



Texture image Classification based on improved local Quinary patterns

Laleh Armi¹ · Shervan Fekri-Ershad^{1,2}

Received: 29 April 2018 / Revised: 4 January 2019 / Accepted: 10 January 2019 /

Published online: 6 February 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Texture image classification is an active research topic in computer vision that play an important role in many applications such as visual inspection systems, object tracking, medical image analysis, image segmentation, etc. So far, there are many descriptors for texture image analysis such as local binary patterns (LBP). LBP is a nonparametric operator, which describes the local spatial structure and the local contrast of an image. Local quinary patterns (LQP) is one of the improved versions of LBP in terms of classification accuracy. Statistic input parameters and don't providing significant binary patterns are some disadvantages of LQP. In this paper a new version of LBP is proposed, which is known as improved local quinary patterns (ILQP). In this paper, a new definition is proposed to divide local quinary codes to four binary patterns. Each extracted binary patterns represent a subset of local features. Also, a new algorithm is proposed here to provide dynamic thresholds in dividing process of LQP. The proposed approach is evaluated using Outex, and Brodatz data sets. Our approach has been compared with some state-of-the-art methods. It is experimentally demonstrated that the proposed approach achieves the highest accuracy in comparison with most of the state-of-the-art texture classification approaches. Low computational complexity, rotation invariant, low impulse-noise sensitivity and high usability are advantages of the proposed texture analysis descriptor.

Keywords Texture image classification · Local Quinary patterns · Local binary patterns · Feature extraction

1 Introduction

The texture is recognizable in both tactile and visual ways. Tactile texture refers to the tangible feel of a surface, and visual texture refers to observation content of the image [20]. Texture

✉ Shervan Fekri-Ershad
fekriershad@pco.iaun.ac.ir

¹ Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

² Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran

detection is easy in human vision; however, it has some complexities in machine vision and image processing.

As the order of warp and woof forms the structure of the fabric, in the image-processing field, the texture of the image refers to the intensity distribution of the pixels in an image next to each other [37]. Therefore, textural images in the image processing refers to the images in which a specific pattern of distribution and dispersion of the intensity of the pixels is repeated sequentially throughout the image [37]. For example, in Fig. 1, an example of a texture image with a repetitive pattern is shown.

The texture of the image can provide useful information about the content, objects within the image and their relationship with the background. Therefore, the analysis of texture and the classification of texture images has become one of the most important issues in the field of image processing.

The texture analysis usually tends to solve the four main texture problems, which includes texture classification, texture segmentation, texture synthesis, and texture shape extraction [8, 20, 37]. In our research, we will address the texture classification. Texture classification means assigning a texture image sample to a predefined texture class. In texture classification, the purpose is to assign an unknown sample image to one of the known texture classes. Texture classification approaches usually involves a two-phase process:

- a) Feature Extraction Phase: At this phase, the texture features are extracted from within the image.
- b) The Classification Phase: At this phase, using a classifier, the extracted features from the test image are compared with the training images, and the test image class is determined.

The most challenging part of texture image analysis is the feature extraction. There are various approaches to extract texture features. Choosing the discriminative features will increase the performance of the classifications.

In real applications, there are three major challenges in texture analysis includes rotation, scale sensitivity and noise in the image. In the event that the methods used to classify and extract features are not sustainable against these three common phenomena, the accuracy of the results may be severely reduced in practice; therefore, in real circumstances, the methods used for texture classification must be, as far as possible, robust and stable against these two



Fig. 1 An example of a textural image. (a) Texture Image (b) Repetitive pattern of texture

phenomena. For this reason, various methods have been presented so far to extract the features of textural images, each of which attempts to cover these aspects. The methods presented in this field can be divided into four main groups: Statistical methods, structural methods, model-based methods, and transform-based methods [8, 10, 11, 28]. Among the most popular statistical methods, can refer to the Gray level co-occurrence matrix (GLCM) [17], Histogram properties, and local binary patterns (LBP) [24]. Methods based on the edge features [40], morphological operators [5], skeletal representations [6] can be put into structural group. Fractal [32] is one of the existing methods in a model-based category; and as one of the most widely used methods for extracting features in the transformation-based method, one can refer to wavelet transform, Curvelet transform [3], and Gabor transform [14].

