Pattern Recognition 61 (2017) 120–138

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Influence of normalization and color space to color texture classification

E. Cernadas [a,*], M. Fernández-Delgado [a], E. González-Rufino [b], P. Carrión [b]

[a] Centro Singular de Investigación en Tecnoloxías da Información da USC (CiTIUS), Universidade de Santiago de Compostela,
Rúa Xenaro de la Fuente Domínguez, 15782 Santiago de Compostela, Spain
[b] Department of Computer Science, Campus Universitario As Lagoas, University of Vigo, Ourense, Spain

ARTICLE INFO

Article history:
Received 17 September 2014
Received in revised form
29 June 2016
Accepted 3 July 2016
Available online 14 July 2016

Keywords:
Color texture classification
Color space
Image normalization
Wavelet transforms
Gabor transforms
Local binary patterns
Learning methods

ABSTRACT

Color texture classification has recently attracted significant attention due to its multiple applications. The color texture images depend on the texture surface and its albedo, the illumination, the camera and its viewing position. A key problem to get an acceptable performance is the ambient illumination, which can vary the perceived structures in the surface. Given a color texture classification problem, it would be desirable to know which is the best approach to solve the problem making the minimal assumptions about the illumination conditions. The present work does an exhaustive evaluation of the state-of-the-art color texture classification methods, considering 5 different color spaces, 12 normalization methods to achieve illumination invariances, 19 texture feature vectors and 23 pure color feature vectors. Our experiments allow to conclude that parallel approaches are better than integrative approaches for color texture classification achieving the first positions in the Friedman ranking. Multiresolution Local Binary Patterns (MLBP) are the best intensity texture features, followed by wavelet and Gabor filters combined with luminance–chrominance color spaces (Lab and Lab2000$_{HL}$), and for pure color classification the best are First Order Statistics (FOS) calculated in RGB color space. For intensity texture features, the learning methods work better on the four smallest datasets, although they could not be tested in other four bigger datasets due to its huge computational cost, nor with color texture classification. Normalization and color spaces slightly increase the average accuracy of color texture classification, although the differences achieved using normalization are not statistically significant in a paired *T*-Test. *Lab2000*$_{HL}$ and RGB are the best color spaces, but the former is the slowest one. Regarding elapsed time, the best vector features MLBP for intensity texture, Daub4 (Daubechies filters using mean and variance statistics) for color texture and FOS, for pure color are nearly the fastest or are in the middle interval of all the tested methods.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Texture analysis is an area that has been studied extensively [1]. Image textures are defined as visual patterns appearing in the image. Texture classification is concerned with the problem of assigning a sample image to one in the set of known texture classes. Originally, texture classification was performed on grayscale images, thus discarding color information. Many gray level texture descriptors have been developed and successfully used in numerous domains for image classification: industrial inspections [2], food science [3,4], content-based image retrieval [5], medicine [6], face recognition [7] among others. Nowadays, the cameras register RGB color images and there are proved evidences that color information improves the overall classification performance [8–11]. So, color texture analysis, which uses chromatic and textural properties to characterize an image, has recently attracted significant attention [12–15].

The methods used for texture classification usually consist of the following steps: preprocessing to make suitable the image to the next step, feature extraction to transform the image into a texture feature vector, and classification to assign the feature vector to one of the available texture classes. A texture image is a function of the texture surface and its albedo, and the ambient illumination. Illumination variation is a very important issue in color texture classification because it can change the perceived structures in the image. The variation in ambient illumination can be due to the spectral variations in the illuminant and to the camera and its viewing position. So, the performance of a color texture classification approach can be affected by the illumination conditions. Given a color texture classification problem, it would be desirable to know which is the best approach to solve the

[*] Corresponding author.
E-mail addresses: eva.cernadas@usc.es (E. Cernadas), manuel.fernandez.delgado@usc.es (M. Fernández-Delgado), nrufino@uvigo.es (E. González-Rufino), pcarrion@uvigo.es (P. Carrión).

http://dx.doi.org/10.1016/j.patcog.2016.07.002
0031-3203/© 2016 Elsevier Ltd. All rights reserved.

problem making the minimal assumptions about the illumination conditions. The solutions proposed to achieve illumination invariance can be enclosed in two types: choice of a pertinent color space [16,17,8], or image normalization for achieving illumination invariance [12]. Both solutions are included in the pre-processing step.

The approaches to analyze color and texture information in images can be grouped into parallel and integrative [10]. The parallel approach joins the gray-level or intensity texture features of the image and the pure color features. The integrative methods, in their simplest version, use the union of the gray level texture features for each color channel. The more sophisticated integrative methods imply the collective analysis of color and texture properties extracted from the color images. This analysis requires vectorial computations, that are more complex and less intuitive than their scalar equivalents. Consequently, the majority of published works compute the texture features on gray level images [8,18], or analyze gray level texture for each color channel [14].

The intensity texture extraction methods can be categorized into: statistical, spectral, structural, model-based and learning approaches [1]. Statistical and spectral techniques are the most popular in the literature of texture classification. Structural approach usually considers the texture as a composition of texture primitives [19] and it only performs well on very regular textures. Model-based approaches, such as the use of Markov Random Fields (MRF) [20], are not widely extended for image classification [21]. The bag-of-features (BoF) framework [22] and texton dictionary-based methods [23–26] can be considered as learning methods that need a learning and representation stage to extract the feature vector that represents the image.

The literature provides several experiments comparing several aspects of color texture classification. Drimbarean and Whelan [8] conducted experiments on 16 images of VisTex dataset to conclude that the use of color improves the performance of gray level texture classification. They use five color spaces (RGB, HSI, CIE-XYZ, CIE-LAB and YIQ) and three texture features: Discrete Cosine Transform (DCT), Gabor filters and Co-occurrence approach, concluding that the best is YIQ color space using integrative approaches and DCT features. Paschos [27] experimentally analyzed the impact of color space (RGB, Lab and HSV) on color texture classification using a dataset of 50 color images and as texture descriptors the Gabor filter. He concluded that HSV is the best, followed by Lab and RGB. Mäenpää and Piatikäinen [17] experimentally compared integrative and parallel approaches on the datasets VisTex, Outex13 and Outex14 using the color spaces RGB, HSV, Lab and $I_1I_2I_3$. They use Local Binary Patterns and Gabor filter as gray level texture features, and color and color ratio histograms as color features. Kandaswamy et al. [9] analyzed the performance of nine different texture features when varying illumination conditions and degrees of affine transformation to classify color texture images in RGB color space.

All these works are limited in the number of datasets, feature descriptors, color spaces and image normalizations used. The objective of the current paper is to develop a wider comparison of the state-of-art color texture features for different combinations of image normalizations and color spaces. We use a wide range of color texture datasets acquired under variable illumination conditions, composed by the public benchmark datasets CUReT, Outex, Vistex, USPTex and ALOT. Sections 2 and 3 briefly describe the techniques used for image normalization, color space transformation and color texture feature extraction. Section 4 describes the datasets and experimental setup, and discusses the results. The main conclusions are summarized in the last section.

2. Image pre-processing

The pre-processing transforms the original texture images into a more suitable form to be used in color and texture feature extraction. This process includes: (1) the image normalization to standardize the image color range, in such a way that the extracted properties from the images are comparable; and (2) the transformation of the image to the working color space.

2.1. Image normalization

The RGB image recorded by a camera changes significantly under different imaging settings, depending on the illumination conditions in the environment, the sensing device and the physical properties of the materials. In general, the texture information of an observed object is severely affected by changes in the illuminant color, and these variations are different among color texture surfaces [12]. Normalization can be defined as any technique that aims to produce a description of an image that is invariant to the illumination conditions under which the image is taken. Some previous gray level texture classification research applied normalization before computing texture descriptors in order to minimize the effect of intensity variations [23,24,9]. More recently, normalization is also applied to color images before other processings [12,28]. The image normalization literature assumes Lambertian surfaces to model the image formation process, in which the response of the k-th sensor is given by the following equation:

$$q_k = \int_\omega E(\lambda)S(\lambda)Q_k(\lambda)d\lambda, \quad k = 1, \ldots, m \tag{1}$$

where $E(\lambda)$ is the spectral power distribution of the light source which specifies how much energy the source emits at each wavelength (λ) of the electromagnetic spectrum; $S(\lambda)$ characterizes the reflectance properties of the surface, which defines what proportion of light incident upon it is reflected by the surface; and $Q_k(\lambda)$ characterizes the sensor, which specifies its sensitivity to light energy at each wavelength of the spectrum. The integral is taken over the range of wavelength λ. Normally, the acquisition devices have three sensors, which are commonly denoted by R, G and B, corresponding to colors red, green and blue respectively.

This image formation model has inspired a number of different techniques for achieving color normalization [29], by finding a procedure that cancels out all variables in the model that are dependent on illumination. In this work we use the following invariant color representations (normalizations): Chroma, GWN, CGWN, HEQ, CLAHE, RGBcb, RGBib, Retinex, MV and $L_{\max}$. Besides, the image is labelled as WN when any normalization is applied.

Chroma: One of the simplest invariants is a *Chromaticity* representation of the image data, derived from a RGB image by:

$$(R', G', B') = \left(\frac{R}{RGB}, \frac{G}{RGB}, \frac{B}{RGB}\right) \tag{2}$$

where $RGB = R + G + B$. A chromaticity vector (R', G', B') is invariant to a change in the intensity of the illuminant.

GWN: An invariant to changes in illumination color is achieved with the transformation *GrayWorld Normalization*:

$$R' = \frac{R}{R_{\text{avg}}}, \quad G' = \frac{G}{G_{\text{avg}}}, \quad B' = \frac{B}{B_{\text{avg}}} \tag{3}$$

where the triplet $(R_{\text{avg}}, G_{\text{avg}}, B_{\text{avg}})$ denotes the mean of all RGBs in the image.

CGWN: Finlayson et al. [30] have shown that successive and repeated application of Eqs. (2) and (3) converges to an image representation (they call a *Comprehensive Gray World Normalized* image), which is both intensity and illuminant color invariant.

HEQ: *Histogram EQualization* is an image enhancement technique originally developed for a gray-scale images. The aim is to increase the overall contrast in the image. To achieve this, the image is transformed to have uniform histogram, which maximizes its entropy and image information content. Based in this theory, Finlayson et al. [28] propose to use a histogram equalization of each channel in the color image as a representation invariant to imaging conditions (lighting or device), applying satisfactory this method to image indexing. This technique is inappropriate if we are interested in improving the visual quality of the image but, in the current context, we are interested in obtaining a representation which is illuminant and/or device invariant.

CLAHE: Contrast Limited Adaptive Histogram Equalization (CLAHE) was originally developed for medical imaging and has proven to be successful for enhancement of low-contrast images. CLAHE is an extension of histogram equalization (HEQ), which attempts to overcome its noise amplification problem [31]. To achieve this goal, CLAHE introduces two key parameters: block-size and clip-limit. This method operates on a small region (whose size is fixed by block-size parameter) instead on a whole image and it limits the amplification by clipping the histogram (fixed by the clip-limit parameter) to a predefined value before computing the cumulative distribution function. The CLAHE method was also developed for gray-scale images and it can be extended to color images in two ways: (a) CLAHE3 applies CLAHE to every color channel of RGB color image providing a representation which is expected to be meaningful for texture discrimination, although it may be not suitable for human perception. We use the default values of the *adapthisteq* Matlab function: block-size=8 pixels and clip-limit=0.01. (b) CLAHEG transforms the RGB image to the Lab color space, applies the original gray-scale CLAHE method to the intensity channel (L channel) and reverses the transformation (this approach is recommended by the Matlab software help with parameters clip-limit=0.005 and block-size=8 pixels). The last approach (CLAHEG) tries to improve the human perception of the image.

RGBcb and RGBib: Color balance algorithms attempt to correct images, which are underexposed, or images taken in artificial or special natural lights, such as sunset. The assumption underlying this algorithm is that the highest values of R, G, B observed in the image must correspond to white, and the lowest values to obscurity. The algorithm simply stretches, as much as it can, the values of the three channels R, G, B, so that they occupy the maximal possible range [0, 255]. The simplest way to do this is to apply an affine transform $x \mapsto ax + b$ to each channel, computing a and b so that the maximal value in the channel becomes 255 and the minimal value 0. We use the implementation provided by Lumire et al. [32]. Often a spectacular improvement in the perceived image is obtained by “clipping” a small percentage of the pixels with the highest values to 255 and a small percentage of the pixels with the lowest values to 0, before applying the affine transform. Notice that this saturation can create flat white regions or flat black regions that may look unnatural. Thus, the percentage of saturated pixels must be as small as possible. We use a percentage of 5% for both algorithms RGBcb and RGBib. For the RGBcb (*color balance*), the algorithm is applied independently on each channel of the RGB color image. For the RGBib (*intensity balance*), the gray level intensity $I = (R + G + B)/3$ is balanced and transformed into I' by the affine transformation with saturation. Finally, for each pixel, the three color channels are multiplied by I'/I in order to correct the intensity of a color image without modifying the $R/G/B$ ratio of the pixels.

Retinex: The Retinex theory was the first computational model to explain and simulate color constancy. These algorithms modify the RGB values at each pixel to give an estimate of color sensation without *a priori* information of the illumination. They try to solve one of the main enigmas of perception: the discrepancy between the physical reflectance of objects and the colors perceived by the human visual system (eye, retina and visual cortex), which is called color sensation. Morel et al. [33] formalized the original Retinex algorithm as a (discrete) partial differential equation. We use the implementation provided by the same authors in [34].

MV: In gray-scale texture classification, the intensity images are commonly normalized to have zero Mean and unit Variance. This normalization gives invariance to global affine transformations in the illuminant intensity. The MV extends this normalization to color images applying the zero mean and unit variance to each color channel in the RGB image.

$L_{\max}$: Similarly to MV, we apply the normalization $L_{\max}$ to each channel of the color image in order to minimize the effect of illumination changes. Let $L_{i,\max}$, $i \in \{r, g, b\}$ be the maximum of each band i over all pixels, this maximum is used to scale each color band:

$$(R', G', B') = \left(\frac{R}{L_{r,\max}}, \frac{G}{L_{g,\max}}, \frac{B}{L_{b,\max}} \right) \tag{4}$$

Some of the above normalizations (Retinex, RGBcb, RGBib and CLAHEG) provide resulting representations of more quality for the human perception, while others provide representations, inappropriated for visualization, which may improve color texture classification.

2.2. Color space

The RGB color space is the most frequently used for color texture classification, because most acquisition devices take images in RGB. Thus, good color texture discrimination can be achieved using RGB under controlled illumination conditions. Some previous works investigate the effect of using different color space for color texture classification [27,8]. Nevertheless, the choice of the best performing color space is an open question: good choice may bring considerable improvements in certain applications of image processing [29]. The most classical color spaces can be grouped in four main families [35]:

- The primary color spaces (RGB, XYZ, etc.), based on the trichromatic theory, which assume that it is possible to match any color by mixing appropriate amounts of three primary colors.
- The luminance–chrominance color space (Lab, Luv, YUV, etc.), where one component represents the luminance and the other two the chrominance.
- The perceptual color spaces (HSI, HSV, HSL, etc.), which try to quantify the subjective human color perception using intensity, hue and saturation.
- The independent axis color spaces (such as $I_1I_2I_3$), resulting from statistical methods which provide components as less correlated as possible.

In this work, we chose one color space of each family, specifically, the RGB, Lab, HSV and $I_1I_2I_3$. We also considered the recently proposed color space $LAB2000_{HL}$ [36], which is perceptually more uniform than the others. We use the Matlab code provided by the authors on their website to convert images to this color space.

