



A Multiscale Hierarchical Threshold-Based Completed Local Entropy Binary Pattern for Texture Classification

Xiaochun Xu¹ · Yibing Li¹ · Q. M. Jonathan Wu²

Received: 21 May 2019 / Accepted: 25 July 2019 / Published online: 6 August 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Over the year, visual texture analysis has come to be recognized as one of the most important methods in the area of medical image analysis and understanding, face description and detection, and so on. The goal of texture descriptors is to capture the general characteristic of textures such as dependency as well as invariance properties. Among all the texture descriptors, the binary pattern family of algorithms achieves a great trade of representation efficiency and complexity. This work introduces an efficient discriminative texture descriptor for visual texture classification. Its main contribution is twofold: a multiscale thresholding framework based on hierarchical adaptive local partition to binary encoding and an efficient completed local entropy binary pattern (CLEBP) descriptor. The basic completed local entropy binary pattern is extended by multiscale thresholding framework with hierarchical thresholding to capture not only microstructure local patterns but also macrostructure texture information. Such extension improves the quality and discriminative factor of texture classification. Extensive experiments on three widely used benchmark texture databases (Outex, UIUC, and KTH-TIPS) proof the efficiency of the proposed visual texture descriptor and hierarchical thresholding strategy. Compared with some classical local binary pattern variants and many state-of-the-art methods, the proposed descriptor achieves competitive and superior texture classification performance. The results prove that the proposed method is a powerful and effective texture descriptor for visual texture classification.

Keywords Feature extraction · Multiscale thresholding · Completed local entropy binary pattern · Texture classification

Introduction

VISUAL texture classification is a challenging task in the field of pattern recognition, image processing, and computer vision. It is also a popular research topic during the past decades due to its applications such as face description and detection [1, 2], medical image analysis and understanding [3, 4], material analysis and classification [5], remote sensing [6, 7], and so on.

Feature descriptor and classifier designation are two key sub-problems for visual texture classification. As elaborated in [8], feature descriptors play a more critical role in texture classification. Several texture feature descriptors were proposed to explore effective texture representation. In many cases, high quality and robustness of texture representation

are of considerable importance in texture analysis. It is an intriguing issue to extract the completed representation information of texture classes as far as possible.

Binary pattern family methods have demonstrated tremendous success in texture analysis due to their excellent performance. As one of the most prominent texture descriptors, local binary pattern (LBP) methods were firstly introduced in [9] by Ojala et al. The basic idea of LBP is to extract local texture feature by encoding the sign of the difference vector. It has a lot of outstanding advantages, such as rotation invariance, robustness to illumination changes, ease of implementation, and no need for pre-training. Motivated by Ojala's [9] work in 2002, many LBP variants have been developed to be applied in diverse range of problems, such as texture classification [10], content-based image retrieval [11, 12], face recognition [13–15], medical image analysis [3, 4, 16], and object recognition [17]. In particular, Guo et al. [18] proposed the completed modeling of LBP operator for texture classification, named completed local binary pattern (CLBP). The main idea of CLBP is to represent the local region by three components: the sign of difference vector, the magnitude of difference vector, and the center pixel. It combines the three operators for

✉ Yibing Li
liyibing0920@126.com

¹ Department of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

² Department of Electrical and Computer Engineering University of Windsor, Windsor, ON, Canada

completed texture representation in joint and hybrid ways. For robustness to noise, Tan et al. [15] introduced local ternary patterns (LTP) for practical face recognition systems, which quantized and encoded the local difference into three levels. The experiments showed that it was more discriminative and less sensitive to noise than the original LBP. Others tried to explore some new ideas for texture representation. For example, Guo et al. [19] developed the local binary pattern variance (LBPV) to explore the local contrast information based on the variance weight. And Y. Zhao et al. [20] presented local binary count (CLBC) for rotation invariant texture classification, which abandoned the local binary structural information and extracted the local binary grayscale difference information directly. Although this method can represent the local texture effectively, it cannot express the visual microstructure. In addition, Zhao et al. [21] introduced a robust LBP framework named as completed robust local binary pattern (CRLBP) by replacing the center pixel by the average local gray level.

More recently, Song et al. [22] presented a simple and robust texture descriptor named adjacent evaluation local binary pattern (AELBP) for texture classification, which used an adjacent evaluation window to develop the threshold scheme. Guo et al. [23] proposed the scale selective LBP (SSLBP) to address scale variation for texture classification. R. Mehta et al. [24] developed a rotation-invariant and computationally efficient dominant rotated local binary pattern (DRLBP) and increased the discriminative power by discarding the redundant patterns. Pan et al. [25] introduced a feature-based local binary pattern (FbLBP). For FbLBP, the difference magnitude was described by mean and variance. And an adaptive local threshold was developed to encode the two features into binary codes. Wang et al. [26] proposed a discriminative and robust descriptor—local binary circumferential and radial derivative pattern (CRDP)—by fusing both local circumferential and radial derivative features. By extending the non-uniform patterns mapping technique, M. Shakoob et al. [27] presented the extended mapping local binary pattern (EMLBP) method for texture classification. In order to handle the complex local patterns, Z. Zhang et al. [28] developed an information-theoretic framework, namely completed discriminative local features (CDLF), which can learn completed discriminative local features for better representation of texture images.

In recent years, there has been an increased interest in deep features for image representation. And they have achieved significant performance improvement in various fields, especially visual feature extraction [29–32], such as image classification and retrieval [33–38], object recognition and detection [39, 40], place recognition [41, 42], human pose recovery [43, 44], shape recognition [45], emotion recognition [46], etc. The key to successful CNNs is that they can learn high-quality features from lots of annotated training data. However, the main drawbacks of CNNs applied in visual texture classification are the limited labeled training data. In addition, most

deep learning methods have difficulties concerning the high computational complexity and parameters tuning [47]. In 2017, Huang et al. [48] indicated that a structured and robust local descriptor can bring additional benefits to CNNs. Especially for texture classification, some classical methods can be successful complementary solutions to deep learning.

Great efforts have been contributed to satisfy the demands of discriminative and robust texture representation. However, there still remain some inborn defects of LBP. For instance, most of LBP variants were sensitive to noise. They lose local or global texture information. Lots of LBP variants used the simple pixel-level information to explore the local pattern, and the simple and direct representation was difficult to extract some deep abstract texture information effectively. So how to describe different local texture information and how to extract the information are the two fundamental issues for comprehensive understanding and representing the real-world textures.

In this paper, we aim to extract comprehensive and robust texture information. For keeping the simplicity and effectiveness of the original CLBP and addressing its weakness, we propose an efficient completed local entropy binary pattern descriptor and extend it by using the multiscale hierarchical threshold. The proposed methods have two advantages: (1) As it will be shown further, the multiscale hierarchical threshold easily incorporates the different scale texture information by hierarchical adaptive local partition, which makes the local binary more effectively. (2) The entropy binary pattern can reflect the spatial distribution characteristics between the center pixel and neighbor pixels effectively, which is complementary to the original CLBP descriptors. The main contributions of this paper include the following:

1. We introduce a multiscale thresholding scheme based on hierarchical adaptive local partition, which provides different scale texture information to help the optimization of the binary coding.
2. We propose an efficient completed local entropy binary pattern descriptor. As a statistical measure of randomness, the local entropy binary pattern can be used to characterize the spatial distribution between the center pixel and neighbor pixels.
3. We extend the CLEBP by using the multiscale hierarchical threshold to achieve a comprehensive understanding and representation, which can be more robust, discriminative, and comprehensive.
4. Extensive experimental evaluation on three representative texture datasets (Outex, UIUC, and KTH-TIPS) is conducted, showing that the proposed methods demonstrate superior performance to the classical LBP variants and recent state-of-the-art texture descriptors.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we introduce novel

multiscale thresholding. Then, we produce a novel completed local entropy binary pattern descriptor and extend it by using the multiscale thresholding. Experimental results of our proposed methods are presented in Section 4. Finally, Section 5 concludes the paper.

Related Work

Local Binary Pattern

Ojala et al. [9] proposed the LBP operator for texture analysis, which combined structural and statistical texture information by taking the difference of the center pixel with respect to its neighbors. In general, as illustrated in Fig. 1, the classical LBP operator can be defined as below:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad (1)$$

where g_c and g_p represent the gray value of the center pixel and circularly symmetric neighborhood pixels, respectively. P is the total number of the neighborhood pixels on a circle of $R(R > 0)$. $s()$ is the sign function. Figure 1 shows the standard pipeline of the LBP operator ($P = 8$), including the original pattern, binary pattern, and the weights corresponding to these neighbors. If the coordinates of g_c are (a, b) , then the coordinates of neighborhood pixels g_p can be given by $(a - R \sin(2\pi p/P), b + R \cos(2\pi p/P))$ for $p = 0, 1, \dots, P - 1$. It is worth noting when the neighbor point does not fall at integer coordinates, we can compute its value by interpolation.

