Multi Level Directional Cross Binary Patterns New handcrafted descriptor for SVM-based texture classification

M. Kas, I. El Khadiri, Y. El Merabet, Y. Ruichek, R. Messoussi

Version of Record: https://www.sciencedirect.com/science/article/pii/S0952197620301548
Manuscript_8b3304fefff54bd215e31b9bd058fb04

Multi Level Directional Cross Binary Patterns
New handcrafted descriptor for SVM-based texture classification

M. Kas[a,b,*], I. Elkhadiri[b], Y. El merabet[b], Y. Ruichek[a], R. Messoussi[b]

[a] *CIAD UMR 7533, Univ. Bourgogne Franche-Comte, UTBM, F-90010 Belfort,France*
[b] *Laboratoire LASTID, Département de Physique, Faculté des Sciences, Université Ibn Tofail, BP 133, 14000 Kenitra, Maroc*

Abstract

The pattern recognition and computer vision have experienced a prominent progress in feature extraction techniques, judged by the extensive proposed methods in the literature. A big part of these works was devoted to enhance the texture classification performance, regarding the important role of textural analysis in various real-world and challenging applications. Developing discriminant feature extractors requires solid knowledge in machine learning and applied mathematics. However, Local Binary Patterns (LBP) offered much more space to develop enhanced handcrafted descriptors thanks to its simplicity and flexibility. In this paper we introduce a brand new LBP variant referred to as Multi Level Directional Cross Binary Patterns (MLD-CBP). The proposed representation is training-free, low-dimensional, yet discriminative and robust handcrafted operator for texture description. The concept of the proposed MLD-CBP descriptor is based on encoding the most informative directions contained within multi radiuses, which helps in detecting the gray level variations that may occure in different directions. Moreover, the proposed MLD-CBP handcrafted is combined with an automated SVM classifier based on the RBF Kernel, where the γ parameter is calculated automatically according to the training images. Conducted experiments on 15 well known and challenging databases of the literature, demonstrate prominent performance and stability compared to the results achieved by 30 recent and most powerful descriptors of the state-of-the-art. This paper provides also a comparative study on the effect of γ parameter to show the benefits of automatically tuning this parameter value considering the nature of the database and its size.

Keywords: LBP; texture recognition; feature extraction; SVM classifier; MLD-CBP;

1. Introduction

Texture is widely present in our real world and plays a major role in various critical applications. The visual apparent of any object or pattern can be represented in a texture form at a certain level by its size, shape, organization, and proportions of its parts. Texture is detected on both artificial and natural objects such as those on wood, plants, materials and skin. Texture analysis become a fundamental branch of image processing and computer vision, by exploring the fact that any object can be textured. Hence, many applications can be redefined as texture classification tasks, that include, among others, face recognition and content-based image retrieval [1]. Moreover, texture classification has been adopted in medical image analysis and helps to achieve good results in congestive heart failure [2], human skin analysis [3], brain degenerative diseases [4], etc. During past decades, texture classification gained too much attention due to its difficulties in terms of variability and inhomogeneity, such as scale changes, variable illumination, surface shape variability and imaging conditions. Many state-of-the-art researches had proposed frameworks to analyze the classification process using existing texture databases, which cover all texture materials. As reported in [5], the entire published works share a common structure which can be simplified in a generic two steps architecture as illustrated in Figure 1. The first step consists of representing the texture images to be proceeded and extracting discriminant features, while the second step relays on classifying the calculated features according to the learning model using one of the state-of-the-art classifiers including multi-class Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Describing the textural image and extracting its invariant features phase remains the distinguishing element in the classification process based on signal-processing methods and good features should be discriminative, robust and easy to compute [6]. Notable methods include statistical and model-based ones such as co-occurrence matrices, hidden Markov model, and filter-based methods. The representative filter-based methods include wavelet sub-bands [7], Gabor filters [8]. Back to 2000, the computer vision world experienced the birth of a brand new description and feature extraction technique, introducing the concept of local patterns, which was introduced by Ojala et al. [9] referenced as Local Binary Patterns (LBP). The idea behind this concept is to encode the texture image and transform it to a more representative space

*Corresponding author
Email addresses: mohamed.kas@utbm.fr (M. Kas), elkhadiri.issam@uit.ac.ma (I. Elkhadiri), y.el-merabet@univ-ibntofail.ac.ma (Y. El merabet), yassine.ruichek@utbm.fr (Y. Ruichek), messoussi@gmail.com (R. Messoussi)

based on a specified kernel function, which integrates and considers the neighbors of every pixel in calculating its transformed value. Being simple and flexible in addition to its success realized in many fields, LBP constructed the background to develop other variants; many researchers become motivated to propose their own local operators. Tan and Triggs proposed in [10] local ternary patterns (LTP) for face recognition which is based on encoding the difference between the center pixel and its neighbor pixels by three values (1, 0 or -1) giving a user specified threshold. Recently, based on the LTP variant, [11] proposed adaptive local ternary patterns (ALTP) feature descriptor based on an automatic strategy selecting the threshold for LTP calculated using Weber's Law. Inspired by LBP, Ahonen et al. [12] introduced a combination of LBP concepts with Fourier transform to design a new operator referenced as LPQ for Local Phase Quantization, which is based on detecting the blur invariance characteristic of the Fourier phase spectrum. LPQ utilizes the 2-D short-term Fourier transform (STFT) to extract the information about the local phase information. The authors in [13] proposed local directional ternary pattern (LDTP) for texture classification. The LDTP operator, basically, encodes at the same time the information related to image contrast and the directional pattern features based on local derivative variations. LDTP conveys valuable information about the nature of textures by capturing local structures using both LTP's [10] and LDP's [14] concepts simultaneously. Given that the necessity to design a new brand local texture operator with better discriminative ability is no longer to be demonstrated, the extensive research on application of local descriptors in pattern recognition is still ongoing, which is proved by the recent published works including quad binary pattern QBP [15], local quadruple pattern (LQPAT), local neighborhood difference pattern (LNDP) [16], Principal Curvatures-LBP (PC-LBP) [17], local concave-and-convex micro structure patterns (LCCMSP) [18], repulsive-and-attractive local binary gradient contours (RALBGC) [19], directional coding (DC) method [20] etc.

Even if the literature abounds with great number of texture descriptors, choosing an appropriate feature extraction operator for a large number of challenging representative texture databases presenting several properties remains a real challenge. Furthermore, although significant progress has been made, most LBP-like descriptors still have prominent limitations linked to noise sensitivity and contrast information [21]. In section 5.2, an important number of recent LBP variants that have been used for the comparison purposes in this study are presented. In order to avoid the limitations of these methods and keep the simplicity and effectiveness of the basic LBP while achieving high performance, we design a computationally and conceptually simple yet efficient texture operator referred to as Multi Level Directional Cross Binary Patterns (MLD-CBP) for texture recognition. The essence of MLD-CBP descriptor is to capture higher-order local information by encoding various distinctive spatial micro-structures within 5×5 local regions. Differing from LBP that treats the values of differences between the central pixel and each neighborhood pixel independently of their other neighbors pixels, we adopted a sophisticated ker-

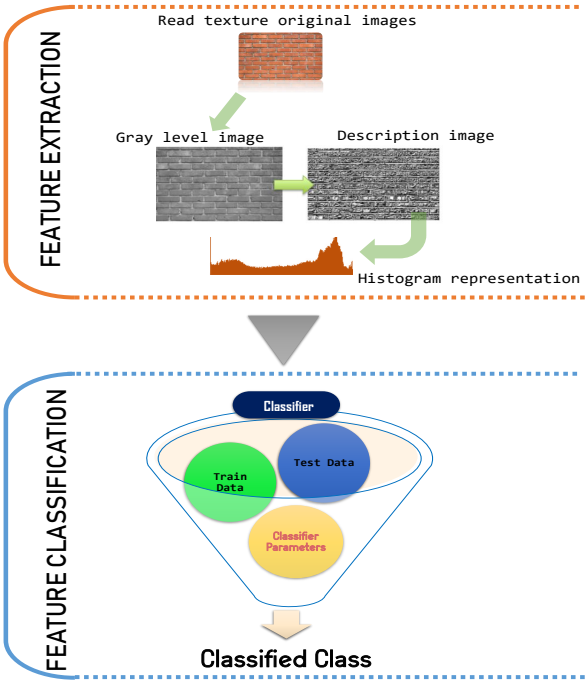

Figure 1: Texture classification basic steps.

nel function where each pixel's value of the adopted sampling groups, is compared to the average gray level of its 3×3 neighbor pixels within its 5×5 neighborhood to make the thresholding process more accurate. Therefore, when defined in a large neighborhood considering two-level radius (i.e., R = 1 and R = 2) and four direction angles (i.e., 0°, 45°, 90° and 135°), the designed descriptor can carry more local information than LBP and many of its variants, and thus the capability of texture representation can be strengthened. The four orientations, i.e.,0°, 45°, 90° and 135° are the most commonly employed ones in the literature [22, 16, 23] to design LBP-like texture operators able to extract prominent features. They are used successfully to construct texture descriptors with high capability to elicit stable and discriminative feature representation while keeping low computational complexity. Considering more directions could lead to additional discriminating information but at the expense of additional computation cost. However, in our approach, besides the pixels belonging to the four directions and in the objective to consider additional discriminative information, the remaining pixels within the 5×5 neighborhood and belonging to other directions are used to calculate some average gray levels, which are involved in the modeling of the proposed descriptor. Note that MLD-CBP operator labels each central pixel with a ten bits code by manipulating only 16 pixels among the 25 ones in the 5×5 neighborhood while the remaining ones are considered in the calculation of average gray levels of each 3×3 sub-block within the 5×5-grayscale image patch in question. The main benefits of the proposed descriptor are: 1) implementation simplicity; 2) low computational complexity 3) free of tuning parameter setup; 4) high ability to extract discriminative and stable texture representation involving high performance compared to old and recent state-of-the-art methods; 5) consid-

erable enhancement of both the discriminative power of LBP variants and their robustness to small variations. After extracting the feature images, we compute their histogram representation which will be fed to an SVM based-classifier set up on the Radial basis function kernel in which the γ parameter is configured automatically according to the nature/particularities of each evaluated descriptor and tested dataset. The process for calculating the γ parameter, which is generally found experimentally, is discussed laterally in this paper. The main contributions introduced in this paper can be summarized and briefed in the following points:

- We design, based on a novel sampling scheme which can encapsulate both microstructure and macrostructure information, a new brand local image descriptor for texture classification, referred to as Multi Level Directional Cross Binary Patterns descriptor (MLD-CBP).
- MLD-CBP operator integrates the available information within a 5×5 grayscale image patch in a compact way which provides an important gray-scale local micro-pattern allowing thus to extract discriminating feature vector keeping a low dimension than traditional LBP and a large number of recent most promising LBP-like descriptors.
- This paper provides systematic, fair and comprehensive comparison between the proposed MLD-CBP descriptor and 30 recent most powerful state-of-the-art methods on 15 publicly available and widely used texture databases representing different textures and class types.
- The proposed model, which is a non-parametric method, performed well when compared with the evaluated state-of-the-art methods.
- In order to make automatic the SVM based classification process, a novel γ parameter calculation formula is proposed. This new formula shows good performance stability over all the tested texture datasets.

The rest of this paper is organized as follows. Section 2 reviews briefly some existing texture descriptors. Section 3 describes the proposed Multi Level Directional Cross Binary Patterns (MLD-CBP) descriptor. Section 4 explains the process of calculating the γ parameter of the SVM classifier. Comprehensive experimental results and discussions are given in Section 5. Section 5.4 investigates the impact of the proposed γ parameter calculation formula on the SVM classifier based classification results leading to some conclusions and perspectives in Section 6.

2. Related Work

In this section, we briefly review the basic Local Binary Patterns (*LBP*) operator and two of its variants: Directional Local Binary Patterns (*nLBPd*) and Local Binary Patterns by Neighborhoods (*dLBP*α) from which we are inspired to develop our handcrafted descriptor. Moreover, *nLBPd* and *dLBP*α illustrate the difference between defining pixel neighbors according to distance and angles within a square sub-block.