3. Color texture features

Color texture extraction can be tackled from different perspectives: simple color features, gray level texture features and

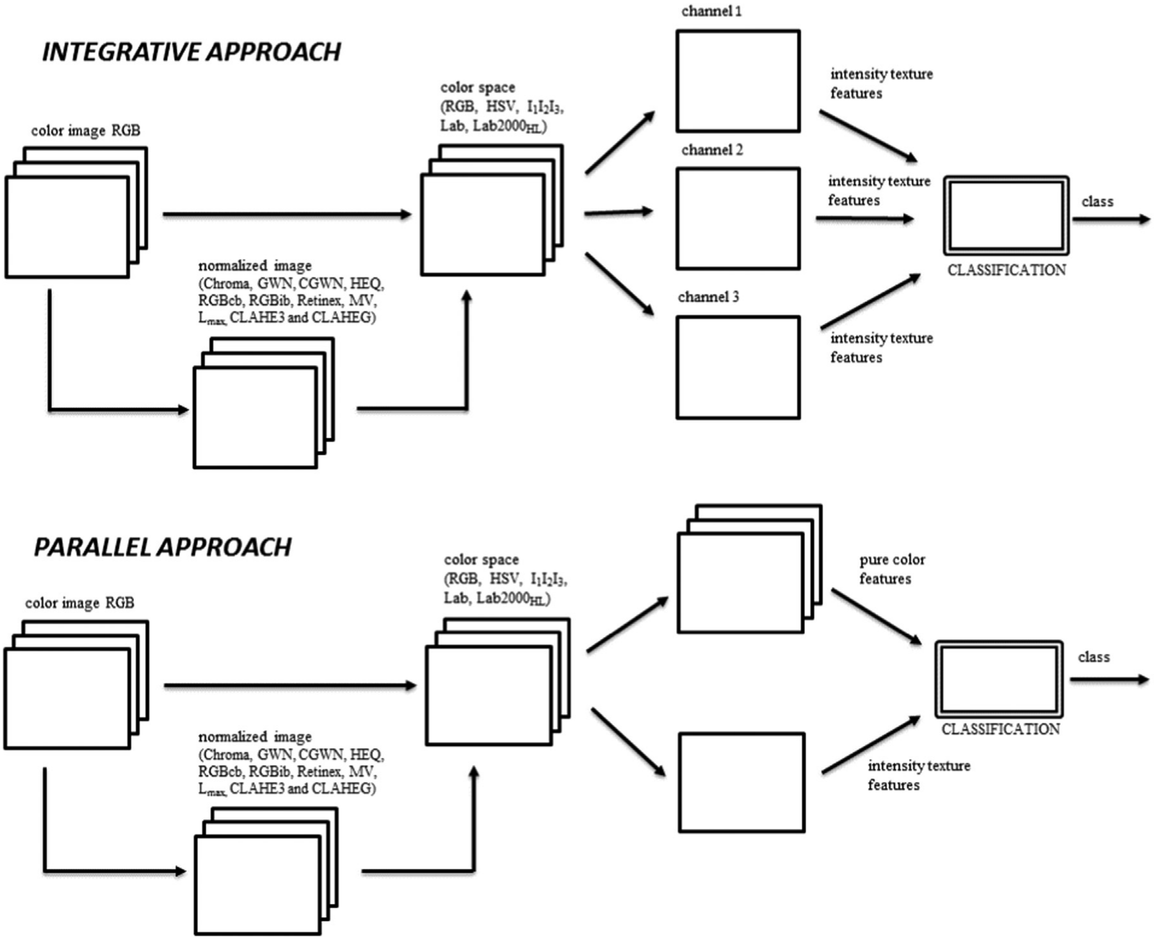
INTEGRATIVE APPROACH
color image RGB
color space (RGB, HSV, I1I2I3, Lab, Lab2000HL)
normalized image (Chroma, GWN, CGWN, HEQ, RGBcb, RGBib, Retinex, MV, Lmax, CLAHE3 and CLAHEG)
channel 1
channel 2
channel 3
intensity texture features
intensity texture features
intensity texture features
CLASSIFICATION
class
PARALLEL APPROACH
color image RGB
color space (RGB, HSV, I1I2I3, Lab, Lab2000HL)
normalized image (Chroma, GWN, CGWN, HEQ, RGBcb, RGBib, Retinex, MV, Lmax, CLAHE3 and CLAHEG)
pure color features
intensity texture features
CLASSIFICATION
class

Fig. 1. Illustration of the integrative (upper panel) and parallel (lower panel) approaches to color texture analysis.

integrative color texture analysis. Parallel approaches are directly derived as the union of pure color and gray level texture features (lower panel of Fig. 1). The simplest integrative approach is also derived as the union of the gray level texture features for each color channel (upper panel in Fig. 1). The normalization process and color space transformation are common to both approaches. In the current paper, we compare the performance of statistical and spectral methods. In the statistical approach, spatial distribution of gray values is defined by several representations such as co-occurrence matrices, sum and difference of histograms, local binary patterns and fractal dimension. Regarding spectral methods, we consider the wavelet and Gabor transforms, which are among the most popular ones. In both cases, each texture image is described by a feature vector which represents a point in a multidimensional feature space.

3.1. Pure color features

We include only the methods that consider color in a *strict* sense. The chromatic channels for the different colors spaces are: the red, blue and green channels for RGB; the a and b channels for Lab and Lab2000HL; the H and S channels for HSV; and I_2 and I_3 channels for $I_1I_2I_3$. Color features computed are: CM, FOS, and CH. The CM (Color Mean) features average the levels on each color channel yielding a vector of 3 values for RGB and 2 values for the rest of color spaces. The FOS (First-Order Statistics) features provide information about the distribution of the levels in the image. They are the simplest features to characterize images, with interesting properties such as invariance to geometric transformations. The vector FOS includes five statistics on each chromatic channel of the image: three channels for RGB color space (15 features) and two channels to the rest of color spaces (10 features). The features computed are [11]: mean level (μ), variance (σ^2), third (m_3) and four (m_4) statistical moments, and entropy (H).

The CH (Color Histogram) provides a compact summarization of the color distribution in an image. It is relatively invariant with translation and rotation about the viewing axis, and it varies only slowly with the angle of view. Color histograms were used as simple color features [17]. A histogram of each chromatic channel of the image is produced by first discretizing the colors in the channel into a number of bins (n_b), and counting the number of image pixels in each bin. We test with $n_b \in \{3, 6, 9, 12, 15, 18, 21\}$ bins. We count the number of pixels in three ways developing three different color feature vectors, namely CCH, JCH and CHP. The CCH (Concatenated Color Histograms) features separately compute the histogram of each chromatic channel (2 or 3 channels depending on the color space considered), quantized in n_b levels, and then concatenate the histograms in a feature vector. The number of color features is $3 \times n_b$ (for RGB) and $2 \times n_b$ for the rest of color spaces. In the JCH (Joint Color Histograms) features, a joint chromaticity histogram is produced by discretization of the two chromaticity channels in the image into n_b bins and counting the number of image pixels in each pair of bins. All the colors spaces considered have two chromaticity channels excepting RGB. A RedGreen chromaticity histogram is formed by first normalizing color pixel values by dividing RGB values by R+G+B, then quantizing the normalized R and B coordinates into n_b bins each one. The number of color features generated is $n_b \times n_b$. In the CHP

(Color Histogram Percentiles) features, a percentile is one of the values of a variable that divides the distribution of the variable into 100 groups having equal frequencies. The percentiles are calculated from a cumulative histogram of each chromatic channel of color image. In our experiments we test the use of n_b percentiles of each chromatic channel to model the color distribution. The number of color features is $3 \times n_b$ (for RGB) and $2 \times n_b$ for the rest of color spaces.

3.2. *Intensity texture features*

The methods included in this section only consider the luminance of the color images. We consider the following intensity images for each color space:

- RGB: The RGB values are converted to intensity values by forming a weighted sum of the R, G and B components given by $0.2989R + 0.5870G + 0.1140B$.
- Lab and Lab2000HL: The intensity image is the lightness channel L, which ranges from 0 (black) to 100 (white).
- HSV: The intensity image is the value channel V, defined as the maximum intensity of the three RGB components.
- $I_1I_2I_3$: The intensity image is the I_1 channel, which is the average of the three RGB components.

Now we will briefly describe the intensity texture features which will be tested on the datasets. They are included in the category *statistical*, like gray level co-occurrence matrix and sum and difference histograms (both second-order statistical methods), local binary patterns and fractal analysis, the category *spectral methods* like wavelets and Gabor filters, and in the category of learning methods like the image patches proposed by Varma and Zisserman [24].

3.2.1. *Second-order statistical features*

The Gray Level Co-occurrence Matrix (GLCM) describes the probability of finding two given pixel values in a predefined relative position in the image [37,38]. The spatial displacement describes the scale and orientation between two points in the image lattice. A matrix is obtained for each scale and orientation. The main problem of GLCM is to choose the appropriate set of scale and orientation parameters that effectively capture the structural information of texture. We average the matrices for each scale and all orientations. From the GLCM matrices, we compute the following features for each scale: contrast, homogeneity, correlation and energy. We construct the vectors: COMS, which includes four features for scale 1; MCOMS with four features for scales $\{1, 2, 3, 4, 5, 6, 7, 8\}$, summing 32 features. The Sum and Difference Histograms were introduced by Unser [39] as an alternative to the usual co-occurrence matrices used for texture analysis with the advantage of decreasing the computation time and memory storage. We use two vectors: SDH, which includes five features (energy, correlation, entropy, contrast and homogeneity) calculated for scale 1, and MSDH (multidimensional SDH), including 40 features, 5 for each scale in the set $\{1, 2, 3, 4, 5, 6, 7, 8\}$.

3.2.2. *Local Binary Patterns (LBP)*

The LBP operator describes each image pixel by comparing each pixel with its neighbors [40]. Precisely, for each neighboring pixel, the result will be set to one if its value is higher than the value of central pixel, otherwise the result will be set to zero. The LBP code of the central pixel is then obtained by multiplying the results with weights given by powers of two, and summing then up together. The histogram of the binary patterns computed over all pixels of the image is generally used for texture description. The final LBP feature vector is very fast to compute and is invariant to monotonic illumination changes. The main drawback of LBP features lies in the high dimensionality of histograms produced by LBP codes (if P is the number of neighboring pixels, then the LBP feature will have 2^P distinct values, resulting in a 2^P-dimensional histogram). Many classifiers cannot operate with high dimensional patterns.

Some variants have been proposed to reduce the dimensionality of original LBP [41,40]: LBP with *uniform patterns* and CLBC (Completed Local Binary Count). The uniform patterns are binary patterns with only two transitions (from 0 to 1 and vice versa). It was found that most of the micro-structures such as bright/dark spots and flat regions can be successfully represented by uniform patterns. In a circularly symmetric neighbor set of P pixels can occur $P + 1$ *uniform* binary patterns. The number of “1's” in the binary pattern is the label of the pattern, while the nonuniform patterns are labelled by $P + 1$. The histogram of the pattern labels accumulated over the intensity image is employed as texture feature vector. We use two texture feature vectors: LBP^{riu} use P=8 and radius $R = 1$ and contains 10 features; and MLBP^{riu} (multiresolution LBP_{riu}) concatenates the features for $(P, R) \in \{(8, 1), (16, 2), (24, 3)\}$ summing 54 features. We use the LBPMatlab[1] implementation.

Local Binary Count [42] (LBC) counts the number of value “1's” in the binary neighbor sets instead of encoding them like LBP. This vector measures how many pixels have comparatively higher value than the central one in a local area, i.e. only focus on the local binary value difference information. We use two texture feature vectors: CLBC and MCLBC (multiresolution LBC) with 9 and 53 features respectively. The vector CLBC use $P = 8$ and radius $R = 1$, while MCLBC concatenates the features for $(P, R) \in \{(8, 1), (16, 2), (24, 3)\}$. We use the publically available[2] LBC implementation.

3.2.3. *Fractal texture features*

The fractal dimension is a useful metric for the analysis of images with self-similar content such as textures [43,44]. For images with limited resolutions and size, the fractal dimension D is related with the concept of self-similarity at some scales. The most wide-spread literature methods can be grouped in three families: box-counting methods, fractional Brownian motion methods and area measurement methods. Based on our previous experience [11], we use the probability method [45] from the first family: let $P(m, L)$ be the probability that there are m surface points within a box of size $L \times L \times L$ centered about an arbitrary point of the region surface. For any value of L we have $\sum_{m=1}^{N_p} P(m, L) = 1$, being $N_p = L^2$ the number of possible points in the box. The number of boxes (N_b) with size L needed to cover the whole image is computed as $N_b(L) = \sum_{m=1}^{N_p} \frac{P(m,L)}{m}$. The fractal dimension D is estimated from $N_b(L) \propto L^{-D}$. We compute the number of boxes for different scales. The main problem is to choose the appropriate set of scales. We carried out some experiments varying the scales to test their influence on the classification accuracy, selecting the following scales: (1) the box size $L = \{3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$ pixels for the probability method, yielding a vector of 10 features called MFP (MultiFractal Probability). Xu et al. [46] introduce a texture descriptor called MFS (MultiFractal Spectrum), which is based on fractal geometry theory and box-counting methods. Furthermore, MFS has low dimension (26 features), has satisfactory performance in handling illumination changes and it is very efficient to compute. We use the publically available MFS[3] code with its default parameter values.

[1] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

[2] http://home.ustc.edu.cn/∼zyknight/CLBC.rar

[3] http://www.cfar.umd.edu/∼fer/website-texture/texture.htm

3.2.4. *Gabor filters*

This is a widely adopted technique for texture classification [47]. A two-dimensional Gabor filter consists of sinusoidal wave modulated by a Gaussian envelope. It performs a localized and oriented frequency analysis of the image. The crucial step in the design of Gabor filters is to choose suitably tuned parameters. Bianconi and Fernández [48] investigate the effect of Gabor filter parameters on texture classification. Their conclusions suggest the following Gabor filter parameters: central frequency of the filter at the highest frequency (F_m=0.327), number of frequencies (n_F=4), number of orientations (n_O=6), the frequency ratio is half-octave frequency spacing ($F_r = \sqrt{2}$) and standard deviation of the Gaussian enveloped, i.e. the *smoothing* parameters ($\gamma = \eta = 0.5$). Another common assumption is that the angular spacing among the filters is uniform. We use the Simplegabor Toolbox[4] for multi-resolution Gabor filtering of 2-D signals (images), with the previous configuration to process all the datasets. Given a bank of digital Gabor filters $G_{ij}(x, y)$ with $i \in \{1, \ldots, n_F\}$ and $j \in \{1, \ldots, n_O\}$, the Gabor transform of the input image is computed for each filter of the bank. The Simplegabor toolbox allows the use of Gabor transform Without Normalization (vector GWN), Gabor with Normalization for Illumination invariance (vector GNI) and Gabor with Normalization for Rotation and Illumination invariance (vector GNRI). Afterwards, the mean μ_{ij} and the standard deviation σ_{ij} of the magnitude of each transformed image are used as elements of the texture feature vector. The three texture vectors computed contain 48 features each one.

3.2.5. *Wavelet features*

Discrete wavelet transform (DWT) representation is a theory for multi-dimensional signal decomposition [49]. A two-band Quadrature Mirror Filter (QMF) bank utilizes orthogonal analysis filters to decompose data into low-pass and high-pass frequency bands. Applying the filters recursively to the lower frequency bands produces wavelet decomposition. The wavelet transformation involves filtering and subsampling [50]. A compact representation needs to be derived in the transform domain for texture classification. The mean and variance of the energy distribution for the transform coefficients in each subband and decomposition level are used to construct the feature vector. We compute four feature vectors varying the statistical features calculated (only mean of energy or both mean and variance) and the analysis filter used (Haar or Daubechies). In all the cases we use 3 levels of decomposition. With Daubechies filters we use a filter of size 4. The feature vectors are: Haar (Haar filter and mean and variance statistics, 20 features); HaarME (Haar filtering and MEan statistic, 10 features); Daub4 (Daubechies filtering and mean and variance statistics, 20 features); and Daub4Me (only MEan statistics, 10 features). For the calculation of wavelet texture features, we use the Wavelab Toolbox,[5] which provides a variety of algorithms related to wavelet analysis on 1-D and 2-D signals (images).