In order to enhance the robustness to the rotation, the rotation invariant LBP is proposed by formula (2):

$$LBP_{P,R}^i = \min\{ROR(LBP_{P,R}, i) | i = 0, 1, \dots, P - 1\}, \quad (2)$$

where $ROR(x, i)$ represents an i -step circular bit-wise right shift on x . $LBP_{P,R}^i$ has a crude quantization for occurrence

statistics of rotation invariant patterns, leading to a dramatic reduction in feature dimensionality.

Furthermore, Ojala et al. [9] proposed a uniform rotation invariant texture descriptor $LBP_{P,R}^{riu2}$:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases}, \quad (3)$$

where

$$U(LBP_{P,R}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|, \quad (4)$$

where the superscript $riu2$ represents the rotation-invariant uniform patterns, which has $P + 2$ distinct output values.

In general, the histogram of LBP operator outputs is regarded as the final feature vector. And it is a very powerful tool for texture classification.

Completed Local Binary Pattern

Using the local difference sign-magnitude transform (LDSMT) and the center pixel intensity, Guo et al. proposed CLBP [18], which introduced the three local features including the center gray level (C), the sign (S), and the magnitude (M). Then, as illustrated in Fig. 2, the three operators, named CLBP_Center (CLBP_C), CLBP_Sign (CLBP_S), and CLBP_Magnitude (CLBP_M), were proposed to improve the discriminative capability and complementary information.

The CLBP_C is computed by the difference between the center pixel and the average gray level of the whole image. It is defined as

$$CLBP_C_{P,R} = t(g_c, c_I), t(x, c) = \begin{cases} 1, & x \geq c \\ 0, & x < c \end{cases}, \quad (5)$$

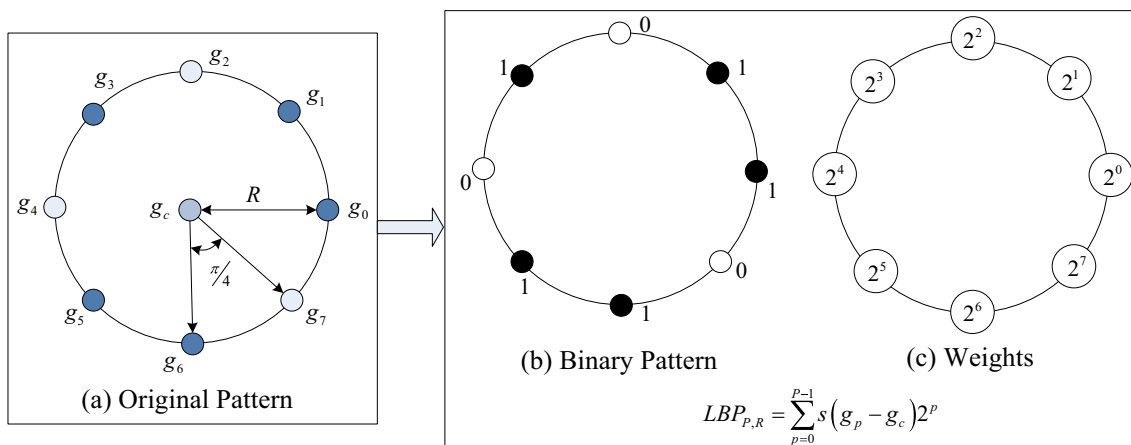


Fig. 1 The standard pipeline of LBP operator

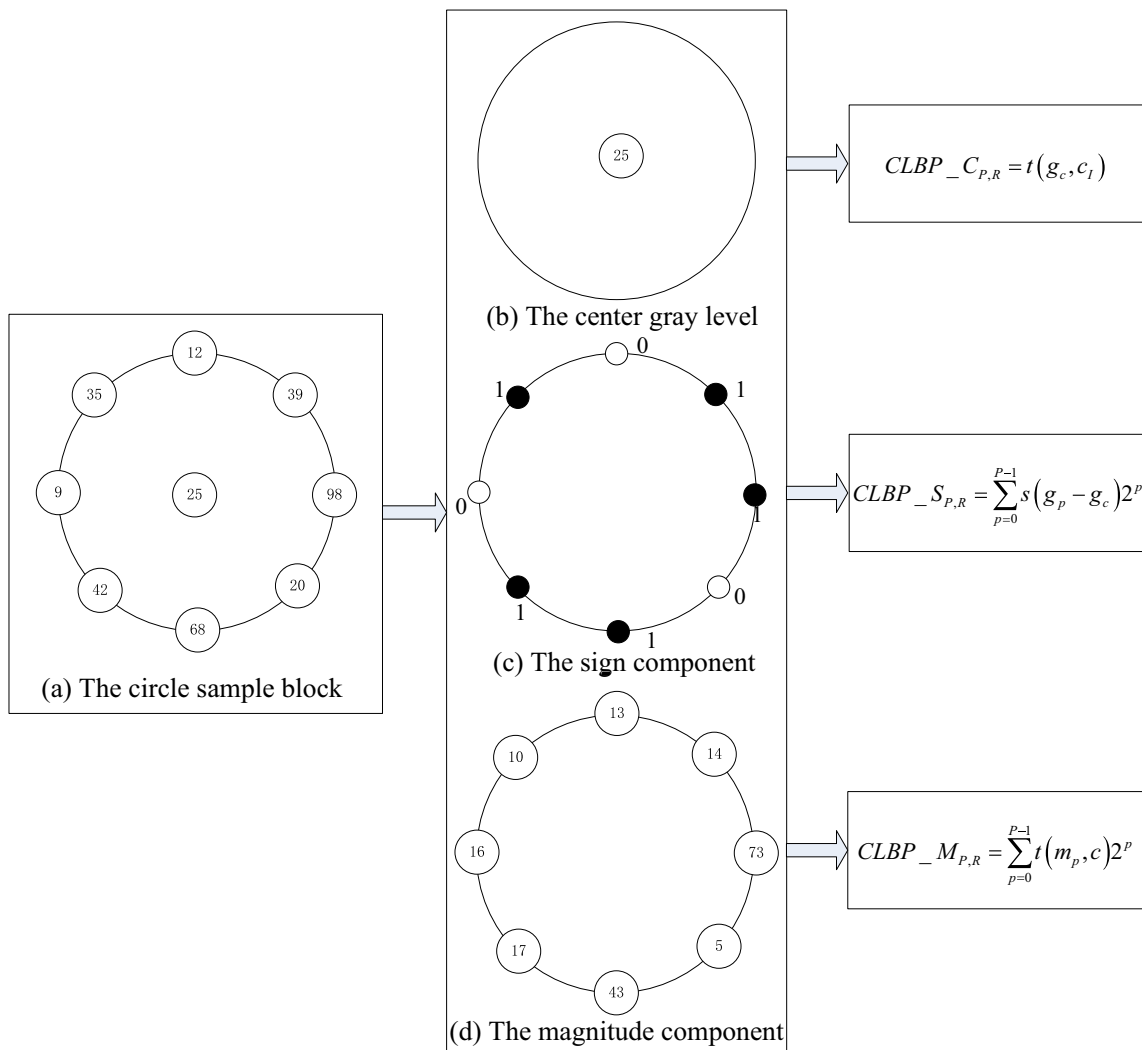


Fig. 2 The standard pipeline of CLBP operator

where the threshold c_l is set as the average gray level of the whole image. And $t(\cdot)$ is the threshold function.

The CLBP_S is equivalent to the classical LBP operator defined in (1).

The CLBP_M is the magnitude LBP operator described as follows:

$$CLBP_{M_{P,R}} = \sum_{p=0}^{P-1} t(m_p, c_m) 2^p, \quad m_p = |g_p - g_c|, \quad (6)$$

where $t(\cdot)$ is defined in (5). g_p and g_c are defined in (1). The threshold c_m is the mean value of m_p for the whole image.

As elaborated in [18], there are two ways to combine the CLBP_C, CLBP_S, and CLBP_M for improving the classification results, including jointly and hybridly. Then, we can get different feature combinations, such as CLBP_M/C, CLBP_S/M, CLBP_S_M/C, and CLBP_S/M/C. By analyzing the occurrence of combined histograms among different texture classes, the texture classification can be realized more exactly.

The main contribution of literature [18] is that it decomposed local difference into two complementary components: the signs and the magnitudes, which can capture more texture information on pixel level than the traditional LBP. As a matter of fact, all the features of CLBP are based on the pixels directly, it cannot represent the abstract and deep information. Noticeably, the CLBP cannot supply the completed texture information, so the completeness and accuracy cannot be guaranteed.

