YC_{IQ}hIQ$, $YC_{UV}h_{UV}$, $L^*C^*_{uv}h_{uv}$, $L^*C^*_{ab}h_{ab}$, $I_1C_{I2I3}h_{I2I3}$, $LC_{UV}h_{UV}$
	Morphological information [229]	*RGB, rgb, HSL,* $L^*a^*b^*$, $I_1I_2I_3$, L^*IV_1 ,
	Gabor Filters [231]	*RGB, YUV, YIQ, HSV, HSL, XYZ, LCH ,* $L^*a^*b^*$, $L^*u^*v^*$, YC_bC_r, YP_bP_r, YD_bD_r, $JPEG-YC_bC_r$

The next section details the color texture feature selection in multiple color spaces.

3.3 Multi color space LBP selection

3.3.1 Considered color spaces

In order to show the interest of the multi color space approaches for color texture classification, $N_S = 9$ color spaces among the spaces have been presented in section 1.1.1 are considered for experiments:

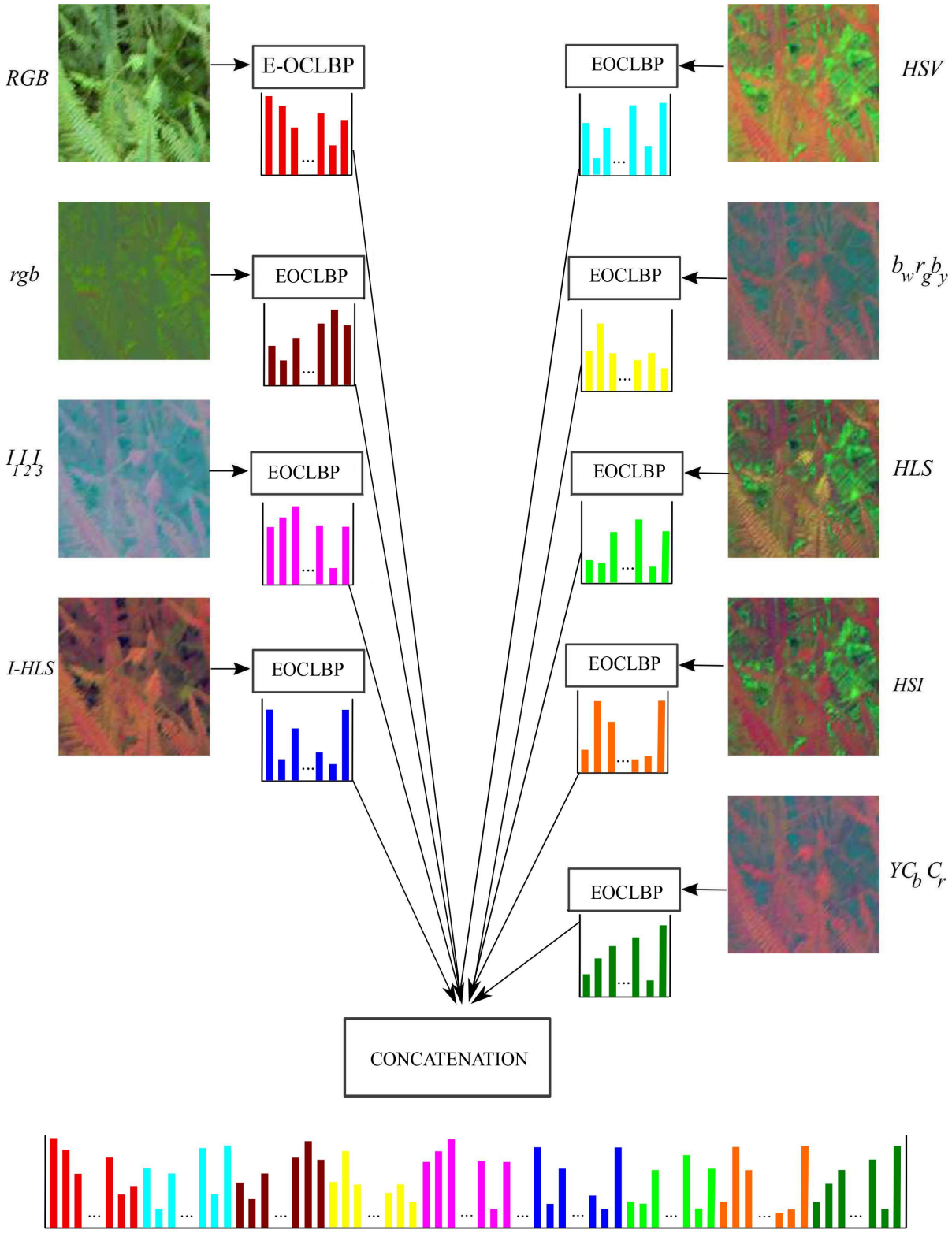
- *RGB* and *rgb*, which belong to the primary space family,
- YC_bC_r and $b_w r_g b_y$, which are luminance-chrominance spaces,
- $I_1I_2I_3$, which is an independent color component space,
- *HSV*, *HSI*, *HLS* and *I-HLS*, which belong to the perceptual space family.

We have chosen these nine color spaces since they do not require to know illumination and image acquisition conditions, like $L^*a^*b^*$ or $L^*u^*v^*$ for instance. They also allow a good representation of the four different color space families, even if a majority of perceptual spaces have been chosen because these spaces are known to obtain a good classification accuracy [126, 139].

3.3.2 Candidate color texture descriptors

To compute the color LBP histograms or bins that are candidate for the selection, each image is first coded in each of the $N_S = 9$ color spaces previously introduced. Then, the $\delta_{max} = 9$ different LBP histograms of EOCLBP descriptors described in section 1.1.4.3 are computed from the so-coded images. A color texture is thus represented by $\delta_{max} \times N_S = 9 \times 9 = 81$ candidate LBP histograms. When the number of bins Q is equal to 256 for each histogram, the total number of bins is $Q \times \delta_{max} \times N_S = 256 \times 9 \times 9 = 20736$ bins. Figure 3.1 illustrates the representation of a texture in multiple color spaces by the EOCLBP descriptor. The considered image is coded in 9 different color spaces in which the EOCLBP is applied separately to compute their histograms and obtain the texture feature vector. Finally, the concatenation of these vectors is achieved in order to provide a multi color space representation of a texture.

We present here the first results obtained on the New BarkTex set by the single color space and the multiple color space approaches. The New BarkTex set is divided into a half for training set and a half for testing set by holdout method. Table 3.3 presents the classification results obtained on the testing set of this database. The purpose of this work being to show the contribution of the multi color space approach, independently of the considered classifier and its parameters (like the metric), the nearest neighbor classifier associated with the L1 distance as a similarity measure is here considered. Obviously, more sophisticated methods such as SVM



Histograms of EOCLBP composed from images coded in 9 color spaces

Figure 3.1: An illustration of EOCLBP feature extraction from 9 color spaces.

and other metrics may provide better results, but at the cost of difficult complicated tuning procedures.

The first column of this table shows the considered color spaces. The second column indicates the rate of well-classified images with the dimension of the subspace. Here no selection is performed. The dimension is thus $9 \times 256 = 2304$ when a single color space is considered, and $9 \times 9 \times 256 = 20736$ when the multi color approach is applied. The results without any selection vary from 70.1% to 74.4% when a single color space is considered. The *rgb* color space gives the best rate (value in box) and the average rate among the different color spaces singly considered is 71.9%. The last row shows that the accuracy obtained in multiple color spaces is 78.2% . We can observe that the multi space approach allows to significantly improve the classification accuracy (+ 6.3% compared with the average value) however at the cost of a nine times longer feature space.

Table 3.3: The classification results of New BarkTex database in nine single color space and in multi color space when no selection method is used.

Color spaces	Without selection	
	Rate	Dimension $(\hat{\mathcal{D}} = \delta_{max} \times Q \times N_S)$
RGB	73.2	$9 \times 256 \times 1 = 2304$
rgb	74.4	
$I_1I_2I_3$	71.7	
HSV	70.5	
$b_w r_g b_y$	72.1	
HLS	70.1	
I-HLS	72.1	
HSI	71.7	
YC_bC_r	71.6	
Average in single space	71.9 ± 1.3	
Multi spaces	**78.2**	$9 \times 256 \times 9 = 20736$

It is well-known that the performance of a classifier is generally dependent on the dimension of the feature space due to the curse of dimensionality (see chapter 2) [257]. To reach a satisfying classification accuracy while decreasing the computation time of the on-line classification, we propose to reduce the dimension of the feature space by selecting the most discriminating features during a supervised learning stage.

3.3.3 Dimensionality reduction

To reduce the dimensionality of the feature space, two main strategies are proposed: feature extraction and feature selection. And in order to evaluate the relevance of the feature subspaces by feature selection methods, different approaches are proposed: filter, wrapper and hybrid [162] (see chapter 2). Among these approaches, hybrid approaches are preferred [172]. These approaches combine a filter approach to select the most discriminating feature subspaces at different dimensions and a wrapper approach to determine the dimension of the selected subspace [175]. To operate a supervised feature selection, it is necessary to extract learning and testing image subsets from the studied database. The learning subset is used to build a low dimensional discriminating feature space during a supervised learning stage and the testing subset is used during the classification stage to evaluate the performances of the proposed approach. Since wrapper approach is a feature selection procedure that uses the classification rate as discrimination power of a feature subspace, it needs to classify the images of a learning subset for all the candidate feature subspaces, that involves an important learning time and classifier-dependent results. When a classifier such as the nearest neighbor is considered, it requires of decomposing the learning subset into a training and a validation subsets.

Applying a multi color space strategy avoids the difficulty of choosing a relevant color space that depends on the considered application and allows to take advantage of the discrimination quality of several color spaces by combining them. It seems thus interesting to compare the approaches of LBP histogram selection and LBP bin selection in a multi color space framework. The first approach is a multi color space extension of the LBP histogram selection proposed by Porebski et al. for a single color space [8]. The second proposed approach is the extension to color of the LBP histogram bin selection proposed by Guo et al. for a gray level analysis by applying a multi color space strategy [10]. The third approach is the extension of the sparsity score to LBP histogram bin ranking in multiple color spaces [197]. The following sections detail these original approaches.

3.4 Multi color space histogram selection

The Multi Color Space Histogram Selection (MCSHS) approach analyzes LBP histograms computed from texture images coded into several color spaces. Indeed, rather than looking for

the best color space, these approaches first compute LBP histograms from several color spaces (see sections 3.3.1 and 3.3.2) and then selects, out of the different candidate LBP histograms, those which are the most discriminant for the considered application in a supervised context.

The MCSHS approach proposed in this section, whose flow chart is represented by figure 3.2, is an extension to the multi-color space domain of the histogram selection approach proposed in 2013 by Porebski et al. [8].

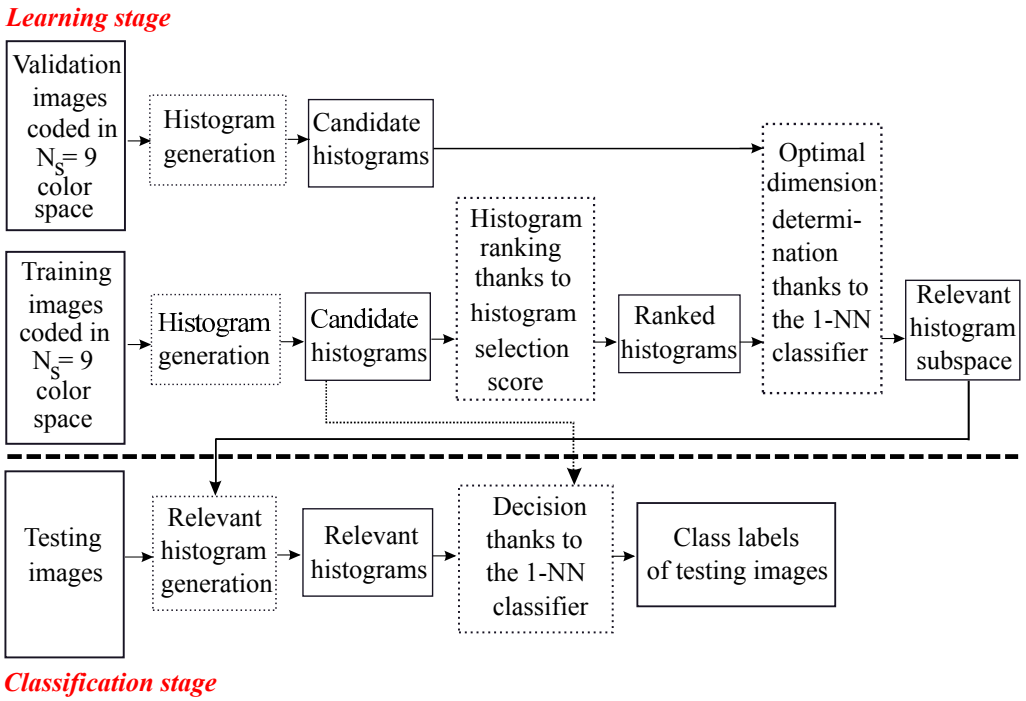


Figure 3.2: An illustration of the multi color space histogram selection aprroach.

MCSHS is an hybrid histogram selection approach that requires to split up the initial image dataset in order to build a training, a validation and a testing image subset, according to a holdout decomposition. During the learning stage, candidate histograms are generated from training images. Let us note that most of considered texture benchmark databases are composed of only two image subsets, whereas the MCSHS approach needs three subsets. In order to compare our experimental results with the same condition of other works, we thus propose to use one subset as the training subset and the second both as the validation and testing subset in order to evaluate and compare the performances of the MCSHS approach.

Then, the proposed histogram selection procedure uses a “feature ranking” algorithm. The selection is based on the histogram score evaluated for each of the 81 available histograms. In this work, four different scores are considered and compared:

- the Intra-Class Similarity score (ICS-score), proposed by Porebski et al. [8] and presented in section 2.4.2.1, which is based on an intra-class similarity measure.

- the Adapted Supervised Laplacian score (ASL-score), proposed by Kalakech et al. [9] and presented in section 2.4.2.2, which evaluates the relevance of a histogram using the local properties of the image data.

- the Simba-2 score, proposed by Mouhajid et al. [227] and presented in section 2.4.2.4, which is based on the hypothesis margin and the χ^2 distance.

- the Sparse Adapted Supervised Laplacian score (SpASL-score), presented in section 2.4.2.3, which is based on ASL-score and sparse representation [258].

Once the score has been computed for each of the $\delta_{max} \times N_S = 81$ candidate histograms, a ranking is performed. The candidate subspaces - composed, at the first step of the procedure, of the histogram with the best score, at the second step, of the two first ranked histograms and so on - are then evaluated to determine the relevant histogram subspace (see Figure 3.2). The stopping criterion of the histogram selection procedure is based on the classification accuracy. For this purpose, a classifier operates in each candidate subspace in order to classify the validation images. The selected subspace, whose dimension is $\hat{\delta} \times Q$, is the one which maximizes the rate R_δ of well-classified validation images noted:

$$\hat{\delta} = \underset{1 \leq \delta \leq \delta_{max} \times N_S}{\operatorname{argmax}} R_\delta \tag{3.1}$$

During the classification stage, the relevant histograms previously selected are computed for each testing image and compared to the training images in the so-selected relevant histogram subspace to determine the testing image label.

Table 3.4 presents the results obtained by using histogram selection in a single color space and in multiple color spaces. The first column of this table represents the color spaces used to code image. The second column recalls the results obtained in table 3.3 without any selection. The third column is divided to four sub-columns corresponding to the four considered scores: ICS, Simba-2, ASL and SpASL-score. By analyzing this table, we see that the histogram selection clearly improves the classification performance when a single or multiple color spaces are considered. These approaches also reduce the number of histograms used in the classification stage. The best rate is 88.0%. It is obtained by ICS-score in multiple color spaces with a nearly reduced half number of used histograms. It improves nearly 10% of the rate obtained when no selection is applied. This result shows that the relevance of considering a histogram selection approach in a single and in multiple color spaces.

We have presented the Multi Color Space Histogram Selection approach. The next section details the proposed Multi Color Space Bin Selection approach.

Table 3.4: Comparison of classification rates obtained with and without LBP histogram selection in a single color space and in multi spaces on the New BarkTex database. The values in boxes represent the best rates obtained with each color space and the boldface indicates the best rate obtained of each approach.

Color spaces	Without selection		Histogram selection							
			ICS		Simba-2		ASL		SpASL	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
RGB	73.2	9	81.3	4	81.3	4	81.3	4	81.3	4
rgb	74.4	9	76.8	7	74.4	9	77.1	3	77.1	3
$I_1I_2I_3$	71.7	9	79.5	7	75.4	8	79.5	7	79.5	7
HSV	70.5	9	81.0	3	76.8	4	81.0	3	81.0	3
$b_w r_g b_y$	72.1	9	80.0	7	72.1	9	80.6	6	80.6	6
HLS	70.1	9	81.0	3	71.2	8	81.0	3	81.0	3
I-HLS	72.1	9	75.9	6	72.1	9	77.1	5	78.8	2
HSI	71.7	9	79.8	3	73.9	7	79.8	3	79.8	3
YC_bC_r	71.6	9	79.3	7	71.6	9	79.3	7	79.3	7
Average in single space	71.9 $\pm$ 1.3	9	79.4 $\pm$ 1.8	5	74.3 $\pm$ 3.2	7	79.6 $\pm$ 1.5	4	79.8 $\pm$ 1.3	4
Multi spaces	**78.2**	81	**88.0**	42	**85.2**	40	**86.8**	29	**87.3**	37

3.5 Multi color space bin selection

We first briefly recall the notations that are introduced in section 2.4.2. In the considered LBP histogram selection context, the database is composed of N color texture images. Each image I_i, $i = \{1,...,N\}$ is characterized by $\delta_{max} \times N_S = 9 \times 9 = 81$ histograms in the multi color space approach and represented by a concatenated histogram $\mathcal{H}_i$. The available data is summarized by the matrix $\mathcal{T}$ defined as:

$$\mathcal{T} = \begin{bmatrix} \mathcal{H}_1 \\ ... \\ \mathcal{H}_i \\ ... \\ \mathcal{H}_N \end{bmatrix} = \begin{bmatrix} \mathcal{H}^1 & ... & \mathcal{H}^r & ... & \mathcal{H}^{\delta_{max} \times N_S} \end{bmatrix} \tag{3.2}$$

where $\mathfrak{T}$ is composed by 81 matrix $\{\mathscr{H}^r\}_{r=1}^{81}$ that represents the data of rth histogram. Let $\mathbf{H}^r$ be one of the 81 histograms to evaluate, the matrix $\mathscr{H}^r$ defined as:

$$\mathscr{H}^r = \begin{bmatrix} \mathbf{H}_1^r \\ \dots \\ \mathbf{H}_i^r \\ \dots \\ \mathbf{H}_N^r \end{bmatrix} = \begin{bmatrix} H_1^r(1) & \dots & H_1^r(k) & \dots & H_1^r(Q) \\ \dots & \dots & \dots & \dots & \dots \\ H_i^r(1) & \dots & H_i^r(k) & \dots & H_i^r(Q) \\ \dots & \dots & \dots & \dots & \dots \\ H_N^r(1) & \dots & H_N^r(k) & \dots & H_N^r(Q) \end{bmatrix} \tag{3.3}$$

where, $Q = 2^{\mathcal{P}}$ is the quantization level that depends on the number of neighbors used to code the local binary patterns of the image. $H_i^r(k)$ represents the values of the kth bin, ($k \in \{1, ..., Q\}$) of the rth histogram of the ith LBP image among N color texture images.

Like the MCSHS approach, the Multi Color Space Bin Selection (MCSBS) analyses LBP histograms computed from texture images coded into several color spaces. Instead of selecting the most discriminating histograms, the MCSBS approach selects the most discriminating bins of these histograms. The first approach, presented in section 3.5.1, is an extension to the multi color space domain of the bin selection approach proposed in 2010 by Guo et al. for gray level image analysis [10]. In the second approach, we consider that the bin of an histogram corresponds to a feature of a vector, and we propose to apply the feature ranking algorithm presented in section 2.3.1 for bin selection. Among the effective supervised filter ranking methods, the supervised sparsity score is outperformed other scores as shown in [197], so we extend this score as the second bin selection approach in the multi color space domain (see section 3.5.2).

The flow chart of the MCSBS approach is represented by figure 3.3.