The dual-tree complex wavelet transform ($DT - CWT$) is a more recent enhancement to the DWT proposed by Selesnick et al. [51] with the following important properties: it is nearly shift invariant and it has good directional selectivity in two dimensions. It can be considered as a special case of Gabor filters with complex coefficients, taking the directional advantages of the Gabor filters. Celik et al. [52] use $DT - CWT$ for texture classification. DT-CWT is obtained by filtering an image separably: two trees are used for the rows of the image and two trees for the columns in a quad-tree structure with 4:1 redundancy. The four quad-tree components of each $DT - CWT$ coefficients are combined by simple arithmetic sum and difference operations to yield a pair of complex coefficients. This produces six directionally selective subbands for each scale of the two-dimensional $DT - CWT$ at approximately $\pm$ 15°, $\pm$ 45° and $\pm$ 75°. Since $DT - CWT$ produces complex coefficients for each directional subband at each scale, Celik et al. [52] use as texture features the variance and entropy of the magnitude of the coefficients. For comparative purpose with the DWT, we use 3 levels of decomposition and compute two feature vectors: DTCWT (mean and variance statistics for 3 scales and 6 directions, 36 features) and DTCWT-VH (variance and entropy statistics for 3 scales and 6 directions, 36 features). For the calculation of $DT - CWT$, we use the Matlab Toolbox[6] provided by the authors.

3.2.6. *Learning methods*

These methods are not direct and they need a learning step to calculate the texture feature vector that represents each image. In general, they consist of the following major stages: train the encoder, encode each image of the dataset and classification. The training of the encoder encloses the computation of image descriptors on the training set and the construction of the texton or visual vocabulary. We evaluate different combinations of state-of-the-art methods for each stage. Common image descriptors include local image patches [24] and Dense Scale Invariant Feature Transform (DSIFT). The image descriptors of the training set are used to construct the visual vocabulary that encodes in a single vector the information contained in the image descriptors. Some encoding methods include clustering of image descriptors, bag-of-visual-words (BoVW), or high-order statistics as the Fisher Vector (FV) or the Vector of Locally Aggregated Descriptors (VLAD) encoders. Specifically, we use the following configurations.

VZ-PATCH [24] uses as image descriptor the raw pixels intensity of an $n \times n$ squared neighborhood around each image pixel converted to a vector of dimension n^2. The visual vocabulary is formed aggregating all the vectors in the training set and applying K-means clustering. The cluster centers are the texton dictionary. The model corresponding to a training image is the histogram of texton frequencies. The classification of a novel image is achieved by a 1-Nearest Neighbor (1-NN) classifier using χ^2 statistic as the distance measure. We use image patches of size n=3 pixels and a dictionary of 10 textons per class, which are the default values for the implementation used.

DSIFT is considered the best local descriptor in the context of object recognition. It is the histogram of the occurrences of image gradients quantized with respect to their location and orientation within a patch. Different scales can be simulated by progressively smoothing the input image or to gradually reducing the image resolution. We use the implementation of the VLFeat[7] open source library [53] with the default parameters: seven scales with a factor $\sqrt{2}$ between successive scales, computed with a 4 pixels step. The SIFT descriptor is reduced to 80 and 100 values respectively to FV and VLAD using Principal Component Analysis (PCA). The encoding methods are also applied using default values: number of words are 1024 for BOVW and 64 for FV and VLAD; the number of samples per word are 200 for BOVW and 1000 for FV and VLAD. We use three combinations: DSIFT-BOVW, DSIFT-FV and DSIFT-VLAD. Finally, each image is encoded and a classifier is used to discriminate among classes. We use two classifiers: (1) 1NN, as in VZ-PATCH for comparative purpose; and (2) linear Support Vector Machine, recommended by the VLFeat library, with the default values.

[4] http://www2.it.lut.fi/project/simplegabor/downloads/src/simplegabortb

[5] http://www-stat.stanford.edu/~wavelab/Wavelab_850/index_wavelab850.html

[6] http://eeweb.poly.edu/iselesni/WaveletSoftware/index.html

[7] http://www.vlfeat.org/index.html

4. Results and discussion

This section outlines the experimental setup and the discussion of the results obtained in our experiments. We use the most popular public color texture datasets: VisTex (the Vision Texture database), Columbia-Utrecht (CUReT), University San Paulo (USPTex), Amsterdan Library of Textures (ALOT) and Outex Texture Database. The VisTex database[8] contains a large set of natural color images taken under noncontrolled conditions. The images are acquired with many different cameras, without any knowledge about the illumination sources, imaging geometries, image scale, gamma correction, etc. VisTex contains 54 natural textures of 512 × 512 pixels, which were split into 16 non-overlapping sub-images of 128 × 128 pixels. A total of 864 samples grouped into 54 classes were considered.

The CUReT[9] database contains 61 textures, each one with 205 images obtained under different viewing and illumination conditions. We chose the 92 images of each texture, whose viewing angle is higher than 60 degrees, similarly to the setup of Varma and Zisserman [23]. A total of 5612 samples grouped into 61 classes are considered. For the same texture, there is a considerable difference in the appearance across images with variation in imaging conditions (illumination and viewing angle).

The USPTex database [13] is a set of natural texture images acquired using a digital camera with 512 × 384 pixel resolution. Texture classes are typically found daily such as beans, rise, tissues, road scenes, various types of vegetation, walls, clouds, soils, blacktop and gravel. Each image is split into 12 non-overlapping sub-images of 128 × 128 pixels. A total of 2292 samples grouped into 191 classes are considered (12 images per class).

The University of Oulu constructed several image databases for texture classification. Each texture available is captured using three different light source: Horizon, Inca, and TL84. Each texture is captured using six spatial resolutions and nine rotation angles. Varying the number of textures, light source, spatial resolution and rotation angles they constructed several test suites. In this study, we use two test suites[10]: Outex_TC_00013 contains 20 images of 68 textures of 128 × 128 pixel resolution captured using illuminants Inca, the spatial resolution 100 dpi and the rotation angle 0° (in total 1360 images); and Outex_TC_00014, which contains 10 images of 68 textures of 128 × 128 pixel resolution captured using all illuminants, spatial resolution 100 dpi and rotation angle 0° (in total 2040 images). The training set includes the images captured by Inca illuminant and the test set includes the images of TL84 and Horizon illuminants (both sets defined by the dataset creators). Within the Outex dataset, we consider two additional sets: (1) Outex_TC_00014_10s with the same images than Outex_TC_00014, but mixing the images from different illuminants; this set includes in total 2040 images (68 classes × 10 images per class × 3 illuminants); and (2) Outex_TC_00014_20s, equal Outex_TC_00014_10s but including 20 images per class as Outex_TC_00013; this set includes in total 4080 images (68 classes × 20 images per class × 3 illuminants).

Finally, the ALOT database [54] contains 250 textures, each one with 100 images obtained under different illumination conditions, i.e. variations in the illumination direction, illumination color and camera viewpoint. A total of 25,000 samples grouped into 250 classes were considered (100 images per class). The image resolution is 384 × 256 pixels.

Although there are many sophisticated classifiers in the literature [55], many of them used to texture classification, the nearest-neighbor is often used as an uncommitted mechanism to compare texture representations due to its simplicity and absence of parameters that need to be tuned. Since our objective is to compare feature vectors for color texture classification, we want to find the one which achieves the best performance, i.e., the one that provides the easiest classification problem. This problem should be the easiest one for any classifier, so we can hypothesize that the conclusions do not depend on the specific classifier. Therefore, we limit our study to the 1NN (*k-Nearest Neighbor* classifier with $k=1$), thus avoiding the fine tuning of the number of neighbors k. Nevertheless, in Section 4.9 we test our hypothesis comparing the conclusions achieved using 1NN and Random Forest, which according to [55] is one of the best available classifiers. The experiments use 4-fold cross validation: each dataset is randomly divided into four equal-sized sets; for each fold, three sets (75% of the available patterns) are used for training and the remaining set (25% of the patterns) is used for testing (being different in each fold), with the same training/test partitions for all the feature vectors in each fold. The only exception is dataset Outex_TC_00014, which provides separated training and testing sets. In each dataset, all the classes are equally populated.

The original RGB color images are preprocessed using the 11 normalization approaches described in Section 2.1: Chroma, GWN, CGWN, HEQ, CLAHE3, CLAHEG, RGBcb, RGBib, Retinex, MV and $L_{\max}$. The label WN (Without Normalization) means that normalization is not applied. The resulting images, originally in the RGB space, are transformed to the 4 remaining color spaces: HSV, $I_1I_2I_3$, Lab and $LAB2000_{HL}$ (labeled Lab2 hereafter). On the transformed images we use the color and texture features described in Section 3.

4.1. Intensity texture and pure color vectors

The first experiment evaluates the performance of gray level texture features and simple color features using the 1NN classifier. The *intensity texture vectors* are obtained computing the texture features described in Section 3.2 to the intensity component of the color images, which is the L channel for Lab and Lab2, the V channel for HSV and the I_1 channel for $I_1I_2I_3$ color space. We generate the following 19 vectors (in brackets the number of features): CLBC (9), COMS (4), Daub4ME (10), Daub4 (20), DTCWT (36), DTCWTVH(36) GWN (48), GNI (48), GNRI (48), HaarME (10), Haar (20), LBP (10), MCLBC (51), MCOMS (32), MFP (10), MFS (26), MLBP (54), MSDHC (40), SDHC (5). The *pure color vectors* are obtained computing the features described in Section 3.1 on color images. We generate the following 23 vectors (the number of features depends on the color space or the number of bins): FOS, CM, CCHxx, JCHxx and CHPxx, where xx is the number of bins tuned using values $n_b \in \{3, 6, 9, 12, 15, 18, 21\}$.

We use a total of 1140 feature vectors for intensity features (12 normalization types considering WN, 5 color spaces considering RGB and 19 texture vectors), and 1380 feature vectors for pure color features (12 normalizations × 5 color spaces × 23 color vectors) summing up to 1140 + 1380 = 2520 accuracy values for each database considered. While the complete results are available online,[11] Table 1 shows the best accuracies reached by some color and intensity texture feature and normalizations for each color space and dataset. As an example for dataset VisTex, if the best accuracies reached by the RGB color space without normalization for intensity (93.6 using GNI vector) and color (89.5 using FOS vector) are considered as a base line, normalization or color space transformation improves accuracies for all cases. Particularly, for VisTex/intensity features the accuracy increases up to 95.4 (RGB–Retinex), up to 94.4 (Lab2–WN) and up to 96.6 (Lab2–RGBcb). For

[8] http://www.cse.oulu.fi/CMV/ImageData, test suite Contrib_TC_00006.

[9] http://www.robots.ox.ac.uk/~vgg/research/texclass/index.html

[10] http://www.outex.oulu.fi

[11] http://persoal.citius.usc.es/eva.cernadas/papers

Table 1
Classification accuracy (in %) of different pure color and intensity texture features for all the color spaces, normalizations and databases.

Color	VisTex		CUReT		USPTex		ALOT	
	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.
	Intensity texture features							
RGB	WN, GNI	93.6	WN, MLBP	94.1	WN, Daub4	80.5	WN, MLBP	92.3
RGB	Retinex, MLBP	95.4	RGBcb, MLBP	95	L_{max}, Daub4	83	RGBcb, MLBP	92.9
HSV	WN, MLBP	94.2	WN, MLBP	95.3	WN, Daub4	80.8	WN, MLBP	**93.6**
HSV	Retinex, MLBP	95.9	RGBcb, MLBP	**95.6**	RGBib, MLBP	82.4		
Lab	WN, MLBP	93.5	WN, MLBP	94.5	WN, Daub4	80.8	WN, MLBP	92.5
Lab	RGBcb, MLBP	96.5	RGBib, MLBP	95	GWN, Daub4	**83.5**	RGBcb, MLBP	92.7
Lab2	WN, MLBP	94.4	WN, MLBP	94.4	WN, Daub4	80.5	WN, MLBP	92.4
Lab2	RGBcb, MLBP	**96.6**	RGBib, MLBP	95.1	GWN, Daub4	82.5	RGBcb, MLBP	93
$I_1I_2I_3$	WN, GNI	93.6	WN, MLBP	94.5	WN, Daub4	80.6	WN, MLBP	92.1
$I_1I_2I_3$	RGBcb, MLBP	95.4	RGBib, MLBP	95.1	GWN, Daub4	82.5	Retinex, MLBP	92.2
	Color features							
RGB	WN, FOS	89.5	WN, FOS	87.2	WN, FOS	77.4	WN, FOS	72.3
RGB	Chroma, FOS	91.6	RGBib, FOS	90.3	Chroma, FOS	82.1	L_{max}, FOS	76.5
HSV	WN, JCH21	87	WN, JCH18	91	WN, JCH18-21	70.7	WN, JCH21	69.8
HSV	GWN, JCH12	**94.4**	CLAHE3, JCH15	**93**	CLAHE3, JCH18	**84**	GWN, JCH21	**78.4**
Lab	WN, FOS	86.6	WN, FOS	89.3	WN, FOS	72.5	WN, FOS	68.1
Lab			RGBib, FOS	89.6			RGBIb, FOS	69.2
Lab2	WN, FOS	89.2	WN , FOS	90	WN, FOS	75.1	WN, FOS	73.1
Lab2								
$I_1I_2I_3$	WN, FOS	85.9	WN, FOS	83.6	WN, FOS	71.6	WN, FOS	73.1
$I_1I_2I_3$	Chroma, FOS	89.1	RGBib, FOS	89.1	Chroma, FOS	71.7		
Color	Outex13		Outex14 20s		Outex14 10s		Outex 14	
	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.
	Intensity texture features							
RGB	WN,Daub4	85.5	WN, Haar	85.8	WN, GNI	80.4	WN, MLBP	65.7
RGB			HEQ, GNI	**91.6**	CLAHE3, GNI	81.7	RGBcb, Daub4	**69.8**
HSV	WN, Daub4	85.5	WN, Haar	86	WN, Daub4	76.3	WN, MLBP	60.5
HSV			RGBcb, GNI	86.5	CLAHE3, DaubME	77	RGBcb, GNI	66
Lab	WN, Daub4	87.1	WN, Daub4	86.4	WN, Daub4	80.7	WN, MLBP	64.6
Lab			HEQ, GNI	89.3	L_{max}, Daub4	80.7	RGBcb, Haar	69.2
Lab2	WN, Daub4	**87.4**	WN, Haar	87.6	WN, Daub4	**82.4**	WN, MLBP	64.2
Lab2			HEQ, GNI	89.6			RGBcb, Haar	68.4
$I_1I_2I_3$	WN, Haar	83.9	WN, Haar	83.8	WN, Haar	75.4	WN, MLBP	62.3
$I_1I_2I_3$			MV, GWN	88	CLAHE3, Daub4	77.7	MV, GWN	67
	Color features							
RGB	WN, FOS	86.6	WN, FOS	84.1	WN, FOS	73	WN, CCH9	30.4
RGB	CLAHE3, FOS	**88.3**	CLAHE3, FOS	**85.5**	CLAHE3, FOS	**76.5**	L_{max}, FOS	34.4
HSV	WN, FOS	76.2	WN, FOS	74.1	WN, FOS	60.2	WN, FOS	5.9
HSV	CGWN, FOS	76.3					HEQ, JCH12	**37.8**
Lab	WN, FOS	76.2	WN, FOS	75.4	WN, FOS	63.2	WN, FOS	13.3
Lab	Chroma, FOS	78	RGBib, FOS	77.7	RGBib, FOS	66	GWN, FOS	34.4
Lab2	WN, FOS	76.5	WN, FOS	75	WN, FOS	63.5	WN, CCH9	14
Lab2	Chroma, FOS	76.9	RGBib, FOS	77.5	RGBib, FOS	65.3	GWN, FOS	35.1
$I_1I_2I_3$	WN, FOS	74.6	WN, FOS	76.5	WN, FOS	64	WN, FOS	11
$I_1I_2I_3$	RGBib, FOS	80.1	RGBib, FOS	80.9	RGBib, FOS	70	RGBcb, FOS	26.4

