Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 11, No. 1, 2024

Texture Classification Method Based on Local Enhancement and Non-local Median Patterns

Yaxin Lian, Yongsheng Dong

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Abstract: Local binary pattern (LBP) serves as a highly powerful method for texture classification. LBP and its variant methods are extensively applied across various domains of image processing.In increasingly complex imaging environments, LBP faces two issues: (1) Loss of detail information during feature extraction. (2) Sensitivity to noise. To mitigate these issues, this paper proposes a method called Local Enhancement and Non-local Median Pattern (LENMP). It consists of two operators: Local Adaptive Contrast Enhancement Pattern (LEP) and Non-local Median Binary Pattern (NMBP). The LEP operator captures the sign and magnitude details of local image characteristics, while the NMBP operator captures the global information of image features. First, the image is processed using the threshold ACE (thACE) algorithm to enhance the contrast of high-frequency information in the image, extracting image contour edge information. Then, median extraction is performed separately on the two operators to capture larger spatial texture detail information. Finally, comparative experiments are conducted on multiple datasets (Outex, CUReT, Brodatz, KTH-TIPS, and Kylberg) with representative texture classification methods. The results indicate that our proposed LENMP texture classification method exhibits good classification performance and remains competitive compared to the latest descriptors.

Keywords: LBP; thACE; Robust noise; Median filtering.

1. Introduction

Texture is represented as local patterns that frequently occur in an image, exhibiting certain arrangement regularities. It is a widely observed visual phenomenon, representing the fundamental characteristics of object surfaces and playing an important role as visual cues in human perception of the world [1][2]. Local binary pattern (LBP) [3], as a powerful texture descriptor, has garnered widespread attention in academia. In addition to successful applications in texture classification [4] and face recognition [5], the LBP technique has also exhibited satisfactory results in different domains of image processing. These include dynamic texture [6], medical image analysis [7], object detection [8], image segmentation [9], fingerprint detection [10], color texture [11], finger vein image restoration [12], and others. Due to its simple principle, rapid computation, low storage requirements, high efficiency, and rotational invariance, researchers have proposed many LBP-based variant methods.

On one hand, to capture more intricate details from descriptors. Guo et al. [13] proposed the CLBP. CLBP had three descriptors in total, namely CLBP_C, CLBP_S, and CLBP_M, aiming to enhance the discriminative power of features by integrating multiple complementary pieces of information. Tan et al. [14] proposed LTP and Zheng et al. [15] introduced circular local ternary pattern (CLTP), extending binary pattern encoding to ternary patterns, which were more discriminative and less sensitive to uniform region noise. Guo et al. [16] introduced the adaptive ALBP. ALBP provided a comprehensive representation of the spatial structure of the entire image. Dong et al. [17] introduced the MSDMD. MSDMD successfully captured the rotation invariance of textures. Hu et al. [18] introduced the ABMV method. ABMV capable of better extracting amplitude features in rough and smooth regions. Ma et al. [19] proposed the MSMDB for palmprint recognition, applying binary patterns to palmprint recognition, forming complementary descriptors, and extracting comprehensive information from multiscale and multidirectional features.

On the other hand, to improve the resilience of descriptors against noise. Liu et al. [20] introduced the BRINT descriptor, which computed the average before binarization. This descriptor was simple, efficient, and robust, maintaining robustness against noise, illumination changes, and rotation. Liu et al. [21] introduced the MRELBP, which compared the median grayscale values of local image regions to capture larger spatial information. MRELBP exhibited good grayscale and rotational invariance as well as robustness against noise. Song et al. [22] proposed the LGONBP descriptors to capture both local and non-local texture features, achieving robustness against image rotation and noise. Khadiri et al. [1] introduced the PGMO-MSTP, which utilized Peterson graphs in four directions for multi-level encoding to capture spatial information of horizontal structures. The approach computed the mean grayscale values to generate diverse global and local average grayscale levels utilized for constructing the PGMO-MSTP code. This technique exhibited robustness to noise and remained unchanged under monotonic grayscale adjustments. However, the aforementioned original LBP and its variant methods mainly focus on extracting local symbol differences, thereby overlooking certain detailed information, often resulting in descriptors that are sensitive to noise. Addressing these issues, to enhance the classification accuracy of texture classification methods, this paper introduces a LENMP descriptor. The primary contributions of this research are outlined below:

(1). We propose a local enhancement and non-local median pattern (LENMP) for texture classification, which includes two operators: local adaptive enhancement pattern (LEP) for extracting local information and non-local median binary pattern (NMBP) for extracting global information. It not only extracts more detailed information but also exhibits good robustness to noise.

(2). We introduce the adaptive threshold contrast

enhancement algorithm (thACE) to enhance high-frequency information, extracting more contour edge details and enhancing the effectiveness of feature extraction.

(3). The LENMP descriptor utilizes a median filter for preprocessing, effectively eliminating noise from the image while capturing larger spatial texture details.

(4). A series of comparative experiments are conducted on several classical datasets (Outex, CUReT, Brodatz, KTH-TIPS, and Kylberg) with representative texture classification methods. The experimental results demonstrate that the proposed LENMP descriptor exhibits favorable classification performance and remains competitive compared to the latest descriptors.

The specific organizational framework of this paper is as follows. Section 2 introduces the concepts of LBP and CLBP. Section 3 elaborates on the LENMP method. Section 4 evaluates our approach and presents the comparative experimental results of LENMP with various descriptors. Section 5 summarizes the proposed method and provides a concluding analysis.