2.1. The original Local Binary Patterns

In 2000, Ojala et al. [9] proposed the LBP operator which is not only a descriptor but it goes beyond to a new concept of extracting the feature image exploring the locality and pixel position. It adopts a 3×3 neighborhood as illustrated in Figure 2 and encodes, based on the Heaviside thresholding function (Eq. 2), the relationship between the central pixel and its 8 neighbors starting from the top-left pixel in the clockwise direction. The LBP operator thresholds the value of the central pixel I_c from each of its eight pixels and the resulting value is encoded by 0 if it's strictly negative, or by 1 otherwise. The elementary LBP code generated for each pixel of the original gray level image is calculated as defined in Eq. 1:

$$LBP(I_c) = \sum_{p=0}^{P-1} \Delta(I_p, I_c) \times 2^p \tag{1}$$

where P refers to the number of neighbors which is 8 adopting a 3×3 block size while I_p refers to values of the neighbor pixels with $p = [0, 1, ...P-1]$. Thus, the basic LBP allows to reach a discriminative power of 256 (2^8) possible different patterns. $\Delta(x, y)$ is Heaviside step function (cf Eq. 2).

$$\Delta(x, y) = \begin{cases} 1 & , x \geq y \\ 0 & , x < y \end{cases} \tag{2}$$

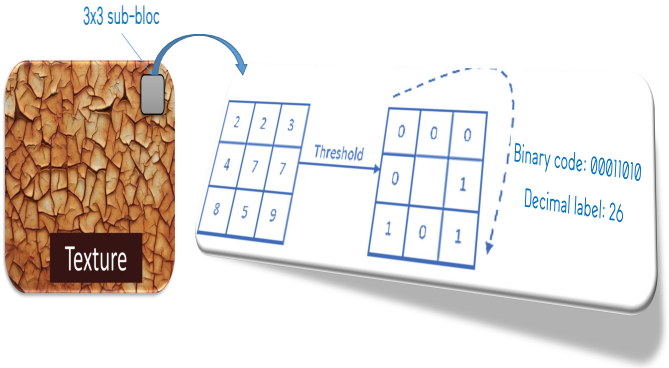

Figure 2: LBP code calculation example.

2.2. Directional local binary patterns and local binary patterns by neighborhoods

The authors in [23] introduced two LBP-like handcrafted descriptors for texture analysis, referred to as local binary patterns by neighborhoods (*nLBPd*) and directional local binary patterns (*dLBP*α). nLBP_d, which uses the same bloc size adopted by the basic LBP operator, consists also in encoding the relationship between the eight peripheral pixels $\{\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_7\}$ around the central pixel $\mathbf{I}_c$. As defined in the kernel function in Eq. 3, the thresholding process is done considering the peripheral pixels by comparing the gray level value of each pixel with the one of the d next pixel, where d is a distance parameter. An example of nLBP_d calculation from a given sub-image with $d = 1$ and $d = 2$ is given in Figure 3.

$$\mathrm{nLBP}_d(\mathbf{I}_c) = \sum_{p=0}^{\mathbf{P}-1} \Delta(\mathbf{I}_p, \mathbf{I}_{(p+d) \bmod \mathbf{P}}) \times 2^p \quad (3)$$

where $\Delta(x, y)$ is the same step function used in LBP operator (cf. Eq. 2).

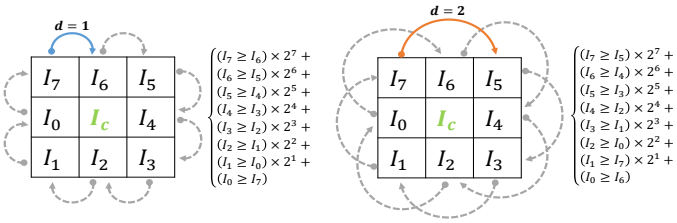

Figure 3: Overview of relations between neighbors in the nLBPd.

In $dLBP\alpha$, the neighborhood topology introduces another concept based on orientation angles. The comparison is done through the pixels belonging to the same straight line with orientation angle α (α=0°, 45°, 90° or 135°) in a counter clockwise according to $+x$-axis as in Gray-Level Co-Occurrence Matrix (GLCM) [24]. After fixing the angle, i.e., when the neighbors are defined, the traditional LBP process is used to compute the label of the central pixel using the thresholding function defined in Eq. 2. The neighborhood vectors in the horizontal orientation (α=0°), vertical orientation (α=90°), and two diagonal orientations (α=45° and α=135°) are shown in Figure 4.

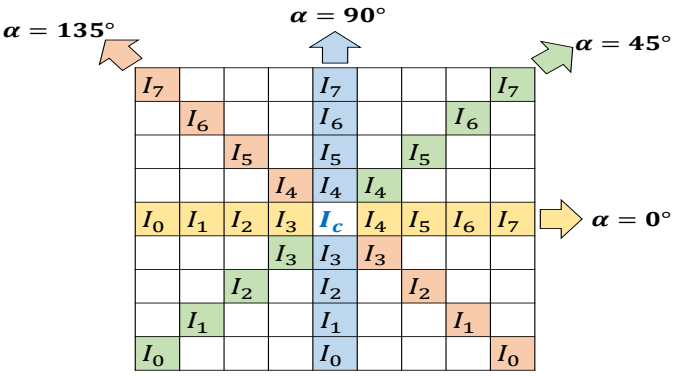

Figure 4: Displacement vectors of dLBPα.

3. Multi Level Directional Cross Binary Patterns

Because of binary coding in the aforementioned texture operators, a large amount of texture information of local spatial patterns is lost. On the one side, due to the simplicity of the bit string coding technique incorporated in the code construction of the basic LBP and a large number of LBP-like methods, it ignores most information of the neighborhood. In addition, when these LBP variants encode micro-level information of edges, spots and other local features in an image using intensity information change around each pixel [13]. On the other side, in the neighborhood topology of $dLBP\alpha$ operator, only the pixels belonging to the same straight line in an orientation angle α are adopted in the code construction, thus, $dLBP\alpha$ misses some directional informations which can be captured by the other orientations. Besides, many LBP variants may suffer from serious weaknesses of noise sensitivity for the local pixel-based texture feature extraction, which means even the slightest fluctuation above or below the gray value of the central pixel $\mathbf{I}_c$ or a neighbor pixel $\mathbf{I}_p$ can change the whole LBP pattern significantly [25]. In this paper, to overcome these disadvantages, we introduce an advanced encoding way involving two different concepts of neighborhood sampling by exploiting simultaneously multi-radial and multi-orientation information. The designed local texture modeling inherits pertinent properties from both compact encoding of directional pattern features and encoding of contrast information. The code construction of the proposed descriptor, referred to as Multi Level Directional Cross Binary Patterns (MLD-CBP), involves three main parts: local sampling (i.e., neighborhood topology), patterns encoding (i.e., defining neighbor pixels and encoding micro-patterns based on thresholding kernel functions), and histogram computing (i.e., feature vector), as illustrated in Figure 5.

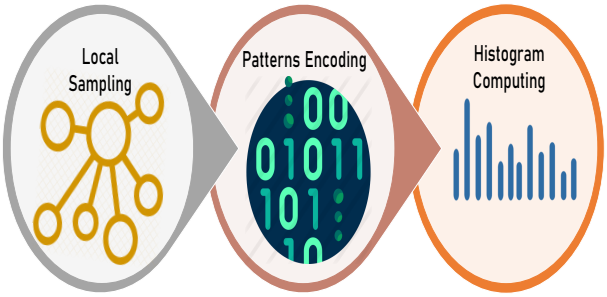

Figure 5: MLD-CBP steps for feature vector extraction.

3.1. Local sampling

In computer vision applications, the choice of neighborhood size is a key factor in the development phase of handcrafted descriptors. The concept is based on encoding the relationship between pixels using the thresholding function. In general, more the pixels used in the kernel function are reasonably numerous (keeping local analysis) more the description is accurate, which is required to improve classification performance. However, adopting higher neighborhood sizes leads to high processing time during the thresholding process and necessitates to manipulating feature vector with high dimension. In our work and in order to accommodate large intra-class variation and low inter-class distinction, we adopted 5×5 neighborhood, which allows combining radii (2) and angles (4) in the construction of the proposed descriptor. This 5×5 configuration should provide more discriminant information, compared to the 3×3 one, which is only supporting angle variation. In addition to the central pixel $\mathbf{I_c}$ of each 5×5 sub-region, we manipulate 16 pixels covering at the same time four orientations (i.e., 0°, 45°, 90° and 135°) and two radiuses (i.e., $\mathcal{R}$=1 and $\mathcal{R}$=2). Hence, the proposed MLD-CBP can achieve continuous

invariance at any rotation angle and capture micro- and macro-structure texture information simultaneously than conventional LBP-based schemes.

Based on the above statements, the local sampling of the proposed MLD-CBP operator is performed as illustrated in Figure 6. For each central pixel $\mathbf{I}_c$ in the 5×5 sub-block, we symmetrically sample eight pixels in each of the two radiuses (i.e., $\mathcal{R}$=1 and $\mathcal{R}$=2). In order to easily manipulate these pixels in the thresholding steps, we defined 3 groups of pixels based on their distance to the central pixel and their angle as can be seen in Figure 7. The first one, denoted by $\mathbf{I}^A$, contains the eight pixels of the first neighborhood which cover the four orientations [0°, 45°, 90° and 135°]. In the second neighborhood ($\mathcal{R}$=2), there are two kind of pixels, eight ones belonging to the four primary orientations forming the $\mathbf{I}^B$ group and the remaining eight pixels distributed on different angles construct the group $\mathbf{I}^C$. The elements of each pixels group are numerated in clockwise order so we exploit their parity in incoming equations and definitions.

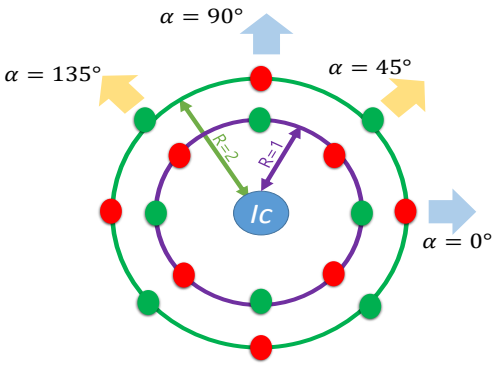

Figure 6: MLD-CBP local sampling topology. Sixteen points are sampled around the central pixel $\mathbf{I_c}$.

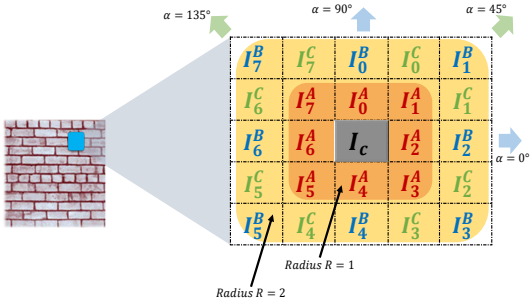

Figure 7: Pixels organization within the 5×5 sub-block.