VisTex/color features, the accuracy increases up to 91.6 (RGB–Chroma), up to 94.4 (HSV-GWN). For CUReT/intensity texture features, the accuracy 94.1 (RGB–WN) grows up to 95 (RGB–RGBcb), up to 94.5 (Lab–WN or $I_1I_2I_3$–WN) and up to 95.6 (HSV–RGBcb). For CUReT/color features, the accuracy 87.2 (RGB–WN) grows up to 90.3 (RGB–RGBib), 91 up to (HSV–WN) and up to 93 (HSV–CLAHE3). For USPTex/intensity texture features, the accuracy 80.5 (RGB–WN) grows up to 83 (RGB–L_{max}), up to 80.8 (Lab-WN or HSV–WN) and up to 83.5 (Lab–GWN). For USPTex/color features, the accuracy 77.4 (RGB–WN) grows up to 82.1 (RGB–Chroma) and up to 84 (HSV–CLAHE3). For ALOT/intensity texture features, the accuracy 92.3 (RGB–WN) grows up to 92.9 (RGB–RGBcb) and up to 93.6 (HSV–WN). For ALOT/color features, the accuracy 72.3 (RGB–WN) grows up to 76.5 (RGB–L_{max}), up to 73.1 (Lab2–WN or $I_1I_2I_3$–WN), and up to 78.4 (HSV-GWN). For Outex datasets family, the CLAHE3 normalization normally improves the accuracies for color features. Specifically, the accuracies of RGB–WN–FOS raise respectively from 86.6, 84.1 and 73 for Outex13, Outex14-20s and Outex14-10s up to 88.3, 85.5 and 76.5 for RGB–CLAHE3–FOS. For Outex13 /intensity features the accuracy increase from 85.5 (RGB–WN) up to 87.4 (Lab2–WN). For Outex14-20s/intensity features the accuracy increases from 85.8 (RGB–WN) up to 91.6 (RGB–HEQ), up to 87.6 (Lab2–WN). For Outex14-10s/intensity features the accuracy increases from 80.4 (RGB–WN) up to 81.7 (RGB–CLAHE3) and to 82.4 (Lab2–WN). For Outex14, in which the testing set includes images capturing by a different illuminants than the training set, the normalization and color space are again very important. The accuracies achieved for RGB without normalization are: 67.5 for intensity features and 30.4 for color features. These values grows up to 69.8 (RGB–RGBcb) for intensity features and 37.8 (HSV-HEQ) for color features. These results lead to conclude that normalization and color spaces are important to increase the accuracy

Table 2
Average accuracy, Friedman rank and p-value over all datasets using 1NN classifier.

Feature	Avg.	Std.	Rank	Pos.	p-value
Intensity feature					
Lab2, RGBcb, MLBP	83.49	10.8	78.7	1	–
Lab, RGBcb, MLBP	83.17	11.1	84.4	2	0.29
Lab2, WN, Daub4	81.47	11.1	84.5	3	0.58
Lab, WN, Daub4	80.92	11.2	91.3	5	0.44
RGB, L_{max}, Daub4	80.91	10.2	93.4	7	0.38
RGB, RGBcb, MLBP	82.79	10.8	94.6	8	**0.030**
Lab2, WN, MLBP	82.56	10.6	94.9	9	0.11
Lab, WN, MLBP	82.35	10.5	106.6	14	0.11
HSV, WN, Daub4	80.00	11.5	107.9	16	0.27
RGB, WN, MLBP	82.29	10.2	111.3	20	0.15
RGB, RGBcb, GNI	80.83	9.4	126.7	44	0.44
$I_1I_2I_3$, WN, MLBP	81.01	11.5	150.5	72	**0.0004**
RGB–L_{max}–DTCWTVH	75.85	11.2	244.1	206	**0.049**
Lab, RGBcb, DTCWT	44.83	11.7	862.9	884	**0.0001**
Color feature					
RGB, L_{max}, FOS	74.08	17.2	29.9	1	
RGB, WN, FOS	74.71	19.8	35.2	2	0.67
Lab2, WN, FOS	69.39	24.4	72.1	5	0.14
RGB, CLAHE3, CCH21	65.41	18.0	76.7	7	**0.013**
RGB, CLAHE3, CCH3	65.39	18.2	76.7	8	**0.016**
RGB, CLAHE3, CCH9	65.11	18.1	76.7	9	**0.014**
RGB, CLAHE3, CCH6	65.24	18.1	76.7	10	**0.012**
HSV, GWN, FOS	63.54	18.8	88.1	16	**0.019**
$I_1I_2I_3$, Chroma, FOS	68.9	23.3	89.4	17	O.074
HSV, RGBcb, JCH3	62.24	18.0	97.6	19	**0.016**
Lab, WN, FOS	67.75	23.9	97.6	20	**0.048**
RGB, CLAHE3, CHP9	67.85	16.2	97.6	22	**0.004**

average for both color and intensity texture classification (overall for intensity texture classification). The best texture features for all color spaces and normalizations are frequently: MLBP (Rotation invariant multidimensional Local Binary Patterns), the wavelet family (vectors Daub4, Daub4Me, Haar and HaarMe) and Gabor vectors (GNI or GWN). The best color feature vector is frequently the FOS (First-Order Statistics), although sometimes the vectors JCHxx (Joint Color Histograms) achieve the best accuracy in the HSV color space.

In order to analyze which is the best combination for all datasets, Table 2 shows the average value, deviation, the ranking given by Friedman test [56], decreasing with the classifier quality, the position attending to this rank (approximately the top 20 feature vectors are shown), and the p-value of a paired T-Test comparing the best of each half of the table and each vector. Since the Friedman test requires an accuracy value for each feature vector, those cases with errors were imputed the accuracy achieved by a random classification. The normalization CGWN is not included because it always provides results very closed to WN (without normalization). As well, a combination of color space, normalization and feature vector is not shown when the same combination without normalization (WN) achieves lower Friedman rank. According to color space, Lab2 with RGBcb normalization and texture features MLBP achieves the best position using intensity images (the mean accuracy is also the best achieving 83.49%). The first four feature vectors use luminance–chrominance color space (Lab or Lab2) and the first 15 feature vectors use color space RGB, Lab and Lab2. The first feature vectors that use other color spaces are in position 16 (HSV) and 72 ($I_1I_2I_3$). A paired T-Test reveals no significant difference between the best vector of Lab2 color space and the best vectors of Lab, RGB and HSV color space (except for the combination RGB, RGBcb, MLBP). However, the difference is significant ($p < 0.05$)with the $I_1I_2I_3$ color space. According to texture feature, the first 20 vectors use the texture features based on wavelets (Daub4) and Local Binary Patterns (MLBP). Among the wavelet family, the DWT (Daub4) is clearly better than the dual-tree complex wavelet transform (DTCWT, position 206). The variance and entropy characteristics (vector DTCWTVH) are clearly better than mean and variance (DTCWT, position 884). A paired T-Test between Lab2–WN–Daub4 and RGB–L_{max}–DTCWTVH gives a p=2.0285e-05, and comparing RGB–L_{max}–DTCWTVH with Lab–RGBcb–DTCWT gives a p=0.0001 (both of them significant). The first Gabor feature vector (GNI) is the position 44 and the best fractal vector (MFS) is in the position 532. Among statistical texture features, MLBP is better than MCLBC (position 157), MCOMS (position 303) and MSDHC (position 267). Thus, multidimensional feature vectors (MLBP, MCLBC, MCOMS and MSDHC) are always better than their corresponding non-multidimensional ones (LBP, CLBC, COMS and SDHC), because they use a bigger local neighborhood to calculate the texture features. According to normalization, although the two best vectors use normalization RGBcb, the third one does not use any normalization (WN). With respect to pure color features (lower half of Table 2), the best vectors are the color space RGB and features FOS using L_{max} normalization (first position and average accuracy 74.08%) and without normalization (second position and average accuracy 74.71%). The remaining vectors work much worse (below 69.39%). The first CCH (CCH21) is in the position 7, the first JCHxx (JCH3) is in the position 9 and the first CHPxx (CHP9) is in the position 22. In fact, their difference with the best (RGB, L_{max}, FOS) is statistically significant. The second best vector RGB–WN–FOS gives is significantly better than RGB–CLAHE3–CCH21, HSV-RGBcb-JCH3 and RGB–CLAHE3–CHP9. So the color space RGB joint to color features FOS are the best choice with L_{max} normalization or without normalization. The fact that RGB is the best for pure color features suggests that some color information is lost in the transformation to other spaces. The reason might be that in RGB color features are calculated on the three channels, while in the other color spaces they are calculated on the two-color channels.

4.2. Comparison with learning methods

The learning methods (Section 3.2.6) do not compute the texture feature vector in a direct way, and they need a learning process to compute the patterns. We tried to run these approaches with the evaluation methodology to classify intensity images using all the color spaces and normalizations, resulting in 60 processes for each dataset (5 color space times 12 normalizations). The huge computational cost of the VZ-PATCH approach made impossible to run it on the eight datasets with our experimental methodology, and we limited our study to the four smaller datasets (VisTex, Outex13, Outex14 and USPTex). For the remaining datasets the executions did not finish after 192 h and they were cut. Table 3 shows the best accuracies reached by the VZ-PATCH vector with some normalization for all the color space and datasets. For

Table 3
Classification accuracy of patch image approach (VZ-PATCH) to classify intensity images for all color spaces and normalizations.

Color	VisTex		Outex13		USPTex		Outex14	
	Norm.	Acc.	Norm.	Acc.	Norm.	Acc.	Norm.	Acc.
RGB	WN	94.1	WN	84.9	WN	87	WN	74.2
RGB	CLAHE3	**97.9**	GWN	**86.2**	CLAHE3	93.7	CLAHE3	74.8
HSV	WN	94.1	WN	84.5	WN	84.6	WN	65.2
HSV	CLAHE3	96.9	CGWN	84.3	CLAHE3	93.5	GWN	69.9
Lab	WN	95.7	WN	83.4	WN	85.6	WN	75
Lab	CLAHE3	97.8	RGBib	84.2	CLAHE3	93.1	MV	76.4
Lab2	WN	95.5	WN	84.4	WN	85.1	WN	76
Lab2	CLAHE3	97.7	CLAHE3	84.9	CLAHE3	92.2	GWN	**77**
$I_1I_2I_3$	WN	94.2	WN	83.1	WN	86.6	WN	72.6
$I_1I_2I_3$	CLAHE3	97.6	gwn	83.6	CLAHE3	**94.2**	GWNI	72.7

Table 4
Classification accuracy of SIFT descriptors to classify intensity images for all color spaces and normalizations.

Color	Clas.	VisTex			Outex13			USPTex			Outex14		
		Norm.	Enc.	Acc.	Norm.	Enc.	Acc.	Norm.	Enc.	Acc.	Norm.	Enc.	Acc.
RGB	1NN	WN	FV	86	WN	FV	71.3	WN	FV	70.5	WN	FV	67.3
RGB		CHROMA	FV	87.5	L_{max}	FV	72.7	Retinex	FV	71.2	L_{max}	FV	67.6
RGB	SVM	WN	FV	95.8	WN	FV	77.3	WN	FV	87.7	WN	FV	72.6
RGB		CHROMA	FV	96.9	Retinex	FV	79.6	L_{max}	FV	**88**	–	–	–
HSV	1NN	WN	FV	86.5	WN	FV	74.5	WN	FV	69.7	WN	FV	65.1
HSV		GWN	FV	87.5	–	–	–	Retinex	FV	72.3	–	–	–
HSV	SVM	WN	FV	96.1	WN	FV	78	WN	FV	87.4	WN	FV	69.4
HSV		GWN	FV	96.8	Retinex	FV	**80.7**	Retinex	FV	87.7	GWN	FV	70.7
Lab	1NN	WN	FV	88.2	WN	FV	71.1	WN	FV	70.8	WN	FV	67
Lab		L_{max}	FV	88.9	CLAHE	FV	72.3	Retinex	FV	71.5	MV	FV	68.2
Lab	SVM	WN	FV	97.1	WN	FV	71.2	WN	FV	87.5	WN	FV	72.4
Lab		Retinex	FV	97.6	Retinex	FV	80.2	L_{max}	FV	87.9	RGBcb	FV	**74.4**
Lab2	1NN	WN	FV	86.2	WN	FV	70.4	WN	FV	72.5	WN	FV	68.4
Lab2		Retinex	FV	88.2	Retinex	FV	73.2	–	–	–	L_{max}	FV	69.1
Lab2	SVM	WN	FV	97.2	WN	FV	78.1	WN	FV	87.9	WN	FV	72.6
Lab2		Retinex	FV	**97.8**	Retinex	FV	79.9	–	–	–	RGBcb	FV	73.5
$I_1I_2I_3$	1NN	WN	FV	85.5	WN	FV	70.7	WN	FV	70.2	WN	FV	66.7
$I_1I_2I_3$		MV	FV	86.8	Retinex	FV	72.7	MV	FV	71.5	–	–	–
$I_1I_2I_3$	SVM	WN	FV	95.8	WN	FV	76	WN	FV	87	WN	FV	71
$I_1I_2I_3$		Retinex	FV	96.4	Retinex	FV	79.6	–	–	–	GWN	FV	71.5

Clas.: classifier, Norm.: normalization, Enc.: encoding method, Acc.: accuracy.

dataset VisTex, if color space RGB without normalization is considered, as it is usual, as a base line(94.1), normalization and color space transformation always improve accuracies for all cases, being RGB–CLAHE3 the best one (97.9). For Outex13, the accuracy 84.9 (RGB–WN) grows up to 86.2 (RGB–GWN). For USPTex, the accuracy 87 (RGB–WN) grows up 93.7 (RGB–CLAHE3) and 94.2 ($I_1I_2I_3$–CLAHE3). For Outex14, the accuracy 74.2 (RGB–WN) grows up to 74.8 (RGB–CLAHE3), 76 (Lab2–WN) and up to 77 (Lab2–GWN). In this case, the influence of normalization is less important than the rest of experiments (lower right column in Table 1) due to patch approach incorporates a normalization to zero mean and unit standard deviation. The normalization CLAHE3 is the best for all color spaces in datasets VisTex and UPSTex, where the texture is less uniform in the image, i. e. there are more intra-class differences than in Outex databases. The accuracies for VZ-PATCH approach are higher in three (in-bold in Table 3) of four datasets compared with the results in Table 1.

Table 4 shows the best accuracies reached by the DSIFT-XXX approaches with some normalization for all color space, datasets and classifiers 1NN or linear SVM. The vector DSIFT-FV with SVM is always better than DSIFT-BOVW and DSIFT-VLAD, but worse than VZ-PATCH. These results lead to the following conclusions: (1) the linear SVM always provides higher accuracies than 1NN; (2) the Fisher Vector (FV) always performs better than the other encoders (BOVW and VLAD); (3) frequently, the normalization, color space or both slightly improves the accuracies for all datasets; and (4) the vector VZ-PATCH is superior than DSIFT for all datasets, even using the linear SVM classifier.

In order to find the best combination of normalization, color space, texture description and classifier, Table 5 shows the average value, deviation, Friedman rank, position and p-value of a paired T-Test with respect to the best combination (lowest Friedman rank) over the datasets VisTex, Outex13, USPTex and Outex14. The best combination is the color space Lab2 with normalization CLAHE3 and VZ-PATCH texture feature. This approach fills the first 27 positions in the Friedman ranking combined with several color spaces and normalizations. The following combinations are DSIFT-FV using linear SVM, Local Binary Patterns (MLBP) using RForest classifier, and Wavelets (Daub4) using 1NN classifier. A paired T-Test comparing the first position in Friedman rank (Lab2-CHAHE3-VZ-PATCH) with the following vectors in the table shows that the difference between VZ-PATCH and the others is not statistically significant. However, we must take into account that: (1) the learning approaches use information from a training set in the calculation of the feature vector, and there are therefore dependent on this set. This does not happen with the remaining feature vectors, which do not use a training set for the feature calculation and do not depend on a specific dataset, being, in general, much faster. (2) The distance measure (χ^2) used by the 1-NN classifier in the VZ-PATCH approach is different to the other vectors, and this might be the cause of its good results. Besides, the learning approaches could not be applied

Table 5
Average accuracy, Friedman rank and p-value of a paired T-Test on datasets VisTex, Outex13, USPTex and Outex14.