2. Related Work

The LBP operator operates within a 3x3 square vicinity, where the grayscale value of the central pixel in the window acts as the threshold. It assesses the grayscale values of the eight adjacent pixels surrounding the central pixel. If a neighboring pixel's grayscale value equals or surpasses that of the central pixel, it is denoted as 1; otherwise, it is designated as 0. This procedure is conducted either clockwise or counterclockwise, yielding a binary sequence, subsequently translated into a decimal figure, known as the LBP code.

$$LBP_{P,R} = \sum_{P=0}^{P-1} s\left(g_p - g_c\right)2^P \tag{1}$$

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

where g_c represents the grayscale value of the center pixel, g_p represents the grayscale value of the surrounding neighboring pixels.

The major difference between CLBP and the original LBP lies in the proposal of three different descriptors, namely CLBP_S, CLBP_M, and CLBP_C. The CLBP_S descriptor corresponds to the original LBP descriptor delineated in Equation (1). CLBP_M is defined as follows:

$$CLBP\_M_{r,p} = \sum_{p=0}^{p-1} s\left(m_p, c\right)2^p \tag{3}$$

$$s(x,c) = \begin{cases} 1, & x \geq c \\ 0, & x < c \end{cases} \tag{4}$$

Where c_I represents the average value of the complete image.

CLBP_C encodes the central pixel, defined as:

$$CLBP\_C_{r,p} = \sum_{p=0}^{p-1} s\left(g_c - c_I\right) \tag{5}$$

where c_I represents the mean grayscale magnitude of all pixels across the image. CLBP_S, CLBP_M, and CLBP_C can all generate binary codes, and combining them enhances texture classification accuracy while also providing robustness against illumination and noise.

3. Local Enhancement and Non-local Median Pattern (LENMP)

In this section, we offer a comprehensive overview of our proposed LENMP descriptor. The central idea of LENMP is as follows: firstly, preprocess the image using thACE and median filtering. Next, the LEP operator captures the sign and magnitude details of local image characteristics Then, the NMBP operator extracts the global information of image features. Finally, the joint histogram generated by combining the two operators forms the joint encoding for texture representation of LENMP.

3.1. Local Adaptive Contrast Enhancement Pattern(LEP)

The original LBP and its variants mostly focus on extracting local feature information, which may result in the omission of numerous texture details. While CLBP aims to extract more texture information, it is highly sensitive to noise. Inspired by CLBP, to extract more detailed information, we propose the Local Adaptive Contrast Enhancement Pattern (LEP). LEP utilizes thACE to enhance the contrast of high-frequency information in the image, magnifying it to extract more edge contour detail information, while also improving robustness to noise. LEP consists of two operators: LEP_S and LEP_M, which extract sign and amplitude information, respectively. The definition of the LEP_S is as follows:

$$LEP\_S = \sum_{p=0}^{p-1} s(g_{med_c} - g_{med_{r,p}}),\ s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{6}$$

Where g_{med_c} represents the median grayscale magnitude of the central pixel, $g_{med_{r,p}}$ represents the median grayscale magnitude of the neighboring pixels, and p denotes the quantity of pixels. After enhancing the image with thACE, the local LEP substitutes the grayscale intensity of each pixel with the median of the neighboring pixels within the local image block. The definition of the amplitude pattern LEP_M is as follows:

$$LEP\_M = \sum_{p=0}^{p-1} s(M_p, c)2^p \tag{7}$$

Where $M_p = |g_p - g_c|$ represents the median of the absolute the variation in the neighboring pixel value g_p and the central pixel value g_c, and $c$represents the mean value of M_p across the entire image. This value follows the encoding method of CLBP, and $s(\cdot)$ denotes the sign function.

3.2. threshold Adaptive Contrast Enhancement (thACE)

In images, useful texture information often corresponds to high-frequency information, which is more representative compared to low-frequency information. However, high-frequency information is often mixed into the low-frequency background signal, which reduces the efficiency of extracting texture details from the image and impacts the visual presentation of the image. To extract more high-frequency information, we introduce the threshold adaptive contrast enhancement algorithm (thACE). The thACE algorithm enhances high-frequency information in the image, while NSCT transformation continuously decomposes the image to extract more information and enhance robustness against noise. Therefore, thACE and NSCT are complementary, and combining them for image preprocessing is a suitable choice. The core of the thACE algorithm is to boost the high-frequency information of the samples. The thACE algorithm follows the process outlined below:

(1) Extracting low-frequency background information. Firstly, the mean value of the window is used to determine the background data of the image, i.e., the low-frequency information within the image, calculated as follows:

$$m_y(a,b) = \frac{1}{(2n+1)^2}\sum_{g=a-n}^{a+n}\sum_{h=b-n}^{b+n} y(g,h) \quad (8)$$

Where $m_x(a,b)$ denotes the average magnitude of the window of size $(2n+1)*(2n+1)$, which can also be rectangular. In this paper, a square window with a radius of *n* is utilized. The value of *n* is typically recommended to be between 20 and 25, although the specific value can be chosen according to user requirements. Generally, a larger *n* value leads to more lost image details during the process of extracting features. In this paper, *n* is set to 20.

(2) Compute the standard deviation. Calculate the standard deviation for the window of size $(2n+1)*(2n+1)$:

$$f(a,b) = m_y(a,b) + G(a,b)\big[y(a,b) - m_y(a,b)\big] \quad (9)$$

Where $x(a,b)$ represents the original pixel, $f(a,b)$ denotes the pixel value after adaptive contrast enhancement, and $G(a,b)$ represents the scaling factor CG. There are two methods to determine the value of CG:

a) It is set as a constant *C*, which can be adjusted based on the user's own results, typically ranging from 1 to 5. The enhanced grayscale value is defined as:

$$f(a,b) = m_y(a,b) + C\big[y(a,b) - m_y(a,b)\big] \quad (10)$$

b) Adaptive determination of $G(a,b)$ is defined as:

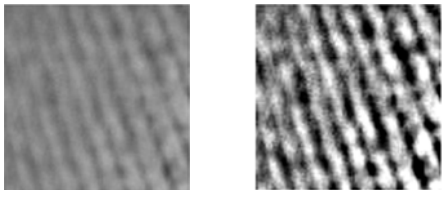
$$G(a,b) = \alpha\frac{M}{\sigma(a,b)} \quad (11)$$

Where $\sigma_y(a,b)$ represents the local standard deviation of the image, M and σ are constants. M can be taken as the global mean of the image or a suitable value set by the user, but in this paper, M is set as the global variance of the image. σ is set as 2. The approach selected in the experiment of this paper is the second adaptive plan.