The concept of the proposed descriptor is based on combining the pixels of the group $\mathbf{I}^A$ with horizontal and vertical directions or diagonal directions along with the other directions from $\mathbf{I}^B$ as illustrated in Figure 6 where the combined pixels have the same color (red/green). This arrangement can be explained based on the parity where we combine the elements of $\mathbf{I}^A$ with the ones of $\mathbf{I}^B$ which have different parity. Therefore, we define two sampling groups $\beth_{SG_1}$ and $\beth_{SG_2}$. $\beth_{SG_1}$ contains the even pixels of $\mathbf{I}^A$ group located in the vertical and horizontal directions $\{\mathbf{I}^A_0,\mathbf{I}^A_2,\mathbf{I}^A_4,\mathbf{I}^A_6\}$ and the odd ones of $\mathbf{I}^B$ group located in the two diagonal directions $\{\mathbf{I}^B_1,\mathbf{I}^B_3,\mathbf{I}^B_5,\mathbf{I}^B_7\}$, while $\beth_{SG_2}$ contains the pixels in green color, i.e., the odd pixels of $\mathbf{I}^A$ group located in the two diagonal directions $\{\mathbf{I}^A_1,\mathbf{I}^A_3,\mathbf{I}_{5^A},\mathbf{I}^A_7\}$ and the even ones of $\mathbf{I}^B$ group located in the vertical and horizontal directions $\{\mathbf{I}^B_0,\mathbf{I}^B_2,\mathbf{I}^B_4,\mathbf{I}^B_6\}$. For a more understandable representation and for the sake of simplicity, the sampling groups $\beth_{SG_1}$ and $\beth_{SG_2}$, illustrated in Figure 8, can be expressed in the following manageable and formal ways (cf. Eq. 4 and Eq. 5):

$$\beth_{SG_1} = \{\mathbf{I}^A_{2i}, \mathbf{I}^B_{2i+1}\}; i = 0, 1..., 3 \tag{4}$$

$$\beth_{SG_2} = \{\mathbf{I}^A_{2i+1}, \mathbf{I}^B_{2i}\}; i = 0, 1..., 3 \tag{5}$$

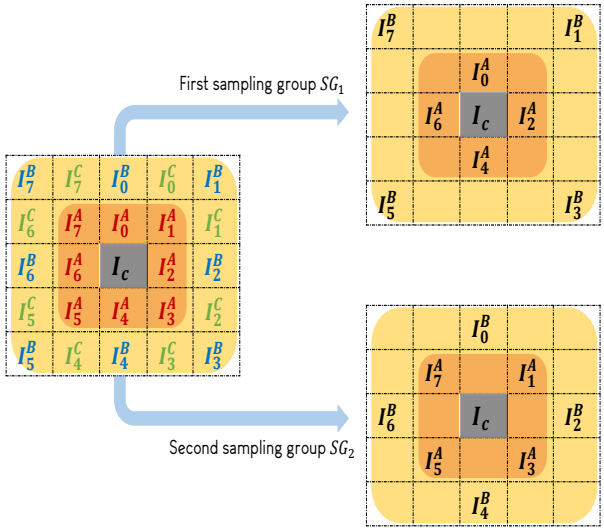

Figure 8: Defining the sampling groups.

As it is confirmed by several researches in pattern recognition based on local binary patterns, the average gray level is a distinguishing statistical parameter for texture analysis, which gives the opportunity of considering a global information within the micro pattern. In view of this and aiming at finding a code which is insensitive to noise and invariant to monotonic gray scale transformation, we adopted, therefore, to integrate average gray level values in the kernel functions in order to enhance the thresholding process. The remaining pixels within the 5×5 neighborhood, labeled as $\mathbf{I}^C_{i;i=0,..,7}$ in Figures 7 and 8, are incorporated, in turn, in the calculation of the average local gray level of the whole 3×3 neighborhood around each pixel having as label $\mathbf{I}^A$, which will be used in the pattern encoding phase. Based on the above neighborhood topology, eight average local gray levels labeled as $\{\widehat{\beta}_{A0},\ldots\ldots\widehat{\beta}_{A7}\}$, are computed as defined in Eqs 6 and 7.

$$\widehat{\beta}_{A_{2i}} = \frac{1}{\mathbf{P}+1}\Big(\sum_{p\in\omega_{2i}} \mathbf{I}^A_p + \sum_{q\in\upsilon_{2i}} \mathbf{I}^C_q + \mathbf{I}^B_{2i} + \mathbf{I}_c\Big) \tag{6}$$

$$\widehat{\beta}_{A_{2i+1}} = \frac{1}{\mathbf{P}+1}\Big(\sum_{p\in\omega_{2i+1}} \mathbf{I}^A{}_p + \sum_{q\in\upsilon_{2i+1}} \mathbf{I}^C{}_q + \mathbf{I}^B{}_{2i+1} + \mathbf{I}_c\Big) \tag{7}$$

where i ∈ [0-3], ω_r is the quintuplet centered at element r and υ_s is the doublet with starting element s, extracted from the **P**-cycle $\mathfrak{G}_{\mathbf{P}}$={0,1,2,....,7} (circular permutation of order **P**), respectively.

3.2. Patterns encoding

After defining the arrangement of the sampled pixels and calculating the average gray level values, we proceed now to pattern encoding scheme of the proposed MLD-CBP descriptor. As presented earlier, we arranged the pixels into two sampling groups based on their radius and angle. Therefore, for each sampling group $\beth_{SG_i}$, we define a cross encoder Ξ_{Ec_i}, which considers the central pixel I_c and the pixels of the sampling groups adopting the basic thresholding function where each pixel of the sampling groups, is compared to the average local gray level of the 3×3 image patch to which the pixel belongs to. The central pixel I_c is compared locally to the average local gray level $\widehat{\beta}_{Ec_i}$ according to each cross encoder $\Xi_{Ec_{i,i=1,2}}$. $\widehat{\beta}_{Ec_1}$ (cf. Eq. 10) and $\widehat{\beta}_{Ec_2}$ (cf. Eq. 11) represent the average of the even pixels $\mathbf{I}^A_{2i;i\in[0-3]}$ and the odd pixels $\mathbf{I}^A_{2i+1;i\in[0-3]}$, respectively which are incorporated in the modeling of the two encoders Ξ_{Ec_1} and Ξ_{Ec_2}, respectively. The codes produced by the two encoders Ξ_{Ec_1} and Ξ_{Ec_2} associated to the two sampling groups $\beth_{SG_1}$ and $\beth_{SG_2}$ are computed as:

$$\Xi_{Ec_1}(\chi) = \sum_{i=0}^{3} \Delta(\mathbf{I}^{\mathcal{B}}_{2i+1}, \widehat{\beta}_{A_{2i+1}}) \times 2^i + \sum_{i=4}^{\mathbf{P}-1} \Delta(\mathbf{I}^{A}_{2(i-4)}, \widehat{\beta}_{A_{2(i-4)}}) \times 2^i + (\mathbf{I}_c, \widehat{\beta}_{\Xi_{Ec_1}}) \times 2^{\mathbf{P}} \tag{8}$$

$$\Xi_{Ec_2}(\chi) = \sum_{i=0}^{3} \Delta(\mathbf{I}^{\mathcal{B}}_{2i}, \widehat{\beta}_{A_{2i}}) \times 2^i + \sum_{i=4}^{\mathbf{P}-1} \Delta(\mathbf{I}^{A}_{2(i-4)+1}, \widehat{\beta}_{A_{2(i-4)+1}}) \times 2^i + \Delta(\mathbf{I}_c, \widehat{\beta}_{\Xi_{Ec_2}}) \times 2^{\mathbf{P}} \tag{9}$$

where

$$\widehat{\beta}_{\Xi_{Ec_1}} = \frac{\sum_{i=0}^{3} \mathbf{I}^A_{2i}}{\mathbf{P}/2} \tag{10}$$

$$\widehat{\beta}_{\Xi_{Ec_2}} = \frac{\sum_{i=0}^{3} \mathbf{I}^A_{2i+1}}{\mathbf{P}/2} \tag{11}$$

In equations 8 and 9, χ is the set of gray-scale values of a 5×5 square neighborhood. $\widehat{\beta}_{\Xi_{Ec_1}}$ and $\widehat{\beta}_{\Xi_{Ec_1}}$ are average local gray levels of even and odd pixels belonging to the group $\mathbf{I}^A$, respectively.

3.3. Histogram computing

Once the Ξ_{Ec_1} and Ξ_{Ec_2} labels are constructed for every pixel of the image, the histograms describing the texture are generated as follows (cf. Eq. 12 and 13):

$$\mathbf{h}_{\Xi_{Ec_1}}(\mathbf{k}) = \sum_{\mathbf{x}} \vartheta(\Xi_{Ec_1}(\mathbf{x})), \mathbf{k}) \tag{12}$$

$$\mathbf{h}_{\Xi_{Ec_2}}(\mathbf{k}) = \sum_{\mathbf{x}} \vartheta(\Xi_{Ec_2}(\mathbf{x})), \mathbf{k}) \tag{13}$$

where $\mathbf{k} \in [0, N_{bins}]$ is the number of Ξ_{Ec_1} and Ξ_{Ec_2} patterns, N_{bins}=2^9 is the number of bins and the delta function $\vartheta(\cdot)$ is defined as below (cf. Eq. 14):

$$\vartheta(\mathbf{a}, \mathbf{b}) = \begin{cases} 1, & \text{if } \mathbf{a} = \mathbf{b}; \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

To make the feature representation more effective and robust, the coarse and fine information can be captured by multi-scale which can be made through a linear combination of different features obtained by several feature extraction operators. In this work, the features captured by the two cross encoders Ξ_{Ec_1} and Ξ_{Ec_2} are combined into hybrid distributions to form the MLD-CBP model (cf. Eq. 15). This hybrid texture description model is powerful because it permits to reduce the noise sensitiveness and increase the discriminative power of Ξ_{Ec_1} and Ξ_{Ec_2} operators as will be shown in the experiment section. The MLD-CBP descriptor is expected to show high capability of encoding image configuration and pixel-wise relationships and thus, high texture classification performance.

$$\mathbf{h}_{\text{MLD-CBP}} = \left\langle \mathbf{h}_{\Xi_{Ec_1}}, \mathbf{h}_{\Xi_{Ec_2}} \right\rangle \tag{15}$$

MLD-CBP operator extracts 1024 (2×2^9) possible patterns in a 5×5 neighborhood. The overall framework of the MLD-CBP feature vector calculation is illustrated in Figure 9 and summarized in Algorithm 1.

Algorithm 1: MLD-CBP based feature extraction

Input: Input Image *Im*
Result: Computed Histogram $h = MLDCBP(Im)$
Functions: index: get the line and column of array's element; **zeros**: initialize a matrix with 0; **hist**: calculate the histogram
read the input image *Im* with m, n size ;
convert the image to grayscale ;
define the set of central pixels :
$\Omega^{I_c} = Im(i, j)/i \in [2, m-2]$ and $j \in [2, n-2]$;
initialize two feature maps :
$\Xi_{Ec_1} = zeros(m-2, n-2)$, $\Xi_{Ec_2} = zeros(m-2, n-2)$;
foreach I_c *in* Ω^{I_c} **do**
  define the sampling groups $\beth_{SG_1}$ (Eq 4) and $\beth_{SG_2}$ (Eq 5) ;
  calculate the average gray level values $\widehat{\beta}_{A0}, \ldots\ldots \widehat{\beta}_{A7}$ corresponding to the neighbor pixels using equations 6 and 7;
  compute the two labels $\Xi_{Ec_1}(index(I_c))$ and $\Xi_{Ec_2}(index(I_c))$ based on equations 8 and 9, respectively;
compute the histogram transformation on the two cross encoders $h_1 = hist(\Xi_{Ec_1})$ and $h_2 = hist(\Xi_{Ec_2})$;
concatenate the two generated histograms $h = [h_1 h_2]$;