Feature	Avg.	Std.	Rank	Pos.	p-value
Intensity feature					
Lab2, CLAHE3, VZ-PATCH	86.97	10.6	21.5	1	–
RGB, CLAHE3, VZ-PATCH	87.54	10.4	34.1	2	0.439
Lab, CLAHE3, VZ-PATCH	87.24	10.4	36.2	3	0.637
RGB, MV, VZ-PATCH	85.53	8.1	43.7	4	0.461
Lab, GWN, VZ-PATCH	85.45	8.4	47.7	5	0.450
HSV, CLAHE3, VZ-PATCH	85.89	12.3	54.1	6	0.365
Lab2, WN, VZ-PATCH	85.25	8.0	54.4	7	0.469
RGB, WN, VZ-PATCH	85.05	8.2	58.2	15	0.279
Lab, WN, VZ-PATCH	84.91	8.5	68.1	19	0.323
$I_1I_2I_3$, WN, VZ-PATCH	84.16	8.9	88	24	0.085
HSV, Retinex, DSIFT-FV, SVM	83.41	11.8	127.3	28	**0.024**
HSV, WN, VZ-PATCH	82.09	12.1	128.7	30	0.073
Lab, Retinex, DSIFT-FV, SVM	83.44	11.8	136.9	32	0.059
RGB, Retinex, DSIFT-FV, SVM	82.92	11.1	160.2	36	**0.021**
Lab2, WN, DSIFT-FV, SVM	83.94	10.9	173	40	0.142
Lab, RGBcb, MLBP, RFOREST	80.28	13.6	219.2	43	0.083
Lab2, RGBcb, MLBP, RFOREST	80.28	13.7	221.1	44	0.082
Lab2, WN, Daub4, KNN	80.45	14.8	224.5	48	0.174
RGB, RGBcb, GNI, RFOREST	80.53	10.4	241.1	53	0.140
Lab, GWN, Daub4, RFOREST	80.25	17.3	245.2	57	0.204
Lab, WN, Daub4, KNN	79.97	15.2	260.4	68	0.161
RGB, RGBcb, MLBP, RFOREST	79.45	14.3	290.9	97	0.066

Table 6
Classification accuracy of integrative color texture features for all the color spaces, normalizations and datasets.

Color	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.	Norm, Feat.	Acc.
	VisTex		CUReT		USPTex		ALOT	
RGB	WN, GNI,MLBP	96.2	WN, MLBP	95.5	WN, Haar	91.1	WN, MLBP	95.7
RGB	Chroma, MLBP	97.3	RGBib, MCOMS	97.2	L_{max}, daub4ME	**96.8**		
HSV	WN, MLBP	98.4	WN, MLBP	97.2	WN, Daub4	91.9	WN, MLBP	95.4
HSV	RGBib, MLBP	99.1	RGBib, Daub4ME	**97.7**	CLAHE3, Daub4	94.4	CLAHE3, MLBP	**95.8**
Lab	WN, GNI	99.1	WN, Daub4	94.8	WN, Daub4	96.3	WN, MLBP	94.7
Lab	Retinex–Daub4	98.8	RGBib, Daub4	97.2	HEQ, Daub4ME	**96.8**	RGBib, MLBP	94.8
Lab2	WN, MLBP	99.2	WN, Daub4	95.3	WN, Daub4	96.7	WN, MLBP	94.1
Lab2	L_{max}–Daub4	99.2	RGBib, Daub4	97.3	HEQ, Daub4	96.7	RGBib, MLBP	94.6
$I_1I_2I_3$	WN, Daub4ME	98.6	WN, Daub4	94.9	WN, Daub4	95.7	WN, MLBP	94.2
$I_1I_2I_3$	CLAHE3, GNI	**99.4**	MV, MCOMS	**97.7**	HEQ, Daub4	96.7		
	Outex13		Outex14 20s		Outex14 10s		Outex 14	
RGB	WN, GWN	**90.4**	WN, GWN	**90.2**	WN, Haar	**81.5**	WN, MCLBC	54.9
RGB							RGBcb, GWN	**71.4**
HSV	WN, Daub4	85.3	WN, GWN	86.6	WN, GWN	76.1	WN, MLBP	43
HSV	RGBib, Daub4	87	RGBib, Daub4ME	86.6	RGBib, Daub4ME	79.4	RGBcb, GWN	66.1
Lab	WN, HaarME	86	WN, GWN	86.4	WN, MFS	75	WN, LBP	49.4
Lab	RGBib, HaarME	88.8	CLAHE3, Haar	88.1	RGBib, Haar	80.4	HEW, GNI	66.9
Lab2	WN, MSDHC	86.3	WN, Daub4ME	84.9	WN, Daub4ME	75.5	WN, LBP	47.1
Lab2	RGBib, Haar	88.5	RGBib, Daub4ME	88.1	RGBib, Daub4ME	80.1	RGBcb, GNI	67.9
$I_1I_2I_3$	WN, MSDHC	83.5	WN, GWN	84.2	WN, GWN	72.3	WN, MLBP	42.8
$I_1I_2I_3$	RGBib, Haar	88	RGBib, Daub4ME	88	RGBib, Daub4ME	80.1	RGBcb, GWN	64.4

Table 7
Intensity texture and pure color features selected to form the parallel color texture feature vectors.

Intensity	Color
Lab2–RGBcb–MLBP	RGB–L_{max}–FOS
Lab–RGBcb–MLBP	RGB–WN–FOS
Lab2–WN–Daub4	Lab2–WN–FOS
Lab–WN–Daub4	RGB–CLAHE3–CCH21
RGB–L_{max}–Daub4	RGB–CLAHE3–CCH3
RGB–RGBcb–MLBP	RGB–CLAHE3–CCH9
Lab2–WN–MLBP	RGB–CLAHE3–CCH6
Lab–WN–MLBP	HSV–GWN–FOS
HSV–WN–Daub4	$I_1I_2I_3$–Chroma–FOS
RGB–WN–MLBP	HSV–RGBcb–JCH3
RGB–WN–Daub4	Lab–WN–FOS
RGB–HEQ–GNI	RGB–CLAHE3–PCTL9
RGB–RGBcb–Daub4	HSV–GWN–JCH21
HSV–WN–MLBP	HSV–GWN–JCH12
HSV–RGBcb–MLBP	HSV–CLAHE3–JCH15
	HSV–CLAHE3–JCH18
	HSV–HEQ–JCH12
	RGB-CLAHE3-FOS

to color images due to its huge computational cost, and to the absence of a version for color images.

4.3. Color texture classification by integrative and parallel approaches

The second experiment compares the integrative and parallel approaches for color texture classification, also using the 1NN classifier. We use the union of the intensity texture features for each color channel to implement the integrative approach. We compute the same 19 feature vectors as in the previous subsection, but now the number of features is three times the number of features for intensity images. Table 6 shows the best accuracies reached by some normalizations and color texture feature vectors for all color spaces and datasets. Considering as a base line the color space RGB without normalization, the use of normalization or other color space improves the performance, except to some datasets of family Outex. Specifically, for Vistex dataset, the accuracy increases from 96.2 (RGB–WN) up to 97.3 (RGB–Chroma) and up to 99.4 ($I_1I_2I_3$–CLAHE3). For dataset CUReT, the accuracy increases from 95.5 for RGB–WN up to 97.2 (RGB–RGBib or HSV–WN) and up to 97.7 (HSV-RGBib or $I_1I_2I_3$–MV). For dataset USPTex, the accuracy increases from 91.1 (RGB–WN) up to 96.8 (RGB–RGBib and Lab-HEQ). For dataset ALOT, the accuracy only increases from 95.7 (RGB–WN) up to 95.8 (HSV–CLAHE3). The color space RGB without normalization achieves the best accuracy in Outex13 (90.4), Outex14-20s (90.2) and Outex14-10s (80.5). Nevertheless, for dataset Outex14 the normalization and color space are again very important: the accuracy achieved for RGB without normalization (54.9) grows up to 71.4 using normalization RGBcb.

The parallel approach joins the intensity texture and the pure color features of the image. Since it is impossible to test all the combinations of intensity and color features vectors ($1140 \times 1380 = 1,573,200$ vectors), we chose the best intensity and pure color features (Table 2) and the vectors which provide the best accuracy for some dataset (Table 1). Overall we include the 15 intensity feature vectors and the 18 color feature vectors shown in Table 7, giving 270 vectors for each dataset. Table 8 shows the feature vector which achieves the best accuracy for each dataset and the average accuracy, Friedman rank and position for all datasets. The parallel approach achieves accuracies higher than the integrative approach (Table 6) in 7 out of the 8 dataset tested (in bold). The FOS color features, combined with different color spaces and normalizations, are the best for five datasets. In the remaining cases, the best results are achieved by color space HSV with features JCHXX.

In order to know what combination is globally (i.e., over the 8 datasets) the best, a Friedman test was developed considering all color texture feature vectors (1140 belonging to the integrative and 270 to the parallel approaches, respectively) on all datasets. Table 9 shows the Friedman rank and position, and p-value of a paired T-Test, of the top 10 for integrative and parallel approaches using the 1NN classifier. The parallel approach is clearly better than integrative approach, achieving the top-20 positions. In fact,

Table 8
The best classification accuracy of different parallel vectors for each dataset.

Dataset	Features		Acc.	Values for all datasets			
	Texture	Color		Avg.	Std.	Rank	Pos.
VisTex	HSV, RGBcb, MLBP	Lab2, WN, FOS	**99.5**	85.22	18.1	121.7	28
CUReT	RGB, RGBcb, Daub4	Lab2, WN, FOS	**98.3**	85.27	16.4	154.4	50
USPTex	RGB, HEQ, GNI	HSV, CLAHE3, JCH15	96.3	79.29	27.2	331.9	207
ALOT	Lab2, RGBcb, MLBP	RGB, L_{max}, FOS	**97.2**	88.50	11.3	52.8	2
Outex13	Lab2, WN, Daub4	RGB, WN, FOS	**90.6**	85.25	13.7	254.1	128
Outex14-20s	RGB, HEQ, GNI	HSV, RGBcb, JCH3	**91**	85.84	8.2	312.7	184
Outex14-10s	RGB, L_{max}, Daub4	Lab2, WN, FOS	**82.9**	84.65	20.0	161.4	57
Outex14	RGB, HEQ, GNI	HSV, RGBcb, JCH3	**73.1**	85.84	8.2	312.7	184

Table 9
Average accuracy and Friedman test on all datasets for integrative and parallel color texture classification using the 1NN classifier.

Features		Avg.	Rank	Pos.	*p*-value
Texture	Color				
Parallel approaches					
Lab2–RGBcb–MLBP	RGB–WN–FOS	88.72	51.9	1	–
Lab–RGBcb–MLBP	RGB–WN–FOS	88.58	53.2	3	0.33
RGB–RGBcb–MLBP	RGB–WN–FOS	88.35	75.1	5	0.088
RGB–WN–MLBP	RGB–WN–FOS	88.66	77.6	6	0.86
Lab2–WN–MLBP	RGB–L_{max}–FOS	88.11	77.7	7	0.12
Lab–WN–MLBP	RGB–L_{max}–FOS	88.16	79.9	8	0.11
Lab2–WN–MLBP	RGB–WN–FOS	88.31	82.6	9	0.18
Lab–WN–MLBP	RGB–WN–FOS	88.40	90.6	10	0.33
HSV–WN–MLBP	RGB–WN–FOS	87.90	93.9	13	0.20
HSV–WN–MLBP	Lab2–WN–FOS	86.31	96	14	0.31
Lab–RGBcb–MLBP	Lab2–WN–FOS	86.33	104.1	15	0.27
Lab2–RGBcb–MLBP	RGB–CLAHE3–PCTL9	87.95	105.1	16	0.12
Lab2–RGBcb–MLBP	Lab2–WN–FOS	86.59	107.7	17	0.30
RGB–RGBcb–MLBP	Lab2–WN–FOS	86.20	109.1	18	0.23
Lab–RGBcb–MLBP	RGB–CLAHE3–PCTL9	87.81	110.7	19	0.12
HSV–WN–MLBP	Lab–WN–FOS	86.16	112.6	20	0.26
RGB–RGBcb–Daub4	Lab2–WN–FOS	85.27	154.4	50	0.23
RGB–HEQ–GNI	RGB–WN–FOS	87.50	173.4	62	0.28
Integrative approaches					
Lab2, RGBib, Daub4		83.92	179.4	68	0.23
Lab2, RGBib, Haar		83.62	197.9	83	0.22
Lab2, RGBib, Daub4ME		83.78	200.9	87	0.21
HSV, RGBib, MLBP		83.31	205.4	88	0.14
Lab2, RGBcb, Daub4		85.26	224.2	100	0.052
Lab2, WN, Daub4		81.95	236.5	110	0.22
Lab, RGBib, Daub4		82.59	236.6	112	0.18
HSV, GWNI, MLBP		83.01	240.1	114	0.092
Lab, RGBib, Daub4		82.49	244.7	120	0.19
$I_1I_2I_3$, RGBib, Daub4		82.38	248.2	124	0.17
HSV, WN, MLBP		82.89	254.6	129	0.092
Lab2, RGBib, MLBP		82.36	257.7	131	0.23
Lab, RGBib, GNI		81.60	323.3	197	0.57

only four integrative vectors are among the first 100 features vectors. For the parallel approach and in relation with color features, the top-10 includes only FOS color features combined by color space RGB with L_{max} normalization or without normalization. The color features CHPxx (Color Histogram Percentiles) appear the first time in the position 16. In relation with intensity texture features, the first 20 positions are occupied by MLBP features combined with different color spaces and with normalization RGBcb, CLAHE3 or without normalization. The best color spaces seem to be Lab2, Lab and RGB. Among wavelet and Gabor feature vectors, Daub4 intensity texture vector appears for the first time in the position 50, and the GNI vector in position 62, even better than integrative approaches. All these results lead to the conclusion that the best performance is achieved using luminance–chrominance color spaces (Lab2 or Lab) to compute the intensity texture features joint to primary color spaces (RGB) to compute the pure color features. For integrative approaches, the first 10 positions are for texture features of the wavelet family with normalization RGBib (except position 4 and 8 of texture features MLBP) and luminance–chrominance color spaces (Lab or Lab2), except the positions 4, 8 and 10, of color space HSV or $I_1I_2I_3$.

As reported by other comparative analysis [9,8], the comparison of Tables 1 (parts labelled by intensity features) and 6 or 8 suggests that color has an important contribution in the discrimination power of texture images. The best accuracy raises for all datasets when considering the color features (in both parallel and integrative approaches), except for Outex14-20s: from 96.6 (in Table 1) to 99.5 (the best of Table 6 or 8) for VisTex, from 95.6 to 98.3 (CUReT), from 83.5 to 96.8 (USPTex), from 93.6 to 97.2 (ALOT), from 87.4 to 90.6 (Outex13), from 82.4 to 82.9 (Outex14-10s) and from 69.8 to 73.1 (Outex14). Nevertheless, the differences between the best vector and the remaining ones in the table are no significant ($p > 0.05$).