Furthermore, unlike the ACE [23] algorithm, in this study, a threshold value, denoted as *th*, is established in $G(a,b)$ to restrict the maximum magnitude of the scaling factor CG (in this paper, *th* is set as 4). Because the value of CG is adaptive and inversely proportional to σ. In regions with large local standard deviations, the corresponding CG values will be smaller. Conversely, in smoother regions of the image where local standard deviations are smaller, the corresponding CG values will be larger. Therefore, in the thACE algorithm, when amplifying the high-frequency information of the image, amplification of noise in the image occurs, which is detrimental to feature extraction of descriptors. Thus, it is necessary to limit the maximum value of CG. This threshold is crucial for enhancing the effect; if $G(a,b)$ exceeds this threshold, it will be set to this threshold value.

The workflow of the thACE algorithm is as follows: the information in the image is divided into two types, one representing the local background of the image, which is the low-frequency information, and the other representing the enhanced image, which is the high-frequency information. The local background information can be obtained through low-pass filtering, while the enhanced image is obtained by subtracting the local background information from the original image. The goal of this algorithm is to enhance the high-frequency part, which represents more texture details. To achieve this, a scaling factor CG utilized to magnify this high-frequency information. Then, the enhanced image is superimposed on the initial image, resulting in an enhanced image where the edge contours are enhanced without losing low-frequency information. After undergoing thACE algorithm processing, the image extracts more high-frequency information, thereby improving the accuracy of texture classification.

Figure 1 illustrates the image enhancement effects of thACE on sample images from the texture dataset. Figure 1-(a) presents the original image from the UIUC dataset, while Figure 1-(b) displays the enhanced image after applying thACE to enhance the high-frequency information. It can be observed that after thACE processing, compared to the primary image, the contrast of the high-frequency information is enhanced, originally blurry details become clearer, and useful high-frequency information is amplified, which is more conducive to feature extraction.



(a) (b)

Figure 1. thACE image enhancement rendering, (a) represents the primary image, (b) represents the enhanced effect image.

3.3. Non-local Median Binary Pattern (NMBP)

Existing LBP and its variants typically focus only on local differences, neglecting the extraction of global information. Therefore, this paper introduces the NMBP operator. The approach in this section mainly involves improvements on the NLBP operator within LGONBP. Herein, we will delve into the NMBP operator. Specifically, the image undergoes median filtering, where the median of local image patches surrounding all neighboring pixels is used as the grayscale value for each pixel. Subsequently, all central pixels are uniformly divided into Q intervals, with an anchor point set in each interval, followed by encoding the intensity difference between all neighboring pixels and anchor points. The distinction from NLBP lies in the fact that NMBP undergoes thACE image processing beforehand. When extracting all central and neighboring pixels, the median grayscale value of their local image regions is utilized as their pixel grayscale values. The operational process of the NMBP operator is as follows:

For the central pixel x located in the image I, the median calculation of all pixels therein is defined as:

$$g_{med_c} = \theta\left(G_{c,w}\right) \tag{12}$$

$$g_{med_{r,p}} = \theta\left(G_{r,p,w}\right) \tag{13}$$

Where $G_{c,w}$ represents the local image block of size $w \times w$ centered at x, $G_{r,p,w}$ represents the local image block surrounding the p-th neighboring pixel, and $\theta(\cdot)$ is the operation function, i.e., the median grayscale value function.

The grayscale magnitudes of all central pixels x within image I are arranged in ascending order, defined as:

$$\left(g'_{med_{c1}}, \ldots, g'_{med_{cM}}\right) \coloneqq sort\left(g_{med_{c1}}, \ldots, g_{med_{cM}}\right) \tag{14}$$

Where, $sort(\cdot)$ denotes the built-in function for ascending sorting, g_{med_c} represents the median grayscale value of central pixel x, M stands for the number of central pixels, and $g_{med_{cm}}$ denotes the median grayscale value of the m-th central pixel.

Next, the sorted pixels mentioned above are equally divided into Q intervals, and then the anchor points for each interval are computed, defined as:

$$g_{A_q} = \frac{1}{M/Q}\sum_{m=(q-1)\lfloor M/Q\rfloor+1}^{q\lfloor M/Q\rfloor} g_{med_{cm}} \tag{15}$$

Where g_{A_q} represents grayscale value magnitude of the q-th anchor point ($q = (1,2,3,\ldots,Q)$), and $\lfloor\cdot\rfloor$ denotes the floor function, indicating the largest integer less than M/Q. Each anchor point is represented by the median grayscale value of all central pixels in each interval. Setting anchor points facilitates the extraction of global image features.

Fig. 2 provides a structural comparison between LBP and NMBP. Fig. 2-(a) illustrates the sampling structure of the original LBP, which simply calculates the differential information between the central pixel and its 8 neighboring pixels, resulting in limited information and inadequate spatial support. In contrast, Fig. 2-(b) depicts the sampling structure of NMBP, which first calculates the median of the pixels surrounding each neighboring pixel and the central pixel. These medians are then used as their respective grayscale values. It is evident that NMBP captures larger spatial scale information compared to LBP.