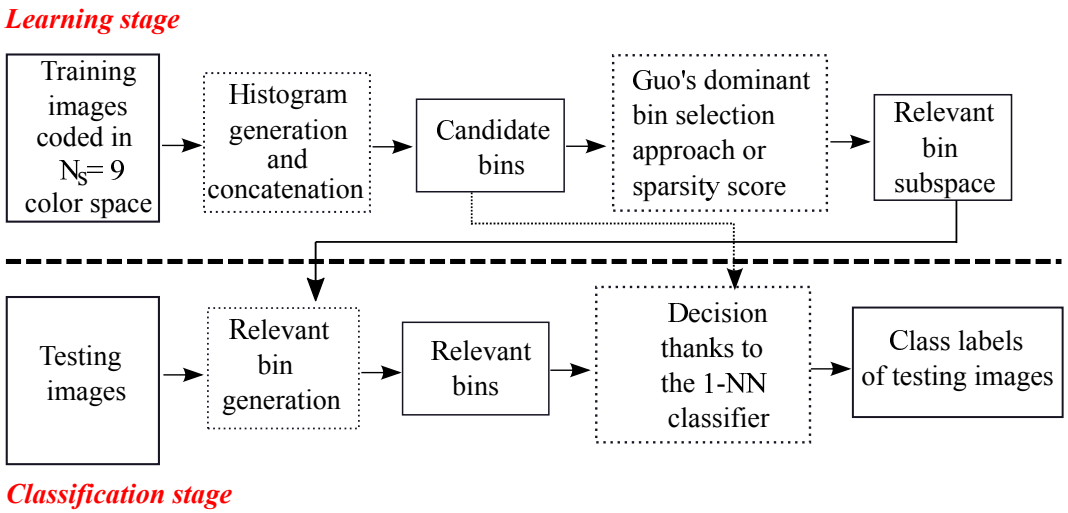


Figure 3.3: An illustration of multi color space bin selection approach.

It is a filter bin selection approach which requires to split up the initial image dataset to build a training and a testing image subset. The bin selection procedure is based on a “feature

ranking" algorithm while the score is based on pattern occurrence frequency by Guo's method or based on the sparsity score. The stopping criterion of bin selection approach depends on a threshold defined by the user. This approach consists in determining the global dominant bin set J_{global} among the $Q \times \delta_{max} \times N_S = 256 \times 9 \times 9 = 20736$ candidate LBP bins of the $\delta_{max} \times N_S$ concatenated histograms (see figure 3.1).

The following subsections present the bin selection based on pattern occurrence and sparsity score.

3.5.1 Occurrence based ranking

During the learning stage, histograms are first generated from each training image I_i. The most reliable and robust dominant bins are then determined among the candidate bins of each concatenated histogram. The dominant bin set J_i of an image I_i is the minimum set of bins which can cover $\mathcal{T}\%$ of all bin occurrences of I_i:

$$J_i = \underset{|J_i|}{\operatorname{argmin}} \frac{\sum_{k \in J_i} \mathcal{H}_i(k)}{\sum_{k=1}^{Q} \mathcal{H}_i(k)} \geq \mathcal{T}\% \quad (i = 1...N) \tag{3.4}$$

where $|J_i|$ denotes the number of elements in the set J_i and Q, the number of bins. Guo proposes to set the threshold $\mathcal{T}$ as 90% [10].

Then, the most reliable and robust dominant bins for each class c are determined. The dominant bin set J_{C_c} of a class c is constituted of the bins that consistently belong to all the dominant bin sets J_i of each image I_i of the class c. It is the intersection of all sets J_i of the images of the class c:

$$J_{C_c} = \bigcap_{I_i \in c} J_i \tag{3.5}$$

Finally, the global dominant bin set J_{global} for the whole database is determined. It is composed of all dominant bins belonging to the sets J_{C_c} $(c = 1...C)$. It is the union of all sets J_{C_c}:

$$J_{global} = \bigcup_{c=1}^{C} J_{C_c} \tag{3.6}$$

During the classification stage, the dominant bins are calculated for each testing image based on the global dominant bin set J_{global} determined during the learning stage. The testing image is then compared to the training images in the relevant bin subspace to determine its label thanks to the nearest neighbor classifier based on the L1 distance.

The following section introduces the bin selection method based on sparsity score.

3.5.2 Sparsity based ranking

The sparsity based ranking approach uses the sparse graph construction approach presented in section 2.3.1. During the learning stage, each training image is represented by a vector composed of the 20736 features (or bins) of the 81 concatenated histograms. The novel proposed method applies the sparsity ranking according two strategies:

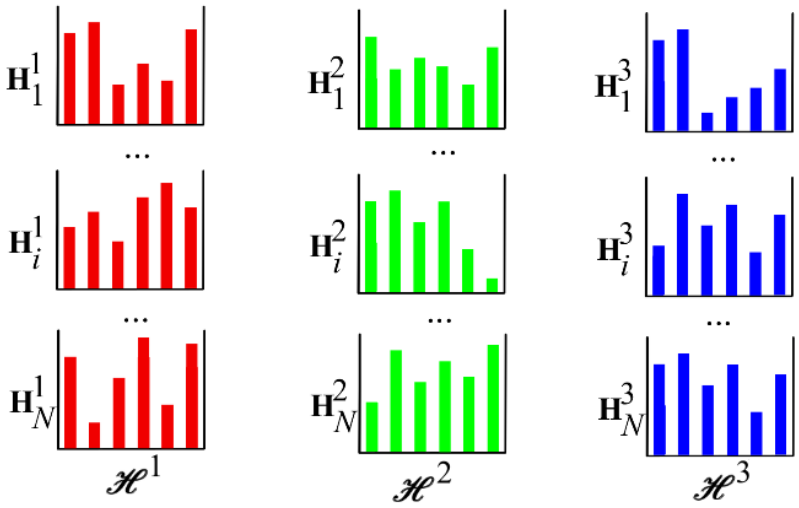
- **Strategy A**: Candidate histograms are generated from training images and concatenated to form a vector with 20736 features. For each class, the sparse similarity matrix is first calculated by using the class label through equation 2.12. Then, the score is assigned for each feature by equation 2.22. The ranked bins are obtained by sorting all bins according to their score in ascending order.

- **Strategy B**: For each candidate histogram generated from training images, the sparsity score is computed for the Q bins of **this** histogram. This differs from the strategy A where the sparsity score is computed for the $Q \times \delta_{max} \times N_S$ bins of the concatenated histogram. The final ranked bins of all candidate histograms are obtained by sorting those bins according to their score in ascending order.

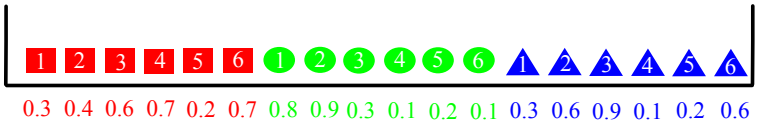
Note that the difference between these two strategies is the dimensional-features input to construct the sparse similarity matrix within each class. So, the score value of each bin within each histogram and the score value of each bin of the concatenated histograms are different. The experimental results conducted in the following section allow us to see this difference.

In order to illustrate the proposed method, we introduce an example with three sample histograms $\mathbf{H}_i^1$, $\mathbf{H}_i^2$ and $\mathbf{H}_i^3$, $i \in \{1,...,N\}$ of the N training images represented by three matrices $\mathscr{H}^1$, $\mathscr{H}^2$ and $\mathscr{H}^3$, respectively, as shown in Figure 3.4. To represent the bins of each histogram, we use three symbols: a square, a circle and a triangle. We consider that each histogram has 6 bins which are numbered from 1 to 6. For example, the square numbered as 1 is represented the first bin of histogram $\mathbf{H}_i^1$.

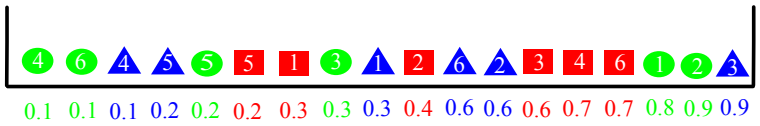
For the strategy A, three histograms are firstly concatenated to form a feature vector with 18 features. We assume the score value of each bin is estimated and illustrated below each symbol. The associated score is computed for each bin by the sparsity score and the bins are ranked in ascending order according to their value as illustrated in figure 3.4a. For the strategy B, the sparsity score is applied on each histogram to compute a score for each bin. For example, the 6th bin of $\mathbf{H}_i^1$ has a smallest sparsity score value and it is more relevant than others. The bins of all histograms are examined in order to be sorted depending on their score value. The illustration of the final ranked bins is at the bottom of the figure 3.4b. We see that all the bins with smallest score values are ranked first, i.e 3th and 5th bin of $\mathbf{H}_i^3$ and 6th bin of $\mathbf{H}_i^2$ are more relevant than others. The order of the bins with the same score value is not considered in this case.



Concatenate three histograms and compute the score for each bin



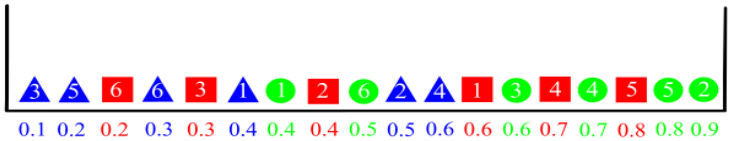
Bin ranking



(a) Bin rank obtained by strategy A

Compute the score

Bin ranking



(b) Bin rank obtained by strategy B

Figure 3.4: An illustration of bin selection by strategy B.

After the bin ranking is finished, the relevant bin subspace can be selected by a threshold defined by user according to the score value. During the classification stage, the features of testing images are determined according to the bin ranked and the threshold defined by user during the learning stage. The decision step is realized as previously presented.

Moreover, the bin selection procedure can be based on the filter or hybrid approaches. For the filter approaches, we can use the threshold parameter $\mathcal{T}$ as Guo does in [10]. On the other hand, the hybrid approaches allow evaluating the relevance of the feature subspaces without using any parameter. That is why we choose hybrid approaches to evaluate our proposed approaches. The following section presents this procedure which is based on sparsity ranking.

3.5.3 Bin selection procedure

The flow chart of an hybrid LBP bin selection approach is represented by figure 3.5. It requires to split up the initial image dataset in order to build a training, a validation and a testing image subset, according to a holdout decomposition. The parts labeled as gray are the differences compared with MCSBS approach shown by figure 3.3. As we mentioned in the previous section, the testing and validation subsets are the same in most of available image database.

During the learning stage, candidate histograms are generated from training images and bin ranking is applied (by the strategy A or B) thanks to the supervised sparsity score presented in section 2.3.1. Once the bin ranking strategy is applied, the final ranked bins are selected to find the relevant subspace and the most discriminant bins have to be selected. The discriminant power of candidate bin subspaces with different dimensions are evaluated to determine the most relevant subspace. At the first step, the candidate subspace composed of the first ranked bin is considered. Then, at the second step, the candidate subspace composed of the two first ranked bins is considered and so on. For this purpose, a nearest neighbor classifier is also considered with the L1 distance. This classifier operates in each candidate subspace to classify the validation images represented by the prototype bins. The selected subspace, whose dimension is $\hat{\mathcal{D}}$, is the one which maximizes the rate of well-classified validation images denoted $R_{\mathcal{D}}$:

$$\hat{\mathcal{D}} = \underset{1 \leq \mathcal{D} \leq Q \times \delta_{max} \times N_S}{\operatorname{argmax}} R_{\mathcal{D}}. \tag{3.7}$$

During the classification stage, the relevant bins previously selected are computed for each testing image and compared to the training images in the relevant bin subspace to determine the testing image label by the nearest neighbor classifier.

3.5.4 Results

Table 3.5 presents the results by the color bin selection approach based on occurrence and sparsity based ranking. We also compare the results obtained in single color space and in

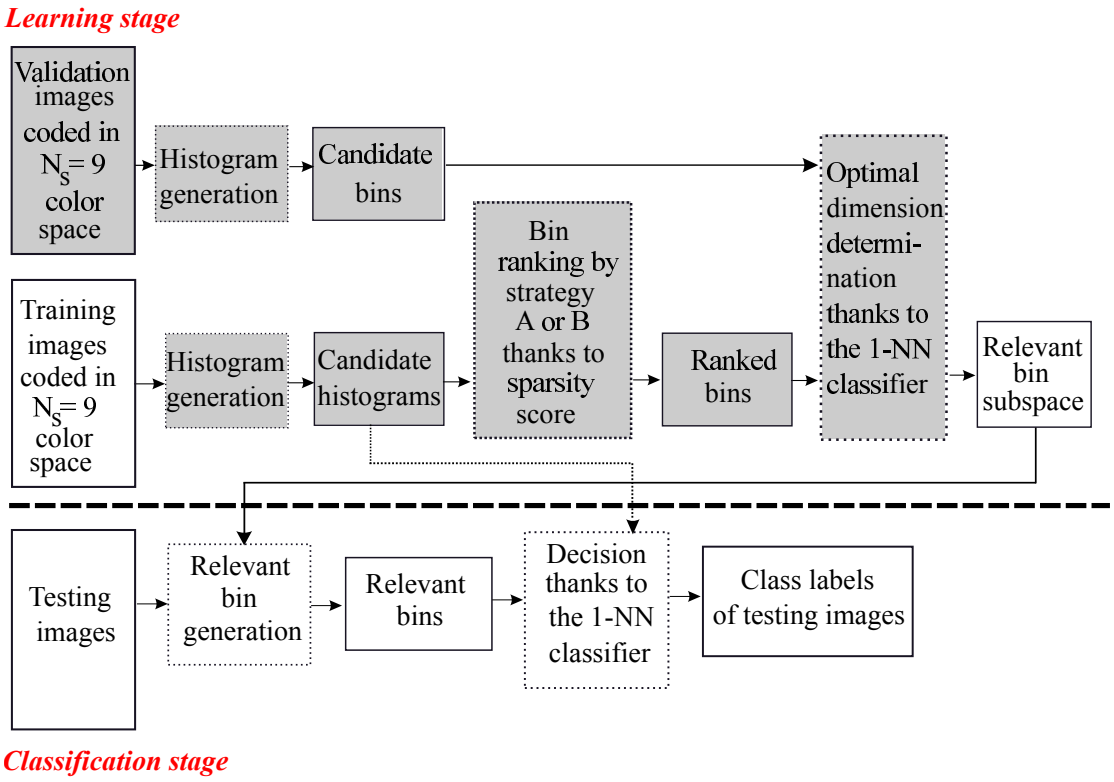


Figure 3.5: Multi color space bin selection approach.

multiple color spaces with and without selection. The third column of this table is divided into three sub-columns corresponding three previously proposed approaches: the occurrence and the two sparsity based ranking strategies. We propose to set the threshold as 90% to select the dominant bins for occurrence based ranking [10]. For the two sparsity ranking approaches, we use the bin selection based on hybrid approaches (figure 3.5) to select the relevant bin subspace. The LBP bin selection approaches improve the classification performance in any cases when a single color space or a multiple color space approaches are considered. The best rate is obtained by occurrence based ranking: 82.5% (with *HSI* space) and 87.8% (with multiple color spaces). The two strategies A and B of the sparsity based ranking give different rates and dimensions. By analyzing the average results in a single color space and in multiple color spaces, we see that the sparsity based ranking with the strategy B gives better results than the strategy A whereas the number of selected bins are also higher. An extended comparison of these strategies will be proposed in the next chapter.

We have presented the LBP bin selection based on the sparsity score. When two bins with the same score, they are ranked randomly. It is interesting to identify which one is more discriminant LBP bins for a better representation. The histogram selection procedure selects the relevant histogram according to their scores. As whole LBP histograms are selected, it is clear that some bins of these histograms are either redundant or even totally uninformative. They

Table 3.5: The classification results of LBP bin selection approaches in a single and multiple color spaces on the New BarkTex database. The values in boxes represent the best rates obtained with each color space and the boldface indicates the best rate obtained of each combination approach.

Color spaces	Without selection		LBP Bin selection					
			Occurence		Sparsity (A)		Sparsity (B)	
	Rate	$\hat{\mathcal{D}}$	Rate	$\hat{\mathcal{D}}$	Rate	$\hat{\mathcal{D}}$	Rate	$\hat{\mathcal{D}}$
RGB	73.2	2304	81.5	281	79.0	1109	78.7	2033
rgb	74.4	2304	74.9	132	74.4	2242	76.0	2274
$I_1I_2I_3$	71.7	2304	74.8	78	74.4	1198	77.2	2011
HSV	70.5	2304	81.0	179	79.5	500	77.6	1544
$b_w r_g b_y$	72.1	2304	74.4	78	77.5	836	78.3	1435
HLS	70.1	2304	81.1	167	78.0	319	76.7	1910
I-HLS	72.1	2304	77.8	129	72.7	1813	78.9	1883
HSI	71.7	2304	82.5	172	79.2	533	79.0	1929
YC_bC_r	71.6	2304	74.5	81	77.0	370	80.6	2116
Average in single space	71.9 $\pm$ 1.3	2304	78.0 $\pm$ 3.4	144	76.9 $\pm$ 2.3	991	78.1 $\pm$ 1.3	1903
Multi spaces	**78.2**	20736	**87.8**	1502	**83.6**	754	**84.4**	16491

can decrease the performances of the learning algorithms. Thus, a dimensionality reduction method for selecting relevant histogram bins of relevant histograms is needed to address these problems. The following section introduces our proposed original approach that combines histogram ranking and bin selection for classification task.

3.6 Combination of histogram ranking and bin selection

We consider that the LBP histogram ranking is important to rank bins as mentioned above. The purpose of this proposition is to filter out the irrelevant bins of the relevant histograms and oppositely find out the relevant bins of the irrelevant histograms. The flow chart of this approach is illustrated by figure 3.6. It is also a hybrid selection method, however the bin ranking strategies are applied after the histogram ranking in this case. Here, we assume that

the bins of discriminant histograms are more relevant than others. During the learning stage, candidate histograms are generated from training images and histogram ranking is applied for each candidate histogram thanks to one of the four scores that are that presented and used in section 2.4.2 and 3.4 (ICS, ASL, SpASL, Simba-2). The differences between figure 3.6 and figure 3.5 are the histogram ranking strategy is applied before bin ranking. In this framework, two more strategies are proposed as follows:

- **Strategy C**: A score for each candidate histogram is computed and the histograms are then ranked according to their score. Next, a bin ranking is achieved for each histogram individually thanks to the sparsity score. We finally concatenate the ranked bins of the ranked histograms.

- **Strategy D**: We assume that the first bin of the most relevant histograms are more relevant than the other bins. So, we propose to rank at first, the group of all the first bins of each histogram in the order of the ranked histograms, then the group of all the second bins are ranked in the second and continuously until the last bin of each histogram. The final bin ranking is a $Q \times \delta_{max}$ uplet vector, where Q is the number of bins of each histogram. The order of the bin in each δ_{max} uplet is based on the ranked histogram.

In order to illustrate the combination of histogram ranking and bin selection approaches (cf. figure 3.7), let us take the same example as in previous section. In this illustration, we assume the histograms are ranked by a consider histogram score as $\mathbf{H}_i^3$, $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$. The bin ranking is achieved for each histogram by the supervised sparsity score. For the strategy C, the bin ranking is obtained by concatenating the three histograms $\mathbf{H}_i^3$, $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$ with previously bin ranking within each histogram as shown in figure 3.7a. The bin ranking obtained by strategy D is a vector composed of the 6 triplet-bins as shown in figure 3.7b. The first triplet is composed by the three first bins of $\mathbf{H}_i^3$, $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$, respectively. This procedure continues to the last triplet is composed by the three last bins of $\mathbf{H}_i^1$, $\mathbf{H}_i^3$ and $\mathbf{H}_i^2$, respectively.

In order find the relevant bin subspace the bin selection procedure is carried out as the same in section 3.5.3.