4.4. Elapsed time

The experiments were executed on the computational cluster of our research center (CiTIUS) which is composed of 8 HP Proliant BL685c G7 nodes, 5 Dell PowerEdge M910 nodes and 5 Dell PowerEdge M620 nodes. Each HP Proliant BL685c G7 node is equipped by 4 processors AMD Opteron 6262 HE (16 cores, 1.6 GHz) and 128 GB RAM memory. Each Dell PowerEdge M910 is equipped with 2 processors Intel Xeon L7555 (8 cores, 1.87 GHz, 16 threads per core) and 64 GB RAM memory. Each Dell PowerEdge M620 node is equipped by 2 processors Intel Xeon E5-2650L (8 cores, 1.8 GHz, 16 threads per core) with 64 GB RAM. The jobs were sent to the queue without specifying the processors. We measure the elapsed time of the normalization step, color space transformation and feature computation. The elapsed time is averaged over all the simulations and normalized by the number of images and image size. For the transformation from RGB to other color spaces, the elapsed time is 0.24 μs for HSV, 1.19 μs for $I_1I_2I_3$, 1.46 μs for Lab and 4.02 μs for Lab2 (which in the implementation that we used requires to transform to Lab space first). Once the image is transformed to a given color space, the calculation of the feature vectors takes similar times in all the color spaces, because the data have the same size. The computation times of the different intensity texture and pure color features for each image pixel are shown in Figs. 2 and 3 respectively. For intensity texture features, the elapsed times sorted in ascending order in microseconds are: LBP (0.5), CLBC (0.6), COMS (0.9), HaarME (1.4), Daub4ME (1.4), MFS (1.9), MLBP (2.3), SDHC 2.6), MCLBC (3.2), Haar (3.3), Daub4 (4.0), GWN (10.4), DTCWT (11.5), GNI (13.3), GNRI (13.6), MCOMS (19.5), MSDHC (55) and MFP (103.2). In general, the non-multiresolution statistical methods are the fastest ones, followed by multiresolution statistical methods, wavelets and Gabor filters. For the pure color features, the fastest method is CM with 0.99 μs on average per image pixel, followed by CCHXX features with 1.24–1.34 μs, CHPXX with 1.47–1.60 μs, FOS with 2.11 μs and JCHXX between 4.49 μs and 4.81 μs.

The elapsed time of learning methods includes the model generation, train and test pattern calculation and classification. For DSIFT descriptors, the model generation includes the train encoder, PCA projection and encoding process. We measure the

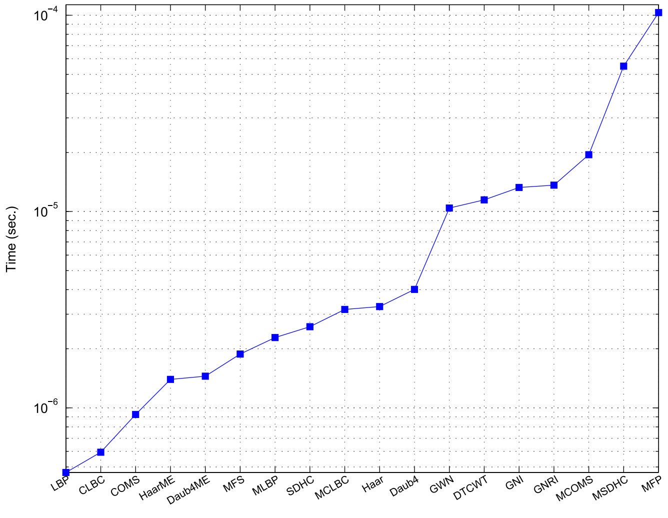

Fig. 2. Elapsed time for intensity texture features averaged on all datasets per image pixel.

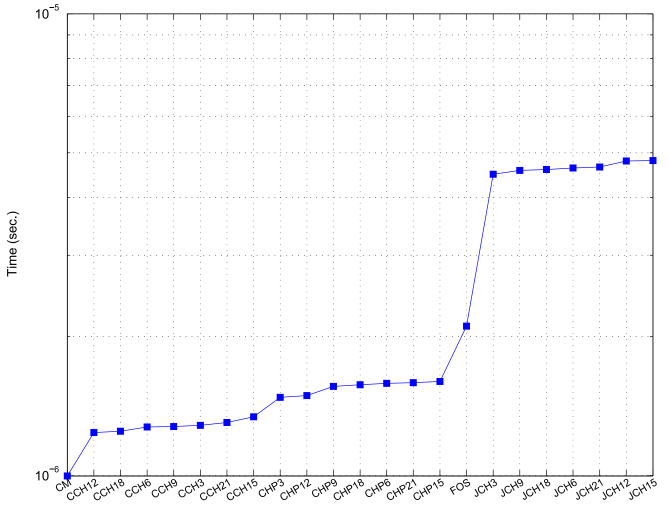

Fig. 3. Elapsed time for pure color features averaged on all datasets per image pixel.

Table 10
Elapsed time (in microseconds per pixel) for learning methods in gray level images averaged over datasets VisTex, Outex13 and USPTex.

Stages	Learning methods			
	VZ-PATCH	DSIFT-BOVW	DSIFT-FV	DSIFT-VLAD
Model generation	2.690	82.9	8.4	6.1
Pattern computation	140	10.9	6.1	5.5

elapsed time for color space RGB without normalization in order to remove the time needed to normalization or color space transformation. Table 10 shows the average elapsed time per pixel for learning methods normalized by number of images and image size. The fastest method is DSIFT-VLAD, which needs 11.6 μs/pixel (6.1 to model generation plus 5.5 for train and test texture patterns calculation), which is similar to some of the slowest direct methods. Since our experiments use 4-Fold cross validation, we must repeat this process 4 times, so the time needed to process one pixel image is multiplied by 4 resulting in methods much slower than direct ones. The DSIFT-FV spends times similar to DSIFT-VLAD, but they are much faster than DSIFT-BOVW (one magnitude order) and VZ-PATCH (2 magnitude orders).

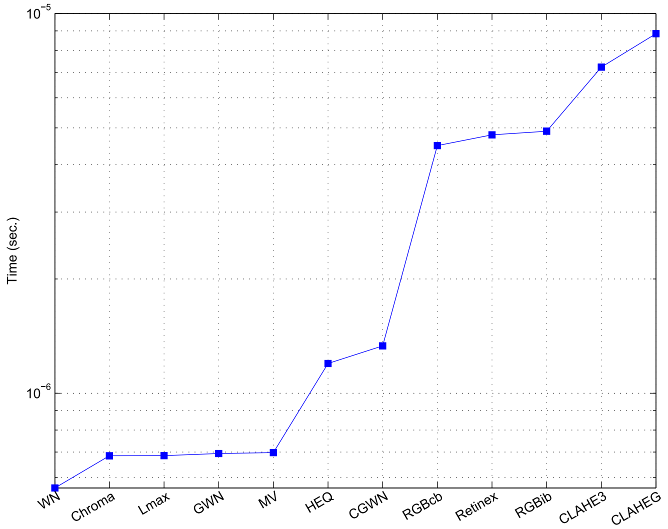

Fig. 4. Elapsed time of normalization methods averaged on all datasets per image pixel.

Fig. 4 shows the elapsed time per image pixel for the normalization methods, ranging from 0.7 μs for Chroma up to 8.8 μs. for CLAHEG. They can be grouped in three intervals: the fastest are Chroma, L_{max}, GWN and MV (about 0.7 μs.), the intermediate methods are HEQ and GWNI (about 1.3 μs), and the slowest methods RGBcb, RGBib, Retinex, CLAHE3 and CHAHEG.

The elapsed time for color texture classification using integrative approaches accounts for the normalization step, color space transformation and three times the calculation of intensity texture features. The time depends on the method used in each step ranging from 1.5 μs (RGB–WN–LBP) to 322.6 μs (Lab2–CLAHEG–MFP). The elapsed time for parallel approaches accounts the normalization step, color space transformation, and the computation of pure color and intensity texture features. This times ranges from 1.49 μs (RGB–WN–CM–LBP) to 120.3 μs (Lab2–CLAHEG–JCHXX–MFP). For both integrative and parallel approaches the ranges are very extensive and they overlap, but comparing the elapsed time for the best scores, we can conclude that parallel are faster than integrative approaches. For example, for the best integrative approach, the time is: $20.92\ \mu s = 4.02$ Lab2 $+ 4.9$ RGBib $+\ 3 \times 4.0$ Daub4, while the best parallel one spends $13.03\ \mu s = 4.02+4.5+2.3+0+0+2.11$ Lab2–RGBcb–MLBP–RGB–WN–FOS. Another example is RGB–WN–MLBP RGB–WN–FOS (parallel, 4.31 μs, sixth position in the Friedman rank) is faster than RGB–WN–Daub4 (integrative, 12 μs, position 202 in the Friedman rank).

4.5. Discussion about color space

In order to discuss the best performing color space, we developed a Friedman test on pure color, intensity texture and color texture (using integrative approach) features, selecting the feature vector with the lowest Friedman rank for each color space. The average accuracy on all datasets for these selected vectors are represented in Fig. 5. The behavior of color space depends on the type of features: RGB is the best space for color texture and pure color features while Lab2 is the best for intensity texture features. The accuracy is almost constant for color texture features (green line) ranging from 83.1 for color space $I_1I_2I_3$ up to 84.9 of RGB. For intensity texture features, the best accuracy is achieved by color spaces Lab2 (83.6) and Lab (83.1), followed by $I_1I_2I_3$ (81.1) and RGB (80.9), being HSV the worst (77). The good results of Lab2 and Lab suggest that transforming to these spaces enhances the texture information and helps the texture classification. The accuracies for

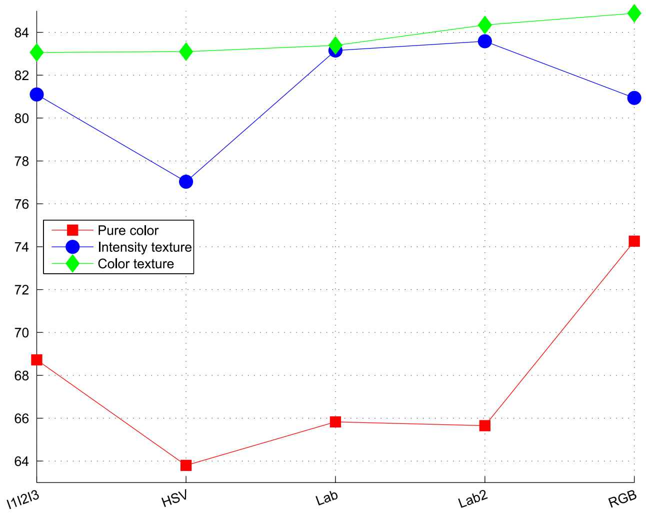

Fig. 5. Highest accuracy achieved by each color space on all datasets.

pure color features change very much with the color space, ranging from 63.8 for HSV up to 74.3 for RGB.

4.6. Influence of image normalization

The feature vectors with the highest Friedman rank for each normalization and feature group are selected to discuss the influence of image normalization on color texture discrimination. Fig. 6 shows the average accuracy on all datasets for the previous vectors. The behavior also depends on the type of features: WN (without normalization) achieves the highest average accuracy for pure color (although L_{max} gets the lowest Friedman rank), L_{max} for color texture features and RGBcb for intensity texture features. For texture features (intensity or color texture), the accuracy is almost constant with normalization, except for CLAHEG and Chroma, which are clearly the worst methods. The accuracy ranges from 80.1 (RGBib-intensity features) and 81.3 (Retinex-color texture) up to 83.6 (RGBcb-intensity) and 85.2 (L_{max}-color texture), and it is very near to WN with 82.6 (intensity) and 83.1 (color texture). For pure color features, the best methods are L_{max} (74.3) and WN or GWN (74.9), followed by Chroma (72.1), being the remaining methods clearly the poorest ones, with accuracies below 68.

A paired *T*-Test comparing the best vector with the remaining ones for each set of feature vectors provides the following results: (1) for pure feature vectors, the normalization L_{max} is not significantly better ($p < 0.05$) than WN and GWN, but it is significantly better than the remaining normalizations; (2) for color texture features, the normalization RGBib shows only significant differences with Chroma and CLAHEG; and (3) for intensity texture features, the normalization RGBcb shows a significant difference with Chroma, HEQ and MV.

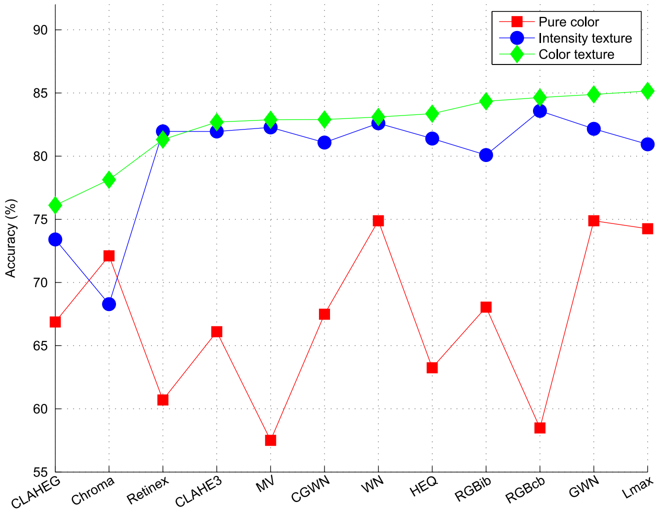

Fig. 6. Highest accuracy achieved by each normalization method on all datasets.

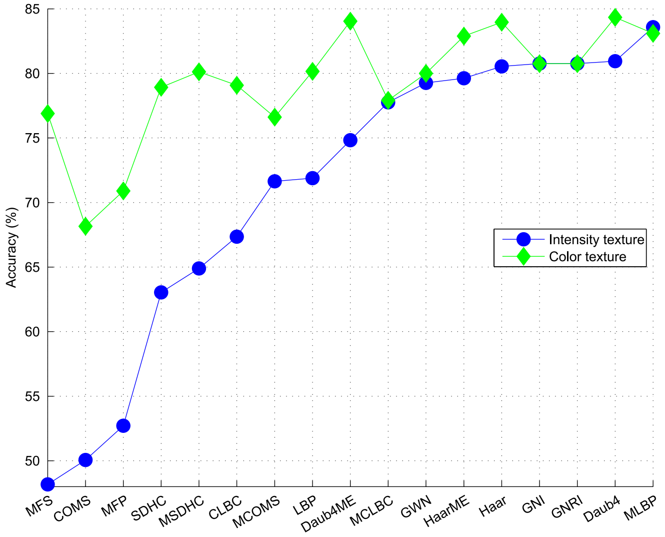

Fig. 7. Accuracies of the best texture features on all datasets. The color of the line shows the type of features considered: intensity texture (blue) and color texture features using the integrative approach (green). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

4.7. Discussion about feature vectors

In order to compare the performance of different families of texture features, we select the vectors with the lowest Friedman rank for each texture feature (in intensity texture and color texture classification problems). Fig. 7 shows the average accuracy on all datasets for each selected vector, in ascending order of accuracy for intensity texture vectors. The best texture vector for intensity texture classification is MLBP (accuracy 86.49), followed by the vectors of spectral analysis based on wavelets and Gabor transforms. The MLBP is clearly better than the other vectors of the local binary pattern family (LBP, MCLBC and CLBC), and than the remaining vectors with statistical (MCOMS, COMS, MSDHC and SDHC) or fractal (MFS and MFP) features. A paired *T*-Test comparing the MLBP vector with the remaining ones shows a significant difference with all of them except Daub4, Daub4ME, GWN, GNI and GNRI. For color texture classification, the feature vectors of the wavelet family (Daub4, Daub4ME, Haar and HaarME) exhibit the best performance, followed very close by MLBP. The Gabor features, LBP and MSDHC vectors are in the middle of the interval and the remaining vectors are clearly worse. A paired *T*-Test comparing the Daub4 vector with the remaining ones shows that it is only significantly better than fractal approaches (MFS and MFP), non-multiresolution statistical features (SDHC, COMS, CLBC), HaarME and GWN.