Today, most of the methods can be assigned to more than one group because the feature extraction methods have become more complicated and combine multiple methods to describe texture images.

Among the existing methods, local binary patterns (LBPs) are known as one of the efficient and most widely used local texture descriptors [24]. LBPs, in their original and modified form, due to their simplicity in implementation, are resistant and sustainable to uniform changes in the intensity, and have been also considered by many experts because of the extraction of appropriate features with high classification accuracy. This operator initially considers a neighborhood of the image, and then compares the intensity of the points in this neighborhood with the intensity of the neighbor center.

In the modified method, the LBP operator first defines the uniformity criterion [25], which indicates the number of mutations in the intensity of the illumination of the neighboring points.

One of the problems with the LBP operator is noise sensitivity, so the central pixel is directly considered as the threshold. Another disadvantage of this operator is low resolution, i.e., local patterns with different structures produce the same binary code, and in a local pattern, it does not matter if the illumination intensity difference of the neighbor pixel and the central pixel is 100 units or just 1 unit. Hence, varieties of enhanced methods for extracting textual images, including Local Quinary Patterns (LQPs) [22], have been developed. LQPs are among the enhanced types of the LBPs. In this method, using two fixed thresholds determined by the user, a five-digit encryption is created, and then using a function, this pattern is divided into four binary patterns. Finally, the histograms obtained from the binary patterns are combined to form the final histogram. This descriptor also has drawbacks that we attempt to overcome in our research. We also attempt to present a method that increases the accuracy of classification and has a higher resistance to noise as compared to other methods.

In this paper, we present a new method called the Improved Local Quinary Pattern with notation ILQP. One of the important drawbacks of the LQP method is the threshold determination that should be set by the user as input parameters. In addition, the accuracy of the LQP analysis is highly sensitive to the threshold.

Using an automatic algorithm to select threshold, instead of fixed thresholds, is one of the innovations of this paper. Fixed thresholds absolutely is not optimal for all image types and datasets. Another drawback of this method (LQP) is the ambiguous definition of the division of the quinary code into four binary patterns, especially in case of the second and third binary patterns, because the resulted binary patterns do not represent any particular occurrence in the image. The zero values in resulted binary code do not have the same attribute and meaning. Therefore, we have solved this problem to a large extent by defining a new function for division quinary pattern into four binary definitive patterns. Motivated by above problems, a texture descriptor is proposed in this paper.

In our descriptor ILQP, first of all a new algorithm is proposed to divide local quinary code to four meaningful binary patterns. Next, two criteria is proposed to compute thresholds of dividing process in LQP dynamically. Also, feature extraction is used based on uniformity labeling to increase discriminatively of the extracted feature vector. According to the results on databases of Brodatz [4], Outex TC_000010 [26], Outex TC_000012 [26] and KTH-TIPS [36] it is clear that the proposed method provide higher classification accuracy in comparison with state-of-the-art methods. Also, the resistance of the proposed method was investigated for impulse noise. The results show that the proposed method is more resistant to noise compared to other methods.

1.1 Texture analysis applications

Image texture, gives us useful information about content of the images, objects within it, the relationship between foreground and background. The analysis and classification of textures will apply in any field of image processing that comparing image contents is the aim. Some applications of image texture analysis are as follows:

- **Medical images analysis:** In some kind of medical images, the difference between healthy and non-healthy cases may be appeared in image textures such as tumor detection, mammogram analysis, pap smear classification, etc.
- **Face recognition:** Texture features in combination of color and shape features, can be used as feature extraction phase for face recognition
- **Objects tracking:** Tracing can be defined as an estimate of the path of an object's movement when the object moves on the scene. The texture feature of an object with a background texture feature is quite different. Therefore, So, texture properties can be used to separate the background area from the object, and it can be used to track objects.
- **Surface Defect detection:** Each abnormality in surface images can be considered as a defect. In many cases, the texture of defect parts is going to be different in comparison of non-defected parts.
- **Image retrieval:** The main aim of the content-based image retrieval systems (CBIR) is to retrieve most similar images to the query one from a huge database. Texture features can be used to detect similarity between database images and query image.