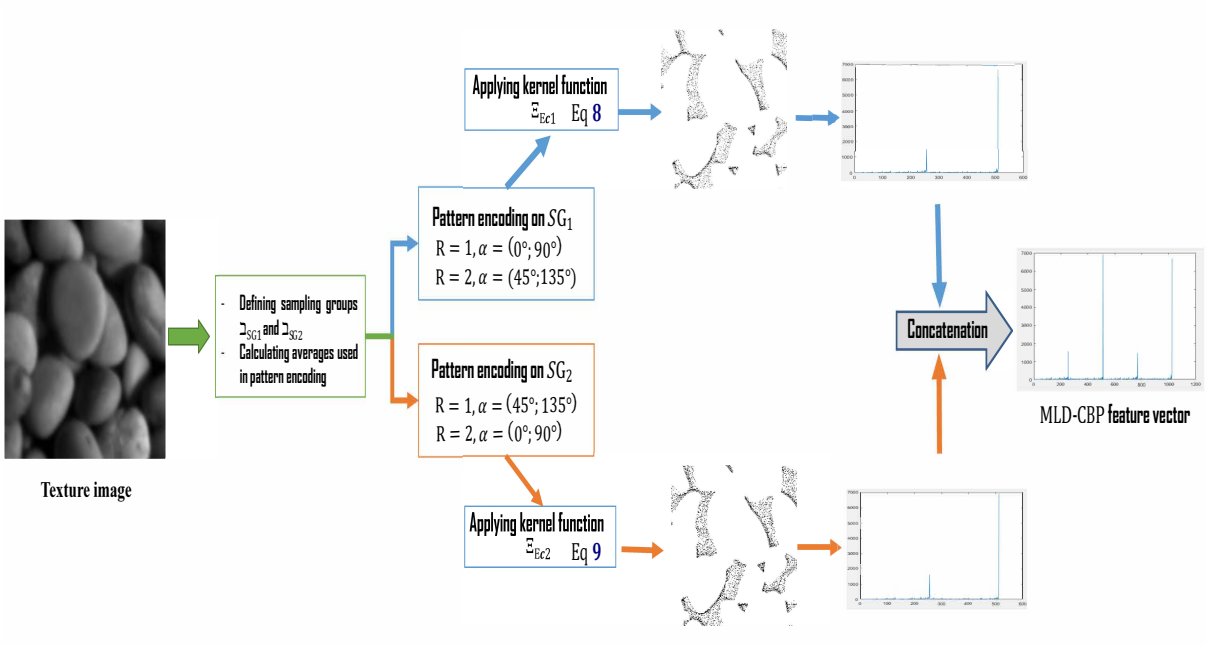

Figure 9: MLD-CBP feature vector calculation.

4. Support Vector Machines (SVM) based texture classification

The Support Vector Machine (SVM) is a discriminative classifier formally based on defining separating hyperplanes. In other words, given labeled training data (supervised learning), the algorithm outputs optimal hyperplanes dividing the two dimensional space generating an SVM model based in the representation of the training data as points in space, mapped so that the examples of the separate categories or classes are divided by a dividing plane that maximizes the margin between different classes which allows to classify the query points and predict their classes.

Figure 10 illustrates the concept behind Support Vector Machines classification. In the left side, we have the input feature vectors calculated earlier using the handcrafted descriptors which are mapped by complex curve. The classifier will rearrange these input points in order to reach a linear separation using a set of mathematical functions called kernels.

In this paper, we adopted lib-svm toolbox [26][1], which became very popular toolbox for SVM classification regarding the complexity of classification in terms of the number of classes. This toolbox supports the four kernels of the binary SVM classifier which are linear, polynomial, radial basis function (RBF) and sigmoid kernels as defined in Eq. 16.

$$\begin{cases} Linear & : u'v \\ Polynomial & : (\gamma u'v + C)^d \\ RBF & : e^{(-\gamma * |u-v|^2)} \\ Sigmoid & : \tanh(\gamma u'v + C) \end{cases} \tag{16}$$

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

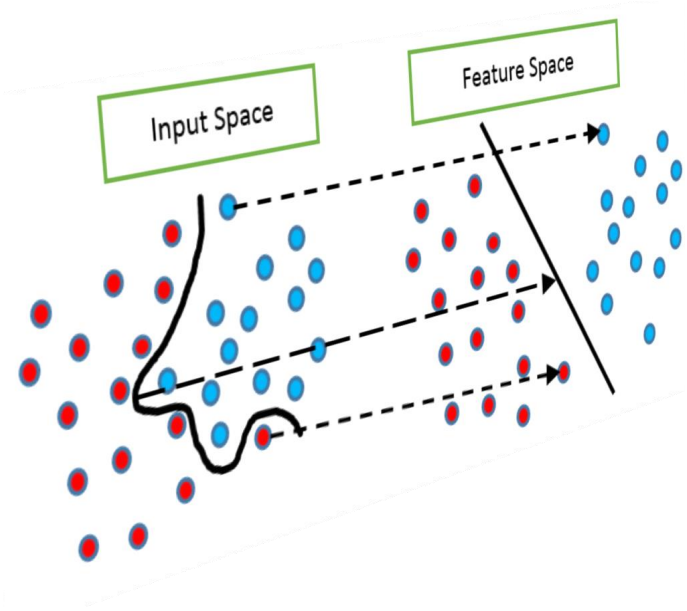

Figure 10: Feature vectors representation both in the input feature spaces.

While training, the multi-class SVM requires two inputs: the feature vectors array X computed by a given descriptor and their corresponding labels vector Y (cd. Eqs. 17 and 18):

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,l} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,l} \end{pmatrix} \tag{17}$$

$$Y = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix} \tag{18}$$

Where l is the length of the computed feature vector and n is the number of observations considered for training. $x_{i,j}$ is the value of the j^{th} bin of the i^{th} train observation, which has the label y_i. This process is performed based on the "**svmtrain**" function available in the LibSVM toolkit. The output of this training stage is an array of parameters and values that will be used in the prediction stage. The latter is performed using the "**svmpredict**" function which takes as input a feature vector of a test sample $X_{test} = [x'_1, x'_2, \ldots, x'_n]$ and outputs a label $y \in Y$. In our classification framework, we implemented the lib-svm toolbox in MATLAB environment, configured on RBF kernel. The RBF is by far the most popular choice of kernel types used in Support Vector Machines believing that it is suitable for very high dimension feature sets. The RBF kernel highly depends on γ parameter which is technically defined as the inverse of the standard deviation of the RBF kernel (Gaussian function), which is used as similarity measure between two points.

Since the setting of this parameter may have significant impact on SVM performance, particular attention should be taken in evaluating the SVM classifier. Generally, the researchers are faced off to three options: 1) the use of the default value $\gamma = \frac{1}{Card(Dataset)}$ determined considering only the size of the dataset, 2) the identification of the optimal γ value for texture classification as it is proposed in [27, 28] and 3) the recourse to the test-error mechanism that consists in adjusting manually the γ parameter until finding the optimal value. To the best of our knowledge, no general criterion to estimate automatically the optimal value for the γ parameter for each evaluated descriptor and tested dataset has been proposed so far. In this paper, to alleviate the problem of selecting the right γ parameter in the SVM and after being inspired by the default γ value defined by lib-svm toolbox, we proposed a new, yet simple γ parameter calculation formula which considers the training set along with the evaluated descriptor. Note that, the default γ value considers only the size of the tested dataset while ignoring its properties and particularities. However, if we consider two different databases with the same size, the SVM classifier will use the same γ value to classify the probes of both databases which makes the SVM kernel less discriminative. In contrary, the proposed formula is based on exploring the representative information contained in the feature vectors rather than considering the number of images. The proposed formula is expressed as follows (cf. Eq. 19):

$$\gamma = \frac{1}{2 \times \sum_i^{Card(Train)} \sum_j^{l} \mathbf{h}_i(x_j)} \tag{19}$$

where $Card(Train)$ refers to the number of images of the train set, l is the number of histogram bins of the histogram $\mathbf{h}_i$ of each feature image extracted using the evaluated descriptor and x_j represents the bins of each histogram. The benefit of this new γ parameter determination the method will be demonstrated through experimentation comparing the results obtained using the proposed Multi Level Directional Cross Binary Patterns descriptor with different γ values reported in earlier works along with the default value and those recorded by the nearest neighbor classifier tested with L1 distance. We highlight in Algorithm 2 the procedure for training the SVM model and in Algorithm 3 the procedure for evaluating its performance. .

Algorithm 2: SVM training

Input: Descriptor (MLDCBP for example), Image database
Result: Trained SVM Model M^{svm}
Functions: index: get the line and column of array's element; **zeros**: initialize a matrix with 0; **hist**: calculate the histogram; **svmtrain**: LibSVM toolbox function to train SVM
get the train images: Ω^{Tr};
define the feature array X and its corresponding labels vector Y:
$X^{Tr} = zeros(Card(\Omega^{Tr}), n)$, Y^{Tr}
n is the number of generated patterns by the descriptor;
foreach *Im in* Ω^{Tr} **do**
  compute the feature vector of the train image:
  $X^{Tr}(index(Im)) = Descriptor(Im)$;
calculate the γ parameter value using equation 19;
train the SVM model using "**svmtrain**" function
M^{svm} =**svmtrain**(X^{Tr}, Y^{Tr}, γ);

Algorithm 3: SVM evaluating

Input: Descriptor (MLDCBP for example), Image database, Trained SVM Model M^{svm}
Result: Classification accuracy Acc
Functions: index: get the line and column of array's element; **zeros**: initiliaze a matrix with 0; **hist**: calculate the histogram; **svmpredict**: LibSVM toolbox function to predict the label of a test image
get the test images: Ω^{Ts};
define the feature array X and the corresponding labels vector Y:
$X^{Ts} = zeros(Card(\Omega^{Ts}), n)$, Y^{Ts};
foreach *Im in* Ω^{Ts} **do**
  compute the feature vector
  $X^{Ts}(index(Im)) = Descriptor(Im)$;
  predict the label using "**svmtpredict**" function :
  $Y^{Pred}(index(Im))$ =**svmpredict**(X^{Ts}, M^{svm});
compare the predicted labels with the groundtruth ones and calculate the accuracy:
$Acc = \frac{\sum_{i=1}^{Card(\Omega^{Ts})} \vartheta(Y^{Pred}(i), Y^{Ts}(i))}{Card(\Omega^{Ts})}$;
ϑ is defined in Eq. 14

5. Experimental analysis

In order to evaluate the performance of the proposed MLDCBP descriptor, we performed extensive texture classification experiments on 15 well-established texture databases routinely used by the texture classification research community. A comparison with a large number of recent state-of-the-art texture descriptors (cf. section 5.2), is performed to show the improvement that our operator provides. Experiments were conducted under the standard half-half configuration based evaluation protocol and all results are reported over 100 random partitionings

of training and testing sets. The following subsections describe: 1) the used texture databases and their properties; 2) the evaluated state-of-the-art LBP-like descriptors; 3) the obtained experimental results and 4) the impact of the γ calculation technique and 5) implementation and execution.

5.1. Texture databases

The discriminating power of the proposed MLD-CBP operator as well as those of 30 evaluated state-of-the-art descriptors are compared on fifteen publicly available datasets gathered in Table 1. The tested texture datasets were chosen to have different characteristics in terms of number of classes, number of samples, noisy conditions, class homogeneity with regards to scale, perspective, and illumination. In the following points, we present the particularities of each database based on the descriptions published in [29] and [30]:

- **2D_Hela**: The 2D_Hela includes 10 classes of fluorescence microscopy, which are DNA (Nuclei), ER (Endoplasmic reticulum), Giantin, (cis/medial Golgi), GPP130 (cis Golgi), Lamp2 (Lysosomes), Mitochondria, Nucleolin (Nucleoli), Actin, TfR (Endosomes), Tubulin. Each class contains 20 samples and the purpose behind this dataset is to evaluate the performance achieved by the MLD-CBP proposed descriptor in bioimage classification task.

- **BonnBTF**: It contains 10 classes and 16 samples for each. This database is obtained by merging two different ones which are part of BTF database Bonn: 'ATRIUM' and 'UBO2003'. The goal behind BonnBTF database is to assess the efficiency and high-fidelity capture of materials appearance.

- **Brodatz**: Despite it is quite old, Brodatz dataset is largely used still today. It includes textures of 13 classes of natural scenes and materials (i.e., grass, bark, sand and straw) as well as artificial manufactured ones (i.e., raffia, pigskin and bricks). Each class is represented over 26 image samples. The particularity of this dataset remains on different background intensities.