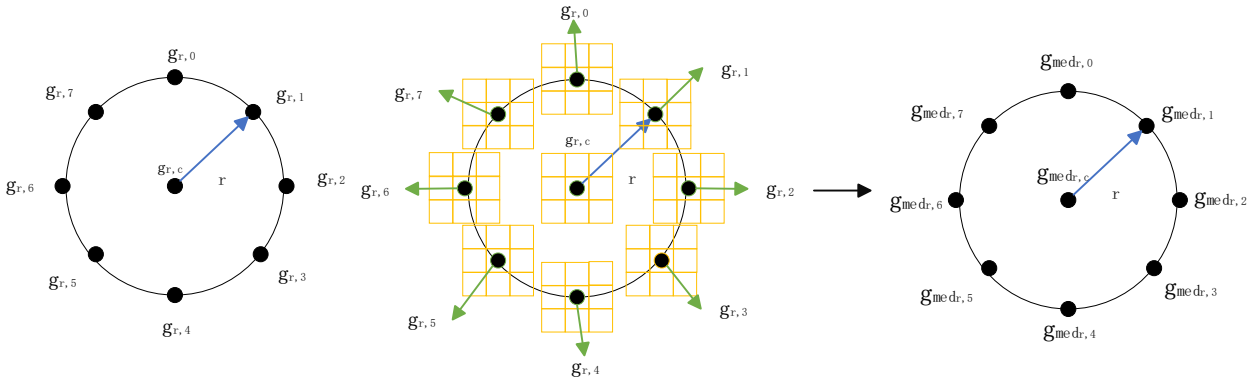


Figure 2. Sampling structures of LBP and LENMP, (a) represents the LBP sampling structure, (b) represents the LENMP sampling structure.

Fig. 3 depicts the workflow of LENMP. Initially, the image undergoes adaptive contrast enhancement through thACE to extract high-frequency information. Subsequently, median extraction is applied separately to the LEP and NMBP operators. The LEP operator extracts symbol and amplitude components, i.e., local information. The NMBP operator captures global image information, with the two operators complementing each other. Finally, they are concatenated to form the LENMP descriptor. The histogram representation of the cascaded LENMP descriptor is as follows (experimentally, in this paper, the scale selection for the LENMP descriptor is set to R=1, P=8):

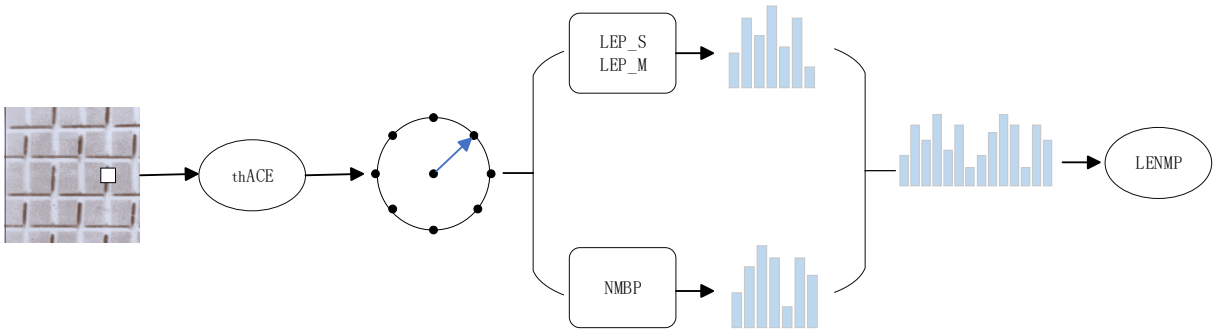


Figure 3. Structure diagram of LENMP

4. Experiments

In this section, we conduct a series of comparative experiments on multiple representative datasets (Outex, CUReT, Brodatz, KTH-TIPS, and Kylberg) with various methods. Among them, four subsets of the Outex dataset (TC0, TC3, TC10, and TC11) are utilized to validate the discriminative ability of the LENMP descriptor in feature extraction and its robustness to noise.

4.1. Texture Datasets and Experimental Settings

4.1.1. Texture Datasets

The Outex texture dataset [24] comprises 17 sub-datasets. In this study, experiments were conducted on OutexTC_00000 (TC0), OutexTC_00003 (TC3), OutexTC_00010 (TC10), and OutexTC_00011 (TC11). The CUReT dataset [25]contains 61 texture types, with each class comprising 205 images, exhibiting rich geometric dimensions and optical characteristics. The Brodatz dataset [26]comprises 112 texture categories, with each category containing 10 black and white texture images sized 512*512 pixels. This study utilized 1792 samples from 112 classes, with 16 images per class featuring different angles and lighting conditions. The KTH-TIPS texture dataset [27], complementing CUReT, introduces scale variations and includes 10 texture categories, each with 81 images sized 200*200 pixels. This study selected 810 color and pre-processed grayscale images from 10 classes. The Kylberg texture dataset [28] comprises 28 texture categories, with each category containing 160 texture images sized 576*576 pixels. This study utilized 1200 image samples from 20 classes.

4.1.2. Experimental Settings

In this paper, there are mainly two parameters of significant importance: the choice of scale and the choice of filter. We conduct a series of comparative experiments on these parameters, as follows:

Fig. 4 illustrates the experiment on the scale selection of LENMP descriptors. The horizontal axis represents 11 datasets, namely KTH-TIPS, Kylberg, Brodatz, CUReT, TC0, TC1, TC3, TC6, TC8, TC10, and TC11. The black line plot corresponds to R=1, P=8, the blue line plot corresponds to R=2, P=16, and the green line plot corresponds to R=3, P=24. It can be observed from the graph that as the radius and the number of sampling points increase, the classification accuracy of the descriptors varies to some extent across the datasets. The black line plot exhibits good performance across multiple datasets, indicating that R=1 and P=8 are chosen as the radius and sampling point size in this paper.