The Multiscale Hierarchical Threshold-Based Completed Local Entropy Binary Pattern

The Multiscale Hierarchical Threshold

In the conventional LBP and CLBP operators, the thresholds used for local binary encode are limited to single scale information (such as the central pixel value for the conventional LBP operator and the average gray level of the whole image

for the CLBP operator). However, they were not reasonable to capture and express texture information accurately and appropriately. Intuitively, as texture is a spatial phenomenon, texture representation should not be based on a single pixel but requires the analysis of pixel information over the local region. In order to solve the aforementioned difficulties, in this section, we propose a robust multiscale thresholding framework based on hierarchical adaptive local partition. The framework incorporates different scale texture information and overcomes the shortcoming of traditional thresholding.

Aiming at finding a threshold which contains multiscale information to encode the local binary pattern, in this study, we design a simple and robust multiscale thresholding framework based on hierarchical adaptive local partition. Figure 3 shows an illustrative example of the proposed multiscale hierarchical threshold (scale $N = 3$). Firstly, given a texture image, a preprocessing step is derived by standardization. Secondly, for each texture image, a multiscale hierarchical grid partitioning is built according to the value of N . Then, we calculate the mean of each grid block as hierarchical threshold based on multiscale grid partitioning. Finally, we average over all hierarchical thresholds as the multiscale threshold. The proposed multiscale hierarchical threshold framework extracts multiscale texture information from different hierarchical partitions, so it can capture more comprehensive and discriminative information for texture information encoding.

In order to give a compact mathematical formulation of the proposed multiscale thresholding framework, we define the multiscale thresholding as follows:

$$thre_{N,g_{i,j}} = \frac{1}{N} \sum_{k=1}^N lm_k(g_{i,j}), \tag{7}$$

where $g_{i,j}$ is the gray pixel of the pixel in the coordinates (i,j) . And $thre_{N,g_{i,j}}$ represents the multiscale thresholding in the coordinates (i,j) with scale N . $lm_k(g_{i,j})$ denotes the local mean of $g_{i,j}$ based on grid partitioning in scale k . $k \in [1, N]$ is the multiscale grid partitioning parameter, and N is the total number of the multiscale hierarchical partitioning.

Taking $N = 3$ as an example, Fig. 3 provides an illustration to explain the proposed multiscale hierarchical threshold. As shown in Fig. 3, the multiscale hierarchical threshold framework contains more comprehensive and robust texture information than traditional thresholding obviously. Note that, when $N = 1$, the multiscale thresholding framework is equivalent to the threshold settings in traditional CLBP.

In this paper, a set of experiments were conducted in the texture database (Outex_TC_00010) to select the optimal parameter N , which will be discussed later in Section 4.2.

The Local Entropy Binary Pattern

As we all know, how to extract full discriminative texture information is an important task for texture classification. However, in the conventional CLBP operators, the three local features (CLBP_C, CLBP_S, and CLBP_M) are directly extracted based on pixel original values. Intuitively, they are too humdrum to present comprehensive and robust texture information. Particularly, with respect to CLBP_C, it has two kinds of values and contains only one bit. Actually, it is not enough to build the local pattern for texture classification.

In order to overcome the aforementioned difficulties, we propose an efficient local entropy binary pattern (LEBP) by introducing the image entropy, which plays an auxiliary role to supply complementary texture information for CLBP_C. Image entropy characterizes textures by the statistical measure of randomness characteristics, which reflects how much information in an image. The more random the values of pixels in a region are, the higher the entropy of the region will be. It is constructed based on the image histogram. So, in this section, we use the image entropy to evaluate the stability of the image. In this section, we use the two-dimensional image entropy to build the local entropy binary pattern, which reflects the spatial characteristic of grayscale distribution.

In the definition of two-dimensional image entropy, the neighborhood pixel gray mean and grayscale of pixel make up two-tuples spatial characteristics.

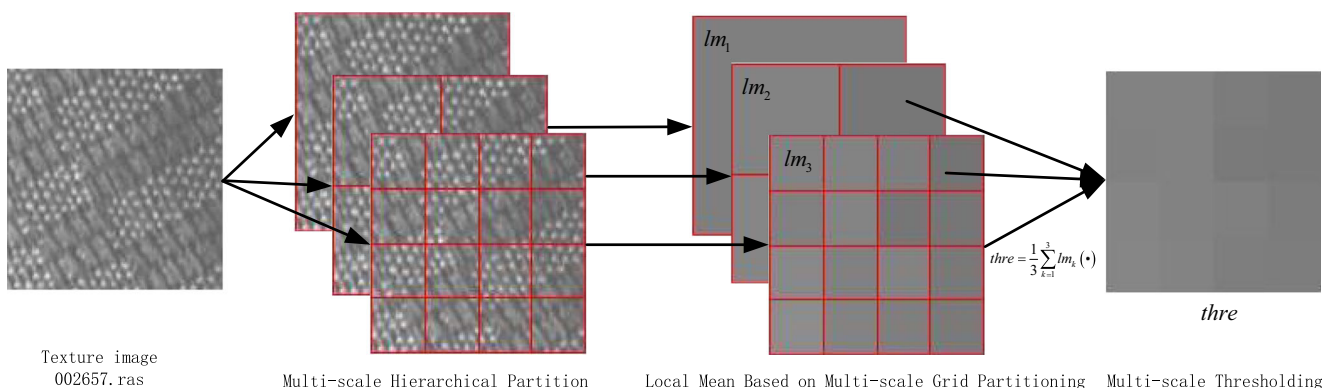


Fig. 3 An illustration of multiscale thresholding (taking $N = 3$ as an example)

The two-tuples is set as (g, m) , which expresses the comprehensive features with respect to the value of the pixel and the neighborhood pixel gray mean, where $g(0 \leq g \leq 255)$ is the value of the pixel and $m(0 \leq m \leq 255)$ is the neighborhood pixel gray mean. We represent the frequency of occurrence for two-tuples spatial characteristics as $f(g, m)$, and it describes the comprehensive gray-value distribution features between a pixel and its neighbors. The two-dimensional image entropy is defined as

$$p_{g,m} = \frac{f(g, m)}{D^2} \tag{8}$$

$$EN_{g,m} = - \sum_{m=0}^{255} \sum_{g=0}^{255} p_{g,m} \ln p_{g,m}, \tag{9}$$

where D is the dimensions of the given texture image.

We perform the LEBP encoding based on the local region two-dimensional image entropy. Similar to the CLBP_C, the LEBP is developed by the difference between the two-dimensional image entropy of center local region, and multiscale entropy thresholding. Formally, the LEBP is built by

$$LEBP = t(EN_c, thre_{N,EN_c}), \tag{10}$$

where $t(\cdot)$ is defined in (5). EN_c is two-dimensional image entropy of center local region and $thre_{N,EN_c}$ devotes the multiscale entropy thresholding corresponding to the center local region. Figure 4 illustrates the operation of LEBP on a given image. LEBP encodes the local two-dimensional image entropy, which is more abstract to express the texture information than the pixel original values. So, it can supply more complementary information for CLBP_C.

The Multiscale Hierarchical Threshold Completed Local Entropy Binary Pattern

In order to overcome the inherent limitations of CLBP, we propose the CLEBP descriptor by combining LEBP to represent the comprehensive and robust texture information. Meanwhile, we extend it based on the multiscale hierarchical threshold framework to enhance the performance of texture descriptor. For the sake of convenience in writing, we abbreviate the completed local entropy binary pattern based on multiscale thresholding as CLEBP. As shown in Fig. 4, the CLEBP contains four components: the central pixel component (CLEBP_C), the local entropy component (CLEBP_E), the sign component (CLEBP_S), and the magnitude component (CLEBP_M).

The CLEBP_C is defined as follows:

$$CLEBP_C_{P,R} = t(\phi(g_c), thre_{N,g_c}) \tag{11}$$

$$\phi(g_c) = \frac{1}{M} \sum_{q=1}^M g_q, \tag{12}$$

$\phi(g_c)$ is g_c -centric local square region operator, which can improve the robustness of the descriptor and show better performance. M is the number of local square region pixels, and g_q denotes the gray value of each pixel in the local square region. g_c represents the center pixel. $thre_{N,g_c}$ denotes the multiscale thresholding of the center pixel, and N is the total number of the multiscale hierarchical partitioning.

CLEBP_E is defined in (10).

CLEBP_S is generated as follows:

$$CLEBP_S_{P,R} = \sum_{p=0}^{P-1} s(\phi(g_p) - \phi(g_c)) 2^p, \tag{13}$$

where $\phi(\cdot)$ is defined in (12). g_p and g_c are defined in (1).