- **CUReT** [2]: The Columbia-Utrecht Reflectance and Texture (CUReT) database represents an improvement over the Brodatz collection and contains 61 real world textures (classes), with 205 images for each. In the subset, we considered 92 samples for each class as adopted in [19]. The images are taken at different illumination orientations and view-points, which results a total of 6512 images and makes this database a challenging one.

- **JerryWu**: The database name refers to the researcher Jerry Wu who built it. The database contains a total of 39 natural and artificial texture classes combining surface rotation, illumination and imaging directions properties to challenge the classification frameworks. We adopted the same experimental setup presented in [29].

- **KTH-TIPS and KTH-TIPS2b**: The KTH-TIPS and KTH-TIPS2b databases provide the possibility of investigating the effect of real-world imaging conditions on material classification. They are extended versions of CUReT database. The KTH-TIPS database contains the ten classes of CUReT database and adds new photographing samples in terms of rotation angles and lighting directions, giving 81 samples for each texture class. The KTH-TIPS2b database consists of 11 classes and 16 images for each. The differences between the two databases are relative to image scale and illumination.

- **Kylberg**[3]: The Kylberg database maintained at the Centre for Image Analysis of Swedish University of Agricultural Sciences & Uppsala University, presents 28 textured surfaces, with 160 samples for each class. Each texture class was imaged under only one light setting from one direction with the same distance. This database is available in two versions: without rotated texture patches and with rotated texture patches. In our experiments, we used the first one.

- **MondialMarmi**: The MondialMarmi database features 12 classes of granite tiles, representing each class over 64 samples. These samples have been acquired under controlled illumination conditions.

- **OuTeX TC_00000, 00001 and 00013** [4]: OuTex databases stand for University of Oulu Texture database. They are a well-known benchmarks for evaluating texture classification and segmentation algorithms. These databases provide a wide variety of surface textures, which have been acquired in multiple illumination directions (three lighting sources), surface rotations, and spatial resolutions. OuTeX TC_00000 and OuTeX TC
_00001 have the same 24 texture classes except that the first one presents each class over 20 samples only and 88 for the second database. The OuTeX TC_00013 test suite is composed of 1360 samples representing 68 classes with 20 samples for each.

- **UIUCTex**: This database combines 25 natural and artificial texture classes, containing 40 samples for each class covering variable and uncontrolled imaging conditions. UIUCTex offers challenging conditions in terms of affine transforms, viewpoint and illumination configurations.

- **VisTex** [5]: The Vision Texture database is formed by the Vision and Modeling group at the MIT Media Lab. It contains 167 classes of natural texture and represents each class over 16 samples. The goal behind developing VisTex database is to provide texture images that are simulating real world conditions.

[2] http://www.cs.columbia.edu/CAVE/software/curet/html/about.php

[3] http://www.cb.uu.se/ gustaf/texture/

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

[5] http://vismod.media.mit.edu/vismod/imagery/VisionTexture
/vistex.html

Table 1: Description of the selected texture databases.

No.	Name	Classes	Samples per class	Total samples	Sample resolution (pixels)
1	2D-HeLa	10	20	200	Variable
2	Bonn BTF	10	16	160	200×200 and 64×64
3	Brodatz	13	16	208	256×256
4	CUReT	61	92	5612	200×200
5	Jerry Wu	39	4	156	256×256
6	KTH-TIPS	10	4	40	100×100
7	KTH-TIPS2b	11	16	176	100×100
8	Kylberg	28	160	4480	576×576
9	MondialMarmi	12	64	768	136×136
10	OuTeX TC-00000	24	20	480	128×128
11	OuTeX TC-00001	24	88	2112	64×64
12	OuTeX TC-00013	68	20	1360	128×128
13	UIUCTex	25	40	100	640×480
14	Vistex	167	16	2672	128×128
15	XU HR	25	40	1000	1280×960

- **XU HR** [6]: This database consists of 1000 different texture images : 40 samples for each of 25 classes. Moreover, it provides significant viewpoint changes and scale differences.

5.2. *Evaluated LBP-like handcrafted descriptors*

In order to fairly judge the performance of the proposed descriptor and to disclose meaningful reviews, we compared its achieved accuracies to those recorded by 2 baselines (i.e., LBP and LTP) and 28 recent and well performing state-of-the-art descriptors. The LBP based methods, used for extensive evaluation and comparison with our method, are summarized in chronological order in Table 2.

Among the 30 evaluated LBP-like descriptors summarized in Table 2, some operators like AELTP, ALTP, CSALTP, DBC, dLBPα, nLBPd, LECTP, LESTP, LTP and QBP are parametric methods which require user defined values to perform the thresholding process. Since the setting of these user-specified parameters may have significant impact on their performance, particular attention should be taken in evaluating these parametric descriptors. Unlike the majority of the state-of-the-art works which adopt one fixed value for each evaluated parametric method over many tested databases, we performed independent classification experiments for each method to pick out its best parameter value over each database. Table 3 reports the obtained optimal values.

5.3. *Experimental results*

The comparative assessment is based on the data reported in Tables 4 and 5. Table 4 summarizes the recorded average accuracies (i.e., over 100 subdivisions) of each descriptor and for each tested texture dataset, while Table 5 illustrates the ranking for each dataset based on the average accuracy recorded by the evaluated methods. Note that the highest classification rates reported in Table 4 are highlighted with a green background. Moreover, Table 6 presents global ranking metrics allowing to analyze more deeply the overall performance of the evaluated handcrafted descriptors. We performed, as for several recent works [18, 45, 13], the Wilcoxon Signed Rank method to compare the performance of each descriptor with the rest on all the adopted databases. Since we considered 31 descriptors (including our proposed MLDCBP) and 15 datasets, each descriptor will have 450 comparisons and the score is the ratio of the victories over the total comparisons. Also, we included the "Average rate" and the "Standard Deviation (std)" columns as complementary criteria to rank the methods that reached the same score. Based on the analysis of the obtained results, we can readily make the following observations:

- It emerges from Table 4, that, except some descriptors like XCS-LBP, LOOP, dLBP$_\alpha$ and CSALTP which do not produce good results, all the other evaluated descriptors, including the proposed MLD-CBP operator get very promising classification results on KTH-TIPS dataset (dataset 6 in Table 4) where their score is above 96%. Impressively, some methods like LBP, LTP, LESTP, LECTP LDTP and LCCMSP as well as the proposed descriptor manage to differentiate all classes perfectly leaving then essentially no room for improvement, i.e., the perfect recognition rate of 100% has been achieved on KTH-TIPS dataset. This same remark can also be expressed for Bonn BTF dataset (dataset 2 in Table 4) where some evaluated methods like LESTP, LECTP and WLD as well as the proposed MLD-CBP descriptor manage to differentiate all classes perfectly while the remaining tested methods perform worse.

- 2D_Hela database which includes only ten classes, demonstrated a real challenge to all evaluated methods as they experienced performance drop compared to the other tested

[6] http://legacydirs.umiacs.umd.edu/fer/High-resolution-database/hr_database.htm

Table 2: Summary of texture descriptors tested and compared with the proposed descriptor.

N°	Complete name	Abbreviation	Application	Year	Ref
1	Local Binary Pattern	LBP	Texture classification	2002	[9]
2	Local Ternary Pattern	LTP	Face recognition	2007	[10]
3	Local Phase Quantization	LPQ	Texture classification	2008	[12]
4	Weber Local Descriptor	WLD	Texture classification	2010	[31]
5	Directional Binary Code	DBC	Face recognition	2010	[32]
6	Local Directional Number Pattern	LDN	Face Expression Analysis	2013	[33]
7	Multi-scale Joing Encoding of Local Binary Pattern	MSJLBP	Texture classification	2013	[34]
8	Complete Robust Local Binary Pattern (S-MxC)	CRLBP-S-MxC	Texture classification	2013	[35]
9	Local Extreme Complete Trio Pattern	LECTP	Image retrieval	2014	[36]
10	Local Extreme Sign Trio Pattern	LESTP	Image retrieval	2014	[36]
11	Local Binary Patterns by neighborhoods	*nLBPd*	Texture classification	2015	[23]
12	Directional Local Binary Patterns	dLBPα	Texture classification	2015	[23]
13	Difference Symmetric Local Graph Structure	DSLGS	Finger vein recognition	2015	[37]
14	Magnitude Maximum Edge Position Octal Pattern	MMEPOP	Image retrieval	2015	[38]
15	Multi-Orientation Weighted Symmetric Local Graph Structure	MOW-SLGS	Finger vein recognition	2015	[39]
16	Sign Maximum Edge Position Octal Pattern	SMEPOP	Image retrieval	2015	[38]
17	eXtended Center-Symmetric Local Binary Pattern	XCS-LBP	Texture classification	2015	[40]
18	Adjacent Evaluation LTP	AELTP	Texture classification	2015	[41]
19	Center-Symmetric adaptive LTP	CSALTP	Face recognition	2016	[11]
20	Dominant Rotated Local Binary Patterns	DRLBP	Texture classification	2016	[42]
21	Extended Local Graph Structure	ELGS	Texture classification	2016	[43]
22	Adaptive Local Ternary Pattern	ALTP	Face recognition	2016	[11]
23	Quad Binary Pattern	QBP	Target tracking	2016	[15]
24	Rotation-invariant features based on directional coding	DC	Texture classification	2018	[22]
25	Local Optimal Oriented Pattern	LOOP	Spieces recognition	2018	[44]
26	Local Neighborhood Difference Pattern	LNDP	Face recognition	2018	[16]
27	Local Concave-and-Convex Micro-Structure Patterns	LCCMSP	Texture classification	2018	[18]
28	Local directional ternary pattern	LDTP	Texture classification	2018	[13]
29	Repulsive-and-attractive local binary gradient contours	RALBGC	Texture classification	2018	[19]
30	Attractive-and-Repulsive Center-Symmetric Local Binary Patterns	ARCS-LBP	Texture classification	2019	[45]

Table 3: Optimal values of the parameter of each parametric method found on each database.

Parametric Descriptor	**Dataset N°**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
AELTP	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2
ALTP	0.003	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.009	0.009	0.009	0.006	0.006	0.006
CSALTP	0.012	0.012	0.009	0.006	0.009	0.012	0.009	0.009	0.012	0.009	0.009	0.006	0.003	0.003	0.009
DBC	90°	135°	90°	135°	90°	135°	135°	90°	90°	135°	135°	135°	90°	45°	135°
dLBPα	45°	45°	90°	90°	135°	90°	45°	0°	90°	45°	90°	45°	45°	90°	135°
nLBPd	2	2	1	1	1	1	2	1	3	1	1	1	1	1	1
LECTP	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
LESTP	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
LTP	3	3	2	3	3	2	2	3	2	3	3	3	4	1	3
QBP	2	1	1	1	2	1	3	2	2	1	1	2	3	2	2

databases. Indeed, it can be readily inferred from Table 4 that there is a significant performance drop for all the tested operators on 2D_Hela database (dataset 1 in Table 4) where their achieved recognition rate is below 75% and this despite the use of the sophisticated SVM classifier. This same finding can also be drawn for KTH-TIPS2b and Vistex datasets (datasets 7 and 14 in Table 4) where the classification rates obtained by the best performing descriptors are below 94% (93.02% by LCCMSP on KTH-TIPS2b and 91.08% by LECTP on Vistex).

- XCS-LBP, LOOP, LPQ, SMEPOP, QBP, LDN and many other evaluated methods are often ranked among the lowest performing descriptors on almost all the used texture databases.

- The results emerged from Table 4 reveals clearly that none of the evaluated state-of-the-art operators performs well over all the used texture datasets. The majority of them achieves good classification results on some datasets, while on the remainders ones, they perform worse. Considering for example the DRLBP operator, it achieves good performance on Bonn BTF, Brodatz and KTH-TIPS datasets, but performs worse on the other used datasets.

Table 4: Average accuracy (%) over 100 splits on all databases calculated as summarized in Algorithm 3.