Fig. 5 presents the filter selection experiment for LENMP descriptors. To capture more spatial detail information, we conducted comparative experiments using five different methods, namely median filtering, mean filtering, maximum filtering, minimum filtering, and no preprocessing. The comparisons were performed across nine datasets. From the graph, it can be observed that except for the Kylberg dataset where all five preprocessing methods yield the same results, median filtering consistently outperforms other preprocessing methods across the remaining datasets, demonstrating stable and satisfactory performance with minimal overall fluctuation. Therefore, in this paper, median filtering is chosen for image preprocessing.

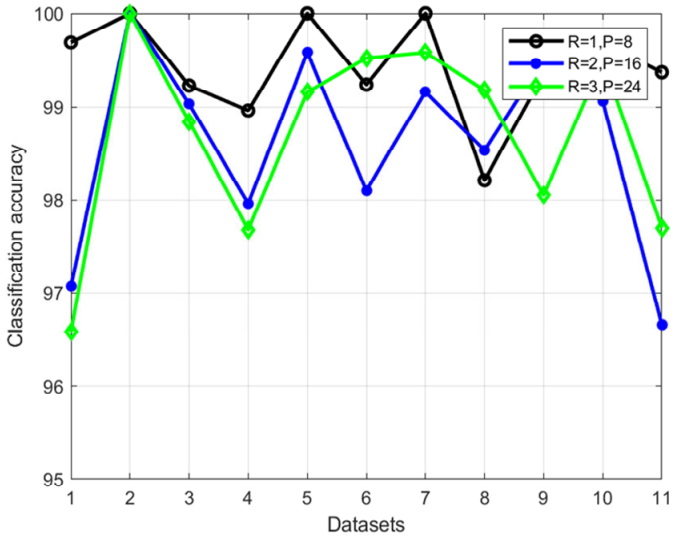


Figure 4. LENMP scale selection

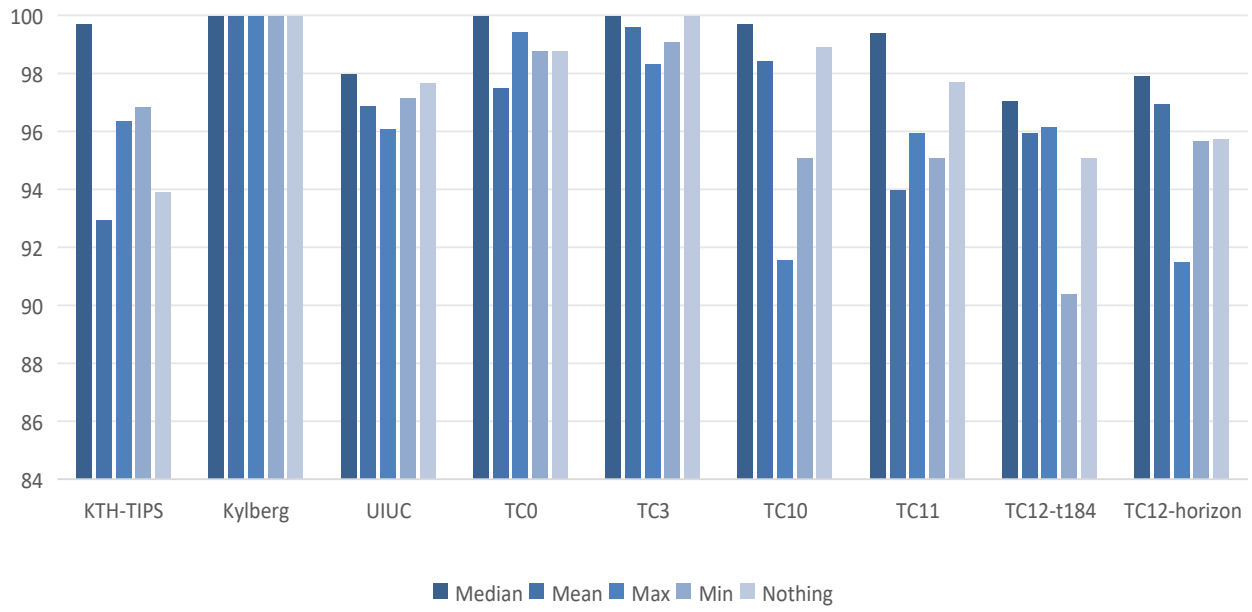


Figure 5. LENMP filter selection

4.2. Classification Performance & Comparison

In this subsection, we conduct multiple comparative experiments between the LENMP descriptor and eleven representative descriptors. The datasets used include Outex, CUReT, Brodatz, Kylberg, and KTH-TIPS. The eleven descriptor methods include LBP[3], CLBP[13], CLBC[29], ECLBP[30], MCDR[31], JRLP[32], MSDMD[17], SMDCP[33], LETRIST[34], SSLBP[35], and LGONBP[22].

4.2.1. Performance and Comparison on Noise-Free Datasets

Table 1 presents the outcomes of comparative experiments conducted on eight datasets without noise. From Table 1, it can be observed that, except for the CUReT dataset, our proposed LENMP outperforms other descriptors on the remaining seven datasets. On the CUReT dataset, the classification performance of LENMP is only slightly lower than that of SMDCP. Additionally, compared to the LGONBP descriptor, LENMP shows a certain degree of improvement. LENMP achieves 100% classification accuracy on the Kylberg, TC0, and TC3 datasets.