4.8. Analysis by datasets

Fig. 8 compares the effect of normalization, color space or both in the performance improvement for each dataset. As mentioned, if the color space RGB without normalization (RGB–WN) is considered as a base line, the use of another color space,

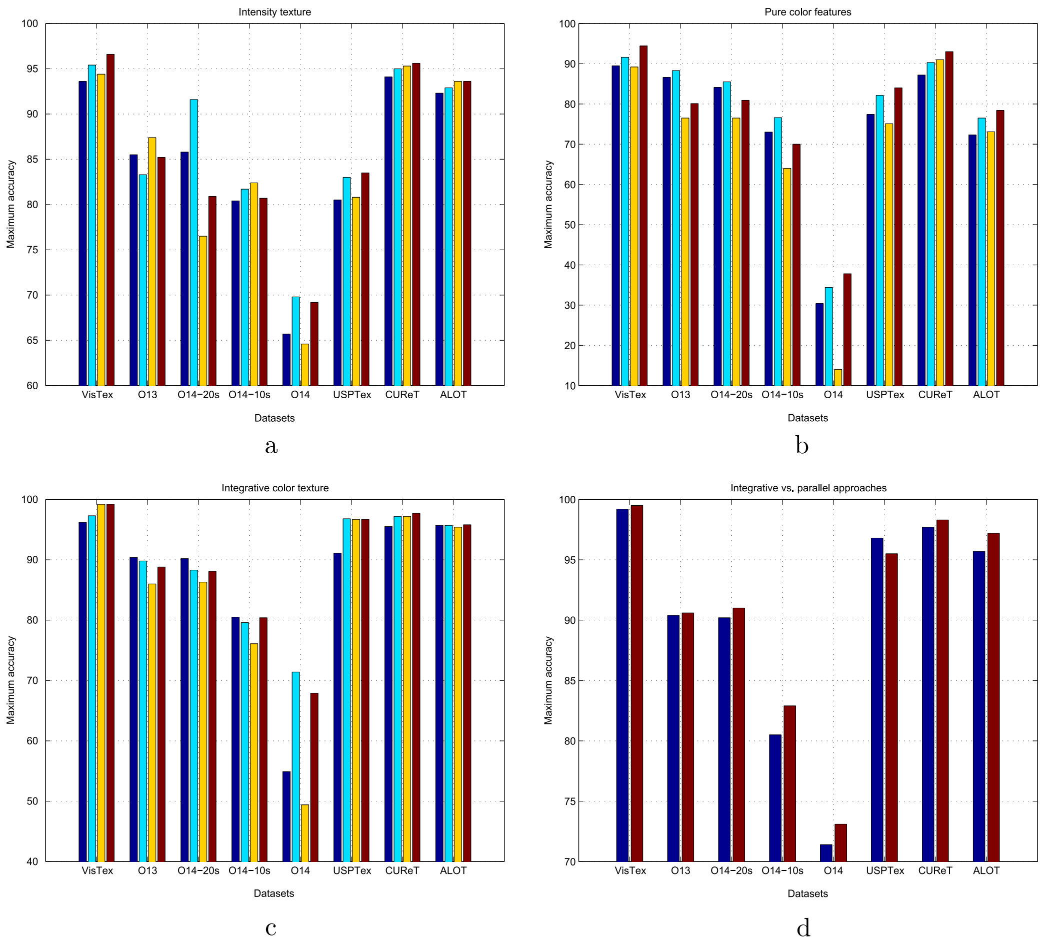

Fig. 8. Comparison of the maximum classification accuracy for each dataset and intensity texture features (a), pure color features (b) and integrative color texture features (c): RGB–WN (dark blue), RGB + any normalization (blue), other color space without normalization (yellow) and color space with normalization (brown). Graph d compares integrative (blue) vs. parallel (brown) color texture classification approaches for every dataset. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

normalization or both very often improves the classification for intensity texture features (graph a), pure color features (graph b) and integrative color texture features (graph c), except for the Outex datasets. For Outex13 (referred as O13 in the graphs), where the acquisition conditions are very controlled and constant, the RGB–WN normally provides the best performance for color texture features (graph c). This behavior is extrapolated to datasets Outex14-20s (referred as O14-20s) and Outex14-10s (referred as O14-10s) because of, although images come from 3 different illuminants, there are images of all types of illuminants in the training set. The number of patterns in the training set seems to be very important to represent the problem. Hence, the performance for Outex13 (20 images per class and one illuminant) and the dataset Outex14-20s (20 samples per class for each of the 3 illuminants) is very similar for intensity (graph a), pure color (graph b) and integrative color texture (graph c), independently of the number of illuminants. A different case is found with Outex14-10s, which has half samples of Outex14-20s, and its performance decreases nearly 10 points for intensity, pure color and color texture classification (labels O14-20s and O14-10s in graphs a, b and c). Outex14 has equal samples than Outex14-10s, but training with the samples of Inca illuminant and testing with samples of the other two illuminants. In this cases, the normalization, color space or both are of vital importance to increase the performance (see dark blue bar in comparison with the other bars for label O14 in graphs a, b and c). Nevertheless, the maximum accuracy achieved is always rather lower than Outex14-10s: intensity 69.8 (O14) vs. 82.4 (O14-10s) in graph (a), pure color 37.8 (O14) vs. 76.5 (O14-10s) and color texture 71.4 (O14) vs. 80.5 (O14-10s) in graph (b). This fact leads to the conclusion that the preprocessing step is necessary when the images are acquired under variant conditions in order to attenuate this effect if this variation is not represented in the training set (case of dataset Outex14). Nevertheless, the preprocessing step is not able to completely remove the effects of variations in the acquisition conditions. CUReT and ALOT databases are obtained by varying viewing and illumination conditions. In both cases, there are many images per class, so probably some of them are acquired under very similar conditions. Therefore, the preprocessing step improves the performance (see dark blue bar in comparison with the other bars for labels CUReT and ALOT in graphs a, b and c), but the improvement is not so dramatic as in Outex14 because for a test pattern there are similar patterns in the training set. This reasoning is also valid for datasets VisTex and USPTex, where the image acquisition conditions are not controlled, but the samples for each class are obtained splitting an image into several non-overlapping sub-images. In this case, the acquisition conditions among the sub-images of a class are constant, although they may change among images. The graph (d) of Fig. 8 compares integrative and parallel approaches for color texture classification. Parallel approaches provide better results for all datasets, except for

Table 11
Average accuracy, Friedman rank and *p*-value of a paired *T*-Test on all datasets using Random Forest classifier for intensity and pure color features.

Feature	Avg.	Std.	Rank	Pos.	*p*-value
Intensity feature					
Lab, RGBcb, MLBP	83.70	10.7	60.3	1	–
Lab2, RGBcb, MLBP	83.61	10.9	11.0	2	0.50
Lab, GWN, Daub4	82.21	12.0	71.75	3	0.47
Lab, Lmax, Daub4	82.11	11.9	76.3	4	0.48
Lab2, WN, Daub4	82.11	12.2	80.4	5	0.52
RGB, Lmax, Daub4	82.08	10.7	85.2	8	0.41
RGB, RGBcb, MLBP	83.08	11.3	92.2	9	0.060
Lab2, CLAHE3, Daub4	82.58	10.1	95.9	10	**0.050**
Lab, WN, MLBP	82.79	10.7	101.2	13	**0.039**
Lab, WN, Daub4	80.97	13.2	101.9	14	0.26
HSV, WN, MLBP	82.90	11.3	110.3	22	0.14
RGB, MV, DTCWTVH	79.62	10.6	161.3	96	0.15
RGB, RGBib, GNI	79.76	9.6	163.7	98	0.19
Color feature					
RGB, GWNI, FOS	78.15	16.6	45.6	1	–
RGB, WN, FOS	77.85	16.8	48.0	2	0.18
Lab2, GWN, FOS	74.62	16.8	61.1	3	0.10
RGB, Lmax, FOS	76.86	15.0	69.3	4	0.17
Lab, GWN, FOS	73.25	17.2	73.6	5	**0.045**
HSV, RGBcb, JCH6	69.86	18.1	95.7	7	0.060
HSV, RGBcb, JCH9	69.80	18.3	95.9	8	0.062
$I_1I_2I_3$, WN, FOS	75.45	25.7	99.6	9	0.46
HSV, RGBcb, JCH3	69.80	17.8	101	10	0.081
Lab, Lmax, FOS	69.46	18.8	101.4	11	0.051
HSV, HEQ, JCH21	70.41	20.0	103.8	12	0.18
HSV, HEQ, JCH3	70.41	20.0	104.1	13	0.18
HSV, HEQ, JCH18	70.54	19.3	106.1	15	0.19
RGB, CLAHE3, CCH9	70.24	15.8	106.4	16	**0.022**
Lab, GWNI, FOS	75.64	27.0	108.5	18	0.53
RGB, CLAHE3, CCH6	69.88	16.5	108.6	19	**0.018**
Lab, WN, FOS	75.7	27.1	109	20	0.54

Table 12
Average accuracy, Friedman rank and *p*-value of a paired *T*-Test all datasets for integrative and parallel color texture classification using Random Forest.

Features		Avg.	Rank	Pos.	*p*-value
Texture	Color				
Parallel approaches					
Lab2–RGBcb–MLBP	RGB–WN–FOS	90.67	88.2	1	–
Lab–RGBcb–MLBP	RGB–WN–FOS	90.62	90	2	0.64
RGB–HEQ–GNI	Lab–WN–FOS	89.67	90.3	3	0.36
RGB–RGBcb–MLBP	RGB–WN–FOS	90.62	96.4	4	0.68
Lab2–RGBcb–MLBP	Lab–WN–FOS	88.83	98.5	5	0.35
Lab2–RGBcb–MLBP	$I_1I_2I_3$–Chroma–FOS	89.22	100.6	6	0.30
HSV–WN–MLBP	Lab–WN–FOS	88.81	103.4	7	0.34
RGB–WN–MLBP	RGB–WN–FOS	90.74	103.8	8	0.83
RGB–WN–MLBP	$I_1I_2I_3$–Chroma–FOS	89.17	104.4	9	0.26
Lab–RGBcb–MLBP	Lab–WN–FOS	88.84	104.7	10	0.33
RGB–HEQ–GNI	RGB–WN–FOS	91.88	105.1	11	0.30
RGB–RGBcb–Daub4	$I_1I_2I_3$–Chroma–FOS	89.25	105.5	12	0.45
HSV–WN–MLBP	$I_1I_2I_3$–Chroma–FOS	89.22	107.5	13	0.31
RGB–HEQ–GNI	Lab2–WN–FOS	89.12	108.5	14	0.32
Lab–RGBcb–MLBP	$I_1I_2I_3$–Chroma–FOS	88.97	109.4	15	0.30
Lab–WN–MLBP	RGB–WN–FOS	90.69	109.8	16	0.95
Lab–WN–MLBP	$I_1I_2I_3$–Chroma–FOS	89.04	110.6	18	0.27
Lab2–WN–MLBP	RGB–WN–FOS	90.53	113.8	19	0.33
RGB–RGBcb–Daub4	Lab–WN–FOS	88.58	115.1	20	0.39
Integrative approaches					
Lab2, RGBib, Daub4		85.67	216.2	112	0.24
Lab, RGBib, Daub4		84.76	247.9	148	0.25
Lab2, RGBcb, GNI		88.53	251.5	152	0.073
$I_1I_2I_3$, RGBib, Daub4		84.55	252.1	154	0.21
Lab2, RGBib, Haar		84.85	252.4	156	0.20
Lab2, RGBcb, GNI		88.85	253.2	157	0.11
Lab, RGBcb, GNI		88.25	267.2	173	**0.041**
Lab, HEQ, GNI		88.64	275.8	181	0.10
Lab2, WN , Daub4		84.44	281.6	187	0.19
Lab2, RGBib, Haar		83.92	282.1	188	0.22
Lab2, RGBcb, Daub4		87.33	288.7	190	**0.020**

USPTex. Recent advances in psychophysiology [57,58] show that color and texture are processed independently in the brain, even in different zones of the medial occipitotemporal cortex. This human perception evidence is in accordance with the separated computation of color and texture (parallel approaches).

4.9. Comparison between 1NN and Random Forest

To find out whether the conclusions depend on the classifier, we repeated all the experiments using a Random Forest (RF) classifier instead of 1NN. We used the RF implementation provided by the randomForest R package [59] with default parameter values: 500 trees in the forest, and the number of inputs randomly sampled equal to the square root of the number of inputs. Table 11 shows the average accuracy, Friedman rank and a paired *T*-Test on all datasets using RF for different intensity texture and pure color features. Comparing to 1NN results (Table 2), the four best vectors are almost the same (excepting minor changes in normalization) for intensity and pure color features. The best intensity texture vector (Lab, RGBcb, MLBP) is only significantly better than Lab2, CLAHE3, Daub4 and Lab, WN, MLBP (10th and 13th positions, respectively). The best pure color vector (RGB, GWNI, FOS) is significantly better than vectors Lab, GWN, FOS and CCHxx. Many vectors in the following positions are similar for 1NN and RF. The best color spaces (Lab, Lab2 and RGB) and texture features (MLBP, Daub4 and GNI) are also the same.

Table 12 shows the average accuracy, Friedman rank and a paired *T*-Test on all datasets using Random Forest for different integrative and parallel color texture features. The parallel are again better than integrative approaches, whose best position is 112. Similarly to 1NN, within parallel approaches the MLBP texture features occupies the first positions, followed by Daub4 and GNI, combined with FOS for pure color features. However, the differences between the best combination and the 20 first vectors are not significant. Among color spaces, Lab2, Lab and RGB occupy the best positions, while HSV continues in position 13. The behavior of integrative approaches is also similar to 1NN. These issues confirm that these vectors provide the easiest classification problem independently of the classifier used (1NN or RF).

4.10. Comparison between learning and direct methods

The results showed in Section 4.2 lead to the conclusion that learning methods classify gray level texture images better than direct methods. The main drawback of learning methods is their high computational cost (see Section 4.4). Hence, the elapsed time needed to compute the best parallel color texture feature Lab2–RGBcb–MLBP–RGB–WN–FOS is 13.02 μs, similar to a good performance learning method as DSIFT-FV (8.4 + 6.1 = 14.5 μs) and much faster than the first method VZ-PATCH (2830 μs, see Table 10). Then, we do a new experiment comparing all methods on the four smallest datasets, independent of classifier and type of features in order to known which is the best solution for color image classification. Table 13 shows the highest accuracies (in bold) provided by a combination feature-vector and classifier in the four smallest datasets. For three out of four datasets the parallel approaches using RF classifier are the best and only for dataset Outex14 the VZ-PATCH is the best. In order to analyze which

Table 13
Best accuracies in the four smallest datasets. The specific DM can be located in Table 1 for gray level and Tables 6 and 8 for color texture.

Features		Datasets			
Type	Classifier	VisTex	Outex13	USPTex	Outex14
Gray level features					
DM	1NN	96.6	87.4	83.5	69.8
DM	RF	94.8	87.6	85.2	69.1
VZ-PATCH		97.9	86.2	94.2	**77**
DSIFT-FV	SVM	97.8	80.7	88	74.4
Color texture features					
DM	1NN	99.5	90.6	96.3	73.1
DM	RF	**99.7**	**92.3**	**98.2**	76.3

DM: direct methods.

Table 14
Average accuracy, Friedman rank and p-value of a paired T-Test on datasets VisTex, Outex13, USPTex and Outex14 comparing direct and learning methods for different classifiers.