Descriptor	Database number														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
ARCS-LBP	48.26	96.13	93.38	7.38	96.42	98.15	61.42	35.27	54.68	86.44	80.39	74.17	36.07	42.91	69.71
AELTP	59.01	99	99.73	95.15	97.18	97.7	88.2	99.87	89.34	99.38	99.17	86.94	69.69	77.32	91.88
ALTP	68.91	99.34	99.99	97.46	94.72	**100**	92.27	99.69	92.98	99.93	99.39	84.68	77.28	77.51	94.42
CRLBP-S-MxC	37.01	92.75	97.56	7.87	93.78	98.75	61.26	29.47	57.63	85.78	79.35	69.23	37.93	41.47	67.94
CSALTP	48.96	94.01	99.59	87.21	86.67	92.8	78.93	98.46	73.23	95.55	90.34	74.99	59.91	49.71	84.96
DBC	25.34	92.66	85.4	7.17	90	97.75	43.34	11.96	40.84	87.11	78.37	64.39	41.33	18.79	45.81
DC	67.42	97.39	99.81	96.9	97.21	99.75	86.98	99.61	93.28	99.3	99.26	83.45	72.76	72.26	91.94
dLBPα	31.65	87.81	93.15	2.5	81.91	90.5	50.43	18.48	45.02	87.89	74.51	61.76	35.26	22.61	43.33
DRLBP	33.06	95.24	97.01	3.38	90.37	98.85	51.85	11.46	47.47	88.27	82.94	70.61	43.54	38.61	60.01
DSLGS	18.42	98.86	96.74	3.24	94.59	96.85	48.7	7.77	48.42	94.76	92.07	75.95	51.34	26.76	58.94
ELGS	21.06	94.81	96.05	5.1	93.13	96.9	50.93	9.15	49.95	93.4	91.34	76.13	52.9	27.13	61.18
LBP	25.04	91.94	98.37	5.88	92.41	**100**	47.1	10.9	48.27	94	88.64	72.88	46.05	27.85	61.36
nLBPd	17.94	96.29	97.58	1.98	93.28	97.75	55.64	7.36	46.52	95.75	92.93	74.21	50.84	24.89	54.51
LCCMSP	67.97	99.65	99.96	97.72	97.31	**100**	**93.02**	99.93	94.17	99.82	99.72	87.8	78.93	84.58	95.82
LDN	13.22	90.84	86.98	5.11	80.71	93	44.5	7.89	43.11	80	72.06	62.79	38.47	20.5	52.27
LDTP	34.87	99.24	99.51	5.27	94.86	**100**	54.3	11.48	51.21	95.39	93.72	76.74	52.38	28.55	58.4
LECTP	69.31	**100**	99.98	92.52	97.15	**100**	84.47	99.85	84.29	97.99	96.51	83.53	74.08	**91.08**	95.99
LESTP	69.96	**100**	99.99	92.61	97.09	**100**	84.66	99.85	84.73	98.05	96.77	83.7	74.32	91.07	95.98
LNDP	27.46	96.13	97.12	3.72	90.24	96.65	56.09	13.66	45.47	90.42	86.42	67.98	47.85	25.75	51.16
LPQ	8.38	97.42	95.6	1.59	96.42	96.5	52.57	3.37	49.36	94.72	92.93	74.43	50.27	35.67	64.42
LOOP	6.81	72.66	67.52	2.3	78.5	88.8	48.75	7.72	35.53	62.91	52.59	35.58	21.44	31.52	52.55
LTP	68.21	99.49	99.99	97.41	95.01	**100**	92.2	99.88	91.88	99.86	99.48	85.52	75.53	79.54	93.73
MMEPOP	10.51	95.81	93	3.95	90.82	97	42.32	5.48	44.48	84.87	80.14	64.04	41.73	24.3	54.36
MOW-SLGS	17.67	98.01	97.52	2.96	94.6	99.4	52.03	5.94	50.14	95.97	93.48	75.98	51.19	28.6	62.77
MSJLBP	7.68	98.85	97.8	96.74	83.41	96.95	90.81	99.86	93.31	99.06	98.95	86.36	71.32	69.52	74.96
MLD-CBP	**71.58**	**100**	**100**	**97.88**	**100**	**100**	90.32	**99.96**	**95.29**	**99.97**	**99.73**	**88.7**	**81.22**	85.05	**96.59**
QBP	23.85	83.28	89.37	9.83	91.19	97	56.19	27.4	53.23	79.65	71.84	48.8	33.92	28.34	52.96
RALBGC	37.38	98.42	99.96	6.91	94.86	98.2	69.3	16.83	54.75	95.09	93.17	74.67	53.54	34.31	66.19
SMEPOP	15.54	89.71	97.07	4.08	91.42	97.85	43.41	8.46	45.68	89.53	86.2	72.27	43.82	22.34	57.65
WLD	57.63	**100**	99.25	90.78	94.68	98.4	86.07	99.6	92.58	98.38	98.32	88.18	74.68	82.72	94.17
XCS-LBP	26.56	77.84	85.13	11.01	68.33	86.3	52.1	39.22	41.11	75.21	72.72	29.19	35.99	34.13	37.81

We can express the same remark for many other methods like ELGS, DSLGS, CSALTP, AELTP, etc.

- Regarding the proposed MLD-CBP operator, it is clearly noticeable that it provides better classification performances which are competitive or better than all the tested state-of-the-art methods. Indeed, the MLD-CBP descriptor is emerged to perform well on almost all the tested datasets as it is often found among the best performing methods regarding overall recognition rate, allowing to achieve good classification results on almost all the tested texture datasets. Note that, MLD-CBP operator manages to differentiate all classes perfectly over 4 databases which are BonnBTF, Brodatz, JerryWu and KTH-TIPS (datasets number: 2, 3, 5 and 6 in Table 4). Moreover, it keeps its strengths realizing a score which is, on the one side, very close to 100% on Kylberg, Outex TC_00000 and Outex TC_00001 datasets (99.96% on dataset 8, 99.97% on dataset 10 and 99.73% on dataset 11 in Table 4) and on the other side, higher than 95% on CureT, MondialMarmi and XU HR datasets (97.88% on dataset 4, 95.29% on dataset 9 and 96.59% on dataset 15, respectively). It is worth mentioning that although the parametric methods, like AELTP, ALTP, CSALTP, DBC, dLBPα, nLBPd, LECTP, LESTP, LTP and QBP, regarded as "optimized" (since their parameter values are tuned for each tested dataset), are markedly less performing than the proposed method.

- Considering the ranking between the evaluated descriptors within each used texture dataset (cf. Table 5), the best operator in terms of average accuracy is MLD-CBP which performs significantly and consistently the best for thirteen datasets: 2D-HeLa, Bonn BTF, Brodatz, CUReT, Jerry Wu, KTH-TIPS, Kylberg, MondialMarmi, OuTeX TC-00000, OuTeX TC-00001, OuTeX TC-00013, Vis-

Table 5: Top Down ranking results of the evaluated methods on each database to highlight the top 5 and the worst performing descriptors

2D-HeLa	BonnBTF	Brodatz	CUReT	JerryWu	KTH-TIPS	KTH-TIPS2b	Kylberg	MondialMarmi	Outex-TC-00000	Outex-TC-00001	Outex-TC-00013	UIUCTex	VisTex	XU HR
MLD-CBP	MLD-CBP	MLD-CBP	MLD-CBP	MLD-CBP	MLD-CBP	LCCMSP	MLD-CBP	MLD-CBP	MLD-CBP	MLD-CBP	MLD-CBP	LECTP	MLD-CBP	MLD-CBP
LESTP	LESTP	ALTP	LCCMSP	LCCMSP	ALTP	ALTP	LCCMSP	LCCMSP	ALTP	LCCMSP	WLD	LESTP	LCCMSP	LECTP
LECTP	LECTP	LESTP	ALTP	DC	LBP	LTP	LTP	MSJLBP	LTP	LTP	LCCMSP	MLD-CBP	ALTP	LESTP
ALTP	WLD	LTP	LTP	AELTP	LECTP	MSJLBP	AELTP	DC	LCCMSP	ALTP	AELTP	LCCMSP	LTP	LCCMSP
LTP	LCCMSP	LECTP	DC	LECTP	LESTP	MLD-CBP	MSJLBP	ALTP	AELTP	DC	MSJLBP	WLD	WLD	ALTP
LCCMSP	LTP	LCCMSP	MSJLBP	LESTP	LTP	AELTP	LECTP	WLD	DC	AELTP	LTP	LTP	LESTP	WLD
DC	ALTP	RALBGC	AELTP	ARCS-LBP	LCCMSP	DC	LESTP	LTP	MSJLBP	MSJLBP	ALTP	ALTP	LECTP	LTP
AELTP	LDTP	DC	LESTP	LPQ	LDTP	WLD	ALTP	AELTP	WLD	WLD	LESTP	AELTP	DC	DC
WLD	AELTP	AELTP	LECTP	LTP	DC	LESTP	DC	LESTP	LESTP	LESTP	LECTP	DC	MSJLBP	AELTP
CSALTP	DSLGS	CSALTP	WLD	RALBGC	MOWSLGS	LECTP	WLD	LECTP	LECTP	LECTP	DC	MSJLBP	AELTP	CSALTP
ARCS-LBP	MSJLBP	LDTP	CSALTP	LDTP	DRLBP	CSALTP	CSALTP	CSALTP	MOWSLGS	LDTP	LDTP	CSALTP	CSALTP	MSJLBP
RALBGC	RALBGC	WLD	XCS-LBP	ALTP	CRLBP	RALBGC	XCS-LBP	CRLBP	*n*LBP*d*	MOWSLGS	ELGS	ARCS-LBP	RALBGC	ARCS-LBP
CRLBP	MOWSLGS	LBP	QBP	WLD	WLD	ARCS-LBP	ARCS-LBP	RALBGC	CSALTP	RALBGC	MOWSLGS	CRLBP	ELGS	CRLBP
LDTP	LPQ	MSJLBP	CRLBP	MOWSLGS	RALBGC	CRLBP	CRLBP	ARCS-LBP	LDTP	*n*LBP*d*	DSLGS	DRLBP	LDTP	RALBGC
DRLBP	DC	*n*LBP*d*	ARCS-LBP	DSLGS	ARCS-LBP	QBP	QBP	QBP	RALBGC	LPQ	CSALTP	LPQ	DSLGS	LPQ
*d*LBPα	*n*LBP*d*	CRLBP	DBC	CRLBP	SMEPOP	LNDP	*d*LBPα	LDTP	DSLGS	DSLGS	RALBGC	RALBGC	MOWSLGS	MOWSLGS
LNDP	ARCS-LBP	MOWSLGS	RALBGC	*n*LBP*d*	DBC	*n*LBP*d*	RALBGC	MOWSLGS	LPQ	ELGS	LPQ	XCS-LBP	*n*LBP*d*	LBP
XCS-LBP	LNDP	LNDP	LBP	ELGS	*n*LBP*d*	LDTP	LNDP	ELGS	LBP	CSALTP	*n*LBP*d*	LOOP	LPQ	ELGS
DBC	MMEPOP	SMEPOP	LDTP	LBP	AELTP	LPQ	DBC	LPQ	ELGS	LBP	ARCS-LBP	MOWSLGS	LNDP	DRLBP
LBP	DRLBP	DRLBP	LDN	SMEPOP	MMEPOP	XCS-LBP	LDTP	DSLGS	LNDP	LNDP	LBP	LDTP	LBP	DSLGS
QBP	ELGS	DSLGS	ELGS	QBP	QBP	MOWSLGS	DRLBP	LBP	SMEPOP	SMEPOP	SMEPOP	QBP	SMEPOP	LDTP
ELGS	CSALTP	ELGS	SMEPOP	MMEPOP	MSJLBP	DRLBP	LBP	DRLBP	DRLBP	DRLBP	DRLBP	LBP	DRLBP	SMEPOP
DSLGS	CRLBP	LPQ	MMEPOP	DRLBP	ELGS	ELGS	ELGS	*n*LBP*d*	*d*LBPα	ARCS-LBP	CRLBP	ELGS	MMEPOP	*n*LBP*d*
*n*LBP*d*	DBC	ARCS-LBP	LNDP	LNDP	DSLGS	*d*LBPα	SMEPOP	SMEPOP	DBC	MMEPOP	LNDP	DSLGS	DBC	MMEPOP
MOWSLGS	LBP	*d*LBPα	DRLBP	DBC	LNDP	LOOP	LDN	LNDP	ARCS-LBP	CRLBP	DBC	LNDP	LDN	QBP
SMEPOP	LDN	MMEPOP	DSLGS	CSALTP	LPQ	DSLGS	DSLGS	*d*LBPα	CRLBP	DBC	MMEPOP	*n*LBP*d*	CRLBP	LOOP
LDN	SMEPOP	QBP	MOWSLGS	MSJLBP	LDN	LBP	LOOP	MMEPOP	MMEPOP	*d*LBPα	LDN	MMEPOP	ARCS-LBP	LDN
MMEPOP	*d*LBPα	LDN	*d*LBPα	*d*LBPα	CSALTP	LDN	*n*LBP*d*	LDN	LDN	XCS-LBP	*d*LBPα	*d*LBPα	XCS-LBP	LNDP
LPQ	QBP	DBC	LOOP	LDN	*d*LBPα	SMEPOP	MOWSLGS	XCS-LBP	QBP	LDN	QBP	SMEPOP	*d*LBPα	DBC
MSJLBP	XCS-LBP	XCS-LBP	*n*LBP*d*	LOOP	LOOP	DBC	MMEPOP	DBC	XCS-LBP	QBP	LOOP	LDN	QBP	*d*LBPα
LOOP	LOOP	LOOP	LPQ	XCS-LBP	XCS-LBP	MMEPOP	LPQ	LOOP	LOOP	LOOP	XCS-LBP	DBC	LOOP	XCS-LBP