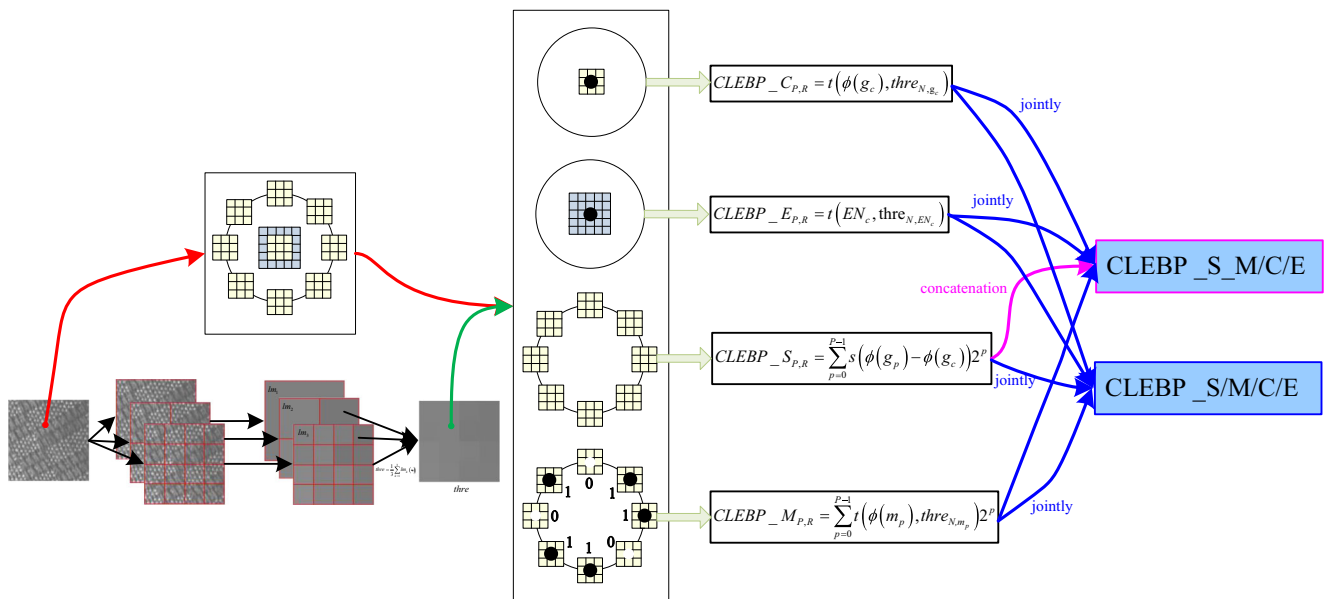


Fig. 4 An illustration of CLEBP operator

It is worth noting that the CLEBP_S does not use the multiscale thresholding to encode the local sign component. The main reasons are as follow: On one hand, as elaborated in [18], the sign component is more informative to represent the original local difference. For texture description, there is no doubt that the microstructure texture information is more important than the macrostructure one, even if we hope for more comprehensive information. On the other hand, from a viewpoint of multiscale information, we hope the new operators contain multiscale texture information, so we apply the sign component to represent the pixel-level texture details. Then, the CLEBP_S keeps the micro thresholding scheme.

CLEBP_M can be computed as

$$CLEBP_{M_{P,R}} = \sum_{p=0}^{P-1} t(\phi(m_p), thre_{N,m_p})2^p, \quad (14)$$

where $\phi(\cdot)$ is defined in (12). $t(\cdot)$ is defined in (5). $thre_{N,m_p}$ denotes the multiscale thresholding of m_p , and N is the total number of the multiscale hierarchical partitioning.

As elaborated in [8], there are two ways to combine four local binary patterns CLEBP_C, CLEBP_E, CLEBP_S, and CLEBP_M for texture classification, including concatenation and jointly. In this paper, two fusion features are used to the following experimental evaluation, which are represented as “CLEBP_S_M/C/E” and “CLEBP_S/M/C/E”. Then, we extend them by the multiscale hierarchical threshold, abbreviated as “CLEBP_S_Mth/Cth/Eth” and “CLEBP_S/Mth/Cth/Eth”. By analyzing the occurrence of combined histograms among different texture classes, the texture classification can be realized more exactly. It has been shown that approaches based on joint distributions can achieve better results for texture classification.

Dissimilarity Metric

In this study, we use the nearest neighborhood classifier (NNC) to test the performance of our proposed descriptor for texture classification. There are various distance metrics to evaluate the goodness-of-fit, including chi-square, histogram intersection, log-likelihood, cosine distance, etc.

In this paper, we utilize the chi-square distances as the dissimilarity between the testing samples and the training samples. Then, the chi-square distance could be calculated as follows:

$$D(T, M) = \sum_{h=1}^H \frac{(T_h - M_h)^2}{T_h + M_h}, \quad (15)$$

where H is the number of bins and T_h and M_h correspond to the values of the testing sampling and the training sampling at the h^{th} bin respectively.

Experimental Evaluation

In this section, we validate the effectiveness of the proposed CLEBP descriptor and multiscale thresholding framework with comprehensive experiments on three large and representative databases: the Outex database [49], UIUC database [50], and KTH-TIPS database [51]. We describe the details in the following section and summarized in Table 1. In addition, the proposed CLEBP descriptor is compared with most of the state-of-the-art texture descriptors for texture classification.

The Image Databases and Experimental Setup

We demonstrate the performance of CLEBP descriptor on three large and representative databases. The texture image databases used in this section are introduced below.

The Outex database [34] collects a large number of surface textures and natural scenes, which provides a rich foundation for processing the variations of illumination/rotation/ resolution. The Outex database also provides a series of test suits by varying different parameters of the experimental design for empirical evaluation of texture classification. In this section, we use the Outex test suits including Outex_TC_0010 (Outex_TC10) and Outex_TC_0012 (Outex_TC12), which contain 24 different homogeneous texture classes (shown in Fig. 5) with dimensions of 128×128 . The Outex_TC10 and Outex_TC12 were captured under three illuminations (“inca,” “t184,” and “horizon”) and nine rotation angles (0° , 5° , 10° , 15° , 30° , 45° , 60° , 75° , and 90°).

The UIUC texture database [35] has 25 classes of images with 40 samples, which was captured under significant viewpoint changes and scale variations. And all images are in grayscale JPG format, 640×480 pixels, which contains non-planarity of the textured surface (bark), significant non-rigid deformations between different samples of the same class (fur, fabric, and water), in-homogeneities of the texture pattern (bark, wood, and marble), and viewpoint-dependent appearance variations (glass). The result on UIUC database is reported as the mean classification accuracy over 50 random splits into training and testing dataset, wherein each texture class, Nt (Nt = 20, 15, 10, 5) images are used for training and the remaining 40-Nt images are used as the test set. Figure 6 shows the samples of 25 texture classes on UIUC database.

The KTH-TIPS database [36] includes 10 texture classes captured in the real-world textures under different conditions. There are 81 images (200×200) per class. The variety of textures present in the database is shown in Fig. 7. In this paper, 40, 20, 10, and 5 images of each texture class are utilized for training and the remaining images are used for testing.

Table 1 Summary of texture database used in experiments $\Theta_1 = \{5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}, \Theta_2 = \{0^\circ, 5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$

Texture database	Texture classes	Sample size (pixels)	Images per class	Training images number	Test images number	Description of training and testing	Experiment evaluation
Outex_TC10	24	128×128	180	480 (20×24)	3810 (160×24)	Training: illuminants(inca), Rotation(0°) Testing: illuminants(inca), Rotation(Θ_1)	Rotation invariant evaluation
Outex_TC12_000	24	128×128	200	480 (20×24)	4320 (180×24)	Training: illuminants(inca), Rotation(0°) Testing: illuminants(t148), Rotation(Θ_2)	Illumination and rotation invariant evaluation
Outex_TC12_001	24	128×128	200	480 (20×24)	4320 (180×24)	Training: illuminants(inca), Rotation(0°) Testing: illuminants(horizon), Rotation(Θ_2)	Illumination and rotation invariant evaluation
UIUC	25	640×480	40	500 (20×25)	500 (20×25)	Training and testing: image rotation and viewpoint variation.	Robustness to complex variations evaluation
KTH-TIPS	10	200×200	81	400 (40×10)	410 (41×10)	Training and testing: orientation, scale, and illumination variation	Robustness to complex variations evaluation

Experimental Results on Outex Database

Testing the Performance of the Proposed Multiscale Hierarchical Threshold Framework

In order to evaluate the performance of the proposed multiscale hierarchical threshold framework and give reasonable parameters settings, we carried out some experiments on Outex database, and more precisely on the text suit: Outex_TC_00010 (TC10).

For the purpose of removing the impact of feature extraction, we use the CLBP operator, which was published in [18], to verify the effectiveness of multiscale thresholding framework. It is worth reminding that we do not use the multiscale

thresholding to encode the local sign component. The magnitude component and center pixel are encoded by the multiscale thresholding framework. In this experiment, we set the number of layers (N in (7)) for multiscale hierarchical partitioning from 1 to 5. We evaluate the parameters settings at a single scale (P, R) = (8, 1).