Table 3.6 presents the results obtained with the two strategies C and D. For the purpose of combination of bin and histogram selection of each strategy C or D, four different histogram scores have been used in the combination of the bin selection (BS) by supervised sparsity score:

1. ICS score and sparsity score,

2. ASL score and sparsity score,

3. SpASL score and sparsity score,

4. Simba-2 score and sparsity score.

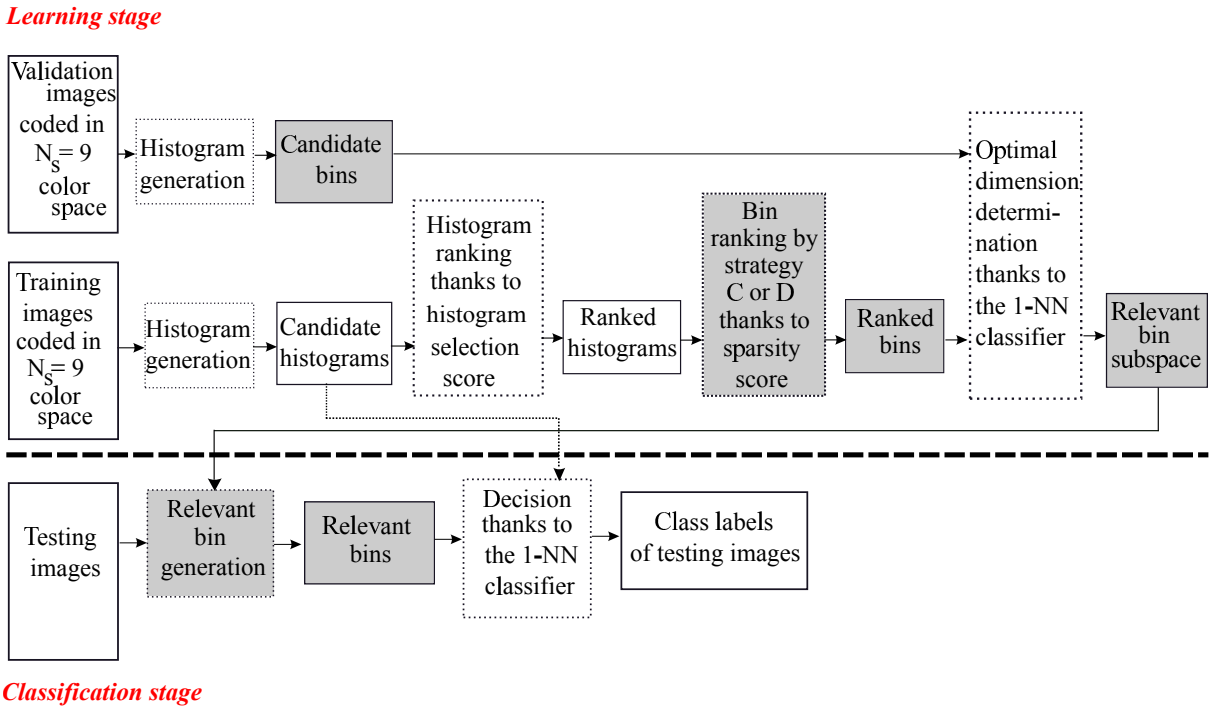
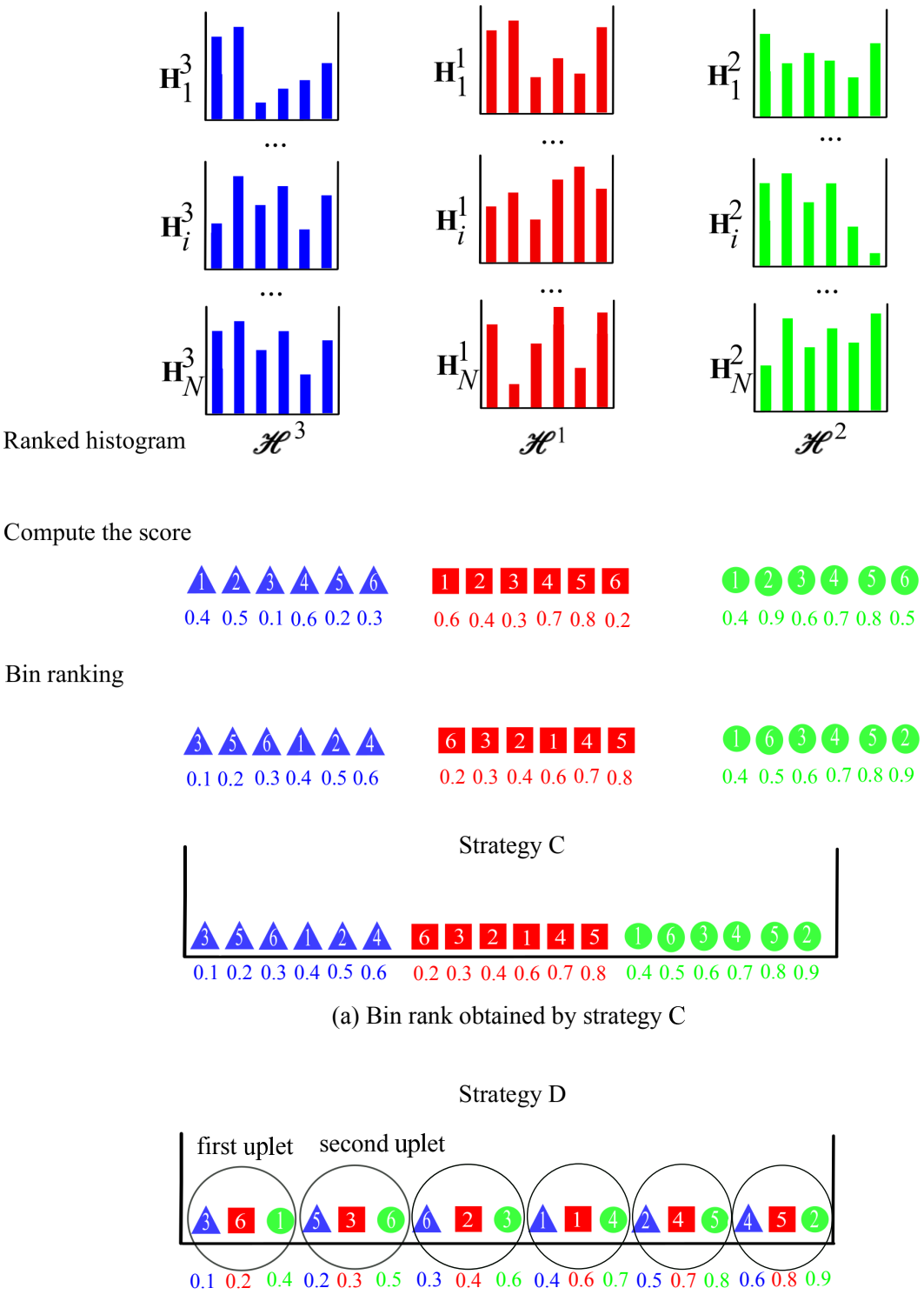


Figure 3.6: The combination of histogram ranking and bin selection in the multiple color spaces.

This table confirms that the combination of histogram ranking and bin selection significantly improves the classification compared with the results obtained without applying the selection scheme in both single color space (+ 9.1%) and in multiple color spaces (+ 10.8%). The best results obtained are 84.2% witht the *RGB* space by the strategy C when a single color space is considered, and 89.0% in multiple color spaces with the strategy D by combining ICS-score and bin selection.



(a) Bin rank obtained by strategy C

(b) Bin rank obtained by strategy D

Figure 3.7: An illustration of the histogram ranking and bin selection approaches.

Table 3.6: The classification results of the combination of histogram ranking and bin selection approaches in a single and multiple color spaces on the New BarkTex database. The values in boxes represent the best rates obtained of each color space and the boldface indicates the best rate obtained of each approach

Color spaces	Without selection		Strategy C								Strategy D							
			ICS & BS		Simba-2 & BS		ASL & BS		SpASL & BS		ICS & BS		Simba-2 & BS		ASL & BS		SpASL & BS	
	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$	Rate	$\hat{D}$
RGB	73.2	2304	84.2	822	84.2	822	84.2	822	84.2	1016	83.7	822	83.7	1016	83.7	1016	83.7	1016
rgb	74.4	2304	77.2	1789	74.4	2303	77.8	709	77.8	709	77.8	1785	74.6	2286	77.9	1530	77.9	1530
$I_1I_2I_3$	71.7	2304	81.4	1581	75.6	2021	80.8	857	80.2	1803	80.6	1564	78.6	2016	80.5	1764	80.5	1764
HSV	70.5	2304	81.6	779	77.6	1165	82.5	851	82.5	851	81.0	768	79.5	1250	81.0	768	81.0	768
$b_w r_g b_y$	72.1	2304	81.0	1721	73.0	2004	81.1	1490	81.1	1490	81.9	1708	77.8	1521	82.1	1524	82.1	1524
HLS	70.1	2304	81.5	824	72.1	2098	81.5	824	81.7	782	81.0	768	81.0	768	81.0	768	81.0	768
I-HLS	72.1	2304	77.4	1575	72.2	2297	78.3	1288	80.8	427	76.8	2032	76.7	729	77.3	1270	79.1	762
HSI	71.7	2304	81.1	793	74.6	1756	81.1	793	81.3	781	79.8	768	77.9	1456	79.8	768	79.8	768
YC_bC_r	71.6	2304	82.0	1639	72.0	2300	80.2	824	79.4	1793	82.5	1778	79.8	1575	82.5	1778	82.5	1778
Average in single space	71.9 ± 1.3	2304	80.8 ± 2.2	1280	75.1 ± 3.9	1862	80.8 ± 1.9	939	81.0 ± 1.8	1072	80.5 ± 2.1	1332	78.9 ± 2.6	1401	80.6 pm 2.0	1242	80.8 ± 1.7	1186
Multi spaces	**78.2**	20736	**88.1**	11501	**85.4**	10251	**87.0**	7460	**87.5**	9580	**89.0**	11457	**87.4**	9912	**88.1**	12466	**88.4**	11985

3.7 Conclusion

In this chapter, a multi color space approach is proposed to overcome the drawback of choosing the single color space best suited for the considered application. We firstly proposed to code images with nine different color spaces and to extract EOCLBP descriptor to characterize the color textures. However, this multi color space approach encountering the problem of high-dimensional feature space, we propose to extend the histogram selection and bin selection approaches to the multi color space domain. These approaches consist in selecting discriminating LBP histograms or bins computed from images coded in several color spaces and thus allow to overcome the difficulty of choosing a well-suited color space to discriminate the considered color texture classes. Four histogram scores are considered for LBP histogram selection in multiple color spaces. Two approaches based on the occurrence and sparsity score are then extended for multi color space bin selection.

In addition, we have proposed to combine the histogram ranking and bin selection according to two selection strategies. These methods aims to select the discriminant LBP bins by using the histogram ranking. In order to illustrate our original approach, we have presented first results obtained on the New BarkTex database. These results show that the multi color space approach clearly improve classification rate instead of using a single color space. Secondly, the proposed selection approaches not only improve the classification in both single and in multiple color spaces but also reduce the dimension of the feature space.

The complete experimental results on other databases are further presented and analyzed in the following chapter.

Chapter 4

Experimental evaluation

Contents

4.1 Impact of similarity measure 115

4.2 Impact of the histogram score 116

4.3 Impact of the LBP-based feature selection strategy 117

4.3.1 Classification results detailed on the New BarkTex 117

4.3.2 Validation on STex, OuTex-TC-00013 and USPTex sets 121

4.3.3 Processing times 125

4.4 Conclusion 127

In this chapter, we propose to compare the strategies of LBP histogram selection and LBP bin selection in the multi color space framework. The approaches presented in the previous chapter will be applied and analyzed thanks to four image databases: New BarkTex, OuTex-TC-00013, USPTex and STex (presented in section 1.1.3). Each database is divided into a half for the training set and a half for the testing set by the holdout method. Let us summarize theses databases by table 4.1.

Table 4.2 lists the methods that will be analyzed. We divide those methods in four categories: no selection, histogram selection, bin selection and combination of histogram ranking and bin selection.

Table 4.1: Summary of image databases used in experiments.

Dataset name	Image size	# class	# training	# testing	Total
New BarkTex	64×64	6	816	816	1632
OuTex-TC-00013	128×128	68	680	680	1360
USPTex	128×128	191	1146	1146	2292
STex	128×128	476	3808	3808	7616

Table 4.2: Summary of the proposed methods used in experiments.

No	Name	Method	Category
1	MCWS	Multi color space without selection	No selection
2	MCSHS-ICS	Multi color space histogram selection based on ICS-score	Histogram selection (c.f section 3.4)
3	MCSHS-Simba-2	Multi color space histogram selection based on Simba-2-score	
4	MCSHS-ASL	Multi color space histogram selection based on ASL-score	
5	MCSHS-SpASL	Multi color space histogram selection based on SpASL-score	
6	MCSBS-Occurrence	Multi color space bin selection based on occurrence (c.f section 3.5.1)	Bin selection
7	MCSBS-Sparsity (A)	Multi color space bin selection based on sparsity-score by strategy A (c.f section 3.5.2)	

Continued on next page . . .

Table 4.2: Summary of the proposed methods used in experiments. continued...

No	Name	Method	Category
8	MCSBS-Sparsity (B)	Multi color space bin selection based on sparsity-score by strategy B (c.f section 3.5.2)	
10	ICS & BS-(C)	Multi color space bin selection by combining ICS-score and bin selection by strategy C	Combination of histogram ranking and bin selection (c.f section 3.6)
11	Simba-2 & BS-(C)	Multi color space bin selection by combining Simba-2-score and bin selection by strategy C	
12	ASL & BS-(C)	Multi color space bin selection by combining ASL-score and bin selection by strategy C	
13	SpASL & BS-(C)	Multi color space bin selection by combining SpASL-score and bin selection by strategy C	
14	ICS & BS-(D)	Multi color space bin selection by combining ICS-score and bin selection by strategy D	
15	Simba-2 & BS-(D)	Multi color space bin selection by combining ICS-score and bin selection by strategy D	

Continued on next page . . .

Table 4.2: Summary of the proposed methods used in experiments. continued...

No	Name	Method	Category
16	ASL & BS-(D)	Multi color space bin selection by combining ASL-score and bin selection by strategy D	
17	SpASL & BS-(D)	Multi color space bin selection by combining SpASL-score and bin selection by strategy D	

In a single color space approach, each texture is coded by $N_\delta = 9$ histograms while $N_\delta = 9 \times 9 = 81$ histograms are considered in the multiple color space approach. In chapter 3, we have shown that this last approach clearly improves the classification rates compared with a single color space approach on the New BarkTex database. Furthermore, when we have compared the result of the multi color space selection approach with those obtained without applying any selection method, we have shown that the multi color space selection not only improves the classification performance but also reduces the dimension of the feature space. In this chapter, we propose to continue to analyze the proposed approaches by applying them on three additional databases, in addition to New BarkTex.

Although OuTex-TC-00013 is widely used, this image set presents a major drawback like USPTex and STex sets. The partitioning used to build these three sets consists in extracting training and testing sub-images from a same original image. However, such a partitioning, when it is combined with a classifier such as the nearest neighbor classifier, leads to biased classification results [139]. Indeed, testing images are spatially close to training images. They are thus correlated and a simple 3D color histogram reaches a high classification accuracy whereas it only characterizes the color distribution within the color space and does not take into account the spatial relationships. On the other hand, the New BarkTex has been create to overcome that drawback (see section 1.2.4). That is the reason why more details and analysis will be proposed in the following on the New BarkTex database.

This chapter is organized in as follows. We first study the impact of the distance used for measuring the similarity between two histograms during the computation of the proposed SpASL-score in order to choose the appropriate distance (see section 4.1). In section 4.2, we evaluate the impact of the histogram scores on the four considered databases to validate the

relevance of the proposed SpASL-score. Next, the different strategies are evaluated, analyzed and compared with the results of the state-of-the-art in section 4.3. Finally, we summarize and conclude the chapter in section 4.4.

4.1 Impact of similarity measure

The ICS-score uses histogram intersection to evaluate the similarity between two histograms, while ASL uses the Jeffrey distance and Simba-2 uses χ^2, respectively. In order to keep the same condition as ASL-score, the proposed SpASL uses the Jeffrey distance to measure the similarity. However, there is an open question regarding the considered distance:"*Does the selected distance has an impact on the histogram score* ?". We propose here to study this question for the SpASL-score.

In order to study the robustness of the SpASL-score, three common distances such as histogram intersection, χ^2 and Jeffrey are compared. For each of the four considered database, the training set is used for the histogram ranking procedure by applying equation 2.65. Then, ranked histograms are used as inputs of the classification process which is performed on the testing set. The L1-distance is associated with the 1-NN classifier while the classification performance is evaluated by the accuracy rate. It is worth to note that our works focus on the feature selection step so that is why we use a basic and non-parametric classifier with a simple distance frequently used.

Table 4.3 shows the classification rates obtained with the different distances associated with the MCSHS-SpASL approach. The first column represents the name of the used database. The second column is divided into three sub-columns corresponding to the three associated distances. The value $\hat{\delta}$, that represents the number of selected LBP histograms for which the well-classified image rate is reached, is also shown in this table. Note that the best results of each row are shown in boldface. This table globally shows that the SpASL-score associated with histogram intersection does not give good results compared with the two other distances whatever the datasets. This is due to the fact that, to extend the distance $(f_i^r - f_j^r)^2$ to the histogram selection context, Kalakech et al. shows that it is necessary to consider a measure which has to be minimized to select the most relevant histograms [9]. The histogram intersection, which has to be maximized, is thus not relevant to compute the ASL-score, and so the SpASL-score. The average result on the four databases also indicates that Jeffrey and χ^2 distance reach the same performance. As Jeffrey and χ^2 give the results close, we propose to use Jeffrey for the proposed SpASL-score hereafter.

We have presented the impact of the similarity measure to the SpASL-score. The next section is dedicated to illustrate the impact of the histogram score.

Table 4.3: The classification obtained by SpASL-score with three measurements of similarity between two histograms by histogram intersection, χ^2, Jeffrey on four texture databases. The value in boldface indicates the best obtained rate of each row.

Database	Distance					
	Histogram intersection		χ^2		Jeffrey	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
New BarkTex	78.4	77	**87.3**	38	**87.3**	37
OuTex-TC-00013	95.2	68	**95.6**	62	**95.6**	62
USPTex	93.7	78	**97.6**	39	97.4	31
STex	93.2	81	96.6	29	**96.8**	32
Average on the four databases	90.1 $\pm$ 7.8		**94.3** $\pm$ 4.7		**94.3** $\pm$ 4.7	

4.2 Impact of the histogram score

Table 4.4 presents the results obtained by MCSHS and MCWS on OuTex-TC-00013, New BarkTex, USPTex and STex database. The first column of this table shows the name of the database. The second column shows the results when no selection is applied. The last column is divided into four sub-columns corresponding to the four histogram scores ICS, Simba-2, ASL and SpASL. The average rate obtained for each approach on the four databases is presented in the last row.

By analyzing this table with the best rate of each row, we can see that the MCSHS approaches improves (+ 0.6%), (+ 9.8%), (+ 3.9%), (+ 3.5%) for OuTex-TC-00013, New BarkTex, USPTex and STex respectively. The best result reached on New BarkTex is obtained by MCSHS-ICS which uses only 42 histograms instead of 81 histograms when no selection is applied. Similarly for other databases, this can confirm that the MCSHS approach not only improves the classification rate but also reduces the subspace dimension. Moreover, the obtained results validate the relevance of the proposed SpASL-score. Indeed, based on the average results obtained on the four databases, this table shows that the SpASL-score improves the results compared with the three other scores and the number of histograms selected by SpASL is each time among the lower.

Table 4.4: The classification results of the multi color space approach without and with histogram selection, on the four texture databases. The value in boldface indicates the best rate obtained for each row.

Database	MCWS		MCSHS							
			ICS		Simba-2		ASL		SpASL	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
OuTex-13	95.0	81	**95.6**	59	95.0	79	95.3	62	**95.6**	62
New BarkTex	78.2	81	**88.0**	42	85.2	40	86.8	29	87.3	37
USPTex	93.7	81	93.7	40	93.8	80	**97.6**	41	97.4	31
STex	93.3	81	94.1	47	95.8	31	96.1	38	**96.8**	32
Average on the four databases	90.0 ± 7.9		92.8 ± 3.2		92.4 ± 4.9		93.9 ± 4.8		**94.3** ± 4.7	

After having presented the results obtained with the multi space histogram selection by the four considered histogram scores, the following section analyses the impact of the LBP-based feature selection approaches.

4.3 Impact of the LBP-based feature selection strategy

This section presents the results reached by the different strategies proposed in chapter 3 and is organized as follows. Section 4.3.1 details the classification results obtained on the New BarkTex database. Section 4.3.2 compares the results reached by our approach with those obtained in the state-of-the-art on the further OuTex-TC-00013, USPTex and STex databases. Finally, the processing times of the proposed approaches is presented and analyzed in section 4.3.3.

4.3.1 Classification results detailed on the New BarkTex

Table 4.5 shows the classification results obtained by the different proposed approaches on the New BarkTex database. The first and second columns represent the categories and the approach

names. The third and fourth column indicate the rate of well-classified images reached and the subspace dimension. The last column shows the average result of each category.

Table 4.5: Classification results of the proposed approaches in the multi color space framework on the New BarkTex database. The value in box indicates the highest rate within each category while the value in boldface indicates the best rate obtained among all approaches.