Features		Class.	Avg.	Rank	Pos.	p-value
Texture	Color					
Parallel approaches						
RGB–HEQ–GNI	HSV–RGBcb–JCH15	RF	89.47	99.2	1	–
RGB–HEQ–GNI	RGB–WN–FOS	RF	90.33	108	2	0.58
Lab2–RGBcb–MLBP	RGB–WN–FOS	RF	88.35	122.1	3	0.67
Lab–RGBcb–MLBP	RGB–WN–FOS	RF	88.15	130	4	0.60
RGB–RGBcb–MLBP	RGB–WN–FOS	RF	88.10	139.2	5	0.59
RGB–WN–MLBP	RGB–WN–FOS	RF	88.35	147.6	7	0.59
Lab2–RGBcb–MLBP	RGB–WN–FOS	1NN	87.08	162.7	11	0.30
Learning methods						
RGB, CHAHE3,VZ-PATCH			87.54	245.1	68	**0.039**
Lab2, CLAHE3,VZ-PATCH			86.97	269	83	**0.033**
Lab, CLAHE3,VZ-PATCH			87.24	277.9	89	**0.029**
HSV, CLAHE3,VZ-PATCH			85.89	321.1	141	**0.046**
$I_1I_2I_3$, CLAHE3,VZ-PATCH			86.48	350.5	182	**0.050**
Lab2, WN,VZ-PATCH			85.25	286.9	218	0.17
Lab2, Retinex,DSIFT-FV		SVM	83.56	494.2	335	**0.045**

is the best solution over all the datasets, Table 14 shows the average accuracy, Friedman rank, position and p-value respect with the first vector comparing the combination for parallel approaches, learning methods and direct methods on gray level images. The first 67 positions are for parallel approaches, almost all using RF classifier. Hence, the first combination using 1NN classifier is in position 11. Among parallel approaches the difference with the first one are not significant (p-value >0.05). The first position for learning methods is 68 (VZ-PATCH) and 335 (DSIFT-FV). In most cases, the differences with the best combination (RGB–HEQ–GNI HSV–RGBcb–JCH15) is significant.

5. Conclusions and future work

Color texture classification plays an important role in real world computer-vision applications, where the images are acquired under different ambient illuminations. The current work tests different normalization algorithms to achieve illuminance invariance, combined with different color spaces, and different color and texture features. We also compare the integrative and parallel color texture classification approaches on the 6 most popular public color texture datasets. The experiments outlined illustrate that parallel approaches are superior to integrative approaches for color texture classification. For parallel approaches, the best options use different color spaces and normalization to compute the intensity texture and pure color features: specifically, the illuminance–chrominance color spaces (Lab or Lab2) combined with Local Binary Patterns (vector MLBP) and normalization RGBcb to compute the intensity texture features, and the color space RGB combined with First Order Statistics (FOS) and normalization L_{max} to compute pure color features. For integrative approaches, the luminance–chrominance color spaces (Lab and Lab2) combined with wavelet texture features and normalization RGBib achieve the best performance. Regarding computational efficiency, the parallel vectors are also faster than the integrative vectors.

The behavior in relation to texture features depends of the type of approach used to compute the color texture features. Hence, the features based on Local Binary Patterns are the best among parallel approaches, which compute texture features on a intensity version of the color image, while the wavelet family features provide the best results for integrative approaches. The good results of MLBP and wavelet feature vectors are due to their ability to capture efficiently the local spatial relation defined by a texture and to discriminate between different relations. Gabor filters provide intermediate results for both approaches, and fractal features work very bad in both cases. From the elapsed time viewpoint, MLBP and Daub4 are relatively fast, only four times slower than the fastest method (LBP, which works bad), while Gabor filter are twenty six times slower. In relation to pure color features, the discrimination power of FOS (First Order Statistics) is superior to the features based on histograms (Table 2). When the pure color features are used in parallel approaches, a paired T-Test shows that FOS is significantly better than the histogram features. The elapsed time of FOS vector is in intermediate, much faster than JCHXX and slower than CCHXX and CHPXX.

Regarding the normalization of a color image before transforming to other color space, the color balance (RGBcb) in the parallel approach and the intensity balance (RGBib) in the integrative approach seem to be superior to other normalizations, although the differences with respect to absence of normalization are not statistically significant, and they are the slowest ones. Nevertheless, this results might be due to the fact that the datasets are acquired under very controlled conditions (Outex, ALOT and CUReT datasets) or maximizing contrast (VisTex and USPTex), so that the training set contains samples with the same, or very close, acquisition conditions as the test set. Hence, for dataset Outex14, where the training set do not contain images with the illuminance of testing test, the normalization increases the accuracy nearly 5 points for intensity texture classification for all color spaces, from 4.4 up to 31.9 points for pure color and more than 16 points for color texture classification and all color spaces (see the last column in Tables 1 and 6).

Although the classification was developed using a 1-Nearest Neighbor classifier, the conclusions of our study have been confirmed by further experiments using a Random Forest classifier. Among learning methods, VZ-PATCH gives the best results on the four smallest datasets (see Section 4.2) for intensity texture classification, but its high computational cost avoided its application to the four bigger datasets. However, for color image classification (see Section 4.10) the direct methods achieve better performance than learning methods on gray level images with much lower computational cost.

A future research should: (1) develop further insights into the application of learning methods for color texture images; (2) study the influence of normalization on ill-defined problems, underexposed, overexposed, different illuminance, scales and view angles images, without incorporating similar samples in the training set; (3) analyze the influence of the “clipping” percentage (5% of pixels in this work) in normalization methods RGBcb and RGBib, which gave the best results; and (4) extend this comparison applying convolutional neural networks for image classification.

Acknowledgments

We thank the financial support to Spanish Ministry of Science and Innovation (MICINN) under projects TIN2012-32262 and TIN2011-22935.

References

[1] M. Petrou, P. García-Sevilla, Image Processing: Dealing With Texture, Wiley, England, 2006.
[2] H.Y.T. Ngan, G.K.H. Pang, N.H.C. Yung, Automated fabric defect detection—a review, Image Vis. Comput. 29 (2011) 442–458.
[3] C. Zheng, D.-W. Sun, L. Zheng, Recent applications of image texture for evaluation of food qualities—a review, Trends Food Sci. Technol. 17 (2006) 113–128.
[4] E. Cernadas, P. Carrión, P.G. Rodríguez, E. Muriel, T. Antequera, Analyzing magnetic resonance images of Iberian pork loin to predict its sensorial characteristics, Comput. Vis. Image Underst. 98 (2005) 345–361.
[5] C.-H. Lin, R.-T. Chen, Y.-K. Chan, A smart content-based image retrieval system based on color and texture feature, Image Vis. Comput. 27 (2009) 658–665.
[6] Y. Song, W. Cai, W. Zhou, D.D. Feng, Feature-based image patch approximation for lung tissue classification, IEEE Trans. Med. Imaging 33 (4) (2013) 797–808.
[7] D. Huang, C. Shan, M. Ardabilian, Y. Wang, L. Chen, Local binary patterns and its applications to facial image analysis: a survey, IEEE Trans. Syst. Man Cybern.—Part C: Appl. Rev. 41 (6) (2011) 765–781.
[8] A. Drimbarean, P.F. Whelan, Experiments in color texture analysis, Pattern Recognit. Lett. 22 (2001) 1161–1167.
[9] U. Kandaswamy, S.A. Schuckers, D. Adjeroh, Comparison of texture analysis schemes under nonideal conditions, IEEE Trans. Image Process. 20 (8) (2011) 2260–2275.
[10] C. Palm, Color texture classification by integrative co-occurrence matrices, Pattern Recognit. 37 (2004) 965–976.
[11] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, R. Domínguez-Petit, Exhaustive comparison of color texture features and classification methods to discriminate cells categories in histological images of fish ovary, Pattern Recognit. 46 (2013) 2391–2407.
[12] U. Kandaswamy, D. Adjeroh, S.A. Schuckers, A. Hanbury, Robust color texture features under varying illumination conditions, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 42 (1) (2012) 58–68.
[13] A.R. Backes, D. Casanova, O.M. Bruno, Color texture analysis based on fractal descriptors, Pattern Recognit. 45 (2012) 1984–1992 ⟨http://fractal.ifsc.usp.br/dataset/USPtex.php⟩.
[14] J. Martínez-Alajarín, D. Luis-Delgado, M. Tomás-Balibrea, Automatic system for quality-based classification of marble textures, IEEE Trans. Syst. Man Cybern.—Part C: Appl. Rev. 35 (2005) 488–497.
[15] J.Y. Choi, Y.M. Ro, K.N. Plataniotis, Color local texture features for color face recognition, IEEE Trans. Image Process. 21 (3) (2012) 1366–1380.
[16] I.-U. Quazi, O. Alata, J.-C. Burie, A. Moussa, C. Fernández-Maloigne, Choice of a pertinent color space for color texture characterization using parametric spectral analysis, Pattern Recognit. 44 (2011) 16–31.
[17] T. Mäenpää, M. Piatikäine, Classification with color and texture: jointly or separatety, Pattern Recognition 37 (2004) 1629–1640.
[18] A. Tabesh, M. Teverovskiy, H. Pang, V.P. Kumar, D. Verbel, A. Kotsianti, O. Saidi, Multifeature prostate cancer diagnosis and Gleason grading of histological images, IEEE Trans. Med. Imaging 26 (10) (2007) 1366–1378.
[19] M. Vanrell, R. Baldrich, A. Salvatella, R. Benavente, F. Tous, Induction operators for a computational color-texture representation, Comput. Vis. Image Underst. 94 (2004) 92–114.
[20] H. Deng, D.A. Clausi, Gaussian MRF rotation-invariant features for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 26 (7) (2004) 951–955.
[21] P. Campisi, A. Neri, G. Panci, G. Scarano, Robust rotation-texture classification using a model based approach, IEEE Trans. Image Process. 13 (6) (2004) 782–791.
[22] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: a comprehensive study, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 493–506.
[23] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vis. 62 (2005) 61–81.
[24] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2032–2047.
[25] J.Z.H. Zhao, J. Liang, Continuous rotation invariant local descriptors for texton dictionary-based texture classification, Comput. Vis. Image Underst. 117 (2013) 56–75.
[26] S. Alvarez, M. Vanrell, Texton theory revisited: a bag-of-words approach to combine texton, Pattern Recognit. 45 (2012) 4312–4325.
[27] G. Paschos, Perceptually uniform color spaces for color texture analysis: an empirical evaluation, IEEE Trans. Image Process. 10 (6) (2001) 932–937.
[28] G. Finlayson, S. Hordley, G. Schaefer, G.Y. Tian, Illuminant device invariant color using histogram equalization, Pattern Recognit. 38 (2005) 179–190.
[29] M. Ebner, Color Constancy, Wiley, England, 2007.
[30] G.D. Finlayson, B. Schiele, J. Crowley, Comprehensive color image normalization, in: ECCV 98, 1998, pp. 475–490.
[31] K. Zuiderweld, Contrast limited adaptive histogram equalization, Academic Press 1994, pp. 474–485.
[32] N. Limare, J.-L. Lisani, J.-M. Morel, A.B. Petro, C. Sbert, Simplest Color Balance, IPOL: Image Processing On Line, 2011, http://dx.doi.org/10.5201/ipol.2011.llmps-scb.
[33] J.M. Morel, A.B. Petro, C. Sbert, A PDE formalization of Retinex theory, IEEE Trans. Image Process. 19 (11) (2010) 3825–3837.
[34] N. Limare, A.B. Petro, C. Sbert, J.-M. Morel, Retinex Poisson Equation: A Model for Color Perception, IPOL: Image Processing On Line, 2011, http://dx.doi.org/10.5201/ipol.2011.lmps_rpe.
[35] N. Vandenbroucke, L. Macaire, J.-G. Postaire, Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis, Comput. Vis. Image Underst. 90 (2003) 190–216.
[36] I. Lissner, P. Urban, Toward a unified color space for perception-based image processing, IEEE Trans. Image Process. 21 (3) (2012) 1153–1168, URL ⟨http://www.idd.tu-darmstadt.de/color/papers⟩.
[37] R.M. Haralick, K. Shanmugan, I. Dinstein, Textural features for image classification, IEEE Trans. Man Cybern. 3 (6) (1973) 610–621.
[38] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, International Thomson Publishing (ITP), USA, 1999.
[39] M. Unser, Sum and difference histograms for texture classification, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1) (1986) 118–125.
[40] T. Ojala, M. Piatikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary pattern, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[41] C. Zhu, C.-E. Bichot, L. Chen, Image region description using orthogonal combination of local binary patterns enhanced with color information, Pattern Recognit. 46 (2013) 1949–1963.
[42] Y. Zhao, D.-S. Huang, W. Jia, Complete local binary count for rotation invariant texture classification, IEEE Trans. Image Process. 21 (10) (2012) 4492–4497.
[43] M. Ivanovici, N. Richard, Fractal dimension of color fractal images, IEEE Trans. Image Process. 20 (1) (2011) 227–235.
[44] S. Jansson, Evaluation of methods for estimating fractal properties of intensity images (Ph.D. thesis), Umea University, Umea, Sweden, 2006.
[45] N. Sarkar, B.B. Chaudhuri, An efficient approach to estimate fractal dimension of textural images, Pattern Recognit. 25 (9) (1992) 1035–1041.
[46] Y. Xu, H. Ji, C. Fermuller, Viewpoint invariant texture description using fractal analysis, Int. J. Comput. Vis. 83 (2009) 85–100.
[47] T. Randen, J.H. Husoy, Filtering for texture classification: a comparative study, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 291–310.
[48] F. Bianconi, A. Fernández, Evaluation of the effects of Gabor filter parameters on texture classification, Pattern Recognit. 40 (2007) 3325–3335.
[49] J.S. Walker, A Primer on Wavelets and Their Scientific Applications, Chapman Hall, London, 2008.
[50] A. Laine, J. Fan, Texture classification by wavelet packet signatures, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1186–1191.
[51] I.W. Selesnick, R.G. Baraniuk, N.C. Kingsbury, The dual-tree complex wavelet transform, IEEE Signal Process. Mag. 22 (6) (2005) 123–151.
[52] T. Celik, T. Tjahjadi, T. Celik, T. Tjahjadi, Multiscale texture classification using dual-tree complex wavelet transform, Pattern Recognit. Lett. (2009) 331–339.
[53] A. Vedaldi, B. Fulkerson, VLFeat: an open and portable library of computer vision algorithms, ⟨http://www.vlfeat.org/⟩, 2008.
[54] G.J. Burghouts, J.M. Geusebroek, Material-specific adaptation of color invariant features, Pattern Recognit. Lett. 30 (2009) 306–313, URL http://staff.science.uva.nl/~aloi/public_alot/.
[55] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (2014) 3133–3181, URL ⟨http://jmlr.org/papers/v15/delgado14a.html⟩.
[56] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, London, 2006.
[57] J. Cant, M. Large, L. McCall, M.A. Goodale, Independent processing of form, color, and texture in object perception, Perception 37 (2008) 57–78.
[58] C. Cavina-Pratesi, R.W. Kentridge, C.A. Heywood, A.D. Milner, Separate channels for processing form, texture, and color: evidence from MRI adaptation and visual object agnosia, Cereb. Cortex 20 (2010) 2319–2332.
[59] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

E. Cernadas was born in A Coruña, Spain, in 1969. She received the B.S. and Ph.D. degrees in Physics from the University of Santiago de Compostela, in 1992 and 1997, respectively. She is a Lecturer of Computer Science in the University of Santiago de Compostela, Spain. She is also a researcher of the Centro de Investigación en Tecnoloxías da Información da USC (CITIUS). Her research focuses on Image Processing and Pattern Recognition, mainly applied in the food technology, robotic and biological domains.

M. Fernández-Delgado was born in A Coruña, Spain, in 1971. He received the B.S. in Physics and the Ph.D. in Computer Science from the University of Santiago de Compostela (USC) in 1994 and 1999, respectively. He is currently a Lecturer of Computer Science and researcher of the Centro de Investigación en Tecnoloxías da Información da USC (CITIUS). His research interests include Neural Computation, Machine Learning and Pattern Recognition.

E. González-Rufino was born in Granada (Spain) in 1967. She received the B.S. in Computer Science in 1991 from the University of Granada, Spain. Since November 1992 she has been a Lecturer of Computer Science at the Faculty of Computer Science at University of Vigo, Spain. Her research interest is in image processing (mainly in texture analysis) and pattern recognitions.

P. Carrión was born in Málaga (Spain) in 1968. She received the B.S. in Computer Science in 1991 from the University of Granada, Spain. Since November 1995 she has been an Assistant Professor of Computer Science at the Faculty of Computer Science at University of Vigo, Spain. She also received the Ph.D. in Computer Science in 2004 from this university. Her research interest is in image processing (mainly in texture analysis) and pattern recognitions.