Table 1. Comparison of noise free classification results between LENMP and various methods on different datasets (%)

Methods	Brodatz	KTH-TIPS	CUReT	Kylberg	TC10	TC11	TC0	TC3
LBP	77.50	74.70	71.62	94.80	84.80	71.04	91.25	91.16
CLBP	96.46	94.60	93.26	98.80	96.56	95.62	98.34	92.31
CLBC	97.50	95.34	97.68	97.28	97.10	96.73	97.70	84.30
ECLBP	98.82	78.71	82.51	98.71	73.07	62.70	65.00	65.83
MCDR	94.85	95.60	91.66	99.30	99.50	95.72	97.70	97.62
JRLP	97.66	95.70	98.54	99.60	97.10	97.08	98.75	99.58
SMDCP	98.71	98.71	99.10	97.60	95.76	97.71	99.16	99.58
MSDMD	96.32	97.78	96.79	100	-	-	-	-
LGONBP	90.89	95.76	97.81	100	99.48	98.95	99.16	99.58
LENMP	99.23	96.58	98.96	100	99.68	99.38	100	100

4.2.2. Robustness Analysis to Gaussian & Salt-and-Pepper Noise

Table 2, 3, and 4 respectively present the comparative experimental results on the KTH-TIPS, Brodatz, and CUReT datasets with different levels of salt-and-pepper and Gaussian noise, denoted as 0.01, 0.05, and 0.1. Observing the KTH-TIPS dataset, under various noise conditions, our proposed LENMP descriptor generally exhibits higher classification accuracy compared to other descriptors. For instance, under 0.01 Gaussian noise, LENMP falls slightly behind LETRIST and JRLP; however, as noise increases, while the classification results of other descriptors decline, LENMP remains stable. On the Brodatz dataset, LENMP ranks second only to LETRIST under 0.05 salt-and-pepper noise. Similarly, on the CUReT dataset, LENMP trails behind SSLBP without noise; however, it performs well under different levels of noise.

Table 2. The comparative results with noise added on the KTH-TIPS dataset (%)

Methods	Noise-free	Salt & Pepper noise (σ)			Gaussian noise (ρ)		
		0.01	0.05	0.1	0.01	0.05	0.1
CLBP	94.60	94.20	92.70	92.90	83.15	86.10	84.58
CLBC	95.34	92.9	93.46	92.54	91.17	93.10	89.54
MCDR	95.60	92.2	91.22	90.83	82.56	84.39	81.12
ECLBP	78.71	82.12	87.10	87.61	82.76	81.63	79.46
JRLP	95.70	93.78	93.60	93.61	97.46	97.21	97.41
LETRIST	97.70	97.38	97.83	98.38	97.53	97.22	96.91
LGONBP	95.76	96.29	96.36	96.15	96.97	97.08	96.93
LENMP	99.69	98.46	98.46	98.77	97.22	97.22	99.07

Table 3. Comparison results of different methods for adding noise on the Brodatz dataset (%)

Methods	Noise-free	Salt & Pepper noise (σ)			Gaussian noise (ρ)		
		0.01	0.05	0.1	0.01	0.05	0.1
CLBP	96.46	97.16	98.54	98.38	74.53	41.15	32.21
CLBC	97.50	98.20	98.57	98.59	97.45	91.07	83.09
MCDR	94.85	94.10	95.48	99.20	90.64	85.68	77.01
JRLP	97.66	98.32	99.18	99.30	98.09	96	95.25
LETRIST	97.90	97.81	100	99.06	97.81	98.12	98.12
SMDCP	98.71	98.68	99.37	98.68	98.43	98.30	98.43
LGONBP	90.89	97.79	98.32	98.07	98.34	97.39	96.07
LENMP	99.23	99.06	99.69	99.38	99.68	99.06	98.68

Table 4. Comparison results of different methods for adding noise on the CUReT dataset (%)

Methods	Noise-free	Salt & Pepper noise (σ)			Gaussian noise (ρ)		
		0.01	0.05	0.1	0.01	0.05	0.1
CLBP	93.26	93.69	92.73	91.00	83.81	64.62	51.73
CLBC	97.68	96.46	94.64	92.52	85.82	65.99	51.94
MCDR	91.66	91.57	89.06	86.78	68.77	40.51	29.59
ECLBP	82.51	88.13	87.03	86.32	55.72	40.81	36.13
JRLP	98.54	98.46	94.64	92.52	93.65	92.87	91.66
LETRIST	98.54	96.56	94.37	92.18	95.00	88.75	85.62
SSLBP	99.51	98.61	97.34	95.83	98.21	90.91	81.66
LGONBP	96.79	98.39	97.12	95.12	93.13	91.00	94.07
LENMP	98.96	98.75	97.50	95.94	94.69	93.19	94.94

4.2.3. Gaussian noise robustness analysis (SNR)

Tables 5, 6, and 7 present the respective comparative experimental outcomes involving various SNR noise levels introduced to the KTH-TIPS, Brodatz, and CUReT datasets. SNR signifies the proportion of signal power to noise power, with values of SNR=100, 30, 15, 10, and 5 (lower SNR values indicate greater noise contamination). On the KTH-TIPS, Brodatz, and CUReT datasets, our proposed LENMP descriptor consistently outperforms other descriptors in classification accuracy. As noise increases, the classification accuracy of other descriptors tends to decline, while LENMP maintains stability.

Based on all the experiments conducted across multiple typical texture datasets, comparing with the most representative texture classification methods and the latest techniques, it is verified that our proposed LENMP descriptor exhibits robustness against noise and enhances texture classification accuracy. Thus, it demonstrates competitive performance in the field of texture classification.