Category	Approach	Rate	$\hat{D}$	Average
No selection	MCWS	78.2	20736	78.2
Histogram selection	MCSHS-ICS	88.0	10752	86.8 ± 1.1
	MCSHS-Simba-2	85.2	10240	
	MCSHS-ASL	86.8	7424	
	MCSHS-SpASL	87.3	9472	
Bin selection	MCSBS-Occurrence	87.8	1502	85.2 ± 2.2
	MCSBS-Sparsity (A)	83.6	754	
	MCSBS-Sparsity (B)	84.4	16491	
Combination of histogram ranking and bin selection	ICS & BS-(C)	88.1	11501	**87.7** ± 1.1
	Simba-2 & BS-(C)	85.4	10251	
	ASL & BS-(C)	87.0	7460	
	SpASL & BS-(C)	87.5	9580	
	ICS & BS-(D)	**89.0**	11457	
	Simba-2 & BS-(D)	87.4	9912	
	ASL & BS-(D)	88.1	12466	
	SpASL & BS-(D)	88.4	11985	

By analyzing this table, we can see that the classification rate range from 78.2% to 89.0%. The highest rate obtained by a MCSHS approach is 88.0% with the ICS-score. This rate is then

compared with the highest rate of the bin selection category. This rate is obtained by MCSBS-Occurrence which is less good with (- 0.2%). However, MCSBS-Occurrence selects a feature space with a lower dimension than the MCSHS approaches. MCSBS-Sparsity (A) and (B) improve the classification rate compared with the MCWS approach, nevertheless these two approaches give the lowest rates. Based on the average results in the last column, we can observe that the combination of histogram ranking and bin selection gives the best result. It improves (+ 0.9%) and (+ 2.5%) compared with the MCSHS and MCSBS approach, respectively.

The combination of histogram ranking and bin selection guarantees to obtain a rate always greater or equal than those obtained by histogram selection. For example, the MCSHS-ICS approach gives at 88.0% when we combine this score with bin selection, the results obtained by strategies C and D are 88.1% and 89.0%, respectively. We find the same conclusion for the combination with the other scores. The best results depend on the choice of the histogram score used for the combination. Regarding the number of selected bins, the combination methods do not reduce the dimension compared with MCSBS and MCSHS approaches. For example, the combination of SpASL-score and bin selection achieves a rate of 88.4% with the strategy D ad with the number of bins equal to 11985. It improves the rate obtained by MCSHS-SpASL + 1.1% but uses more 2513 bins. This can be explained by the fact that the combination approaches seek the relevant bins for all ranked bins while the MCSHS seeks only the relevant histogram and might miss some relevant bins of the irrelevant histograms. That is the reason why we can improve the rate and sometimes the dimension does not reduce. Globally, we can observe that the combination by the strategy D gives better rates than those obtained by strategy C.

The classification results obtained on the New BarkTex by our proposed approaches are then compared with those obtained by the different studies of the state-of-the-art in table 4.6. The rows labeled as gray correspond to experiments that are carried out in this work whereas the other rows correspond to results published by other authors. The first column refers to the related papers and indicates the used selection method. The considered color spaces used to classify the images are presented in the second column of the table. The third column shows the descriptors which have been analyzed to discriminate the different color texture classes. Finally, the last column shows the rate of well-classified testing images obtained with the classifier presented in the fourth column. The classification rate obtained by other authors ranges from 58.6% to 82.1%. This table indicates that our approaches outperform the state-of-the-art results, improving (+ 6.9%) on New BarkTex database by the combination of ICS-score and LBP bin selection methods.

Table 4.6: Comparison between the well-classified image rates reached with the New BarkTex set. The italic values indicate the results obtained based on our implementation while the underlined values indicate the results extracted from [47].

Reference	Color space	Features	Classifiers	Accuracy
ICS & BS-(D)	9 color spaces	EOCLBP	1-NN	89.0
ICS & BS-(C)	9 color spaces	EOCLBP	1-NN	88.1
MCSHS-ICS	9 color spaces	EOCLBP	1-NN	88.0
MCSBS-Occurrence	9 color spaces	EOCLBP	1-NN	87.8
MCSBS-Sparsity (B)	9 color spaces	EOCLBP	1-NN	84.4
[154]	*RGB*	CLBC	1-NN	84.3
MCSBS-Sparsity (A)	9 color spaces	EOCLBP	1-NN	83.6
[66]	*RGB*	3D-ASDH	SVM	82.1
[8] (with selection method)	*RGB*	EOCLBP	1-NN	81.4
[9] (with selection method)	*RGB*	EOCLBP	1-NN	81.4
[62]	*RGB*	LBP and local color contrast	1-NN	<u>80.2</u>
[56]	*RGB*	Between color component LBP	1-NN	*79.9*
MCWS	9 color spaces	EOCLBP	1-NN	78.2
[47]	*RGB*	Mix color order LBP histogram	1-NN	77.7
[138] (with selection method)	20 color spaces	RSCCM	1-NN	75.9
[259]	*RGB*	CLBP	1-NN	*72.8*
[34]	*RGB*	Color angles LBP	1-NN	<u>71.0</u>
[260]	*RGB*	DRLBP	1-NN	*61.4*
[13]	*RGB*	Color histograms	1-NN	*58.6*

4.3.2 Validation on STex, OuTex-TC-00013 and USPTex sets

Table 4.7, 4.9 and 4.8 present the classification results obtained by our proposed approaches and those obtained by the different studies which have applied a color texture classification algorithm on STex, OuTex-TC-00013 and USPTex respectively.

Table 4.7: Comparison between the well-classified image rates reached with the STex database. The italic values indicate the results obtained based on our implementation.

Reference	Color space	Features	Classifiers	Accuracy
SpASL & BS-(D)	9 color spaces	EOCLBP	1-NN	98.1
SpASL & BS-(C)	9 color spaces	EOCLBP	1-NN	96.9
ASL & BS-(C)	9 color spaces	EOCLBP	1-NN	96.8
MCSHS-SpASL	9 color spaces	EOCLBP	1-NN	96.7
MCSBS-Occurrence	9 color spaces	EOCLBP	1-NN	96.7
MCSBS-Sparsity (A)	9 color spaces	EOCLBP	1-NN	94.7
MCSBS-Sparsity (B)	9 color spaces	EOCLBP	1-NN	94.7
MCWS	9 color spaces	EOCLBP	1-NN	93.7
[260]	*RGB*	DRLBP	1-NN	*89.4*
[134]	$L^*a^*b^*$	Wavelet coefficients	Bayes	77.6
[135]	*RGB*	Color contrast occurrence matrix	1-NN	76.7
[131]	$L^*a^*b^*$	Soft color descriptors	1-NN	55.3

For the three databases, the best rates obtained are given by the combination of the SpASL-score and bin selection with the strategy D. Our approaches outperform all other methods for USPTex and STex by improving the classification rate by (+ 2.4%) and (+ 8.7%), respectively.

On the other hand, our approaches do not outperform two works on OuTex-TC-00013 [138,

Table 4.9: Comparison between the well-classified image rates reached with the OuTex-TC-00013 set.

Reference	Color space	Features	Classifier	Accuracy
[138] (with selection method)	28 color spaces	RSCCM	1-NN	96.6
[66]	*HSI*	3D-ASDH	SVM	95.8
SpASL & BS-(D)	9 color spaces	EOCLBP	1-NN	95.7
SpASL & BS-(C)	9 color spaces	EOCLBP	1-NN	95.7
MCSHS-SpASL	9 color spaces	EOCLBP	1-NN	95.6
MCSHS-ICS	9 color spaces	EOCLBP	1-NN	95.6
[13]	*HSV*	3D Color histogram	1-NN	95.4
SpASL & BS-(C)	9 color spaces	EOCLBP	1-NN	95.3
MCSBS-Sparsity (A)	9 color spaces	EOCLBP	1-NN	95.2
MCSBS-Sparsity (B)	9 color spaces	EOCLBP	1-NN	95.2
MCWS	9 color spaces	EOCLBP	1-NN	95.0
[114]	*RGB*	Fractal descriptors	LDA	95.0
[41]	*RGB*	Haralick features	5-NN	94.9
[56]	*RGB*	3D Color histogram	3-NN	94.7
[126]	*I-HLS*	3D Color histogram	1-NN	94.5
[41]	*RGB*	Haralick features	1-NN	94.1
[122]	*HSV*	EOCLBP	SVM	93.5
[9] (with selection method)	*RGB*	EOCLBP	1-NN	93.4

Continued on next page . . .

Table 4.9: Comparison between the well-classified image rates reached with the OuTex-TC-00013 set continued...

Reference	Color space	Features	Classifier	Accuracy
[8] (with selection method)	*RGB*	EOCLBP	1-NN	92.9
MCSBS-Occurrence	9 color spaces	EOCLBP	1-NN	92.9
[139] (with selection method)	*HLS*	RSCCM	1-NN	92.5
[13]	*RGB*	Between color component LBP histogram	1-NN	92.5
[109]	*RGB*	Quaternion-Michelson Descriptor	1-NN	91.3
[127]	*RGB*	Texton	1-NN	90.3
[261]	*RGB*	Combine color and LBP-based features	1-NN	90.2
[134]	$L^*a^*b^*$	Wavelet coefficients	Bayes	89.7
[152]	*RGB*	Intensity-Color Contrast Descriptor	1-NN	89.3
[260]	*RGB*	DRLBP	1-NN	89.0
[126]	*I-HLS*	Autoregressive models and 3D color histogram	1-NN	88.9

Continued on next page . . .

Table 4.9: Comparison between the well-classified image rates reached with the OuTex-TC-00013 set continued...

Reference	Color space	Features	Classifier	Accuracy
[142]	*RGB*	Halftoning Local Derivative Pattern and Color Histogram	1-NN	88.2
[232]	$L^*a^*b^*$	Autoregressive models	1-NN	88.0
[13]	*RGB*	Within color component LBP histogram	1-NN	87.8
[47]	*RGB*	Mix color order LBP histogram	1-NN	87.1
[62]	*RGB*	Color angles LBP	1-NN	86.2
[34]	*RGB*	LBP and local color contrast	1-NN	85.3
[54]	*RGB*	Features from wavelet transform	7-NN	85.2
[259]	*RGB*	CLBP	1-NN	84.4
[135]	*RGB*	Color contrast occurrence matrix	1-NN	82.6
[131]	*HSV*	Soft color descriptors	1-NN	81.4

Continued on next page ...

Table 4.9: Comparison between the well-classified image rates reached with the OuTex-TC-00013 set continued...

Reference	Color space	Features	Classifier	Accuracy
[144]	*RGB*	HEP	1-NN	80.9
[128]	*RGB*	Fuzzy aura matrices	1-NN	80.2
[148]	*RGB*	Modified LBP	1-NN	67.3

66]. In [138], Porebski et al. use Haralick features extracted from Reduced Size Chromatic Co-occurrence Matrices (RSCCMs) by using 28 color spaces compared with 9 color spaces by our proposed approach. In [66], Sandid et al. use the SVM classifier and characterize the texture by the three-dimensional adaptive sum and difference histograms descriptors. Our results are close to those obtained in [66] (the difference is equal to 0.1%) whereas we use a simple 1-NN classifier. Other studies give results range from 67.3% to 95%. Note that the MCSBS-Occurrence gives in this case a less good result with the rate obtained of 92.9%.

In the next subsection, we propose to compare the computing time of the proposed approaches.

4.3.3 Processing times

We select several approaches from each category to compare the processing times required by the learning and the classification stages on the New-BarkTex set. Tables 4.10 and 4.11 show the processing times of the two stages, respectively. These times are obtained by using the Matlab software and a PC cadenced at 3.20 GHz with 24 Gb RAM. Note that we classify these approaches by the ascending of total times.

When a selection is performed, the learning stage consists in the computation of all available histograms (generation phase) and a selection phase.

For the MCSHS approach, the dimension of the discriminating histogram subspace is determined thanks to several classifications (see figure 3.2). This approach thus requires to compute all the histograms for the training (48.0 s) and the validation (48.0 s) images, and to evaluate the rate of well-classified validation images for the candidate histogram subspaces with different dimensions (630.3 s with the ICS-score). It is thus computationally costly (726.3 s), contrary to the MCSBS approach (78.1 s), which do not require any classification to determine the relevant subspace. Indeed, it only consists in extracting all the histograms for the training images (48.0 s) and determining the dominant bins (30.1 s) as shown in figure 3.3. MCSHS-SpASL

takes longer times than others histogram score (797.8 s) since it has to compute the sparse similarity matrix. MCSBS-Sparsity and the ICS & BS-(D) are approaches that need a learning computation time even more important. Indeed, they take approximately 39 hours to determine the bin subspace because they have to carry out 20736 classifications in order to determine the dimension of the feature space instead of 81 for MCSHS approaches. We only illustrate the processing times of the combination approach ICS & BS-(D) since other combination approaches (i.e ASL & BS, Simba-2& BS, SpASL & BS) is nearly the same.

When no selection is performed, the learning stage only consists in computing the histograms from the training images (48.0 s) in order to compare them to the histograms extracted during the classification stage from the testing images with the nearest neighbor classifier.

The classification stage consists in the computation of the previously selected histograms or bins from testing images (generation phase) and a decision phase.

When no selection is performed, the images are characterized by all available histograms, that is to say in a $81 \times 256 = 20736$ dimensional feature space. This high dimension leads to a high computation time for generation (48.0 s) and decision (15.8 s).

When the MCSHS-ICS approach is considered, a low dimensional histogram subspace is determined during the learning stage. The images are thus characterized by a reduced number of histograms (42 instead of 81) during the classification stage. Operating a selection allows here to reduce the generation and the decision times of the classification stage (31.5 s instead of 63.8 s with the ICS-score).

When MCSBS-Sparsity approach is performed, a low dimensional subspace is also determined during the learning stage and used for classification (754 bins are computed instead of 20736). The dimension of the relevant feature subspace is lower than the dimension obtained with the MCSHS-ICS approach (754 instead of 10752). It allows to assign more quickly the images to the estimated classes (0.7 s instead of 8.2 s) during the classification stage.

In order to compare the efficiency of the proposed approaches, we select three approaches and compare the rate obtained and the processing time by the table 4.12.

According to the accuracy reached and the processing times of learning stage represented in table 4.12, we are interested in the MCSBS-Occurrence and MCSHS-ICS approaches. These approaches allow thus to obtain similar classification performances for the New-BarkTex set, whether in accuracy or classification computation time, with a slight advantage for the MCSHS approach. Obviously, the selection provided by MCSHS is achieved at the price of a costly learning time. However, this learning computation time can be reduced as shown in [105]. Indeed, a way to speed up the histogram selection approach is presented in this paper. It consists in considering during the learning stage a reduced neighborhood or a combination of reduced neighborhoods to compute LBP histogram. The results show that this approach gives as good results as those obtained with the full neighborhood while reducing the learning time, that

reinforces the interest of the proposed MCSHS approach.

There is another way to reduce the times of the learning stage by operating the selection of the bin-uplets. Instead of operating the selection bin to bin by the combination of histogram ranking and bin selection with strategy D, we might think that it can be done by operating a selection of bin-uplets. This approach can reduce the number of classifications operated to determine the optimal subspace in the multi color space.

4.4 Conclusion

This chapter presents the results obtained on the four considered databases (New BarkTex, OuTex-TC-00013, USPTex, STex) by LBP-based features selection in the framework of multi color spaces. We first presented the impact factor of the distance measure between two histograms for the proposed SpASL-score. We then presented the validation of the proposed SpASL-score. The obtained results show that the SpASL-score improves the classification results compared with the three other scores. The detailed results for the four of LBP-based feature selection approach categories are then presented and analyzed. The results obtained are very encouraging since the LBP-based feature selection approaches proposed in this work improve the rates of well-classified images compared to the approaches which consider a single color space or the approach which does not perform any selection.

Finally, we have selected several approaches to analyze the processing times of the learning and classification stage. Our proposed approaches outperform other methods in the state-of-the-art on three benchmark databases, however the combination of histogram and bin selection approaches are costly to compute due to a large classification steps operated.

Table 4.8: Comparison between the well-classified image rates reached with the USPTex database. The italic values indicate the results obtained based on our implementation.

Reference	Color space	Features	Classifiers	Accuracy
SpASL & BS-(D)	9 color spaces	EOCLBP	1-NN	98.1
ASL & BS-(C)	9 color spaces	EOCLBP	1-NN	97.6
MCSHS-ASL	9 color spaces	EOCLBP	1-NN	97.6
MCSHS-SpASL	9 color spaces	EOCLBP	1-NN	97.4
MCSBS-Occurrence	9 color spaces	EOCLBP	1-NN	97.3
[261]	*RGB*	Fusion Color texture and LBP-based features	SVM	95.7
MCSBS-Sparsity (A)	9 color spaces	EOCLBP	1-NN	94.8
MCSBS-Sparsity (B)	9 color spaces	EOCLBP	1-NN	94.7
[116]	Luminance	Local jet and LBP	LDA	94.3
[109]	*RGB*	Quaternion-Michelson Descriptor	1-NN	94.2
[142]	*RGB*	Halftoning Local Derivative Pattern and Color Histogram	1-NN	93.9
MCSHS-Simba-2	9 color spaces	EOCLBP	1-NN	93.8
MCWS	9 color spaces	EOCLBP	1-NN	93.7
[260]	*RGB*	DRLBP	1-NN	*89.4*
[262, 115]	Luminance	Fractal descriptors	LDA	85.6
[62]	*RGB*	Color angles	1-NN	88.8
[156]	Luminance	Local multi-resolution patterns	1-NN	86.7
[47]	*RGB*	Mix color order LBP histogram	1-NN	84.2
[34]	*RGB*	LBP and local color contrast	1-NN	82.9
[259]	*RGB*	CLBP	1-NN	72.3
[131]	$L^*a^*b^*$	Soft color descriptors	1-NN	58.0

Table 4.10: Processing times of the learning stage for 816 images whose size is 64 × 64 pixels.

	Learning stages (s)			
	Generation		Selection	Total
Approach	from training images	from validation images		
No selection	48.0	-	-	48.0
MCSBS-Occurrence	48.0	-	30.1	78.1
MCSHS-ICS	48.0	48.0	630.3	726.3
MCSHS-ASL	48.0	48.0	648.1	744.1
MCSHS-Simba-2	48.0	48.0	665.2	761.2
MCSHS-SpASL	48.0	48.0	701.8	797.8
MCSBS-Sparsity	48.0	48.0	143252.4	143348.4
ICS & BS-(D)	48.0	48.0	144462.2	144548.2

Table 4.11: Processing times for classifying 816 images whose size is 64 × 64 pixels.

	Classification stages (s)		
Approach	Generation from testing images	Decision	Total
MCSBS-Sparsity ($\hat{\mathcal{D}} = 754$)	9.1	0.7	9.8
MCSHS-ASL ($\hat{\mathcal{D}} = 27 \times 256 = 6912$)	17.8	5.6	23.4
MCSHS-SpASL ($\hat{\mathcal{D}} = 37 \times 256 = 9472$)	21.2	7.2	28.4
MCSHS-Simba-2 ($\hat{\mathcal{D}} = 40 \times 256 = 10240$)	23.2	7.8	31.0
MCSBS-Occurrence ($\hat{\mathcal{D}} = 1502$)	30.3	1.2	31.5
MCSHS-ICS ($\hat{\mathcal{D}} = 42 \times 256 = 10752$)	24.3	8.2	32.5
ICS & BS-(D) ($\hat{\mathcal{D}} = 11457$)	31.4	9.1	40.5
No selection ($\hat{\mathcal{D}} = 81 \times 256 = 20736$)	48.0	15.8	63.8

Table 4.12: The summary of selected approaches.

		Time	
Approach	Rate	Learning	Classification
ICS & BS-(D)	89.0	39 h	40.5 s
MCSHS-ICS	88.0	726 s	32.5 s
MCSBS-Occurrence	87.8	78.1 s	31.5 s

Conclusion and perspectives

In this work we present our research on feature selection for supervised color texture classification problem. The main contribution is the development of different strategies for LBP histogram and LBP bin histogram selection in the framework of a multi color space.

We briefly summarized our contribution in the following:

- In chapter 1, the current state-of-the-art of color texture classification in supervised context was organized in order to provide a quick and compact overview for the reader. The principal families of color spaces have been presented as well as the main color texture descriptors. The LBP operator and its variants to color have been discussed. We introduced several key concept of supervised texture classification such as: the context, the commonly supervised classifier and the evaluation methods. Several color texture databases used to carried out in the experiments have been reviewed.