tex and XU HR (datasets 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 15 in Table 4). Furthermore, it is in the top 5 and 3 descriptors on KTH-TIPS2b and UIUCTex datasets (datasets 7 and 13 in Table 4), respectively. The second best method is LCCMSP which is in the top 5 descriptors on twelve datasets (also ranked top 2 descriptors on six datasets) followed by LESTP which is in the top 5 descriptors on six datasets.

- From Table 6, it can be inferred that the MLD-CBP descriptor outperformed 434 times (of 450 comparisons) the evaluated handcrafted methods overall the 15 databases, scoring a win ratio of 96.44% followed by LCCMSP with 91.33% (411 times). They were the only methods that exceeded 400 wins.
- All the top 6 ranked mathods have a standard deviation less than 10% with an overall performance above 90%. The MLD-CBP managed to ensure a good balance between the performance and stability regarding its corresponding highest average accuracy of 93.75% and lowest standard deviation of 8.39%. The most unstable method is *n*LBP*d* with a standard deviation of 33.93%.

Moreover, we run an experiment that calculates the elapsed time to extract the feature vector from a given input image that has a resolution of 256 by 256 pixels. The obtained times are

Table 6: Global metrics and execution time based comparison of the evaluated descriptors to find the overall best performing and stable method.

Descriptor	Rank	Ratio	Wins	Avg Acc	Std	Exec Time (ms)
MLD-CBP	1	0.9644	434/450	93.7527	8.3993	12.8959
LCCMSP	2	0.9133	411/450	93.0933	9.1426	8.7802
LTP	3	0.8600	387/450	91.8487	9.7955	5.2599
ALTP	4	0.8533	384/450	91.9047	9.7149	4.5366
LESTP	5	0.8244	371/450	91.2520	9.3857	9.6884
LECTP	6	0.8111	365/450	91.1167	9.5402	12.282
DCLBP	7	0.7822	352/450	90.4880	10.9388	7.4104
AELTP	8	0.7756	349/450	89.9707	11.9241	10.2032
WLD	9	0.7689	346/450	90.3627	11.2148	26.7729
MSJLBP	10	0.6533	294/450	84.3720	22.8861	19.4633
RALBGC	11	0.5867	264/450	66.2387	30.2856	6.0754
LDTP	12	0.5533	249/450	63.7280	32.1640	10.9535
CSALTP	13	0.5489	247/450	81.0213	16.1151	5.4535
AECLBP-S-MxC	14	0.4800	216/450	65.3853	26.5702	8.1974
CRLBP-S-MxC	15	0.4556	205/450	63.8520	27.2109	6.8522
MOW-SLGS	16	0.4556	205/450	61.7507	34.0842	8.155
LBP	17	0.3911	176/450	60.7127	32.0140	4.2173
ELGS	18	0.3822	172/450	61.2773	32.3889	4.4186
DSLGS	19	0.3756	169/450	60.8940	33.6797	4.0554
LPQ	20	0.3733	168/450	60.9100	34.6120	6.3362
*n*LBP*d*	21	0.3689	166/450	60.4980	33.9375	5.5722
DRLBP	22	0.3667	165/450	60.8447	30.2326	8.2082
LNDP	23	0.3200	144/450	59.7413	31.2260	3.8362
QBP	24	0.2756	124/450	56.4567	26.9429	3.6058
SMEPOP	25	0.2733	123/450	57.6687	32.8940	7.0674
DBC	26	0.2133	96/450	55.3507	30.5476	5.2021
MMEPOP	27	0.1933	87/450	55.5207	32.8497	7.3087
XCS-LBP	28	0.1933	87/450	51.5100	23.2085	5.0564
*d*LBPα	29	0.1778	80/450	55.1207	28.6994	3.9467
LDN	30	0.1356	61/450	52.7633	29.9893	6.4615
LQCH	31	0.0733	33/450	44.3453	26.2586	7.6804

included in Table 6 and illustrated graphically in Figure 11 for more readability. To ensure a fair and comprehensive analysis between the evaluated descriptors, this figure plots also the win ratio of each method, in order to perform a global evaluation based on these two criteria (computation time and win ratio). We can see that our proposed MLDCBP descriptor takes 12ms to compute the feature vector which is very fast, regarding the number of generated patterns (i.e., 1024), with high performance, compared to the LBP descriptor (as reference) requiring 4ms to extract the feature vector of size 256 patterns. Moreover, one can see from Figure 11 that the proposed descriptor presents the best tradeoff as it is the more stable one compared to the other tested descriptors, while requiring acceptable computation time. Indeed, if we consider for example the LCCMSP and LTP methods, they reached the highest win ratios after the MLDCBP method (91.33% and 86%, respectively) with an execution time of 8.78 ms for LCCMSP and 5.26 ms for LTP. Compared to the second ranked method (i.e., LCCMSP), the proposed MLDCBP delivered 5% extra of win ratio (i.e., more stability overall the databases) with only 25% more execution time than LCCMSP. Thus, the MLDCBP ensures a good tradeoff between speed and performance.

All the discussed points lead to the same findings, which prove that the proposed MLD-CBP method, which is free of tuning parameter setup, shows steady classification performance. It presents, in fact, a significant performance stability against the tested state-of-the-art descriptors on all the used texture datasets. These results and their analysis show that the proposed method works very well on a wide selection of different texture datasets which fed to the SVM classifier with relevant and distinguishing features. This firmly demonstrates that the proposed operator makes effective the use of micro-structures and relationships between pixels within 5×5 window.

5.4. The impact of SVM γ value

In this subsection, the objective is to investigate the impact of the proposed γ parameter calculation formula on the SVM based classification results. In order to highlight the effectiveness of the proposed formula, a performance evaluation of the proposed descriptor using, on the one hand, the default γ value, the values reported in [27] and [28] and, on the other hand, the proposed user-specified γ parameter calculating formula (cf. Eq. 19), is more than desirable. Table 7 and Figure 12 illustrate the obtained average accuracies over all the tested texture datasets to show the performance stability and the improvement that our formula provides.

As it's clearly stated in Table 7 and seen in Figure 12, the proposed γ calculation method shows good performance stability as it allows the proposed MLD-CBP operator to reach higher scores than that achieved using the other γ calculation techniques as well as the Nearest Neighbor classifier on almost all the tested datasets (12/15 databases). Note that the γ value reported in [27], unlike that proposed in [28], yields competitive accuracy over only four datasets, and performs well on six databases. However, it seems to be unsuitable for the other nine tested databases. In addition, the default γ value provided with the lib-svm Toolbox realizes promising classification accuracies but it suffers performance drop in some databases. It is worth mentioning that the Nearest Neighbor classifier emerged to reach also good results, which points out that the proposed MLD-CBP descriptor provides distinguishing feature vectors and its performance is not attached to a given classifier.

5.5. Implementation and Execution

The texture classification experiments have been performed on a HP ProDesk with Core i7 Processor 4.0 GHz with turbo boost technology and 16GB of RAM, running with Ubuntu 16.04 LTS (Xenial Xerus) operating system. The descriptors and the classification system have been implemented in MATLAB® R2016b environment. The experiments reported in this paper have taken days of computer time that is due to the diversity of the experiments on 15 databases and the large number of evaluated descriptors and adopting the SVM in the classification phase. For reproducible research results, all data required to replicate the experiments (i.e.: source codes, images and subdivisions into train and validation sets) are available upon request to the corresponding author.

6. Conclusion

In this paper, we have designed a novel texture operator, referred to as Multi Level Directional Cross Binary Patterns (MLD-CBP), which proved to be a computationally and conceptually simple yet efficient descriptor for image texture modeling. MLD-CBP operator, which manipulates a 5×5 sub-block size, is based on combining direction and radius concepts. It

Figure 11: The execution time taken by each evaluated method to extract the feature vector from a 256×256 image vs win ratio. This figure helps to find balance between performance and resources consumption.

Table 7: Average accuracy (%) on each database according to different γ values and NN classifier.

	Database $N°$														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$1/(2 \times 13^{4.1})$ [27]	56.63	**100**	**100**	88.04	99.87	**100**	90.82	97.89	92.09	**99.98**	99.38	84.86	78.14	**85.37**	**97.52**
$1/(2 \times 5^{2})$ [28]	56.51	97.15	80.57	1.5	2.69	97.75	89.94	4.23	92.32	99.78	99.38	84.78	0.34	10.56	12.22
Default value	56.55	**100**	99.95	86.51	91.58	**100**	90.82	97.66	92.55	**99.98**	99.38	84.86	0.36	83.47	90.99
NN City-Block	63.5	**100**	**100**	92.8	**100**	**100**	**91.36**	98.51	92.07	99.91	99.6	83.9	78.84	70.78	94.76
Proposed	**71.58**	**100**	**100**	**97.88**	**100**	**100**	90.32	**99.96**	**95.29**	**99.98**	**99.73**	**88.7**	**81.22**	85.05	96.59

is, in fact, built by considering two cross encoders exploiting simultaneously multi-radial and multi-orientation information and integrating local gray level averages in the thresholding process. Thanks to this fact, the proposed MLD-CBP descriptor has the ability to extract more relevant information and achieves superior performance than the existing descriptors as it has been proved through the comparative assessments. For classification purpose, the test images are classified through a supervised image classification task using the SVM classifier configured on the RBF kernel, by adopting a new technique for calculating the γ value automatically according the considered database and the description method. A comprehensive evaluation of the proposed MLD-CBP descriptor is performed on fifteen challenging representative widely-used texture datasets, with comparison to 30 recent most promising state-of-the-art methods to disclose meaningful statements. As expected, the MLD-CBP descriptor coupled with the automated SVM classifier managed to outperform the 30 evaluated LBP variants showing good stability and proving its high discriminative abilities in describing and then correctly classifying texture images of 15 databases. Moreover, the analysis of the experimental results also indicated that the proposed MLD-CBP description method provided good classification results when coupled with the basic Nearest Neighbor classifier, which proves that the achieved performance is not attached to the nature of the used classifier.