1.2 Paper organization

The rest of this paper is as follows: in the section 2, we will overview the primary version of LBP and its enhanced versions. In section 3, we will propose our method, which includes three sub-sections. Experimental results are discussed in section 4 and finally, a general conclusion is provided in the section 5.

2 Research background

As it is described above, LBP is a non parametric texture analysis operation, which provides discriminative features. It is generally used now to handle feature extraction phase of computer

vision applications. Here, we will overview the primary version of local binary patterns (LBPs) and its enhanced versions, as well as the local quinary patterns (LQPs).

2.1 Primary version local binary pattern

Local binary pattern (LBP) is one of the textural image descriptors that can define the local spatial structure and the local contrast of the image. LBP has become a widely used texture descriptor due to its simple implementation and extraction of proper features, and this is why many researchers have considered it. Rotation invariant, gray scale and low computational complexity are some features that have made LBP one of the most suitable texture analysis methods. Ojala et al., [24] first introduced the LBP operator.

In this descriptor, a neighborhood is considered first for each point of the image. Then the intensity of the central pixel is compared with the intensity of the neighboring pixels. If the intensity of the neighboring pixel is larger than the central pixel, then the value for that neighbor in extracted binary pattern considered as is one, and otherwise it is zero. Finally, with a binary weighted sum of the values in the binary extraction pattern we obtain values at the base of ten, which is called the LBP value. The calculation process of the LBP value is shown in the equation below.

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

$$S(x) = \begin{cases} 1 & g_p - g_c > 0 \\ 0 & otherwise \end{cases} \tag{2}$$

Where, P shows the number of neighbors for central pixel c , g_c represents the intensity of the central pixel and g_p represents the intensity of the p^{th} neighbor. The $S(x)$ sign function is used to generate binary values, and is calculated according to Eq. 2. LBP operator should be applied to all pixels of the image. Finally, eventually the histogram of LBP values is constructed that can be used to extract the features. An example LBP computation process is shown in Fig. 2.

In order to prevent the operator from being sensitive to rotation, the neighborhood is considered as a circle and the intensity of the points whose coordinates do not exactly match the center of the pixel is determined through interpolation. Fig. 3 shows a circular neighborhood with different radius R and neighboring points number P .

Given the definition provided for the LBP, the output of this operator is a binary P -bit number so that 2^P has different values. When an image is rotated, all neighbors will rotate around the central pixel g_c in one direction, and this will result in the production of different values for $LBP_{P,R}$. In other words, the value of $LBP_{P,R}$ depends on the indexing of neighboring

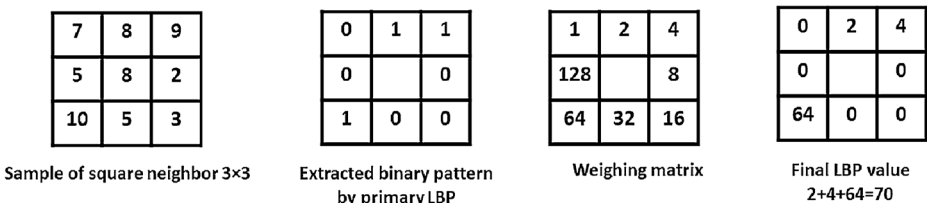


Fig. 2 An example of primary LBP computation process in a 3*3 neighborhood

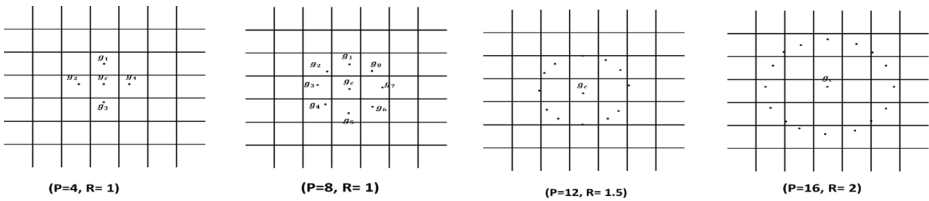


Fig. 3 Circular symmetric neighbors for different values of P and R

pixels. To eliminate rotating destructive effects, the bit rotation to right operator is used so that a unique value is assigned to each of the patterns [27]. This independent rotation operator is calculated as follows:

$$LBP_{P,R}^{ri}(x, y) = \min\{ROR(LBP_{P,R}(x, y), i) | i \in [0, P-1]\} \tag{3}$$

In Eq. 3, the ROR function denotes the rotation to the right on the P-bits binary number, and the ri symbol represents the operator’s insensitivity to the rotation. This is performed i times and the minimum number obtained for each i between 0 to P-1 is the final value of the LBP.

2.2 Modified version of LBP

As indicated in the $LBP_{P,R}^{ri}$ method, its original form, has a high computational complexity and, with increasing number of neighboring points, this computational complexity increases. Many modifications are proposed to improve LBP. The first modification for LBP, called the uniform MLBP, was introduced by Ojala [25].

In MLBP, the uniformity shows the number of mutations occurring between 0 and 1 (and vice versa) in the binary pattern extracted from neighbors, and is represented by the symbol ‘U’. For example, the binary numbers 00000000 and 11,111,111 show no mutation and the binary number 11010110 shows 6 mutations. In this way, the modified LBP operator is rotation-insensitive. In Eq. 4, the number of mutants between the bits is defined. In MLBP, patterns with homogeneity less than or equal to U_T are defined as uniform patterns, and patterns with homogeneity greater than U_T are defined as non-uniform patterns. Given Eq. 5, the labels 0 to P are assigned to uniform neighbors and the label P + 1 is assigned to non-uniform neighbor. After applying this operator to texture images and assigning different labels, the occurrence probability of a particular label can be approximated by the ration of the number of labeled points to the total number of points in the entire image. So eventually, a probability feature vector of P + 2 dimensions for each input image is obtained. In Eq. 5, according to references [25], U_T is usually considered to be P/4, and only a small amount of patterns in the texture is labeled as P + 1.

$$U(LBP) = |s(g_{p-1}-g_c)-s(g_0-g_c)| + \sum_{p=0}^{P-1} |s(g_p-g_c)-s(g_{p-1}-g_c)| \tag{4}$$

$$MLBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p-g_c) & \text{if } U(LBP) \leq U_T \\ P + 1 & \text{otherwise} \end{cases} \tag{5}$$

2.3 A review on some previous versions of local binary patterns

Jin et al. suggested the Improve Local Binary Pattern (ILBP) method which uses the local mean of total neighbors as the threshold instead of the central point of the neighborhood [19]. The Median Binary Pattern (MBP) method was presented by Hafiane et al. [16] This method compared with the ILBP has used the median of total neighboring points instead of the mean of the total neighboring points, and also does not consider patterns 00000000 and 11,111,111 as the same pattern [16]. In 2010, Guo et al. presented a completed binary pattern [15]. In this method, there are three operators: (a) Pixels difference sign (CLBP_S) which includes the sign of the difference between the central pixel and its neighbors, (b) The absolute value of pixels difference (CLBP_M) which includes the absolute value of the difference between the central pixel and its neighbors, and (c) the absolute value of the central pixel (CLBP_C). CLBP has achieved a high percentage of classification by combining these three types of information in comparison with the past versions. Tan and Triggs [35] proposed a local ternary pattern (LTP) to increase noise resistance. Unlike the LBP operator which uses two values of 0 and 1 to generate a binary code, in this method, three values (1, -1, and 0) are used to create a ternary code. Since the histogram size for LTP is large, we split it into two distinct LBP parts and extract the histogram from two local patterns then concatenate them together.