- In chapter 2, the literature review of feature selection methods related to our research have been introduced. The taxonomy of feature selection methods is briefly reviewed according two main axes: the learning contexts and the evaluation strategies. Two main ranking-based approaches are briefly reviewed. These approaches can be achieved by associating a score for each feature or by applying an algorithm which gives weights for a feature subset. Then, the feature selection methods applied to LBP are discussed depending on two groups: LBP bins selection and histogram selection. We present our first contribution in this chapter by proposing a novel histogram score which is based on the sparse similarity matrix.

- In chapter 3, a multi color space approach is designed for color texture classification. The textures are characterized by EOCLBP extracted from images coded in $N_S = 9$ different color spaces and the corresponding histograms defined a color texture feature space. The most discriminating color texture features have then been selected thanks to a feature selection procedure performed during a supervised learning. We extended the LBP histogram selection and LBP bin selection from a single color space to multiple color spaces

by two approaches MCSHS and MCSBS. Two strategies of selection are then proposed by the combination of histogram ranking and bin selection. In order to show the contribution of the proposed approaches, we presented the results on the New BarkTex database in a single color space and in multiple color spaces.

- In chapter 4, all proposed methods in this thesis are evaluated on several benchmark texture databases. We first study the impacts of several measured histogram distances to select the relevant ones. We then validate the SpASL-score on four benchmark texture databases. The results show that this score improved the classification performance in a single and multiple color spaces. The comparison of classification results by different approaches of bin selection and histogram selection are presented and compared with those obtained in the state-of-the-art. The results obtained with the proposed approaches are very encouraging since these methods improve the rates compared to the approaches which consider a single color space or the approach which does not perform any selection. We then compared the processing time of the proposed approaches in order to find a compromise between the performances and the time processing of the proposed approaches.

Based on the results presented in this thesis, we are planning several perspectives for future research directions. The further experimentations can be realized in the short term perspective:

- Although the combination histogram ranking and bin selection has outperformed other methods, it introduces a few limitations about the time processing which should be reduced in the future work. By the first observation, we can improve the proposed strategy D by the bin-uplet selection (c.f section 3.6). This will reduce the number of classification operations to determine the optimal subspace from 20736 to 256.

- In the current work, based on the general graph-preserving feature selection framework, we proposed to use the sparse similarity matrix based on $l1$ graphs for histogram selection, there are other kinds of graphs (e.g. $l2$ graph) that can also be used under general graph-based feature selection framework. It is interesting to investigate whether using other kinds of graphs can also lead to performance improvement.

- Motivated from the developments and utilizations of the different distances to measure similarity between two histograms by the SpASL-score, the comparative work empirically should be explore in order to address the issue: Which distance is appropriate for histogram selection ?

In the long-term perspective, there are two ideas can be explored:

- Many well-known color spaces exist and each one presents specific properties. Moreover, an hybrid color space is defined by selecting a set of three color components from different color space and used in color pixel classification for the segmentation task [256]. We would like to extend this work by using the hybrid color space in order to compare the results obtained in a hybrid color space and in a single color space.

- The first aim of the proposed score is the improvement of ASL-score. Moreover, we notice that the sparse similarity matrix can be constructed in an unsupervised way without using class label, integrated into the ASL-score. In reality, the task of feature selection became more challenging with the so-called “small labeled-sample” problem, in which the amount of data that is unlabeled could be much larger than the amount of labeled data. This allows us to extend this work in the other learning context (semi-supervised or unsupervised) in the framework of feature selection.

Appendices

Appendix A

A summary of studies on color texture classification

Table A.1 summaries the different color texture classification approaches experimented on the four texture databases presented in section 1.2.4:

- The first column of this table gives the name of the author, the year of publication and the reference.

- The second column mentions the color texture features used in the experimentation.

- The third column gives the color spaces used in the experimentation.

- The fourth column presents the name of the test suites.

- The last column indicates the classifiers used in the experimentation.

Table A.1: A summary of studies on color texture classification.

Ref.	Color texture features	Color space used	Database	Classifier
Palm, 2002 [12]	Gabor features	*RGB, HSL*	BarkTex	5-NN
Pietikäinen, 2002 [56]	Image histogram, LBP, Gabor features	*RGB*, $I_1I_2I_3$	Outex-TC-00013	3-NN
Munzenmayer, 2002 [60]	Sum and Difference Histograms	*RGB, HSV, L*u*v**	BarkTex	1-NN
Mäenpää, 2004, [13]	Image histogram, OCLBP	*RGB, HSV*	Outex-TC-00013	1-NN
Palm, 2004, [36]	Image histogram	*RGB, L*u*v**	BarkTex	5-NN
Arvis 2004, [41]	Haralick features from color co-occurence matrices.	*RGB, HSV*, YC_bC_r	Outex-TC-00013	5-NN
Xu, 2005, [54]	Wavelet features	*HSV*, $I_1I_2I_3$	Outex-TC-00013	7-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Iakovidis, 2005, [122]	LBP histogram	*HSV*	Outex-TC-00013	SVM
Aptoula 2007, [232]	Morphological covariance	*RGB, L*a*b*, YUV*	Outex-TC-00013	1-NN
Porebski 2007, [136]	Haralick features from color co-occurrence matrices	*RGB, XYZ, xyz, Irg, L*u*v*, L*a*b*, Y'I'Q', Y'U'V', r*b*b*,* $I_1I_2I_3$, $R_FG_FB_F$, $R_EG_EB_E$, $R_CG_CB_C$, AC_1C_2, $b_wr_gb_y$	BarkTex	1-NN
He, 2009 [125]	LTP histogram	*RGB,* YC_bC_r	Outex-TC-00013	3-NN
Qazi 2011, [126]	Color spectral analysis	*RGB, IHLS, L*a*b**	Outex-TC-00013	1-NN
Alvarez 2012, [127]	Texton features	*HSI, HSV*	Outex-TC-00013	1-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Backles 2012, [117]	Fractal descriptors	*RGB*	Outex-TC-00013	LDA
Cusano 2013, [152]	Intensity-Color Contrast Descriptor	*RGB, HSV, L*a*b*,* $I_1I_2I_3$	Outex-TC-00013	1-NN
Qazi 2013, [263]	Color spectral analysis	*RGB, IHLS, L*a*b*,* $I_1I_2I_3$	Outex-TC-00013	3-NN
Porebski 2013, [8]	E-OCLBP	*RGB, HSV, YUV,* $I_1I_2I_3$	Outex-TC-00013 BarkTex	1-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Porebski 2013, [139]	Haralick features from Reduced Size Chromatic Co-occurrence Matrices	*RGB, XYZ, UVW, HSV, LUV,* AC_1C_2 $b_w r_g b_y$, YC_bC_r, $L^*C_{uv}h_{uv}$, *L*u*v* Y'I'Q', Y'U'V', HSI, HLS, Yxy,* I_1rg, LCh_1Ch_2, $I_1S_2H_1$, , *L*a*b** $AC_{C1C2}h_{C1C2}$, $LC_{Ch1Ch2}h_{Ch1Ch2}$, $I_1S_1H_3$, $L^*C_{ab}h_{ab}$, $L^*S_{uv}h_{uv}$, $I_1C_{I2I3}h_{I2I3}$ $Y'C'_{IQ}h'_{IQ}$, $Y'C'_{UV}h'_{UV}$, $b_wC_{rgby}h_{rgby}$	Outex-TC-00013	1-NN
El Maliani 2014, [134]	Statistical multi-model and geodesic distance	*RGB, HSV, L*a*b**	Outex-TC-00013 STex	Bayesian Naives
Hammouche 2015, [128]	Fuzzy gray-level aura matrices	*RGB*	Outex-TC-00013 STex	1-NN
Kalakech 2015, [9]	E-OCLBP	*RGB, HSV, YUV,* $I_1I_2I_3$	Outex-TC-00013 BarkTex	1-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Martínez 2015, [135]	Color contrast ocurrence matrix	*L*a*b**	Outex-TC-00013 STex	5-NN
Da silva 2015, [116]	Corrosion-Inspired Texture Analysis	*RGB*	USPTex	LDA
Oliveira 2015, [155]	Local jet space	*RGB*	USPTex	LDA
ahmadvand 2016, [129]	Spatial filter banks	*RGB*	Outex-TC-00013	k-NN
Florindo 2016, [264]	Local connectivity index	*RGB*	Outex-TC-00013	LDA
Guo 2016, [142]	Halftoning Local Derivative Pattern and Color Histogram	*RGB*	Outex-TC-00013 USPTex	1-NN
Ledoux 2016, [130]	Color morphological texture features	*RGB*	Outex-TC-00013	3-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Ledoux 2016, [47]	Compact descriptors color LBP	*RGB*	Outex-TC-00013 BarkTex USPTex	1-NN
Sandid 2016, [66]	Three-dimensional adaptive sum and difference histograms	*RGB, XYZ, UVW, HSV, LUV,* AC_1C_2, $b_w r_g b_y$, YC_bC_r, $L^*C_{uv}h_{uv}$, *Yxy, Y'I'Q', Y'U'V',HSI, HLS, L*u*v*,* I_1rg, LCh_1Ch_2, $I_1S_2H_1$, $AC_{C1C2}h_{C1C2}$, $LC_{Ch1Ch2}h_{Ch1Ch2}$, $I_1S_1H_3$, $L^*C_{ab}h_{ab}$, $L^*S_{uv}h_{uv}$, $Y'C'_{IQ}h'_{IQ}$, $Y'C'_{UV}h'_{UV}$, $b_wC_{rgby}h_{rgby}$, $I_1C_{I2I3}h_{I2I3}$, *L*a*b*, LMS,* $O_1O_2O_3$	Outex-TC-00013 BarkTex	SVM
Bello-cerezo 2016, [131]	Soft color descriptors	*RGB, HSV, YUV, YCbCr, L*a*b* YIQ, L*u*v*, XYZ,* $I_1I_2I_3$	Outex-TC-00013 USPTex STex	1-NN

Continued on next page . . .

Table A.1: A summary of studies on color texture classification continued...

Ref.	Color texture features	Color space used	Database	Classifier
Casanova 2016, [114]	Fractal descriptors	*RGB, L*a*b*, HSV, IHLS,* $I_1I_2I_3$	Outex-TC-00013 USPTex	LDA
Lan 2016, [109]	Quaternion-Michelson descriptors	*RGB*	Outex-TC-00013 USPTex	1-NN
Naresh 2016, [148]	Modified Local binary patterns	*RGB*	Outex-TC-00013	1-NN
Cernadas 2017, [132]	Intensity texture	*RGB, L*a*b*, Lab2000HL, HSV,* $I_1I_2I_3$	Outex-TC-00013 USPTex	1-NN

Appendix B

Application of LBP-based features for lace images classification

In the recent years, there is an increasing need to digitize museum lace textile collections. The aim is to preserve and promote the regional industrial heritage by making available to the public, designers and artists, large lace image databases. Calais's International Center for Lace and Fashion is a museum located at Calais in France. This center preserves the history of industrial lace production. The collection of lace consists of about 100,000 production samples which are divided into 253 different volumes. In order to assist the access to these databases, it is necessary to characterize the texture of lace.

In addition, lace is a soft and extensible material and can be easily deformed which makes the texture analysis a challenging problem. Moreover, lace contains a large range of decorative motifs. Figure B.1 illustrates examples of lace with flower and geometric decorative motifs. This figure shows that lace is composed of two parts, a decorative motif (1) on a basic pattern (2).

Although there is a wide range of decorative motifs, there is a limited number of basic patterns. So, we firstly propose to analyze the texture of lace basic patterns. Figure B.2shows examples of these lace basic texture patterns. It is interesting to note that the figures B.2 (a)-(b) and (c)-(d) show basic textures that appear similar although they have different shapes. The lace texture analysis needs to firstly remove the background of the basic patterns since no information about the texture is contained in the background. Due to orientation changes of the laces, their texture has to be represented by rotation invariant descriptors.

In texture analysis literature, the local binary pattern descriptors (LBP) are widely used. However, the parameters of LBP descriptors have to be carefully chosen in order to reveal the texture structure of lace images. In [265], the lace texture is represented by descriptors based on LBP histograms in order to classify lace basic patterns. They used the Fourier Transform applied on LBP histograms, named LBP-HF [94]. Following this work, we propose to adjust

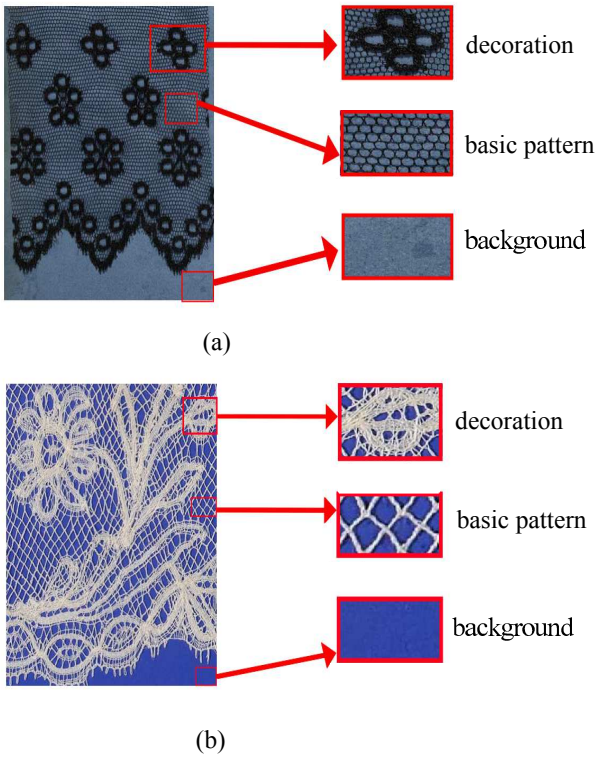


Figure B.1: Examples of lace images in collection with geometric decorations (a) and flower (b).

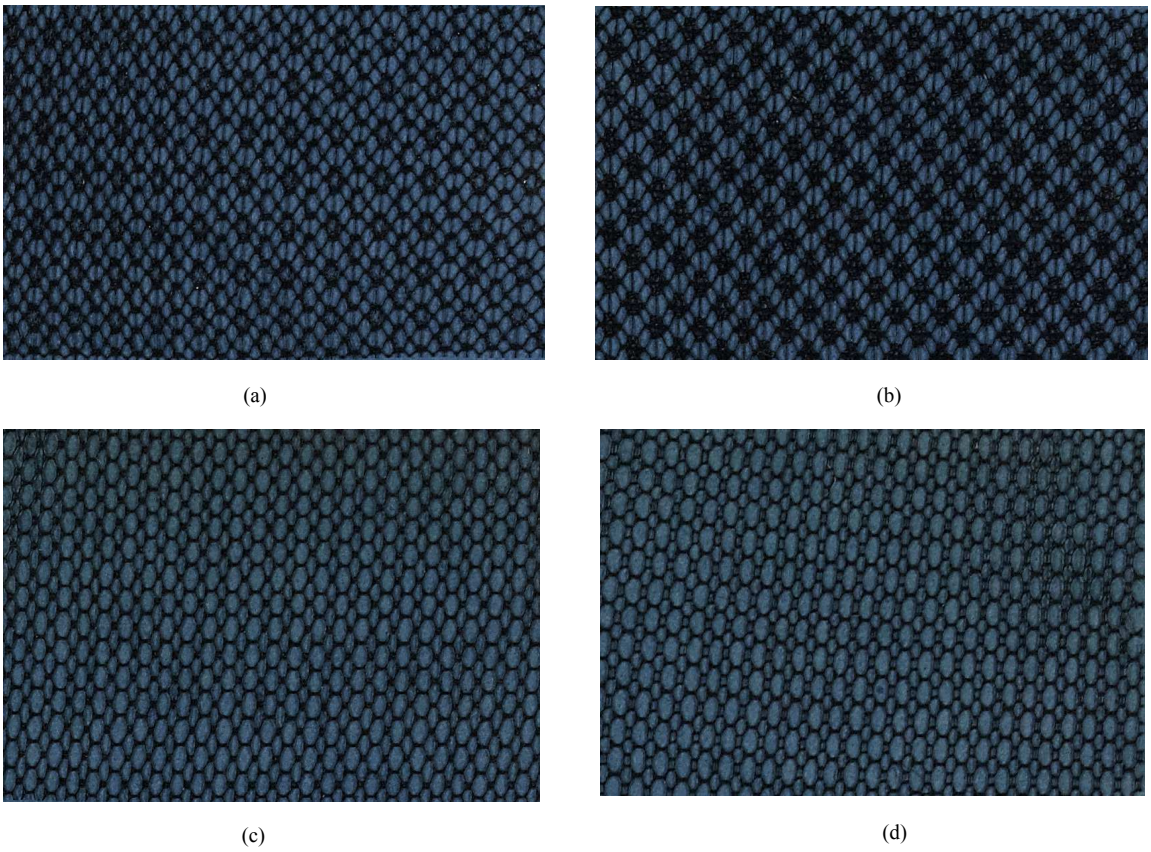
the parameters of LBP descriptors in order to improve texture representation in the lace image.

Preprocessing step and feature extraction

The specificity of lace samples of museum registers is that the scanned images contain paper background. In order to characterize the lace texture, it is necessary to segment the images to remove background and consequently only keep the material. The well-known Otsu's method is then used for this purpose [266]. Features are then extracted from those processed images in order to analyze the different textures.

Preprocessing step

Note that before the image processing step, color images are transformed into gray level images. Otsu's segmentation method assumes that the image contains two groups of pixels following a bi-modal histogram. It then calculates the optimum threshold separating the two groups so that their combined spread (within-class variance) is minimal. Figure B.3 shows lace image before processing (a) and after processing (b). Figure B.3 (d) shows that background mode has been removed from the histogram B.3 (c).



(a) (b)

(c) (d)

Figure B.2: Similarity between basic pattern of laces (a)-(b) and (c)-(d).

Feature extraction

In order to represent the texture of lace in the images, we propose to use LBP descriptors. The definition of the original LBP operator has then been generalized to explore intensity values of points on a circular neighborhoods. Thus, $\text{LBP}_{\mathcal{P},\mathcal{R}}$ is able to take any radius $\mathcal{R}$ and $\mathcal{P}$ neighbors around the central pixel, using a circular neighborhood. For an input image, texture statistics are obtained from LBP codes of all pixels represented by a histogram H. Histogram bins are denoted by $H(q), n = 0,1,...,Q-1$, where $Q = 2^{\mathcal{P}}$. In order to extract texture features that are invariant to rotation, Discrete Fourier Transform (DFT) V is applied on the *LBP* histogram:

$$V(k) = \sum_{q=0}^{Q-1} H(q) \times exp^{-i2\pi kq/Q}, \quad 0 \leq k \leq Q-1 \tag{B.1}$$

The extracted features denoted *LBP-HF* are obtained by the magnitude spectrum and given by the equation $HF(k) = \sqrt{V(k) \times \overline{V(k)}}$ where, $\overline{V(k)}$ is the complex conjugate of $V(k)$. $HF = [HF(0),...,HF(Q-1)]^T$ is the rotation invariant feature vector which will be used for texture analysis.

It is interesting to note that LBP-HF features are generated by the $\text{LBP}_{\mathcal{P},\mathcal{R}}$ descriptors and their performance depends on parameters $\mathcal{R}$ and $\mathcal{P}$ which have to be appropriately adjusted.