Figure 8 shows the classification results with a different number of thresholding layers visually. We can get the following finding from Fig. 8 that most operators significantly improve the best results with four thresholding layers on TC10, expect for CLBP_M. For example, the CLBP_S/M/C obtains the best classification rate (98.20%) with four thresholding layers, and CLBP_M/C, CLBP_S/M, and CLBP_S_M/C reach the classification rates of 93.02%, 95.13%, and 95.99%,



Fig. 5 The 24 textures classes from the Outex database

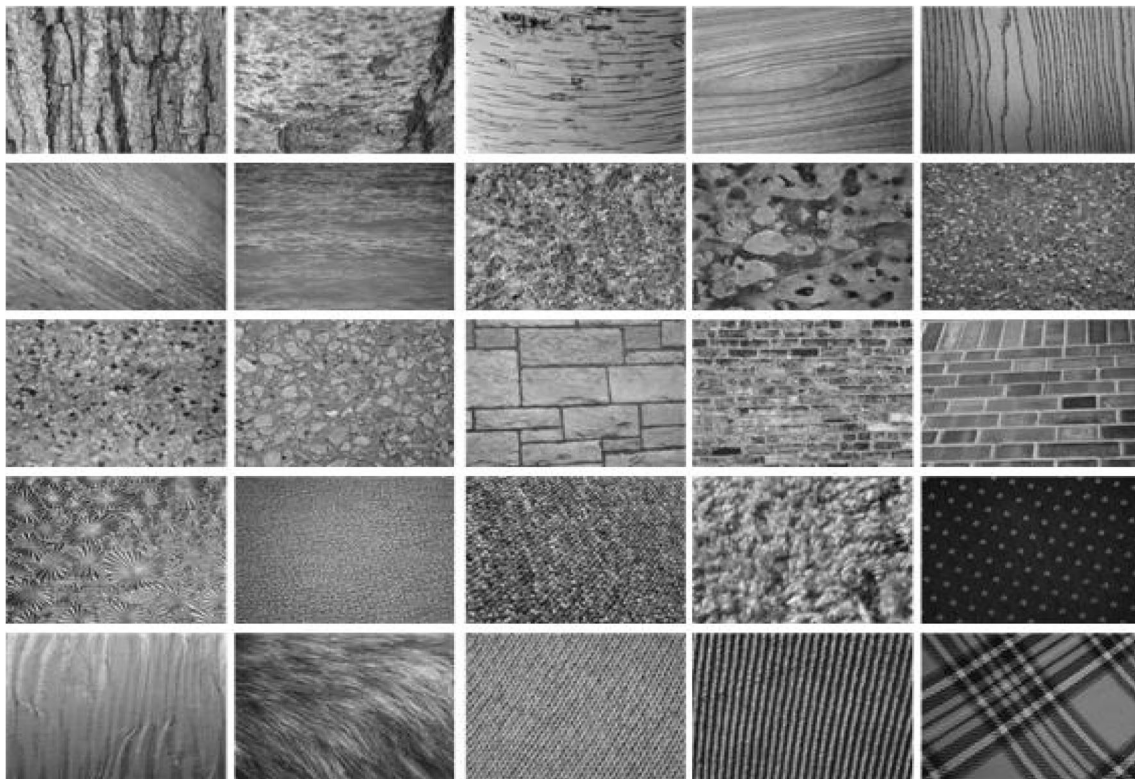


Fig. 6 The 25 textures classes from the UIUC database

respectively. And then, as the number of thresholding layers increases, the classification rate will remain stable or even decrease. According to the results, in order to enhance both the discriminative power and computational efficiency, we limit the multiscale thresholding framework evaluation to three thresholding layers in the following experiments on Outex database. Secondly, the experimental results indicate that the increasement of the number of layers is not always helpful for the improvement of classification performance. On the contrary, when the number of thresholding layers exceeds four, the classification performance would degrade. Obviously, the

number of thresholding layers is a fine-tuned parameter which depends on different texture databases.

Experimental Results on the Outex Database

To fully evaluate the capabilities of the proposed CLEBP with multiscale threshold, some classical texture descriptors and state-of-the-art methods are adopted as the competitors. Table 2 details experimental results compared with the traditional LBP, original CLBP, some classical LBP variants, and recent state-of-the-art methods at a different scale.

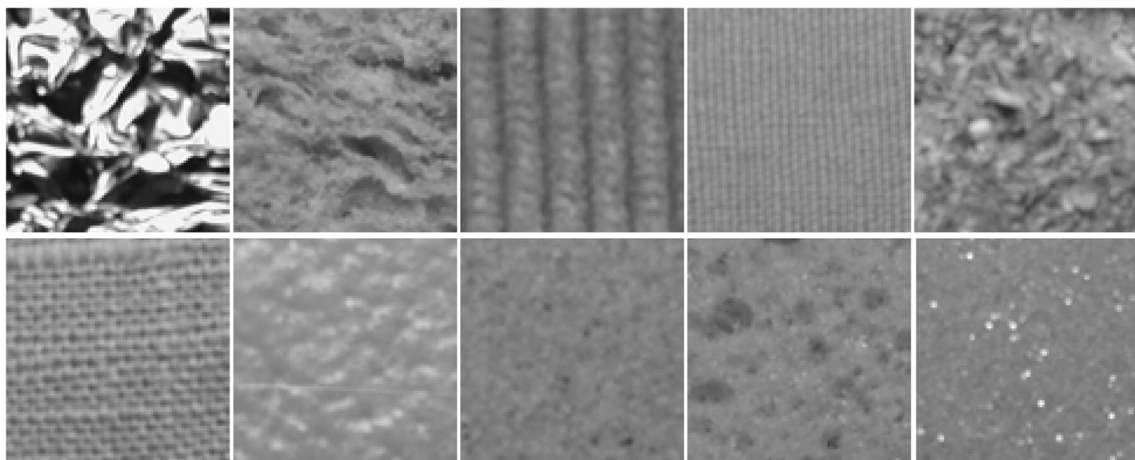


Fig. 7 The 10 textures classes from the KTH-TIPS database

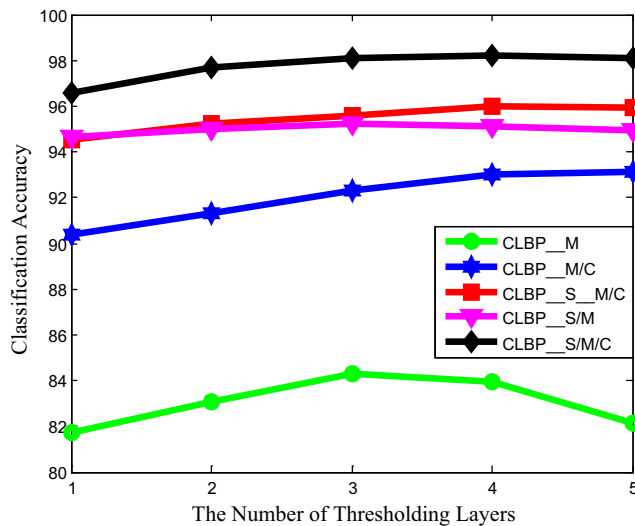


Fig. 8 The classification results with different number of thresholding layers for CLBP operator

Through Table 2, we can draw some interesting conclusions. The proposed multiscale thresholding framework is helpful for the improvement of classification accuracy. As can be seen, on TC10, the CLEBP_S/Mth/Cth/Eth is higher than CLBP_S/M/C by about 1.14%, 0.52%, and 0.42% with $(P, R) = (8, 1), (16, 2),$ and $(24, 3)$, respectively. For

TC12_000, the improvement in accuracy between CLEBP_S/Mth/Cth/Eth and CLBP_S/M/C are 2.04%, 1.48%, and 1.34% with corresponding $(P, R) = (8, 1), (16, 2),$ and $(24, 3)$, respectively. For TC12_001, we can draw similar conclusions. Derived from the differences, we can conclude that the CLEBP_S/Mth/Cth/Eth performs better than corresponding CLBP_S/M/C. The proposed multiscale thresholding framework is very discriminative and powerful, which helps the optimization of binary coding.