Among other noise resistant local binary versions, a fuzzy based method is proposed in [1, 18]. In 2010, Loris Nanni et al. used the idea of modifying LBP to ILBP in [23] to improve the LTP and called it improved local ternary pattern (ILTP). ELTP [41], enhanced the LTP and adopted a self-adaptive strategy for threshold selection using a local mean of grey level instead of the central pixel grey value. This method is not only resistant to changes in the illumination intensity of the grey level, but also is resistant to noise. Shakoor and Tajeripour suggested the method of local entropy pattern (LEP) as a new descriptor for texture images [31]. This method is more resistant to noise compared to the LBP and local variance. The LEP, like the variance, indicates the non-uniformity of the local patterns of each neighborhood. Fekri-Ershad et al., proposed hybrid color local binary patterns (HCLBP) which extract texture and color features jointly [13]. HCLBP consider color information among texture analysis, which makes it useful to describe colored textures. In [13], first of all, modified local binary pattern is applied on color channels separately. Next, extracted binary patterns for each pixel are considered in AND logical format to provide a new binary pattern.

Wang et al., proposed an efficient approach for texture analysis by combining JLDPCP and RCLBP [39]. They extracted jumping local difference count pattern (JLDPCP) consisting of second-order difference count pattern (SDCP) and diagonal difference count pattern (DDCP) to represent the jumping information in a local domain [39]. Then they proposed RCLBP by refining non-uniform pattern in completed local binary pattern [39]. As discussed, MLBP is sensitive to impulse noise. Local ternary pattern (LTP) [35] was proposed by Tan and Triggs to solve this problem. However, both MLBP [25] and LTP [35] treat the corrupted image patterns as they are. In [30], a noise resistant version of LBP (NRLBP) is proposed to preserve the image local structure in presence. Ren et al. [30] proposed a new version of LBP to correct noisy non-uniform patterns back to possible uniform patterns. Deep et al., in [7] proposed a new descriptor named local quantised extrema quinary pattern (LQEQrYP) for medical image indexing and retrieval [7]. They also explored the local texture relationship between pixels in different directions and

extracted the pattern variation in horizontal, vertical, diagonal, and anti-diagonal directions to form the feature vector [7].

Among the enhanced texture analysis descriptors, one can refer to the local quinary patterns (LQP) [22]. The LQP is one of the enhanced types of LBP. In this method, using two fixed thresholds set by the user, a five-digit encryption is created. This descriptor also has some drawbacks that we attempt to overcome in our research. We also attempt to present a method that increases the accuracy of classification and has a higher resistance to noise as compared to other methods.

2.4 Local Quinary patterns

Loris Nanni et al. [16] proposed a quinary code in order to reduce the LBP sensitivity to noise in almost uniform areas of the image. As the primary LBP uses the difference between the intensity of the central point of the neighborhood (g_c) and that of the neighboring points (g_p) for binary coding, LQP uses the difference between the intensity of the central point of the neighborhood and surroundings neighbors, and two fixed thresholds (τ_1 and τ_2) creates a five-digit code. It is shown as follows.

$$s(g_c, g_p, \tau_1, \tau_2) = \begin{cases} 2 & g_p \geq g_c + \tau_2 \\ 1 & g_c + \tau_1 \leq g_p < g_c + \tau_2 \\ 0 & g_c - \tau_1 \leq g_p < g_c + \tau_1 \\ -1 & g_c - \tau_2 \leq g_p < g_c - \tau_1 \\ -2 & \text{Otherwise} \end{cases} \quad (6)$$

The extracted LQP is broken down into four binary patterns using the $b_c(x)$ function and $c \in \{2, 1, 0, -1, -2\}$.

$$b_c(x) = \begin{cases} 1 & x = c \\ 0 & \text{otherwise} \end{cases}, \quad c \in \{2, 1, 0, -1, -2\} \quad (7)$$

Considering Eq. 7, the first, second, third and fourth binary patterns are respectively obtained considering $c = 2$, $c = 1$, $c = -1$ and $c = -2$. Finally, the histograms of these four binary patterns are built. Next, four created histograms are merged to extract features. An example of the division process of the quinary code into four binary patterns is shown in Fig. 4. LQP now is used highly in many image processing applications, for example LQP is used in [29] for Mammograms analysis, in [2] for image-based facial expression recognition, in [38] to extract efficient features for content based retrieval, in [7] for biomedical image indexing, etc.