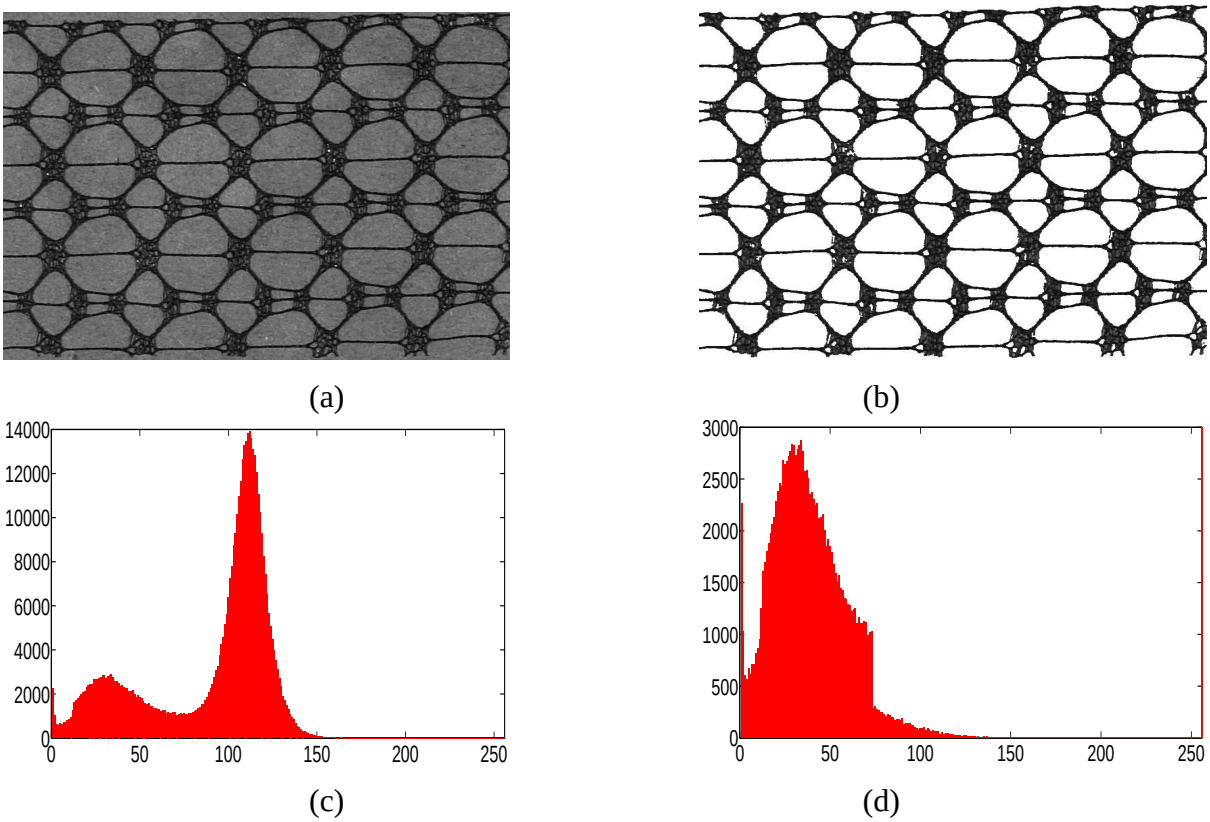


Figure B.3: Example of laces tissues before and after the image processing (a)-(b), and their histograms (c)-(d).

Figure B.4 illustrates $LBP_{8,6}$ image (a) and its histogram (b).

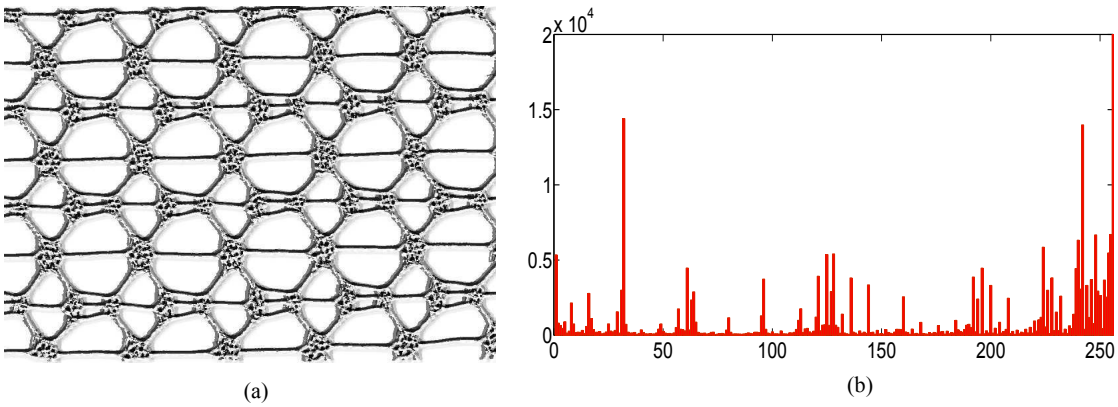


Figure B.4: $LBP_{8,6}$ image (a) and its histogram (b).

Lace images classification

In order to assess the relevance of the extracted features, we propose to measure the classification performances of lace images obtained with these features using 1-NN classifier.

Experimental setup

We consider a small labeled database which consists of 492 images belonging to 41 classes of basic pattern laces (see figure B.2). Each class is composed of 12 images whose size is (150×150) pixels. The considered images are firstly processed in order to eliminate the paper background. The LBP-HF features are then extracted from the considered images.

In the supervised context, we need a database of K classes with a learning set and a testing set. We considered the 1-NN algorithm due to its performance and simplicity. The image dataset is divided into training and testing sets. For each one, we randomly select 6 images for training and 6 ones for testing per class. This process is repeated 20 times in order to evaluate classification performances using the cross-validation technique. The classification performance rate is calculated by comparing the predicted class labels obtained after application of classification algorithm with the true class labels.

Classification results

We propose to find the appropriate value of $\mathcal{R}$ and $\mathcal{P}$ which condition the performance of LBP-HF features. The classification performances of the 1-NN algorithm are computed with different values of $\mathcal{R} \in \{1, 2, 3..., 14\}$ and $\mathcal{P} \in \{4, 8, 12, 14, 16, 20\}$ applied on images after processing. Figure B.5 displays accuracy for these different values.

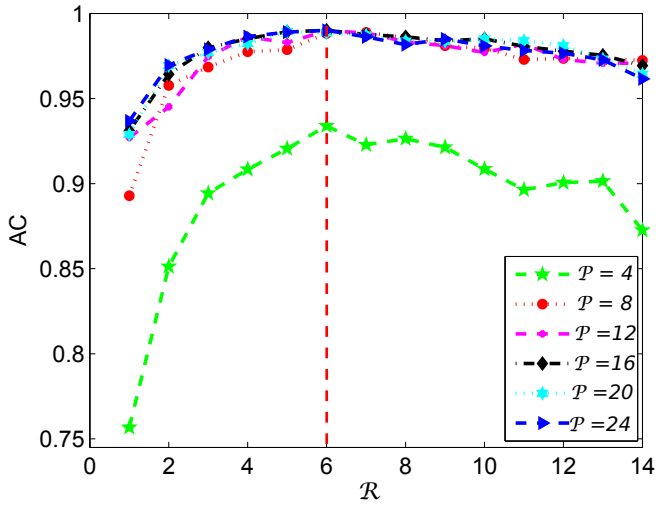


Figure B.5: *AC* vs. *R* for different values of *P* after image processing step in supervised context.

From this figure, we can see that the highest accuracy is obtained when $\mathcal{R} = 6$ for all values of $\mathcal{P}$. Table B.1 summarizes the performance of LBP-HF$_{\mathcal{P},6}$. We can notice that their value change slightly when the values of $\mathcal{P}$ are higher than 4, this value being the spatial resolution of a lace thread. So, we set $\mathcal{R} = 6$ and $\mathcal{P} = 8$ hereafter.

Based on these adjusted values of $\mathcal{R}$ and $\mathcal{P}$, we compare the classification performances of 1-NN algorithm applied on the database before and after image processing (Table B.2). This table confirms that the elimination of the paper background significantly improves the classification performances.

Table B.1: Classification performances after image processing step for different numbers of neighbors $\mathcal{P}$ with $\mathcal{R} = 6$.

LBP-HF$_{\mathcal{P},6}$	Accuracy
$\mathcal{P} = 4$	93.4
$\mathcal{P} = 8$	98.8
$\mathcal{P} = 12$	98.9
$\mathcal{P} = 16$	**99.0**
$\mathcal{P} = 20$	98.9
$\mathcal{P} = 24$	**99.0**

Table B.2: Classification performances before and after image processing step with LBP-HF$_{8,6}$.

Feature	Before processing	After processing
LBP-HF$_{8,6}$	95.2	**98.8**

In addition the classification performances obtained thanks to this tuning are clearly higher (98.8) than those reached in [265] (92.5).

Appendix C

Supplementary results

Table C.1, C.2 and C.3 present the classification obtained of LBP histogram selection approaches by different histogram score in a single color space and in multi space on Outex-TC-00013, USPTex and STex respectively. The first column indicates the color space used to code image. The second column represents the results when no selection is applied. The third column is divided into four sub-columns corresponding to the histogram score used.

Table C.1: The comparison of classification obtained of LBP histogram selection in a single color space and in multi space without and with histogram selection approaches on the OuTex-TC-00013 database. The value in boxes represent the best rates obtained with each color space and the boldface indicates the best rate obtained of each approach.

Color spaces	Without selection		Histogram selection							
			ICS		Simba-2		ASL		SpASL	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
RGB	92.9	9	92.9	9	92.9	6	93.2	8	93.4	8
rgb	87.1	9	87.1	9	87.5	4	87.4	8	87.4	6
$I_1I_2I_3$	88.5	9	89.0	8	88.5	9	88.7	8	88.5	9
HSV	90.4	9	91.0	3	90.4	9	91.3	7	91.3	7
$b_w r_g b_y$	89.6	9	89.9	9	89.9	9	91.8	8	91.8	8
HLS	92.4	9	92.4	9	92.4	9	93.4	6	93.4	6
I-HLS	89.7	9	89.7	9	89.7	9	89.7	9	90.3	7
HSI	92.9	9	92.9	9	92.9	9	93.1	8	93.1	8
YC_bC_r	89.6	9	89.6	9	89.6	9	90.6	8	90.6	8
Average in single space	90.8 ± 2.0	9	91.0 ± 2.1	8	90.4 ± 1.9	8	91.0 ± 2.1	7	91.1 ± 2.1	7
Multi spaces	**95.0**	81	**95.3**	59	**95.0**	79	**95.3**	62	**95.6**	62

Table C.2: The comparison of classification obtained of LBP histogram selection in a single color space and in multi space without and with histogram selection approaches on the USPTex database. The value in boxes represent the best rates obtained with each color space and the boldface indicates the best rate obtained of each approach.

Color spaces	Without selection		Histogram selection							
			ICS		Simba-2		ASL		SpASL	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
RGB	93.0	9	93.3	5	93.1	8	93.6	6	93.6	6
rgb	82.5	9	89.5	3	82.5	9	89.5	3	89.5	3
$I_1I_2I_3$	84.6	9	94.1	3	84.6	9	94.1	3	94.1	3
HSV	88.1	9	92.2	7	88.1	9	94.3	3	94.3	3
$b_w r_g b_y$	85.1	9	94.9	3	85.1	9	94.9	3	94.9	3
HLS	87.2	9	91.6	7	87.2	9	94.4	4	94.8	3
I-HLS	88.0	9	92.5	7	88.0	9	95.0	4	95.1	3
HSI	88.1	9	92.6	7	88.1	9	94.9	3	94.9	3
YC_bC_r	85.3	9	95.7	3	85.3	9	95.7	3	95.7	3
Average in single space	89.6 ± 3.0	9	92.9 ± 1.8	5	86.9 ± 3.0	9	94.0 ± 1.8	4	94.1 ± 1.8	3
Multi spaces	**93.7**	81	**93.7**	40	**93.8**	80	**97.6**	41	**97.4**	31

Table C.3: The comparison of classification obtained of LBP histogram selection in a single color space and in multi spaces without and with histogram selection approaches on the STex database. The value in boxes represent the best rates obtained with each color space and the boldface indicates the best rate obtained of each approach.

Color spaces	Without selection		Histogram selection							
			ICS		Simba-2		ASL		SpASL	
	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$	Rate	$\hat{\delta}$
RGB	91.9	9	91.9	9	93.4	5	93.2	5	93.2	5
rgb	85.2	9	85.2	6	85.6	8	87.8	5	87.8	5
$I_1I_2I_3$	82.8	9	89.6	4	85.9	6	91.3	3	91.3	3
HSV	88.5	9	90.0	7	91.8	4	93.9	3	93.9	3
$b_w r_g b_y$	81.5	9	90.0	4	84.5	7	92.7	3	92.7	3
HLS	89.6	9	90.6	7	92.2	5	93.2	3	93.2	3
I-HLS	90.6	9	91.3	8	94.2	4	94.0	3	94.0	3
HSI	89.4	9	90.6	7	93.6	3	93.6	3	93.6	3
YC_bC_r	82.2	9	89.5	4	88.6	5	92.2	3	92.2	3
Average in single space	89.6 ± 3.0	9	90.5 ± 1.9	6	90.0 ± 3.9	5	92.4 ± 1.9	4	92.4 ± 1.9	4
Multi spaces	**93.3**	81	**94.1**	47	**95.8**	31	**96.1**	38	**96.8**	32

Bibliography

[1] M. Mirmehdi, X. Xie, and J. Suri. *Handbook of texture analysis*. Imperial College Press, London, UK, UK, 2009.

[2] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[3] N. Asada and T. Matsuyama. Color image analysis by varying camera aperture. In *Proceedings of the 11th International Conference on Pattern Recognitio, Computer Vision and Applications*, pages 466–469, August 1992.

[4] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[5] F. Bianconi, R. Harvey, P. Southam, and A. Fernández. Theoretical and experimental comparison of different approaches for color texture classification. *Journal of Electronic Imaging*, 20(4):043006–043006, 2011.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[7] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781, 2011.

[8] A. Porebski, N. Vandenbroucke, and D. Hamad. LBP histogram selection for supervised color texture classification. In *Proceedings of the 20th IEEE International Conference on Image Processing*, pages 3239–3243. IEEE, September 2013.

[9] M. Kalakech, A. Porebski, N. Vandenbroucke, and D. Hamad. A new LBP histogram selection score for color texture classification. In *Proceedings of the 5th IEEE International Conference on Image Processing Theory, Tools and Applications*, pages 242–247, November 2015.

[10] Yimo Guo, Guoying Zhao, Matti Pietikäinen, and Zhengguang Xu. Descriptor learning based on fisher separation criterion for texture classification. In *Proceedings of the 10th Asian Conference on Computer Vision*, pages 185–198. Springer, 2010.

[11] A. Drimbarean and P. F. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161–1167, 2001.

[12] C. Palm and T. M. Lehmann. Classification of color textures by Gabor filtering. *Machine Graphics & Vision International Journal*, 11(2/3):195–219, September 2002.

[13] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, August 2004.

[14] S. Banerji, A. Verma, and C. Liu. LBP and color descriptors for image classification. In *Cross Disciplinary Biometric Systems*, pages 205–225. Springer, 2012.

[15] R. Khan, J. van de Weijer, F. S. Khan, D. Muselet, C. Ducottet, and C. Barat. Discriminative color descriptors. In *Proceedings of 23th IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, June 2013.

[16] F. S. Khan, J. Van de Weijer, S. Ali, and M. Felsberg. Evaluating the impact of color on texture recognition. In *International Conference on Computer Analysis of Images and Patterns*, pages 154–162. Springer, 2013.

[17] G. Wyszecki and W. S. Stiles. *Color Science: concepts and methods, quantitative data and formulae*. John Wiley & Sons, Inc., New York, second edition, 1982.

[18] N. Vandenbroucke, L. Busin, and L. Macaire. Unsupervised color-image segmentation by multicolor space iterative pixel classification. *Journal of Electronic Imaging*, 24(2):023032–023032, 2015.

[19] Internationale Beleuchtungskommission, editor. *Colorimetry*. Number 15 in Technical report / CIE. CIE, Central Bureau, Vienna, 3. ed edition, 2004. OCLC: 255699746.

[20] D. T. Berry. Colour recognition using spectral signatures. *Pattern Recognition Letters*, 6(1):69–75, 1987.

[21] C. Garbay, G. Brugal, and C. Choquet. Application of colored image analysis to bone marrow cell recognition. *Analytical and quantitative cytology*, 3(4):272–280, December 1981.

[22] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[23] Y. Ohta, T. Kanade, and T. Sakai. Color Information for Region Segmentation. *Computer Graphics and Image Processing*, 13(3):222 – 241, July 1980.

[24] P. Lambert and T. Carron. Symbolic fusion of luminance-hue-chroma features for region segmentation. *Pattern recognition*, 32(11):1857–1872, 1999.

[25] T. Y. Shih. The reversibility of six geometric color spaces. *Photogrammmetric Engineering & Remote Sensing*, 61(10):1223–1232, October 1995.

[26] A. Hanbury and J. Serra. A 3d-polar coordinate colour representation suitable for image analysis. In *Proceedings of the 13th Scandinavian conference on Image analysis*, pages 804–811. Springer-Verlag, 2002.

[27] M. Tuceryan, Anil K. Jain, and others. Texture analysis. *Handbook of pattern recognition and computer vision*, 2:207–248, 1993.

[28] S. W. Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5(2):190–202, 1976.

[29] J. Sklansky. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(4):237–247, 1978.

[30] S. Livens, P. Scheunders, G. Wouwer, and D. Van Dyck. Wavelets for texture analysis, an overview. In *Proceedings of the 6th International Conference on Image Processing and Its Applications*, volume 2, pages 581–585. IET, July 1997.

[31] R. M. Haralick and L. G. Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992.

[32] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[33] F. Bianconi and A. Fernández. An appendix to "Texture databases – A comprehensive survey". *Pattern Recognition Letters*, 45:33–38, August 2014.

[34] C. Cusano, P. Napoletano, and R. Schettini. Combining local binary patterns and local color contrast for texture classification under varying illumination. *Journal of the Optical Society of America A*, 31(7):1453, July 2014.

[35] H. Permuter, J. Francos, and I. Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[36] C. Palm. Color texture classification by integrative Co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, May 2004.

[37] O. J. Hernandez, J. Cook, M. Griffin, C. De Rama, and M. McGovern. Classification of color textures with random field models and neural networks. *Journal of Computer Science & Technology*, 5, 2005.

[38] C. Vertan, M. Ciuc, V. Buzuloiu, and C. Fernandez-Maloigne. Compact color-texture run-length description for ornamental stones recognition and indexing. *Machine Vision Applications in Industrial Inspection X*, 2002.

[39] T. Mäenpää, J. Viertola, and M. Pietikäinen. Optimising colour and texture features for real-time visual inspection. *Pattern Analysis & Applications*, 6(3):169–175, 2003.

[40] F. López, J. M. Valiente, J. M. Prats, and A. Ferrer. Performance evaluation of soft color texture descriptors for surface grading using experimental design and logistic regression. *Pattern Recognition*, 41(5):1744–1755, 2008.

[41] V. Arvis, C. Debain, M. Berducat, and A. Benassi. Generalization of the cooccurrence matrix for colour images: application to colour texture classification. *Image Analysis & Stereology*, 23(1):63–72, 2004.

[42] W. Polzleitner and G. Schwingshakl. Real-time color-based texture analysis for sophisticated defect detection on wooden surfaces. In *Proceedings of Intelligent Robotics and Computer Vision XXII : Algorithms, Techniques and Active Vision*, volume 5604, pages 54–69, October 2004.

[43] J. Martinez-Alajarin, J. D. Luis-Delgado, and L. M. Tomas-Balibrea. Automatic system for quality-based classification of marble textures. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(4):488–497, November 2005.

[44] X. Xie and M. Mirmehdi. TEXEMS: Texture Exemplars for Defect Detection on random textured surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1454–1464, August 2007.

[45] M. A. Akhloufi, X. Maldague, and Wael B. Larbi. A new color-texture approach for industrial products inspection. *Journal of Multimedia*, 3(3), July 2008.

[46] A. Porebski, N. Vandenbroucke, and L. Macaire. Haralick feature extraction from LBP images for color texture classification. In *Proceedings of IEEE International Conference on Image Processing Theory, Tools and Applications*, pages 1–8, 2008.

[47] A. Ledoux, O. Losson, and L. Macaire. Color local binary patterns: compact descriptors for texture classification. *Journal of Electronic Imaging*, 25(6):061404, 2016.

[48] K. Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, September 2010.

[49] N. Vandenbroucke, O. Alata, C. Lecomte, A. Porebski, and Imtnan-Ul-H. Qazi. Color texture attributes. In *Digital Color Imaging*, pages 193–240. ISTE - Wiley, May 2012.

[50] C. Zheng, Da-W. Sun, and L. Zheng. A new region-primitive method for classification of colour meat image texture based on size, orientation, and contrast. *Meat Science*, 76(4):620–627, August 2007.

[51] A. Koschan. A comparative study on color edge detection. In *Proceedings of the 2nd Asian Conference on Computer Vision*, pages 574–578, 1995.

[52] A. Sinha, S. Banerji, and C. Liu. Novel color Gabor-LBP-PHOG (GLP) descriptors for object and scene image classification. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, page 58. ACM, 2012.

[53] G. Van de Wouwer, P. Scheunders, S. Livens, and D. Van Dyck. Wavelet correlation signatures for color texture characterization. *Pattern recognition*, 32(3):443–451, 1999.

[54] Q. Xu, J. Yang, and S. Ding. Color texture analysis using the wavelet-based hidden Markov model. *Pattern Recognition Letters*, 26(11):1710–1719, August 2005.