Table 5. Comparison results of different methods for adding different degrees of SNR on the KTH-TIPS dataset (%)

Methods	Noise-free	SNR				
		100	30	15	10	5
CLBP	94.60	94.25	90.29	85.20	80.34	66.64
CLBC	95.34	94.26	90.19	84.40	80.07	67.65
ECLBP	78.71	86.05	78.57	75.25	72.38	66.71
MCDR	95.60	92.82	86.34	78.95	73.95	66.00
JRLP	95.70	94.86	93.98	91.12	87.67	82.89
LETRIST	97.70	98.02	95.85	87.80	86.82	75.36
SSLBP	99.39	95.10	93.07	90.28	85.36	81.94
MSDMD	97.78	59.02	60.07	61.54	62.02	66.44
LGONBP	95.76	92.89	91.62	89.88	88.04	78.78
LENMP	99.69	98.15	97.84	95.68	95.68	93.21

Table 6. Comparison results of different methods for adding different degrees of SNR on the Brodatz dataset (%)

Methods	Noise-free	SNR				
		100	30	15	10	5
CLBP	96.46	91.82	86.25	80.70	76.17	66.30
CLBC	97.50	90.57	86.61	85.75	85.46	78.39
ECLBP	78.71	78.25	71.08	65.62	60.70	51.70
MCDR	94.85	91.98	90.98	89.12	88.90	86.31
JRLP	97.66	91.98	90.98	89.12	88.51	86.32
LETRIST	97.90	97.14	97.14	95.89	94.82	94.28
SSLBP	100	90.57	89.48	88.69	88.69	86.71
MSDMD	96.32	34.93	37.24	38.39	38.88	38.83
LGONBP	90.89	92.32	91.96	93.03	93.21	92.32
LENMP	99.23	98.85	98.17	97.31	97.40	96.63

Table 7. Comparison results of different methods for adding different degrees of SNR on the CUReT dataset (%)

Methods	Noise-free	SNR				
		100	30	15	10	5
CLBP	93.26	94.85	90.17	84.49	79.51	69.03
CLBC	97.68	95.48	91.45	86.55	81.65	71.67
ECLBP	82.51	96.70	91.43	84.78	83.22	83.48
MCDR	91.66	69.60	59.03	49.47	43.62	35.61
JRLP	98.54	96.93	96.20	95.40	94.56	91.68
LETRIST	98.50	97.02	95.85	87.80	86.82	75.36
SSLBP	99.51	96.90	96.53	94.83	92.53	87.56
MSDMD	97.81	53.68	59.21	62.43	63.47	65.07
LGONBP	96.79	96.97	96.52	95.13	93.75	89.56
LENMP	98.96	97.50	96.61	95.36	94.97	92.12

4.2.4. Ablation experiments

Table 8 presents the results of ablation experiments conducted on seven datasets targeting the thACE module. There are three methods of adding the thACE module. Since our method is inspired by CLBP and LGONBP and improves upon these two methods, the comparative methods for the ablation experiments are CLBP, LGONBP, and LENMP, with LNMP representing the operator without the ACE module. CLBP_thACE and LGONBP_thACE respectively combine the LEP module with CLBP and LGONBP. From the table, it can be observed that after adding thACE, CLBP_thACE and LGONBP_thACE exhibit varying degrees of improvement in classification accuracy compared to CLBP and LGONBP. LENMP shows improved classification accuracy compared to LNMP across all datasets, with 100% accuracy achieved on the TC0 and TC3 datasets. Moreover, LENMP outperforms CLBP, CLBP_thACE, LGONBP, LGONBP_thACE, and LNMP in terms of classification accuracy. However, its performance is less satisfactory on the TC10 and TC11 datasets, indicating insufficient robustness to rotation invariance. Nevertheless, on the TC10 and TC11 datasets, the classification accuracy of LENMP with the added thACE module surpasses that of LGONBP and LGONBP_thACE, demonstrating the effectiveness of the thACE module and its ability to provide some degree of rotation invariance.

Table 8. Results of ablation experiments with thACE added on different datasets (%)

Methods	Brodatz	CUReT	KTHTIPS	TC0	TC3	TC10	TC11
CLBP	96.46	93.26	94.60	90.4	97.7	96.56	95.62
CLBP_thACE	96.74	95.51	96.00	91.63	99.00	98.92	98.41
LGONBP	90.89	96.79	95.76	99.17	99.58	99.48	98.95
LGONBP_thACE	94.64	97.39	96.09	99.58	99.73	99.61	99.16
LNMP	98.21	96.47	94.14	99.16	99.16	96.97	96.66
LENMP	99.23	97.96	99.69	100	100	99.68	99.38

5. Conlusion

This paper proposes a method called Local Enhancement and Non-local Median Pattern (LENMP) for texture classification. LENMP captures more edge contour details and enhances robustness to Gaussian and salt and pepper noise. Firstly, thACE is employed to enhance the high-frequency information of the image, extracting more texture edge information. Then, the LEP and NMBP operators are subjected to median sampling to extract larger spatial information. Combining LEP and NMBP, local and non-local information are respectively extracted, and the complementary operators are combined to obtain the LENMP descriptor. A series of comparative experiments are conducted on several public datasets (Outex, CUReT, Brodatz, KTH-TIPS, and Kylberg) with various descriptor methods. The experimental outcomes demonstrate that the LENMP descriptor exhibits excellent classification performance and noise robustness, remaining competitive compared to the latest descriptor methods.

References

[1] Khadiri I E, Merabet Y E, Tarawneh A S, et al. Petersen graph multi-orientation based multi-scale ternary pattern (PGMO-MSTP): An efficient descriptor for texture and material recognition [J]. IEEE Transactions on Geoscience and Remote Sensing, 2021, 30: 4571-4586.