The proposed CLEBP method outperforms classical LBP variants and the state-of-the-art methods in Table 2. Firstly, we can see that the proposed CLEBP method, especially for CLEBP_S/Mth/Cth/Eth, is the best in comparison with LBP and CLBP. For example, with $(P, R) = (24, 3)$, the CLEBP_S/Mth/Cth/Eth operator obtains the best classification accuracy of 99.87% on TC10, which has 4.8% and 0.94% improvement compared with the classification rate of LBP and CLBP_S/M/C, respectively. Similarly, with the same scale, the CLEBP_S/Mth/Cth/Eth operator also reaches the best result (98.03% and 97.92%) on TC12_000 and TC12_001, which maintains the great advantage of 12.99% and 17.14% over LBP, and shows a performance boost of 2.71% and 3.39% over CLBP_S/M/C, respectively. In addition, the classification rate by CLEBP_S/Mth/Cth/Eth gets to 99.58% on TC10 with $(P, R) = (8, 1)$, 3.02% higher than the CLBP_S/M/C operator, and even

Table 2 Classification rate (%) on Outex_TC10 and Outex_TC12

	$(P, R) = (8, 1)$				$(P, R) = (16, 2)$				$(P, R) = (24, 3)$			
	Outex_TC10	Outex_TC12_000	Outex_TC12_001	Average	Outex_TC10	Outex_TC12_000	Outex_TC12_001	Average	Outex_TC10	Outex_TC12_000	Outex_TC12_001	Average
LBP [9]	84.81	65.46	63.68	71.31	89.40	82.26	75.20	82.28	95.07	85.04	80.78	86.96
CLBP_S_M/C [18]	94.53	81.87	82.52	86.30	98.02	90.99	91.08	93.36	98.33	94.05	92.40	94.92
CLBP_S/M/C [18]	96.56	90.30	92.29	93.05	98.72	93.54	93.91	95.39	98.93	95.32	94.53	96.26
CLBC [20]	97.16	89.79	92.92	93.29	98.54	93.26	94.07	95.29	98.78	94.00	93.24	95.67
CRLBP ($\alpha = 8$) [23]	97.55	91.94	92.45	93.98	98.59	95.88	96.41	96.96	99.35	96.83	96.16	97.45
CRLBP ($\alpha = 1$) [23]	96.54	91.16	92.06	93.25	98.85	96.67	96.97	97.50	99.48	97.57	97.34	98.13
BRINT [52]	91.87	86.46	88.50	88.94	96.43	93.38	93.98	94.59	96.04	94.24	94.35	94.87
AECLBP_S_M/C [22]	94.35	84.19	84.28	87.61	96.43	90.39	89.35	92.06	97.92	92.38	91.39	93.90
AECLBP_S/M/C [22]	97.58	91.83	91.81	93.74	98.80	95.42	94.70	96.31	99.19	96.83	95.05	97.02
DRLBP ₃ [24]	88.54	84.25	83.19	85.33	99.19	97.15	95.37	97.24	–	–	–	–
LQC_C(2)N(4) [36]	97.16	–	–	–	98.46	–	–	–	98.54	–	–	–
LQC_C(6)N(4) [36]	97.80	–	–	–	98.80	–	–	–	98.96	–	–	–
CDLF_S/M/C [28]	97.29	91.06	93.31	93.89	99.48	94.24	95.02	96.25	99.11	95.65	95.25	96.67
CDLF_S/M/C_AHA [28]	97.45	91.62	94.03	94.37	99.64	94.40	95.44	96.49	99.22	96.02	95.67	96.97
FbLBP [25]	98.85	92.66	93.82	95.11	99.64	95.42	95.69	96.92	99.58	96.00	94.75	96.78
CLEBP_S_M/C/E	97.81	90.16	90.95	92.97	98.80	93.91	94.33	95.68	99.11	94.65	95.79	96.52
CLEBP_S/M/C/E	98.44	92.75	94.79	95.33	99.22	96.69	96.64	97.52	99.45	96.69	97.11	97.75
CLEBP_S_Mth/Cth/Eth	98.23	89.35	91.78	92.5	99.22	94.79	94.40	95.52	99.64	96.34	96.64	97.34
CLEBP_S/Mth/Cth/Eth	99.58	94.79	94.93	96.85	99.74	98.17	97.62	98.68	99.87	98.03	97.92	98.77

14.77% better than the accuracy of traditional LBP. With the same scale on TC12_000, CLEBP_S/Mth/Cth/Eth has significant 29.33% and 4.49% improvement over the accuracy of the original LBP and CLBP_S/M/C respectively. Similarly, on TC12_001, it maintains the obvious advantage of 31.25% and 2.54% over LBP and CLBP_S/M/C, respectively. Similar conclusions can be obtained with $(P, R) = (16, 2)$.

Secondly, the CLEBP_S/Mth/Cth/Eth operator outperforms some classical LBP variants. For example, on TC10 with $(P, R) = (8, 1)$, CLEBP_S/Mth/Cth/Eth achieves the best accuracy of 99.58%, which has about 2.42%, 2.03%, and 2% improvements over CLBC, CRLBP ($\alpha = 8$), and AECLBP_S/M/C, respectively, and even 7.71% higher than the accuracy by BRINT.

Thirdly, compared with recent state-of-the-art approaches, similar conclusions can be made from the experimental results. More specifically, the proposed CLEBP_S/Mth/Cth/Eth reaches the classification rate of 99.58% on TC10 with $(P, R) = (8, 1)$, also maintaining the distinct advantage of 11.04%, 1.78%, 2.13%, and 0.73% over DRLBP₃, LQC_C(6)N(4), CDLF_S/M/C_AHA, and FbLBP, respectively. Likewise, for TC12_000 and TC12_001, we can reach similar conclusions. In addition, these groups of comparisons imply that the proposed CLEBP_S/Mth/Cth/Eth outperforms the recent state-of-the-art methods on the Outex database and has a more comprehensive understanding and representation for texture information.

It is noteworthy that the computational cost of CLEBP_S/Mth/Cth/Eth is 0.060 s for a texture image on TC10, which is about twice as much as the CLBP_S/M/C descriptor (0.032 s). There is no doubt that the operation of multiscale thresholding framework and local entropy binary pattern increase the expense in computational cost. However, the proposed CLEBP_S/Mth/Cth/Eth still has the same order of magnitudes for computational time as CLBP_S/M/C. The additional computational cost does not affect the effectiveness of the algorithm too much. Moreover, it is worthwhile that the CLEBP_S/Mth/Cth/Eth uses the additional computational cost in return for the improvement of accuracy. As a result, The CLEBP_S/Mth/Cth/Eth operator obtains the best classification accuracy of 99.58% on TC10 with $(P, R) = (8, 1)$, which has about 3.02% improvement over the CLBP_S/M/C descriptor.

Experimental Results on the UIUC Database

Table 3 compares classification results between different methods on UIUC database. It can be observed that the proposed multiscale thresholding strategy significantly enhances the classification performance of the texture descriptors. Through analyzing the classification performance of CLEBP_S/Mth/Cth/Eth and CLEBP_S/M/C/E, we can get similar findings to the Outex database. On the other hand, as

we can see from the table, the CLEBP_S/Mth/Cth/Eth shows a significant performance boost compared with traditional LBP and CLBP. For instance, with $(P, R) = (1, 8)$, the classification rates obtained by CLBP with different numbers of training images ($N_t = 40, 20, 10,$ and 5) are 87.64%, 85.70%, 82.65%, and 75.05%, respectively. With the same configuration setting, the CLEBP_S/Mth/Cth/Eth can reach the accuracy of 93.54%, 91.65%, 89.01%, and 82.29%, which has 38.89%, 38.71%, 41.87%, and 42.57% improvement over the traditional LBP. Similar conclusions to the proposed CLEBP_S/Mth/Cth/Eth can be found for other configuration settings.

To further highlight the performance of the proposed CLEBP_S/Mth/Cth/Eth, we select some classical LBP variants and state-of-the-art methods as the competitors. For example, with $(P, R) = (16, 2)$ and $N_t = 20$, the CLEBP_S/Mth/Cth/Eth can reach the accuracy of 95.54%, which has 4.5%, 2.62%, 11.4%, and 4.86% improvement over CLBC, CRLBP ($\alpha = 1$), BRINT, and AECLBP_S/M/C, respectively. In addition, compared with some state-of-the-art methods, the CLEBP_S/Mth/Cth/Eth significantly improves the best classification accuracy of recent texture descriptors on UIUC database. Specifically, with $N_t = 20$ and $(P, R) = (1, 8), (16, 2),$ and $(24, 3)$, the best classification accuracy obtained by CLEBP_S/Mth/Cth/Eth are 93.54%, 95.54%, and 96.40%, which outperform FbLBP by 1.5%, 1.4%, and 2.23%, and have 3.38%, 1.44%, and 2.63% improvement over CDLF_S/M/C_AHA, respectively. The experimental results demonstrate that the proposed CLEBP_S/Mth/Cth/Eth is a very discriminative and powerful texture descriptor for texture classification.

Experimental Results on the KTH-TIPS Database

In this section, we perform some comparative evaluations of the proposed CLEBP descriptors against some classical LBP variants and state-of-the-art methods in terms of classification rates on KTH-TIPS database. We could get the following finding from Table 4. Firstly, through the comparison of CLEBP_S/M/C/E and CLEBP_S/Mth/Cth/Eth, CLEBP_S/M/C/E and CLEBP_S/Mth/Cth/Eth, the same conclusion has been drawn that the classification accuracy has a significant improvement with the proposed multiscale thresholding strategy. In particular, with $(P, R) = (8, 1)$, and $N_t = 40$, the CLEBP_S/Mth/Cth/Eth shows a performance boost of 1.2% compared with CLEBP_S/M/C/E. And similarly, the CLEBP_S/Mth/Cth/Eth achieves 1.04% improvement over CLEBP_S/M/C/E. Similar conclusions can be found in other cases. Once again, it is verified that the proposed multiscale thresholding strategy efficiently enhances the classification performance of texture descriptors and yields the best results in most cases.