[55] P. S. Hiremath, S. Shivashankar, and J. Pujari. Wavelet based features for color texture classification with application to CBIR. *International Journal of Computer Science and Network Security*, 6(9A):124–133, 2006.

[56] M. Pietikäinen, T. Mäenpää, and J. Viertola. Color texture classification with color histograms and local binary patterns. In *Workshop on Texture Analysis in Machine Vision*, pages 109–112, 2002.

[57] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, November 1973.

[58] A. Rosenfeld. Multispectral texture. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(1):79–84, January 1982.

[59] R. Pydipati, T. F. Burks, and W. S. Lee. Identification of citrus disease using color texture features and discriminant analysis. *Computers and Electronics in Agriculture*, 52(1-2):49–59, June 2006.

[60] C. Münzenmayer, H. Volk, C. Küblbeck, K. Spinnler, and T. Wittenberg. Multispectral texture analysis using interplane sum- and difference-histograms. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 42–49, 2002.

[61] T. Mäenpää, M. Pietikainen, and J. Viertola. Separating color and pattern information for color texture discrimination. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, volume 1, pages 668–671, 2002.

[62] S. H. Lee, J. Y. Choi, Y. M. Ro, and Konstantinos N. Plataniotis. Local color vector binary patterns from multichannel face images for face recognition. *IEEE Transactions on Image Processing*, 21(4):2347–2353, April 2012.

[63] P. Vácha, M. Haindl, and T. Suk. Colour and rotation invariant textural features based on Markov random fields. *Pattern Recognition Letters*, 32(6):771–779, April 2011.

[64] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *European conference on computer vision*, pages 517–530. Springer, 2006.

[65] A. Sinha, S. Banerji, and C. Liu. New color GPHOG descriptors for object and scene image classification. *Machine Vision and Applications*, 25(2):361–375, February 2014.

[66] F. Sandid and A. Douik. Robust color texture descriptor for material recognition. *Pattern Recognition Letters*, 80:15–23, September 2016.

[67] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition*, 62:135–160, February 2017.

[68] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 168–182. Springer, 2007.

[69] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, March 2009.

[70] K. Wang, Charles-E. Bichot, Chao Z., and L. Bailin. Pixel to patch sampling structure and local neighboring intensity relationship patterns for texture classification. *IEEE Signal Processing Letters*, 20(9):853–856, September 2013.

[71] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.

[72] L. Li, L. Yunli, P. W. Fieguth, L. Songyang, and Z. Guoying. BRINT: Binary Rotation Invariant and Noise Tolerant Texture classification. *IEEE Transactions on Image Processing*, 23(7):3071–3084, July 2014.

[73] L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth. Extended local binary patterns for texture classification. *Image and Vision Computing*, 30(2):86–99, February 2012.

[74] Z. Baochang, G. Yongsheng, Z. Sanqiang, and L. Jianzhuang. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Transactions on Image Processing*, 19(2):533–544, February 2010.

[75] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikainen. Median robust extended Local Binary Pattern for texture classification. *IEEE Transactions on Image Processing*, 25(3):1368–1381, March 2016.

[76] N. S Vu, H. M. Dee, and A. Caplier. Face recognition using the POEM descriptor. *Pattern Recognition*, 45(7):2478–2488, July 2012.

[77] G. Zhao, T. Ahonen, and J. Matas. Rotation-Invariant Image and Video Description With Local Binary Pattern Features. *IEEE Transactions on Image Processing*, 21(4):1465–1477, April 2012.

[78] J. Kannala and E. Rahtu. Bsif: Binarized statistical image features. In *Proceedings of the 21th IEEE International Conference on Pattern Recognition*, pages 1363–1366, 2012.

[79] M. Pietikäinen, T. Ojala, and Z. Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[80] S. Liao and A. C. Chung. Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *Proceedings of the Asian Conference on Computer Vision*, pages 672–679. Springer, 2007.

[81] L. Nanni, A. Lumini, and S. Brahnam. Local binary patterns variants as texture descriptors for medical image analysis. *Artificial Intelligence in Medicine*, 49(2):117–125, June 2010.

[82] T. Mäenpää and M. Pietikäinen. Multi-scale binary patterns for texture analysis. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, pages 885–892. Springer Berlin Heidelberg, 2003.

[83] L. Zhang, R. Chu, S. Xiang, S. Liao, and Stan Z. Li. Face detection based on multi-block LBP representation. In *Proceedings of the 2007 International Conference on Advances in Biometrics*, pages 11–18. Springer-Verlag, 2007.

[84] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1978–1990, October 2011.

[85] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved LBP under Bayesian framework. In *Proceedings of the Third International Conference on Image and Graphics*, pages 306–309, December 2004.

[86] A. Hafiane, G. Seetharaman, and B. Zavidovique. Median binary pattern for textures classification. In *Proceedings of the 4th International Conference on Image Analysis and Recognition*, pages 387–398. Springer Berlin Heidelberg, 2007.

[87] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.

[88] S. Liao, M. W. K. Law, and A. C. S. Chung. Dominant Local Binary Patterns for texture classification. *IEEE Transactions on Image Processing*, 18(5):1107–1118, May 2009.

[89] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, volume 1, pages 786–791, October 2005.

[90] X. Wang, Tony X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 32–39, 2009.

[91] S. U. Hussain and W. Triggs. Feature sets and dimensionality reduction for visual object detection. In *Proceedings of the British Machine Vision Conference*, pages 112–1. BMVA Press, 2010.

[92] B. F. Klare and A. K. Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1410–1422, June 2013.

[93] A. Roy and S. Marcel. Haar Local Binary Pattern Feature for Fast Illumination Invariant Face Detection. In *Proceedings of the The British Machine Vision Conference*, pages 19.1–19.12. BMVA Press, 2009.

[94] T. Ahonen, J. Matas, C. He, and M. Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Proceedings of the 16th Scandinavian Conference on Image Analysis*, pages 61–70. Springer Berlin Heidelberg, 2009.

[95] V. T. Hoang, A. Porebski, N. Vandenbroucke, and D. Hamad. LBP parameter tuning for texture analysis of lace images. In *Proceedings of the Second IEEE International Conference on International Image Processing, Applications and Systems*, pages 1–6, 2016.

[96] S. Brahnam, L. C. Jain, L. Nanni, and A. Lumini, editors. *Local Binary Patterns: New variants and applications*, volume 506 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[97] U. Kandaswamy, S. A. Schuckers, and D. Adjeroh. Comparison of texture analysis schemes under nonideal conditions. *IEEE Transactions on Image Processing*, 20(8):2260–2275, August 2011.

[98] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, and R. Domínguez-Petit. Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary. *Pattern Recognition*, 46(9):2391–2407, September 2013.

[99] J. Ning, L. Zhang, D. Zhang, and C. Wu. Robust object tracking using joint color-texture histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1245–1263, 2009.

[100] G. Han and C. Zhao. A scene images classification method based on local binary patterns and nearest-neighbor classifier. In *Proceedings of the Eighth IEEE International Conference on Intelligent Systems Design and Applications*, pages 100–104, November 2008.

[101] J.Y. Choi, K. N. Plataniotis, and Yong M. Ro. Using colour local binary pattern features for face recognition. In *Proceedings of the 17th IEEE International Conference on Image Processing*, pages 4541–4544, 2010.

[102] C. Zhu, Charles-E. Bichot, and L. Chen. Image region description using orthogonal combination of local binary patterns enhanced with color information. *Pattern Recognition*, 46(7):1949–1963, July 2013.

[103] Chi-H. Chan, J. Kittler, and K. Messer. Multispectral local binary pattern histogram for component-based color face verification. In *Proceedings of the First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–7, 2007.

[104] F. Z. Chelali and A. Djeradi. CSLBP and OCLBP local descriptors for speaker identification from video sequences. In *Proceedings of IEEE International Conference on Complex Systems*, pages 1–7, 2015.

[105] A. Porebski, N. Vandenbroucke, and D. Hamad. A fast embedded selection approach for color texture classification using degraded LBP. In *Proceedings of IEEE International Conference on Image Processing Theory, Tools and Applications*, pages 254–259, 2015.

[106] N. L. Bihan and S. J. Sangwine. Quaternion principal component analysis of color images. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages I–809, 2003.

[107] R. Lan, Y. Zhou, Y. Y. Tang, and C. P. Chen. Person reidentification using quaternionic local binary pattern. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1–6, 2014.

[108] C. Chahla, H. Snoussi, F. Abdallah, and F. Dornaika. Discriminant quaternion local binary pattern embedding for person re-identification through prototype formation and color categorization. *Engineering Applications of Artificial Intelligence*, 58:27–33, February 2017.

[109] R. Lan and Y. Zhou. Quaternion-Michelson descriptor for color image classification. *IEEE Transactions on Image Processing*, 25(11):5281–5292, November 2016.

[110] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[111] Tjen-S. Lim, Wei-Y. Loh, and Yu-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 40(3):203–228, 2000.

[112] K. Fukunaga. *Introduction to statistical pattern recognition*. Computer science and scientific computing. Academic Press, Boston, 2nd ed edition, 1990.

[113] R. A. Johnson and D. W. Wichern, editors. *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[114] D. Casanova, J. B. Florindo, M. Falvo, and O. M. Bruno. Texture analysis using fractal descriptors estimated by the mutual interference of color channels. *Information Sciences*, 346-347:58–72, June 2016.

[115] J. B. Florindo and O. M. Bruno. Texture analysis by fractal descriptors over the wavelet domain using a best basis decomposition. *Physica A: Statistical Mechanics and its Applications*, 444:415–427, February 2016.

[116] N. R. da Silva, P. Van der Weeën, B. De Baets, and Odemir M. Bruno. Improved texture image classification through the use of a corrosion-inspired cellular automaton. *Neurocomputing*, 149:1560–1572, February 2015.

[117] A. R. Backes, D. Casanova, and Odemir M. Bruno. Color texture analysis based on fractal descriptors. *Pattern Recognition*, 45(5):1984–1992, May 2012.

[118] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[119] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[120] Z. Wang and Z. Xue. Multi-class support vector machine. In *Support Vector Machines Applications*, pages 23–48. Springer, 2014.

[121] Chih-W. Hsu and Chih-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[122] D. Iakovidis, D. Maroulis, and S. Karkanis. A comparative study of color-texture image features. In *Proceedings of IEEE International Workshop on Systems, Signal and Image Processing*, pages 205–209, 2005.

[123] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[124] D. W. Aha, D. Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[125] D. He and N. Cercone. Local triplet pattern for content-based image retrieval. In *International Conference Image Analysis and Recognition*, pages 229–238. Springer, 2009.

[126] Imtnan-Ul-H. Qazi, O. Alata, J. C. Burie, A. Moussa, and C. Fernandez-Maloigne. Choice of a pertinent color space for color texture characterization using parametric spectral analysis. *Pattern Recognition*, 44(1):16–31, January 2011.

[127] S. Alvarez and M. Vanrell. Texton theory revisited: A bag-of-words approach to combine textons. *Pattern Recognition*, 45(12):4312–4325, December 2012.

[128] K. Hammouche, O. Losson, and L. Macaire. Fuzzy aura matrices for texture classification. *Pattern Recognition*, 53:212–228, December 2015.

[129] A. Ahmadvand and M. Daliri. Invariant texture classification using a spatial filter bank in multi-resolution analysis. *Image and Vision Computing*, 45:1–10, January 2016.

[130] A. Ledoux and N. Richard. Color and multiscale texture features from vectorial mathematical morphology. *Signal, Image and Video Processing*, 10(3):431–438, March 2016.

[131] R. Bello-Cerezo, F. Bianconi, A. Fernández, E. González, and F. Di Maria. Experimental comparison of color spaces for material classification. *Journal of Electronic Imaging*, 25(6):061406, June 2016.

[132] E. Cernadas, M. Fernández-Delgado, E. González-Rufino, and P. Carrión. Influence of normalization and color space to color texture classification. *Pattern Recognition*, 61:120–138, January 2017.

[133] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[134] A. D. El Maliani, M. El Hassouni, Y. Berthoumieu, and D. Aboutajdine. Color texture classification method based on a statistical multi-model and geodesic distance. *Journal of Visual Communication and Image Representation*, 25(7):1717–1725, October 2014.

[135] R. A. Martínez, N. Richard, and C. Fernandez. Alternative to colour feature classification using colour contrast ocurrence matrix. In *The International Conference on Quality Control by Artificial Vision*, pages 953405–953405. International Society for Optics and Photonics, 2015.

[136] A. Porebski, N. Vandenbroucke, and L. Macaire. Iterative feature selection for color texture classification. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages III – 509–III – 512, September 2007.

[137] A. Porebski, N. Vandenbroucke, and L. Macaire. A multi color space approach for texture classification: experiments with Outex, Vistex and Barktex image databases. In *Proceedings of the International Conference on Colour in Graphics, Imaging, and Vision*, volume 2010, pages 314–319. Society for Imaging Science and Technology, 2010.

[138] A. Porebski, N. Vandenbroucke, L. Macaire, and D. Hamad. A new benchmark image test suite for evaluating colour texture classification schemes. *Multimedia Tools and Applications*, 70(1):543–556, 2014.

[139] A. Porebski, N. Vandenbroucke, and L. Macaire. Supervised texture classification: color space or texture feature selection? *Pattern Analysis and Applications*, 16(1):1–18, 2013.

[140] M. Kalakech, P. Biela, L. Macaire, and D. Hamad. Constraint scores for semi-supervised feature selection: A comparative study. *Pattern Recognition Letters*, 32(5):656–665, April 2011.

[141] Y. Guo, G. Zhao, and M. Pietikäinen. Discriminative features for texture description. *Pattern Recognition*, 45(10):3834–3843, October 2012.

[142] Jing-M. Guo, H. Prasetyo, H. Lee, and C. Yao. Image retrieval using indexed histogram of Void-and-Cluster Block Truncation Coding. *Signal Processing*, 123:143–156, June 2016.

[143] M. Paci, L. Nanni, and S. Severi. An ensemble of classifiers based on different texture descriptors for texture classification. *Journal of King Saud University - Science*, 25(3):235–244, July 2013.

[144] A. Fernández, M. X. Álvarez, and F. Bianconi. Texture description through histograms of equivalent patterns. *Journal of Mathematical Imaging and Vision*, 45(1):76–102, January 2013.

[145] J. J. M. Sá Junior and A. R. Backes. ELM based signature for texture classification. *Pattern Recognition*, 51:395–401, March 2016.

[146] F. Bianconi, R. Bello-Cerezo, P. Napoletano, and F. Di Maria. Improved opponent colour Local Binary Patterns for colour texture classification. In *Proceedings of the 6th International Workshop on Computational Color Imaging*, pages 272–281. Springer International Publishing, 2017.

[147] X. Chen, Z. Zhou, J. Zhang, Z. Liu, and Q. Huang. Local convex-and-concave pattern: An effective texture descriptor. *Information Sciences*, 363:120–139, October 2016.

[148] Y. G. Naresh and H. S. Nagendraswamy. Classification of medicinal plants: An approach using modified LBP with symbolic representation. *Neurocomputing*, 173:1789–1797, January 2016.

[149] S. Hossain and S. Serikawa. Texture databases – A comprehensive survey. *Pattern Recognition Letters*, 34(15):2007–2022, November 2013.

[150] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 1, pages 701–706 vol.1, 2002.

[151] R. Lakmann. Barktex benchmark database of color textured images. *Koblenz- Landau University*, (ftp://ftphost.uni-koblenz.de/outgoing/vision/Lakmann/ BarkTex), April 1998.

[152] C. Cusano, P. Napoletano, and R. Schettini. Illuminant invariant descriptors for color texture classification. In *Computational Color Imaging*, pages 239–249. Springer, 2013.

[153] F. Bianconi, R. Bello-Cerezo, and P. Napoletano. Improved opponent color local binary patterns: an effective local image descriptor for color texture classification. *Journal of Electronic Imaging*, 27(01):1, December 2017.

[154] J. Wang, Y. Fan, and N. Li. Combining fine texture and coarse color features for color texture classification. *Journal of Electronic Imaging*, 26(06):1, December 2017.

[155] M. W. S. Oliveira, N. R. da Silva, A. Manzanera, and O. M. Bruno. Feature extraction on local jet space for texture classification. *Physica A: Statistical Mechanics and its Applications*, 439:160–170, December 2015.

[156] W. N. Gonçalves, N. R. da Silva, L. da Fontoura Costa, and Odemir M. Bruno. Texture recognition based on diffusion in networks. *Information Sciences*, 364-365:51–71, October 2016.

[157] X. He and P. Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[158] S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

[159] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, October 2004.

[160] M. Yang, F. Wang, and P. Yang. A novel feature selection algorithm based on hypothesis-margin. *Journal of Computers*, 3(12):27–34, 2008.

[161] W. Megchelenbrik. Relief-Based feature selection in bioinformatics: detecting functional specificity residues from multiple sequence alignments. *Radboud University, Nijmegen*, 2010.

[162] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[163] Y. Sun, S. Todorovic, and S. Goodison. Local-learning-based feature selection for high-dimensional data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1610–1626, 2010.

[164] D. Dernoncourt, B. Hanczar, and J. D. Zucker. Analysis of feature selection stability on high dimension and small sample data. *Computational Statistics & Data Analysis*, 71:681–693, March 2014.

[165] G. Roffo, C. Segalin, A. Vinciarelli, V. Murino, and M. Cristani. Reading between the turns: statistical modeling for identity recognition and verification in chats. In *Proceedings of the 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 99–104, August 2013.

[166] G. Roffo, C. Giorgetta, R. Ferrario, W. Riviera, and M. Cristani. Statistical analysis of personality and identity in chats using a keylogging platform. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 224–231. ACM Press, 2014.

[167] G. Roffo and S. Melzi. Online feature selection for visual tracking. In *Proceedings of the The British Machine Vision Conference*. BMVA Press, 2016.

[168] G. Roffo, M. Cristani, L. Bazzani, H. Q. Minh, and V. Murino. Trusting skype: Learning the Way People Chat for Fast User Recognition and Verification. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 748–754, December 2013.

[169] J. Yuan and F. B. Bastani. Robust object tracking via online informative feature selection. In *Proceedings of the IEEE International Conference on Image Processing*, pages 471–475, 2014.

[170] K. Zhang, L. Zhang, and M. H. Yang. Real-Time Object Tracking Via Online Discriminative Feature Selection. *IEEE Transactions on Image Processing*, 22(12):4664–4677, December 2013.

[171] George H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of 11th International Conference on Machine Learning*, pages 121–129. Morgan Kaufmann, 1994.

[172] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

[173] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[174] K. Benabdeslem and M. Hindawi. Constrained laplacian score for semi-supervised feature selection. In *Machine Learning and Knowledge Discovery in Databases*, pages 204–218. Springer, 2011.

[175] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications. Editor: Charu Aggarwal, CRC Press In Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*, 2014.

[176] J. C. Ang, A. Mirzal, H. Haron, and H. Hamed. Supervised, unsupervised and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2015.

[177] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157. ACM, 2007.

[178] R. Liu, N. Yang, X. Ding, and L. Ma. An unsupervised feature selection algorithm: laplacian score combined with distance-based entropy measure. In *Proceedings of the Third IEEE International Symposium on Intelligent Information Technology Application*, pages 65–68, 2009.

[179] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*, 2005.

[180] D. Zhang, S. Chen, and Zhi-H. Zhou. Constraint Score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, 41(5):1440–1451, May 2008.

[181] M. Kalakech, A. Porebski, P. Biela, D. Hamad, and L. Macaire. Constraint score for semi-supervised selection of color texture features. In *Proceedings of the Third IEEE international conference on machine vision*, pages 275–279, 2010.

[182] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *Proceedings of the SIAM International Conference on Data Mining*, pages 792–801. SIAM, 2009.

[183] R. Sheikhpour, Mehdi A. Sarram, S. Gharaghani, and Mohammd Ali Z. Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, November 2016.

[184] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th International Conference on International Conference on Machine Learning*, volume 3, pages 856–863. AAAI Press, 2003.

[185] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 266–273. AUAI Press, 2011.

[186] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[187] M. Dash and H. Liu. Feature selection for clustering. In *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pages 110–121. Springer-Verlag, 2000.

[188] L. Talavera. An evaluation of filter and wrapper methods for feature selection in categorical clustering. In *Advances in Intelligent Data Analysis VI*, pages 440–451. Springer, 2005.

[189] H. Yoon, K. Yang, and C. Shahabi. Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1186–1198, 2005.

[190] D. Rodrigues, L. A. M. Pereira, R. Y. M. Nakamura, K. A. P. Costa, Xin-S. Yang, A. N. de Souza, and J. P. Papa. A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest. *Expert Systems Application*, 41(5):2250–2258, 2014.