[2] Xiao B, Wang K, Bi X, et al. 2D-LBP: An enhanced local binary feature for texture image classification [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 9(29): 2796-2808.

[3] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987.

[4] Liao S, Law M W K, Chung A C S. Dominant local binary patterns for texture classification[J]. IEEE Transactions on Image Processing, 2009, 18(5): 1107-1118.

[5] Deng W, Hu J, Guo J. Compressive binary patterns: Designing a robust binary face descriptor with random-field eigenfilters [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(3): 758-767.

[6] Barua P D, Keles T, Dogan S, et al. Automated EEG sentence classification using novel dynamic-sized binary pattern and multilevel discrete wavelet transform techniques with TSEEG database [J]. Biomedical Signal Processing and Control, 2023, 79: 104055.

[7] Zhong Y, Deng W, Hu J, et al. SFace: Sigmoid-constrained hypersphere loss for robust face recognition [J]. IEEE Transactions on Image Processing, 2021, 30: 2587-2598.

[8] Yu X, Lu X. Domain adaptation of anchor-free object detection for urban traffic [J]. Neurocomputing, 2024: 127477.

[9] Girum K B, Créhange G, Lalande A. Learning with context feedback loop for robust medical image segmentation [J]. IEEE Transactions on Medical Imaging, 2021, 40(6): 1542-1554.

[10] Xia Z, Yuan C, Lv R, et al. A novel weber local binary descriptor for fingerprint liveness detection [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 50(4): 1526-1536.

[11] Dong Y, Jin M, Li X, et al. Compact interchannel sampling difference descriptor for color texture classification [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(5): 1684-1696.

[12] He J, Shen L, Yao Y, et al. Finger vein image deblurring using neighbors-based binary-GAN (NB-GAN) [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2021, 7(2): 295-307.

[13] Guo Z, Zhang L, Zhang D. A completed patternling of local binary pattern operator for texture classification [J]. IEEE Transactions on Image Processing, 2010, 19(6): 1657-1663.

[14] Tan X, Triggs B. Enhanced local texture feature sets for face recognition under difficult lighting conditions [J]. IEEE Transactions on Image Processing, 2010, 19(6): 1635-1650.

[15] Zheng Z, Xu B, Ju J, et al. Circumferential local ternary pattern: New and efficient feature descriptors for anti-counterfeiting pattern identification [J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 970-981.

[16] Guo Z, Zhang L, Zhang D, et al. Rotation invariant texture classification using adaptive LBP with directional statistical features [C]. IEEE International Conference on Image Processing, 2010: 285-288.

[17] Dong Y, Wu H, Li X, et al. Multiscale symmetric dense micro-block difference for texture classification [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 29(12): 3583-3594.

[18] Hu S, Pan Z, Dong J, et al. A novel adaptively binarizing magnitude vector method in local binary pattern based framework for texture classification [J]. IEEE Signal Processing Letters, 2022, 29: 852-856.

[19] Ma S, Hu Q, Zhao S, et al. Multiscale multidirection binary pattern learning for discriminant palmprint identification [J]. IEEE Transactions on Instrumentation and Measurement, 2023, 72: 1-12.

[20] Liu L, Long Y, Fieguth P W, et al. BRINT: binary rotation invariant and noise tolerant texture classification [J]. IEEE Transactions on Image Processing, 2014, 23(7): 3071-3084.

[21] Liu L, Lao S, Fieguth P W, et al. Median robust extended local binary pattern for texture classification [J]. IEEE Transactions on Image Processing, 2016, 25(3): 1368-1381.

[22] Song T, Feng J, Luo L, et al. Robust texture description using local grouped order pattern and non-local binary pattern [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(1): 189-202.

[23] Narendra P M, Fitch R C. Real-time adaptive contrast enhancement [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1981 (6): 655-661.

[24] Ojala T, Maenpaa T, Pietikainen M, et al. Outex-new framework for empirical evaluation of texture analysis algorithms [C]. International Conference on Pattern Recognition. 2002, 1: 701-706.

[25] Xu Y, Yang X, Ling H, et al. A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid [C]. IEEE Conference on Computer Vision and Pattern Recognition. 2010: 161-168.

[26] Brodatz P. Textures: a photographic album for artists and designers [M]. Dover Publications, 1966.

[27] Dana K J, Van Ginneken B, Nayar S K, et al. Reflectance and texture of real-world surfaces [J]. ACM Transactions on Graphics, 1999, 18(1): 1-34.

[28] Kylberg G. Kylberg texture dataset v. 1.0 [M]. Centre for Image Analysis, 2011.

[29] Zhao Y, Huang D S, Jia W. Completed local binary count for rotation invariant texture classification [J]. IEEE Transactions on Image Processing, 2012, 21(10): 4492-4497.

[30] Sima J, Dong Y, Wang T, et al. Extended contrast local binary pattern for texture classification [J]. International Journal of New Technology and Research, 2018, 4(3): 263120.

[31] Dong Y, Feng J, Yang C, et al. Multi-scale counting and difference representation for texture classification [J]. The Visual Computer, 2018, 34: 1315-1324.

[32] Wang T, Dong Y, Yang C, et al. Jumping and refined local pattern for texture classification [J]. IEEE Access, 2018, 6: 64416-64426.

[33] Dong Y, Zheng B, Liu H, et al. Symmetric mean and directional contour pattern for texture classification [J]. Electronics Letters, 2021, 57(24): 918-920.

[34] Song T, Li H, Meng F, et al. LETRIST: Locally encoded transform feature histogram for rotation-invariant texture classification [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 28(7): 1565-1579.

[35] Guo ZH, Wang X, Zhou J, et al. Robust texture image representation by scale selective local binary patterns [J]. IEEE Transactions on Image Processing, 2016, 2(25): 687-699.