Experimental results in [22] show that the LQP operator has a high ability to analyze and classify textural in comparison with many other versions of LBP. But, some disadvantages can be mentioned for LQP as follows:

- *Use static thresholds*

One of the important drawbacks of the LQP method is the threshold determination that should be set by the user as input parameters. In addition, the accuracy of the LQP analysis is highly sensitive to the threshold. The threshold for each class of texture can be very different.

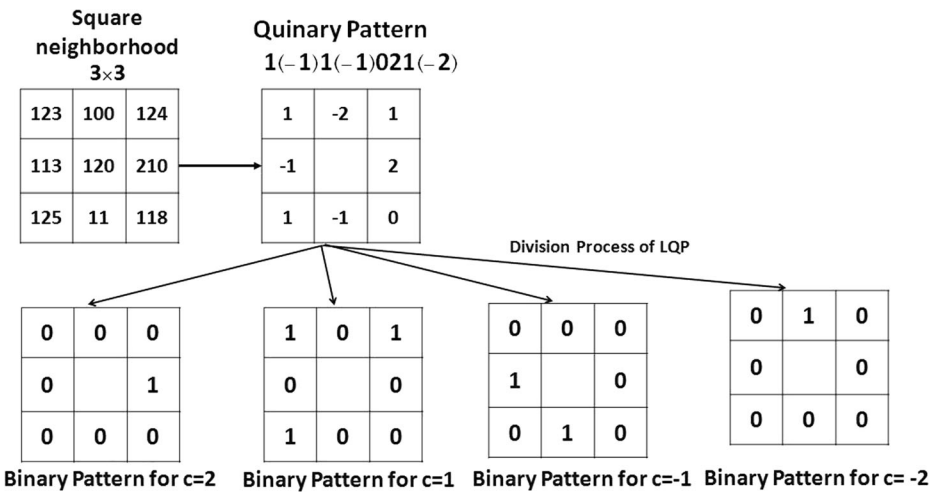


Fig. 4 An example of dividing a quinary pattern code into four binary patterns

- *The lack of a strong theoretical reason for the division process*

Another drawback of this method is the ambiguous definition of the division of the quinary code into four binary patterns, especially in case of the second and third binary patterns (with the observation of $c = 1$ and $c = -1$) because the resulted binary patterns do not represent any particular occurrence in the image. The zero values in resulted binary code do not have the same attribute and meaning. The value 2 indicates greater difference in intensity of the illumination in the image, while -1 and -2 indicate less difference in intensity of the illumination in the image. But these points have a same value zero in extracted binary pattern $c = 1$.

For example, in Fig. 4, with observation of $c = 1$ binary code (10100010), the third bit of this binary code has different meaning in comparison with other zero bits in this pattern. In other words, the third bit shows a neighbor with intensity value very lower than center value. But other zero bits in binary pattern (the first, fourth, fifth, and seventh neighbors) shows some neighbors with same or greater intensity than center pixel. In other words, there is no difference between 2, 0, -1 and -2 when observing value $c = 1$ in the five-digit code, when it divide into a single binary code, while 2 is greater than 1; and 0, -2 , and -1 are less than 1. The generated zeros in the binary code do not express this difference. Similarly, this applies for the third binary code, too. There is no difference between 2, 0, 1 and -2 when observing value $c = -1$ in the five-digit code when it transformed into a binary code, while 0, 1 and 2 are greater than -1 ; and -2 , is less than -1 , and again, the binary code $c = -1$ does not express this difference.

3 The proposed improved local QUINARY patterns

In order to overcome the disadvantages of LQP, we propose a new version of LBP for texture analysis which is called Improved Local Quinary Pattern (ILQP) in this paper. We present a new theory to describe the texture features in a local neighborhood. Low sensitivity to impulse

noise, high classification accuracy, reduction the number of user input parameters, rotation invariant and gray scale invariant are other main goals of our proposed approach.