[191] M. Monirul Kabir, M. Monirul Islam, and K. Murase. A new wrapper feature selection approach using neural network. *Neurocomputing*, 73(16-18):3273–3283, October 2010.

[192] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. A new hybrid filter–wrapper feature selection method for clustering based on ranking. *Neurocomputing*, 214:866–880, November 2016.

[193] S. Yan, D. Xu, B. Zhang, Hong-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[194] Z. Lu and Y. Peng. Exhaustive and efficient constraint propagation: a graph-based learning approach and its applications. *International Journal of Computer Vision*, 103(3):306–325, July 2013.

[195] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, December 1996.

[196] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

[197] M. Liu and D. Zhang. Sparsity score: a novel graph-preserving feature selection method. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(04):1450009, June 2014.

[198] von U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[199] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[200] C. Cortes and M. Mohri. On transductive regression. *Advances in Neural Information Processing Systems*, 19:305, 2007.

[201] J. Xu, G. Yang, H. Man, and H. He. L1 graph based on sparse coding for feature selection. In *Advances in Neural Networks*, pages 594–601. Springer, 2013.

[202] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang. Learning with l-1 -Graph for image analysis. *Image Processing, IEEE Transactions on*, 19(4):858–866, April 2010.

[203] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, December 2006.

[204] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, February 2009.

[205] S. Sun, J. Wang, Mary F.H. She, and L. Kong. Sparse representation with multi-manifold analysis for texture classification from few training images. *Image and Vision Computing*, 32(11):835–846, November 2014.

[206] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[207] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Advances in neural information processing systems*, pages 609–616, 2006.

[208] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk. The smashed filter for compressive classification and target recognition. In *Proceedings of SPIE - The International Society for Optical Engineering*, pages 64980H–64980H. International Society for Optics and Photonics, 2007.

[209] T. T. Cai and A. Zhang. Sparse Representation of a Polytope and Recovery of Sparse Signals and Low-Rank Matrices. *IEEE Transactions on Information Theory*, 60(1):122–132, January 2014.

[210] L. Qiao, S. Chen, and X. Tan. Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, 43(1):331–341, January 2010.

[211] Y. Xie, W. Zhang, Y. Qu, and Y. Zhang. Discriminative subspace learning with sparse representation view-based model for robust visual tracking. *Pattern Recognition*, 47(3):1383–1394, March 2014.

[212] M. Liu, D. Sun, and D. Zhang. Sparsity Score: A new filter feature selection method based on graph. In *Proceedings of the 21st IEEE International Conference on Pattern Recognition*, pages 959–962, 2012.

[213] J. Xu, G. Yang, Y. Yin, H. Man, and H. He. Sparse-representation-based classification with structure-preserving dimension reduction. *Cognitive Computation*, 6(3):608–621, September 2014.

[214] S. C. Shaobing and D. Donoho. Basis pursuit. In *28th Asilomar conf. Signals, Systems Computers*, 1994.

[215] F. Dornaika and A. Bosaghzadeh. Adaptive graph construction using data self-representativeness for pattern classification. *Information Sciences*, 325:118–139, December 2015.

[216] G. Doquire and M. Verleysen. A graph Laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing*, 121:5–13, December 2013.

[217] M. Liu and D. Zhang. Pairwise constraint-guided sparse learning for feature selection. *IEEE Transactions on Cybernetics*, pages 1–1, 2015.

[218] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin analysis of the LVQ algorithm. In *Advances in neural information processing systems*, pages 462–469, 2002.

[219] M. Yang and J. Song. A novel hypothesis-margin based approach for feature selection with side pairwise constraints. *Neurocomputing*, 73(16-18):2859–2872, October 2010.

[220] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Proceedings of the 9th International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[221] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection-theory and algorithms. In *Proceedings of the 21th International Conference on Machine learning*, page 43. ACM, 2004.

[222] R. S. Smith and T. Windeatt. Facial expression detection using filtered local binary pattern features with ECOC classifiers and platt scaling. In *Proceedings of the First Workshop on Applications of Pattern Analysis*, pages 111–118, 2010.

[223] O. Lahdenoja, M. Laiho, and A. Paasio. Reducing the feature vector length in local binary pattern based face recognition. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages II–914, 2005.

[224] D. Maturana, D. Mery, and A. Soto. Learning discriminative local binary patterns for face recognition. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference On*, pages 470–475. IEEE, 2011.

[225] D. Zhao, Z. Lin, and Z. Tang. Laplacian PCA and its applications. In *Proceeding of 11th IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[226] L. Nanni and A. Lumini. RegionBoost learning for 2d+3d based face recognition. *Pattern Recognition Letters*, 28(15):2063–2070, November 2007.

[227] A. Moujahid, A. Abanda, and F. Dornaika. Feature extraction using block-based Local Binary Pattern for face recognition. *Proceedings of Intelligent Robots and Computer Vision XXXIII: Algorithms and Techniques*, 2016(10):1–6, 2016.

[228] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer vision using Local Binary Patterns*, volume 40. Springer London, London, 2011.

[229] F. Cointault, D. Guerin, J. P. Guillemin, and B. Chopinet. In field Triticum aestivum ear counting using color texture image analysis. *New Zealand Journal of Crop and Horticultural Science*, 36(2):117–130, 2008.

[230] S. Chindaro, K. Sirlantzis, and F. Deravi. Texture classification system using colour space fusion. *Electronics Letters*, 41(10):589–590, 2005.

[231] L. Nanni and A. Lumini. Fusion of color spaces for ear authentication. *Pattern Recognition*, 42(9):1906–1913, September 2009.

[232] E. Aptoula and S. Lefèvre. On morphological color texture characterization. In *Proceedings of the International Symposium on Mathematical Morphology*, pages 153–164, 2007.

[233] M. Adel, D. Wolf, R. Vogrig, and R. Husson. Evaluation of colour spaces in computer vision application of wood defects detection. In *Proceedings of the IEEE International conference on Systems Man and Cybernetics Conference*, pages 499–504, 1993.

[234] F. Bianconi, A. Fernández, E. González, and S. A. Saetta. Performance analysis of colour descriptors for parquet sorting. *Expert Systems with Applications*, 40(5):1636–1644, April 2013.

[235] C. C. Brunner, A. G. Maristany, D. A. Butler, D. VanLeeuwen, and James W. Funck. An evaluation of color spaces fordetecting defects in Douglas-fir veneer. *Industrial metrology*, 2(3-4):169–184, 1992.

[236] F. Bianconi, E. González, A. Fernández, and S. A. Saetta. Automatic classification of granite tiles through colour and texture features. *Expert Systems with Applications*, 39(12):11212–11218, September 2012.

[237] F. Bianconi, R. Bello, A. Fernández, and E. González. On comparing colour spaces from a performance perspective: Application to automated classification of polished natural stones. In *Proceedings of International Workshops New Trends in Image Analysis and Processing*, pages 71–78. Springer International Publishing, 2015.

[238] S. Chindaro, K. Sirlantzis, and M. C. Fairhurst. ICA-based multi-colour space texture classification system. *Electronics Letters*, 42(21):1208–1209, 2006.

[239] C. Charrier, G. Lebrun, and O. Lezoray. Evidential segmentation of microscopic color images with pixel classification posterior probabilities. *Journal of Multimedia*, 2(3), 2007.

[240] M. Mignotte. A de-texturing and spatially constrained K-means approach for image segmentation. *Pattern Recognition Letters*, 32(2):359–367, January 2011.

[241] L. Busin, N. Vandenbroucke, and L. Macaire. Color spaces and image segmentation. In *Advances in Imaging and Electron Physics*, volume 151, pages 65–168. Elsevier, 2009.

[242] H. Stern and B. Efros. Adaptive color space switching for tracking under varying illumination. *Image and Vision Computing*, 23(3):353–364, March 2005.

[243] F. Laguzet, M. Gouiffès, L. Lacassagne, and D. Etiemble. Automatic color space switching for robust tracking. In *Proceedings of the IEEE International Conference on Signal and Image Processing Applications*, pages 295–300, 2011.

[244] F. Laguzet, A. Romero, M. Gouiffès, L. Lacassagne, and D. Etiemble. Color tracking with contextual switching: real-time implementation on CPU. *Journal of Real-Time Image Processing*, 10(2):403–422, June 2015.

[245] N. Vandenbroucke, L. Macaire, and J. G. Postaire. Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. *Computer Vision and Image Understanding*, 90(2):190–216, May 2003.

[246] M. Mignotte. A label field fusion bayesian model and its penalized maximum rand estimator for image segmentation. *IEEE Transactions on Image Processing*, 19(6):1610–1624, June 2010.

[247] S. Banerji, A. Verma, and C. Liu. Novel color LBP descriptors for scene and image texture classification. In *Proceedings of the 15th International Conference on Image Processing, Computer Vision, and Pattern Recognition*, pages 537–543, 2011.

[248] S. Banerji, A. Sinha, and C. Liu. New image descriptors based on color, texture, shape, and wavelets for object and scene image classification. *Neurocomputing*, 117:173–185, October 2013.

[249] S. Banerji, A. Sinha, and C. Liu. Haarhog: Improving the hog descriptor for image classification. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 4276–4281, 2013.

[250] R. Khan, A. Hanbury, J. Stöttinger, F. A. Khan, A. U. Khattak, and Amjad Ali. Multiple color space channel fusion for skin detection. *Multimedia Tools and Applications*, 72(2):1709–1730, September 2014.

[251] L. Busin, N. Vandenbroucke, L. Macaire, and J. G. Postaire. Color space selection for unsupervised color image segmentation by histogram multi-thresholding. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 203–206, 2004.

[252] A. Gupta and A. Chaudhary. Robust skin segmentation using color space switching. *Pattern Recognition and Image Analysis*, 26(1):61–68, January 2016.

[253] C. Benedek and T. Szirányi. Study on color space selection for detecting cast shadows in video surveillance. *International Journal of Imaging Systems and Technology*, 17(3):190–201, 2007.

[254] Y. Shan, F. Yang, and R. Wang. Color space selection for moving shadow elimination. In *Proceedings of the Fourth IEEE International Conference on Image and Graph*, pages 496–501, August 2007.

[255] N. Razmjooy, B. S. Mousavi, M. Khalilpour, and H. Hosseini. Automatic selection and fusion of color spaces for image thresholding. *Signal, Image and Video Processing*, 8(4):603–614, May 2014.

[256] N. Vandenbroucke, L. Macaire, and J.-G. Postaire. Color pixels classification in an hybrid color space. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 176–180, 1998.

[257] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, February 1997.

[258] V. T. Hoang, A. Porebski, N. Vandenbroucke, and D. Hamad. LBP histogram selection based on sparse representation for color texture classification. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 4, pages 476–483, 2017.

[259] Z. Guo, D. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, June 2010.

[260] R. Mehta and K. Egiazarian. Dominant Rotated Local Binary Patterns (DRLBP) for texture classification. *Pattern Recognition Letters*, 71:16–22, February 2016.

[261] P. Liu, Jing-M. Guo, K. Chamnongthai, and H. Prasetyo. Fusion of color histogram and LBP-based features for texture image retrieval and classification. *Information Sciences*, January 2017.

[262] E. Achtert, S. Goldhofer, H. P. Kriegel, E. Schubert, and A. Zimek. Evaluation of clusterings – metrics and visual support. In *Proceedings of the 28th IEEE International Conference on Data Engineering*, pages 1285–1288, April 2012.

[263] Imtnan-Ul-H. Qazi, O. Alata, and Z. Kato. Parametric stochastic modeling for color image segmentation and texture characterization. In *Advanced Color Image Processing and Analysis*, pages 279–326. Springer, January 2013.

[264] J. B. Florindo, G. Landini, and O. M. Bruno. Three-dimensional connectivity index for texture recognition. *Pattern Recognition Letters*, 84:239–244, December 2016.

[265] W. Ben Soltana, A. Porebski, N. Vandenbroucke, A. Ahmad, and D. Hamad. Texture analysis of lace images using histogram and local binary patterns under rotation variation. In *Proceedings of the First IEEE International Image Processing, Applications and Systems Conference*, pages 1–5, 2014.

[266] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

Publications

During this thesis, 2 papers have been published in international conferences. In addition, 1 paper has been submitted to international journals for review and 1 paper is preparing to submit to international conference.

Accepted Papers in International Conferences:

1. Hoang, V. T., Porebski, A., Vandenbroucke, N. & Hamad, D. LBP parameter tuning for texture analysis of lace images. in 2nd IEEE International Conference on Image Processing, Applications and Systems (IPAS'16). (2016) doi:10.1109/IPAS.2016.7880063

2. Hoang, V. T., Porebski, A., Vandenbroucke, N. & Hamad, D. LBP Histogram Selection based on Sparse Representation for Color Texture Classification. in 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'17). (2017). doi:10.5220/0006128- 204760483

Accepted Paper in International Journal:

1. Porebski, A., Hoang, V. T., Vandenbroucke, N. & Hamad, D Multi color space LBP-based feature selection for texture classification. Journal of Electronic Imaging 27(1), 011010 (2018). doi:10.1117/1.JEI.27.1.011010

List of Tables

1.1 Characteristics of OuTex-TC-00013 dataset 45
1.2 Characteristics of BarkTex dataset 46
1.3 Characteristics of USPTex dataset 46
1.4 Characteristics of STex dataset 47

3.1 Studies about color space comparison for classification. 84
3.1 Studies about color space comparison for classification continued... 85
3.1 Studies about color space comparison for classification continued... 86
3.2 Studies about multi color space approach. 89
3.2 Studies about multi color space approach continued... 90
3.2 Studies about multi color space approach continued... 91
3.3 The classification results of New BarkTex without selecion 94
3.4 The performance of histogram selectionn on New BarkTex. 98
3.5 LBP bin selection approaches in a single and multiple color spaces on BarkTex 105
3.6 The classification results of the combination methods 109

4.1 Summary of image databases used in experiments. 112
4.2 Summary of the proposed methods used in experiments. 112
4.2 Summary of the proposed methods used in experiments. 113
4.2 Summary of the proposed methods used in experiments. 114
4.3 The classification obtained by SpASL-score with four different distances. . . . 116
4.4 The classification results on four texture database 117
4.5 The classification of the proposed approaches on the New BarkTex database . . 118
4.6 Comparison between the well-classified of New Bartex 120
4.7 Comparison between the well-classified of STex 121
4.9 Comparison between the well-classified of OuTex-13 122
4.9 Comparison between the well-classified image rates reached on OuTex. 123
4.9 Comparison between the well-classified image rates reached on OuTex. 124

4.9 Comparison between the well-classified image rates reached on OuTex. 125
4.8 Comparison between the well-classified of USPTex 128
4.10 Processing times of the learning stage for 816 images 129
4.11 Processing times for classifying 816 images 129
4.12 The summary of selected approaches. 130

A.1 A summary of studies on color texture classification. 138
A.1 A summary of studies on color texture classification continued... 139
A.1 A summary of studies on color texture classification continued... 140
A.1 A summary of studies on color texture classification continued... 141
A.1 A summary of studies on color texture classification continued... 142
A.1 A summary of studies on color texture classification continued... 143
A.1 A summary of studies on color texture classification continued... 144

B.1 Classification obtained after image processing for $\mathcal{R} = 6$ and different $\mathcal{P}$. 150
B.2 Classification performances with $LBP - HF_(6,8)$. 150

C.1 The performance of histogram selectionn on OuTex-TC-00013. 152
C.2 The performance of histogram selectionn on USPTex. 153
C.3 The performance of histogram selectionn on STex. 154

List of Figures

1.1 An example of a color image with its three components image in the *RGB*. . . . 13
1.2 An example of an image coded in different color spaces. 14
1.3 Color distributions of the image of leaves coded in four color spaces. 15
1.4 A variety of textures 16
1.5 Different observation conditions of a texture 18
1.6 Color texture by extracting the features from the luminance and color distribution 19
1.7 Representation of color texture by extracting features within each component . 20
1.8 Representation of color texture from the within and the between-components . 21
1.9 An example of the original LBP computation. 24
1.10 Circular neighborhoods of the center pixel with different neighbors 25
1.11 An example of non-uniform and uniform patterns LBP 26
1.12 An example of rotation LBP patterns 27
1.13 Illustration of the within component color LBP 30
1.14 An example of the Opponent Color LBP operator 31
1.15 Exemple of two different textures 33
1.16 A basic scheme of texture classification. 33
1.17 A general framework of a supervised classification. 34
1.18 A general framework of an unsupervised classification. 34
1.19 A general framework of a semi-supervised classification. 34
1.20 A scheme of texture classification with dimensionality reduction step. 35
1.21 An illustration of LDA 36
1.22 Illustration of different hyperplanes 37
1.23 Example of K-NN classification 38
1.24 The OuTex-TC-00013 dataset includes 68 different texture classes. 40
1.25 Selected textures among 191 classes from USPTex database 41
1.26 Selected textures among 476 classes from STex database 42
1.27 Examples of images of the BarkTex database. 43

1.28 Example of New BarkTex texture test suite 44

2.1 An example of relevant, noisy and redundant features 51
2.2 The different steps of feature selection 51
2.3 Categorization of feature selection (a) context and (b) evaluation strategy. . . . 53
2.4 A general framework of supervised feature selection 54
2.5 A general framework of unsupervised feature selection 54
2.6 A general framework of semi-supervised feature selection 55
2.7 An illustration of filter method 56
2.8 An illustration of wrapper method 56
2.9 An illustration of hybrid method 57
2.10 Illustration of the *nearhit* and *nearmiss* concepts. 66

3.1 An illustration of EOCLBP feature extraction from 9 color spaces. 93
3.2 An illustration of the multi color space histogram selection 96
3.3 Multi color space bin selection 99
3.4 An illustration of bin selection for strategy B 102
3.5 An illustration of different bin selection strategies 104
3.6 LBP bin ranking dependent on the ranked histogram approach 107
3.7 An illustration of bin selection for strategy C and D 108

B.1 Examples of lace images in collection 146
B.2 Similarity between basic pattern of laces 147
B.3 Example of laces tissues before and after the image processing. 148
B.4 $LBP_{8,6}$ image (a) and its histogram (b). 148
B.5 *AC* vs. *R* for different values of *P* after image processing step 149

Abstract

Texture analysis has been extensively studied and a wide variety of description approaches have been proposed. Among of them, Local Binary Pattern (LBP) takes an essential part of most of color image analysis and pattern recognition applications. Usually, devices acquire images and code them in the *RGB* color space. However, there are many color spaces for texture classification, each one having specific proprieties. In order to avoid the difficulty of choosing a relevant space, the multi color space strategy allows using the properties of several spaces simultaneously. However, this strategy leads to increase the number of features extracted from LBP applied to color images. This work is focused on the dimensionality reduction of LBP-based features by feature selection methods. In this framework, we consider the LBP histogram and bin selection approaches for supervised texture classification. Extensive experiments are conducted on several benchmark color texture databases. They demonstrate that the proposed approaches can improve the state-of-the-art results.

Keywords: Texture classification, color spaces, LBP operator, feature selection, histogram selection, bin selection, supervised learning.

Résumé

L'analyse de texture a été largement étudiée dans la littérature et une grande variété de descripteurs de texture ont été proposés. Parmi ceux-ci, les motifs binaires locaux (LBP) occupent une part importante dans la plupart des applications d'imagerie couleur ou de reconnaissance de formes et sont particulièrement exploités dans les problèmes d'analyse de texture. Généralement, les images couleur acquises sont représentées dans l'espace colorimétrique *RGB*. Cependant, il existe de nombreux espaces couleur pour la classification des textures, chacun ayant des propriétés spécifiques qui impactent les performances. Afin d'éviter la difficulté de choisir un espace pertinent, la stratégie multi-espace couleur permet d'utiliser simultanément les propriétés de plusieurs espaces. Toutefois, cette stratégie conduit à augmenter le nombre d'attributs, notamment lorsqu'ils sont extraits de LBP appliqués aux images couleur. Ce travail de recherche est donc axé sur la réduction de la dimension de l'espace d'attributs générés à partir de motifs binaires locaux par des méthodes de sélection d'attributs. Dans ce cadre, nous considérons l'histogramme des LBP pour la représentation des textures couleur et proposons des approches conjointe de sélection de bins et d'histogrammes multi-espace pour la classification supervisée de textures. Les nombreuses expériences menées sur des bases de référence de texture couleur, démontrent que les approches proposées peuvent améliorer les performances en classification comparées à l'état de l'art.

Mots-clés: Classification de textures, espaces couleur, opérateur LBP, sélection d'attributs, sélection d'histogramme, sélection de bins, apprentissage supervisé.