As future work, we intend to enhance the performance of the proposed classification framework by introducing dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to get extra class separation and variance in the feature, which may boost the SVM classifier robustness. Moreover, we plan to develop a deep feature version of the proposed MLD-CBP descriptor based on Pixel Difference Vectors (PDVs), then investigate the performance on other challenging applications such as writer identification, emotional state classification, etc.

7. Acknowledgments

The authors gratefully acknowledge the scholarships received from CNRST-Maroc (Centre National de la Recherche Scien-

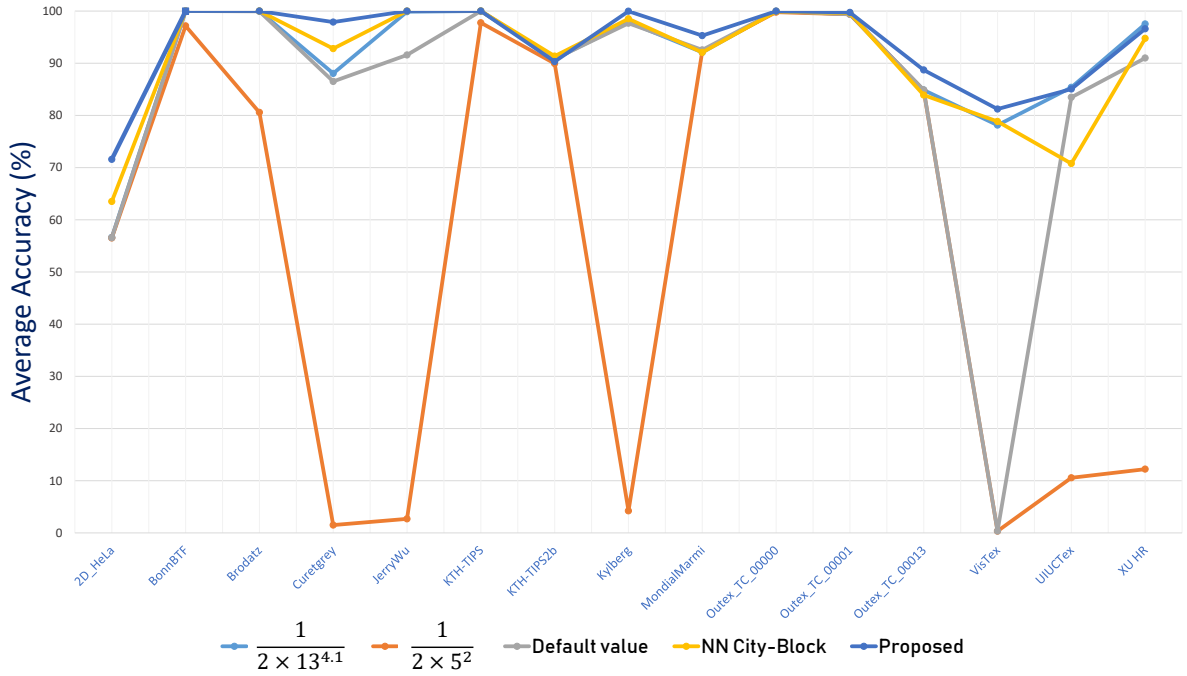

Figure 12: Performance and stability comparison according to γ values obtained using the proposed formula and those of the literature as well as 1-NN classifier.

tifique et Technique) grant number 7UIT2017 and the French government (Eiffel scholarship) grant number 919849L.
The authors would like to thank the anonymous reviewers for their constructive comments.

Conflict of Interest Statement

The authors declare that there is no conflict of interest

References

[1] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, in: International conference on image and signal processing, Springer, 2008, pp. 236–243.

[2] U. Raghavendra, U. R. Acharya, A. Gudigar, R. Shetty, N. Krishnananda, U. Pai, J. Samanth, C. Nayak, Automated screening of congestive heart failure using variational mode decomposition and texture features extracted from ultrasound images, Neural Computing and Applications 28 (10) (2017) 2869–2878. doi:10.1007/s00521-017-2839-5.
URL https://doi.org/10.1007/s00521-017-2839-5

[3] M. Maktabdar Oghaz, M. A. Maarof, M. F. Rohani, A. Zainal, S. Z. M. Shaid, An optimized skin texture model using gray-level co-occurrence matrix, Neural Computing and Applications (Jul 2017). doi:10.1007/s00521-017-3164-8.
URL https://doi.org/10.1007/s00521-017-3164-8

[4] L. Moraru, S. Moldovanu, L. T. Dimitrievici, A. S. Ashour, N. Dey, Texture anisotropy technique in brain degenerative diseases, Neural Computing and Applications 30 (5) (2018) 1667–1677. doi:10.1007/s00521-016-2777-7.
URL https://doi.org/10.1007/s00521-016-2777-7

[5] J. Zhang, T. Tan, Brief review of invariant texture analysis methods, Pattern recognition 35 (3) (2002) 735–747.

[6] O. Tuzel, F. Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: European conference on computer vision, Springer, 2006, pp. 589–600.

[7] Z.-H. Huang, W.-J. Li, J. Shang, J. Wang, T. Zhang, Non-uniform patch based face recognition via 2d-dwt, Image Vis Comput 37 (2015) 12–19.

[8] T. Abhishree, J. Latha, K. Manikantan, S. Ramachandran, Face recognition using gabor filter based feature extraction with anisotropic diffusion as a pre-processing technique, procedia Computer Science 45 (2015) 312–321.

[9] M. Topi, O. Timo, P. Matti, S. Maricor, Robust texture classification by subsets of local binary patterns, in: Pattern Recognition, 2000. Proceedings. 15th International Conference on, Vol. 3, IEEE, 2000, pp. 935–938.

[10] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans Image Process 19 (6) (2010) 1635–1650.

[11] W. Yang, Z. Wang, B. Zhang, Face recognition using adaptive local ternary patterns method, Neurocomputing 213 (2016) 183–190.

[12] T. Ahonen, E. Rahtu, V. Ojansivu, J. Heikkila, Recognition of blurred faces using local phase quantization, in: 19th International Conference on Pattern Recognition, 2008, pp. 1–4.

[13] A. Chahi, Y. Ruichek, R. Touahni, et al., Local directional ternary pattern: A new texture descriptor for texture classification, Computer vision and image understanding 169 (2018) 14–27.

[14] T. Jabid, M. H. Kabir, O. Chae, Local directional pattern (ldp) for face recognition, in: Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on, IEEE, 2010, pp. 329–330.

[15] H. Zeng, J. Chen, X. Cui, C. Cai, K.-K. Ma, Quad binary pattern and its application in mean-shift tracking, Neurocomputing 217 (2016) 3–10.

[16] M. Verma, B. Raman, Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval, Multimedia Tools and Applications 77 (10) (2018) 11843–11866.

[17] Q. Kou, D. Cheng, L. Chen, K. Zhao, A multiresolution gray-scale and rotation invariant descriptor for texture classification, IEEE Access 6 (2018) 30691–30701.

[18] Y. El Merabet, Y. Ruichek, Local concave-and-convex micro-structure patterns for texture classification, Pattern Recognition 76 (2018) 303–322.

[19] I. El Khadiri, M. Kas, Y. El Merabet, Y. Ruichek, R. Touahni, Repulsive-and-attractive local binary gradient contours: New and efficient feature descriptors for texture classification, Information Sciences (2018).

[20] F. Ouslimani, A. Ouslimani, Z. Ameur, Rotation-invariant features based on directional coding for texture classification, Neural Computing and Applications (2018) 1–8.

[21] F. M. Khellah, Texture classification using dominant neighborhood structure, IEEE Trans. Image Process 20 (11) (2011) 3270–3279.

[22] F. Ouslimani, A. Ouslimani, Z. Ameur, Rotation-invariant features based on directional coding for texture classification, Neural Computing and

Applications (2018) 1–8.
[23] Y. Kaya, Ö. F. Ertuğrul, R. Tekin, Two novel local binary pattern descriptors for texture analysis, Applied Soft Computing 34 (2015) 728–735.
[24] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6) (1973) 610–621. doi:10.1109/TSMC.1973.4309314.
[25] Z. Pan, X. Wu, Z. Li, Z. Zhou, Local adaptive binary patterns using diamond sampling structure for texture classification, IEEE signal processing letters 24 (6) (2017) 828–832.
[26] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM transactions on intelligent systems and technology (TIST) 2 (3) (2011) 27.
[27] K. I. Kim, K. Jung, S. H. Park, H. J. Kim, Support vector machines for texture classification, IEEE transactions on pattern analysis and machine intelligence 24 (11) (2002) 1542–1550.
[28] S. Li, J. T. Kwok, H. Zhu, Y. Wang, Texture classification using the support vector machines, Pattern recognition 36 (12) (2003) 2883–2893.
[29] A. Fernández, M. X. Álvarez, F. Bianconi, Texture description through histograms of equivalent patterns, Journal of mathematical imaging and vision 45 (1) (2013) 76–102.
[30] S. Hossain, S. Serikawa, Texture databases–a comprehensive survey, pattern recognition letters 34 (15) (2013) 2007–2022.
[31] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, W. Gao, Wld: A robust local image descriptor, IEEE Trans Pattern Anal Mach Intell 32 (9) (2010) 1705–1720.
[32] B. Zhang, L. Zhang, D. Zhang, L. Shen, Directional binary code with application to polyu near-infrared face database, Pattern Recognit Lett 31 (14) (2010) 2337–2344.
[33] A. R. Rivera, J. R. Castillo, O. O. Chae, Local directional number pattern for face analysis: Face and expression recognition, IEEE Trans Image Process 22 (5) (2013) 1740–1752.
[34] X. Qi, Y. Qiao, C.-G. Li, J. Guo, Multi-scale joint encoding of local binary patterns for texture and material classification., in: BMVC, 2013.
[35] Y. Zhao, W. Jia, R.-X. Hu, H. Min, Completed robust local binary pattern for texture classification, Neurocomputing 106 (2013) 68–76.
[36] S. K. Vipparthi, S. K. Nagar, Local extreme complete trio pattern for multimedia image retrieval system, International Journal of Automation and Computing 13 (5) (2016) 457–467.
[37] S. Dong, J. Yang, C. Wang, Y. Chen, D. Sun, A new finger vein recognition method based on the difference symmetric local graph structure (dslgs), International Journal of Signal Processing, Image Processing and Pattern Recognition 8 (10) (2005) 71–80.
[38] S. K. Vipparthi, S. Murala, S. K. Nagar, A. B. Gonde, Local gabor maximum edge position octal patterns for image retrieval, Neurocomputing 167 (2015) 336–345.
[39] S. Dong, J. Yang, Y. Chen, C. Wang, X. Zhang, D. S. Park, Finger vein recognition based on multi-orientation weighted symmetric local graph structure., KSII Transactions on Internet & Information Systems 9 (10) (2015).
[40] C. Silva, T. Bouwmans, C. Frélicot, An extended center-symmetric local binary pattern for background modeling and subtraction in videos, in: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015, 2015.
[41] K. Song, Y. Yan, Y. Zhao, C. Liu, Adjacent evaluation of local binary pattern for texture classification, Journal of Visual Communication and Image Representation 33 (2015) 323–339.
[42] R. Mehta, K. Egiazarian, Dominant rotated local binary patterns (drlbp) for texture classification, Pattern Recognition Letters 71 (2016) 16–22.
[43] H. K. Bashier, L. S. Hoe, L. T. Hui, M. F. Azli, P. Y. Han, W. K. Kwee, M. S. Sayeed, Texture classification via extended local graph structure, Optik-International Journal for Light and Electron Optics 127 (2) (2016) 638–643.
[44] T. Chakraborti, B. McCane, S. Mills, U. Pal, Loop descriptor: Local optimal-oriented pattern, IEEE Signal Processing Letters 25 (5) (2018) 635–639.
[45] Y. Elmerabet, et al., Attractive-and-repulsive center-symmetric local binary patterns for texture classification, Engineering Applications of Artificial Intelligence 78 (2019) 158–172.