Table 3 Classification accuracy (%) on UIUC

Number of training images Nt =	$(P, R) = (8, 1)$				$(P, R) = (16, 2)$				$(P, R) = (24, 3)$			
	20	15	10	5	20	15	10	5	20	15	10	5
LBP [9]	54.65	52.94	47.14	39.72	61.32	56.42	51.16	42.67	64.05	60.05	54.25	44.59
CLBP_S/M/C [18]	87.64	85.70	82.65	75.05	91.04	89.42	86.29	78.57	91.19	89.21	85.95	78.05
CLBC [20]	87.83	85.66	82.35	74.57	91.04	89.66	86.63	79.48	91.39	90.10	86.45	79.75
CRLBP ($\alpha = 8$) [23]	88.08	86.62	82.97	76.01	91.99	90.41	88.04	81.49	92.83	90.55	88.02	80.54
CRLBP ($\alpha = 1$) [23]	86.91	85.67	82.20	73.95	92.92	91.82	88.15	81.98	93.31	92.03	89.47	81.90
BRINT [52]	79.52	76.14	71.26	61.59	84.14	81.57	77.06	67.96	86.39	83.77	79.33	70.34
AECLBP_S/M/C [22]	88.04	86.54	82.96	75.43	90.68	89.18	85.17	78.13	92.18	90.22	87.08	79.69
LQC_C(2)N(4) [53]	89.45	87.70	83.58	76.23	92.32	90.58	87.67	81.48	92.94	90.39	87.54	81.64
LQC_C(6)N(4) [53]	90.12	88.54	85.62	78.38	92.62	90.76	87.97	81.98	93.17	90.91	88.13	81.77
CDLF_S/M/C[53]	89.57	87.73	84.74	78.39	92.85	90.17	88.73	80.55	92.90	91.18	87.34	81.78
CDLF_S/M/C_AHA[28]	90.16	88.30	85.17	78.82	94.10	90.53	89.58	80.82	93.77	92.83	88.80	82.11
FbLBP[25]	92.04	90.27	87.80	80.10	94.14	92.73	89.79	82.43	94.17	92.58	89.51	81.79
CLEBP_S_M/C/E	81.85	79.46	75.40	67.45	87.69	85.20	81.92	73.43	89.20	86.88	82.74	74.17
CLEBP_S/M/C/E	89.18	87.62	84.18	76.75	92.61	91.03	88.59	81.78	93.47	92.21	89.10	82.13
CLEBP_S_Mth/Cth/Eth	87.89	85.79	82.49	74.74	89.52	87.36	83.59	75.21	91.82	89.47	86.24	77.84
CLEBP_S/Mth/Cth/Eth	93.54	91.65	89.01	82.29	95.54	93.98	91.35	84.75	96.40	94.96	92.67	86.36

Secondly, from Table 3, it can be found that the proposed CLEBP_S/Mth/Cth/Eth is comparable or outperform with most comparison methods. On one hand, the proposed CLEBP_S/Mth/Cth/Eth achieves higher classification accuracy than classical LBP variants. For example, with $(P, R) = (8, 1)$, and $N = 40, 20, 10$, and 5 , the classification accuracies achieved by CLEBP_S/Mth/Cth/Eth are 96.40%, 92.04%, 85.59%, and 75.64%, which could have 3.55%, 7.45%, 10.49%, and 11.95% improvement over the original LBP, 1.07%, 0.97%, 0.93%, and 1.93% higher than CLBP, and

outperform LTP by 4.03%, 5.96%, 10.25%, and 10.51%, respectively. Similar conclusions to $(P, R) = (16, 2)$ and $(24, 3)$ can be found on the database. On the other hand, the CLEBP_S/Mth/Cth/Eth achieves competitive even better classification performance compared with the state-of-the-art methods. In particular, in the case of $N = 40$ and $(P, R) = (8, 1)$, the CLEBP_S/Mth/Cth/Eth achieves the best classification accuracy with 96.40%, which outperforms DRLBP₃ by 2.24%, has 0.57% improvement over LQC_C(2)N(4) and has a little higher than LQC_C(2)N(4), respectively. In other

Table 4 Classification accuracy (%) on KTH-TIPS

Number of training images Nt =	$(P, R) = (8, 1)$				$(P, R) = (16, 2)$				$(P, R) = (24, 3)$			
	40	20	10	5	40	20	10	5	40	20	10	5
LBP [9]	92.85	84.59	75.10	63.69	88.32	82.02	70.59	60.54	79.83	74.87	66.10	57.29
CLBP_S/M/C [18]	95.33	91.07	84.66	73.71	96.15	89.51	83.45	73.51	95.27	89.30	80.94	73.05
LTP [15]	92.37	86.08	75.34	65.13	93.24	86.20	75.73	66.84	90.39	85.79	76.01	64.59
CLBC [20]	95.34	90.18	83.41	72.78	93.68	88.38	79.98	71.16	91.12	85.92	74.89	68.62
LQC_C(2)N(4) [53]	95.83	91.16	84.72	74.71	96.05	90.30	82.87	73.51	94.77	89.67	82.10	71.24
LQC_C(6)N(4) [53]	96.39	91.25	84.82	74.74	96.23	90.38	83.54	74.22	95.31	89.82	82.27	74.53
DRLBP1 [24]	93.17	–	–	–	92.82	–	–	–	–	–	–	–
DRLBP2 [24]	93.78	–	–	–	94.32	–	–	–	–	–	–	–
DRLBP3[24]	94.16	–	–	–	90.16	–	–	–	–	–	–	–
CLEBP_S_M/C/E	92.66	86.79	78.74	68.85	93.49	87.08	79.17	68.63	93.09	86.18	77.27	66.52
CLEBP_S/M/C/E	95.36	91.16	83.88	74.05	95.80	90.31	82.62	72.25	94.58	88.58	80.36	69.96
CLEBP_S_Mth/Cth/Eth	93.86	87.89	80.32	70.18	93.67	87.90	79.96	70.69	94.48	87.75	79.13	69.09
CLEBP_S/Mth/Cth/Eth	96.40	92.04	85.59	75.64	95.97	90.89	83.45	74.10	95.96	89.77	81.48	71.90

cases, it could still keep more or less superiority over the comparison methods. For example, the best classification accuracy of CLEBP_S/Mth/Cth/Eth in the case $(P, R) = (24, 3)$ and $N_t = 40$ is 95.96%, which is 0.65% higher than LQC_C(6)N(4) and outperforms CLBC by 4.84%, respectively. The experimental results demonstrate that the proposed CLEBP_S/Mth/Cth/Eth is a powerful and effective texture descriptor.

Conclusion

In this paper, we develop a multiscale thresholding framework based on hierarchical adaptive local partition for binary encoding and present the CLEBP descriptor to extract comprehensive texture feature, which contains four components. Except with the three patterns from CLBP, we use the two-dimensional image entropy to build a novel local entropy binary pattern, which reflects the spatial characteristic of gray-scale distribution. Then we extend the CLEBP with multiscale thresholding framework and propose CLEBP_S/Mth/Cth/Eth. Experimental results on three representative texture databases of Outex, UIUC, and KTH-TIPS show that the proposed multiscale thresholding framework is very discriminative and powerful, which helps the optimization of the binary coding. With the multiscale thresholding framework, the proposed CLEBP_S/Mth/Cth/Eth achieves competitive and superior texture classification performance, which is a powerful and effective texture descriptor for texture classification. In future works, we plan to develop an adaptive optimization technique to optimize the multiscale hierarchical threshold strategy. In addition, complementary solutions to the binary pattern family are of great interest to texture classification.

Funding Information The paper is funded by the National Natural Science Foundation of China (Grant No. 61701134, 51,809,056), the National Key Research and Development Program of China (Grant No. 2016YFF0102806), and the Natural Science Foundation of Heilongjiang Province, China (Grant No. F2017004).

Compliance with Ethical Standards

Conflict of Interests The author declares that she has no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Boulkenafet Z, Komulainen J, Hadid A. Face spoofing detection using colour texture analysis. *IEEE Trans Inf Forensics Security*. 2016;11(8):1818–30.
- Zhao X, Lin Y, Heikkilä J. Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection. *IEEE Trans Multimedia*. 2017;20(3):552–66.
- Castellano G, Bonilha L, Li L, Cendes F. Texture analysis of medical images. *Clin Radiol*. 2004;59(12):1061–9.
- Torabi M, Ardekani RD, Fatemizadeh E. Discrimination between Alzheimer's disease and control group in MR-images based on texture analysis using artificial neural network. *Proc IEEE Int Conf Biomed Pharmaceutical Eng*. 2006;79–83.
- Wang YN, Huang JC. Texture analysis in hexagonal materials. *Mater Chem Phys*. 2003;81:11–26.
- Akbari V, Doulgeris AP, Moser G, Eltoft T, Anfinsen SN, Serpico SB. A textural-contextual model for unsupervised segmentation of multipolarization synthetic aperture radar images. *IEEE Trans Geosci Remote Sens*. 2013;51(4):2442–53.
- Masjedi A, Zoj MJV, Maghsoudi Y. Classification of polarimetric SAR images based on modeling contextual information and using texture features. *IEEE Trans Geosci Remote Sens*. 2016;54(2):932–43.
- Liu L, Lao S, Fieguth P, Guo Y, Wang X, Pietikainen M. Median robust extended local binary pattern for texture classification. *IEEE Trans Image Process*. 2016;25(3):1368–81.
- Ojala T, Pietikäinen M, Maenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell*. 2002;24(7):971–87.
- Liu L, Fieguth P, Pietikäinen M, and Lao S. Median robust extended local binary pattern for texture classification. in *Proc IEEE Int Conf Image Process (ICIP)*. 2015;2319–2323.
- Subrahmanyam M, Maheshwari R, Balasubramanian R. Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking. *Signal Process*. 2012;92(6):1467–79.
- Murala S, Maheshwari RP, Balasubramanian R. Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Trans Image Process*. 2012;21(5):2874–86.
- Ahonen T, Hadid A, Pietikäinen M. Face description with local binary patterns: application to face recognition. *IEEE Trans Pattern Anal Mach Intell*. 2006;28(12):2037–41.
- Ahonen T, Hadid A, Pietikäinen M. Face recognition with local binary patterns. *Proc Eur Conf Comput Vis*. 2004; 469–481.
- Tan X, Triggs B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans Image Process*. 2010;19(6):1635–50.
- Nanni L, Lumini A, Brahnam S. Local binary patterns variants as texture descriptors for medical image analysis. *Artif Intell Med*. 2010;49(2):117–25.
- Satpathy A, Jiang X, Eng H. LBP based edge texture features for object recognition. *IEEE Trans Image Process*. 2014;23(5):1953–64.
- Guo Z, Zhang L, Zhang D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans Image Process*. 2010;9(16):1657–63.
- Guo Z, Zhang L, Zhang D. Rotation invariant texture classification using lbp variance (LBPV) with global matching. *Pattern Recogn*. 2010;43(3):706–19.
- Zhao Y, Huang DS, Jia W. Completed local binary count for rotation invariant texture classification. *IEEE Trans Image Process*. 2012;20(10):4492–7.
- Zhao Y, Jia W, Hu RX, Min H. Completed robust local binary pattern for texture classification. *Neurocomputing*. 2013;106(1):68–76.
- Song K, Yan Y, Zhao Y, Liu C. Adjacent evaluation of local binary pattern for texture classification. *J Vis Commun Image Represent*. 2015;33:323–39.
- Guo Z, Wang X, Zhou J, You J. Robust texture image representation by scale selective local binary patterns. *IEEE Trans Image Process*. 2016;25(2):687–99.

24. Mehta R, Egiazarian KO. Dominant rotated local binary patterns (DRLBP) for texture classification. *Pattern Recogn Lett.* 2016;71:16–22.
25. Pan ZB, Li ZY, Fan HC, Wu XQ. Feature based local binary pattern for rotation invariant texture classification. *Expert Syst Appl.* 2017;88:238–48.
26. Wang K, Bichot CE, Li Y, Li BL. Local binary circumferential and radial derivative pattern for texture classification. *Pattern Recogn.* 2017;67:213–29.
27. Shakoor M, Boostani R. Extended mapping local binary pattern operator for texture classification. *Int J Pattern Recognit.* 2017;31(6):1–22.
28. Zhang Z, Liu S, Mei X, Xiao B, Zhang L. Learning completed discriminative local features for texture classification. *Pattern Recogn.* 2017;67:263–75.
29. Yu J, Kuang Z, Zhang B, Zhang W, Lin D, Fan J. Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing[J]. *IEEE Trans Inf Forensics Secur.* 2018;13(5):1317–32.
30. Zhang J, Yu J, Tao D. Local deep-feature alignment for unsupervised dimension reduction[J]. *IEEE Trans Image Process.* 2018;24(5):2420–32.
31. Yu J, Rui Y, Tao D. Click prediction for web image Reranking using multimodal sparse coding[J]. *IEEE Trans Image Process.* 2014;23(5):2019–32.
32. Yu J, Yang X, Gao F, et al. Deep multimodal distance metric learning using click constraints for image ranking[J]. *IEEE Trans Cybern.* 2016;1–11.
33. Thangarajah A, Jonathan WQ, Hui Z. Effect of fusing features from multiple DCNN architectures in image classification[J]. *IET Image Process.* 2018;12(7):1102–10.
34. Akilan T, Wu QMJ, Safaei A, Jiang W. A late fusion approach for harnessing multi-CNN model high-level features. 2017 IEEE International Conference on Systems. 2017; 1–8.
35. Lin TY and Maji S. Visualizing and understanding deep texture representations. in Proc IEEE Conf Comput Vis Pattern Recogn. 2016; 2791–2799.
36. Yu J, Tao D, Wang M, Rui Y. Learning to rank using user clicks and visual features for image retrieval[J]. *IEEE Transactions on Cybernetics.* 2015;45(4):767–79.
37. Yu J, Rui Y, Tang YY, Tao D. High-order distance-based multiview stochastic learning in image classification[J]. *IEEE Transactions on Cybernetics.* 2014;44(12):2431–42.
38. Zhong G, Yan S, Huang K, Cai Y, Dong J. Reducing and stretching deep convolutional activation features for accurate image classification[J]. *Cogn Comput.* 2018;10(1):179–86.
39. Srikanth V, Neerja M, Chakravarthy NB, et al. 3D local spatio-temporal ternary patterns for moving object detection in complex scenes[J]. *Cogn Comput.* 2019;11(1):18–30.
40. Toprak S, Nicolás N-G, Wermter S. Evaluating integration strategies for visuo-haptic object recognition[J]. *Cogn Comput.* 2018;10(3):408–25.
41. Yu J, Zhu C, Zhang J, et al. Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition[J]. *IEEE Trans Neural Netw Learn Syst.* 2019;1–14.
42. Zrira N, Khan HA, Bouyakhf EH. Discriminative deep belief network for indoor environment classification using global visual features[J]. *Cogn Comput.* 2018;10(3):437–53.
43. Hong C, Yu J, Chen X, et al. Image-based 3D human pose recovery with locality sensitive sparse retrieval. in Proc. IEEE International Conference on Systems. 2013; 1–6.
44. Hong C, Yu J, Wan J, Tao D, Wang M. Multimodal deep autoencoder for human pose recovery[J]. *IEEE Trans Image Process.* 2015;24(12):5659–70.
45. Yang ZX, Tang L, Zhang K, Wong PK. Multi-view CNN feature aggregation with ELM auto-encoder for 3D shape recognition[J]. *Cogn Comput.* 2018;10(6):908–21.
46. Li J, Zhang Z, He H. Hierarchical convolutional neural networks for EEG-based emotion recognition[J]. *Cogn Comput.* 2017;10(2):368–80.
47. Ren CX, Lei Z, Dai DQ, Li SZ. Enhanced local gradient order features and discriminant analysis for face recognition. *IEEE Trans Image Process.* 2016;22(5):2656–69.
48. Huang W, Yin H. Robust face recognition with structural binary gradient patterns. *Pattern Recogn.* 2017;68:126–40.
49. Ojala T, Maenpaa T, Pietikainen M, Viertola J, Kyllonen J, and Huovinen S. Outex – new framework for empirical evaluation of texture analysis algorithms. In Proc IEEE Int Conf Pattern Recognit (ICPR). 2002; 701–706.
50. Dana KJ, Van Ginneken B, Nayar SK, Koenderink JJ. Reflection and texture of real-world surfaces. *ACM Trans Graph.* 1999;18(1):1–34.
51. Hayman E, Caputo B, Fritz M, and Eklundh J. On the significance of real-world conditions for material classification. In Proc Eur Conf Comput Vis. 2004; 253–266.
52. Liu L, Long Y, Fieguth PW, Lao S, Zhao G. BRINT: binary rotation invariant and noise tolerant texture classification. *IEEE Trans Image Process.* 2014;23(7):3071–84.
53. Zhao Y, Wang R G , Wang W M , et al. Local quantization code histogram for texture classification. *Neurocomputing.* 2016; 354–364.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.