3.1 Proposed dynamic thresholds for the ILQP

To comate the a quinary pattern for each pixel in the image, first we define a circular neighborhood; the definition of the local quinary pattern in the proposed method is the same as in the LQP method in [22], except that the threshold is dynamically defined. In this paper, we use a new strategy to automatically select thresholds instead of using fixed static thresholds set by the user. In this paper, we use the two criteria of absolute standard deviation and the global significant value (GSV) to select the dynamic threshold for quinary pattern creation process of LQP. As it is described above two thresholds τ_1 and τ_2 is used in LQP process to obtain quinary pattern. To obtain the first threshold (τ_1), local MAD is computed for each local neighborhood, and finally the total MAD should be computed on local MAD. (Eqs. 8 and 9)

$$LocalMAD = median(|G - median(G)|), \quad G = \{g_p | p = 0, 1, \dots, P-1\} \quad (8)$$

$$\tau_2 = MAD = median(|lmad - median(lmad)|) \quad (9)$$

$$lmad = \{LocalMAD_k | k = 1, 2, \dots, M \times N\}$$

Where, P is the number of local neighbors and g_p shows the grey value of the neighborhoods. G represents a set of grey surface values defined in this neighborhood. MAD is the abbreviation for the Median Absolute Deviation. MAD is a measure for the accurate measurement of how a set of data is developed. Also, this benchmark is not noise sensitive; and in fact, the existence of outlier data has negligible effect on the final result.

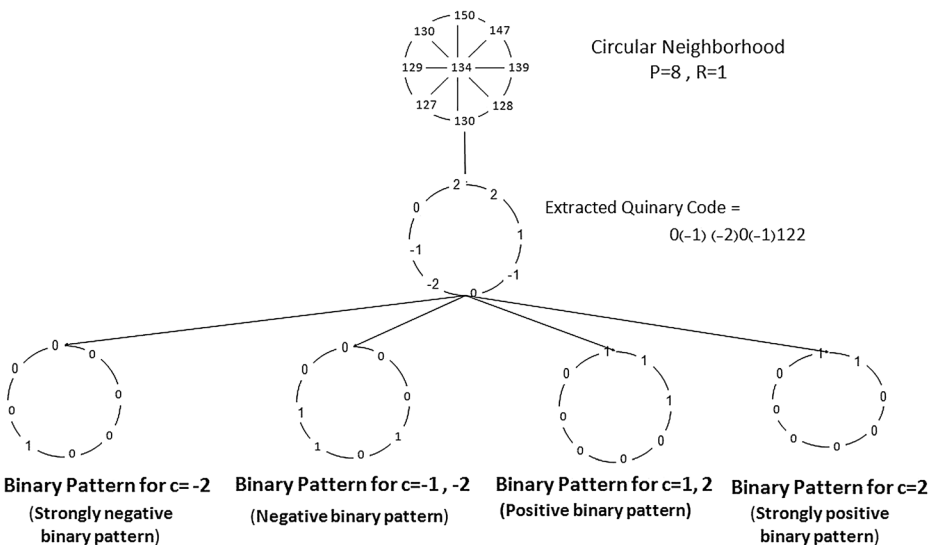


Fig. 5 An example of the proposed division algorithm

To select the second threshold (τ_2), we use the global significant value (GSV). This criterion is first presented in [12] to extract portions of the image texture for which the difference in the intensity of the points is higher than the neighbors. To calculate the GSV criterion, the absolute value mean is first obtained from the difference between intensity of the neighboring center and the intensity of the neighboring pixels. Given the Eq. 10, we calculate LSV for each neighborhood in the whole image. At the end, an average of the values of LSV is obtained, which is called GSV. The calculation of GSV is shown in the following equations below.

$$LSV_c = \frac{1}{P} \sum_{p=0}^{P-1} (|g_c - g_p|) \tag{10}$$

$$GSV = \frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M LSV_{i,j} \tag{11}$$

Where, M and N are the size of input image and $LSV_{i,j}$ shows the local significant value of the neighborhood with center pixel c with coordination (i,j).

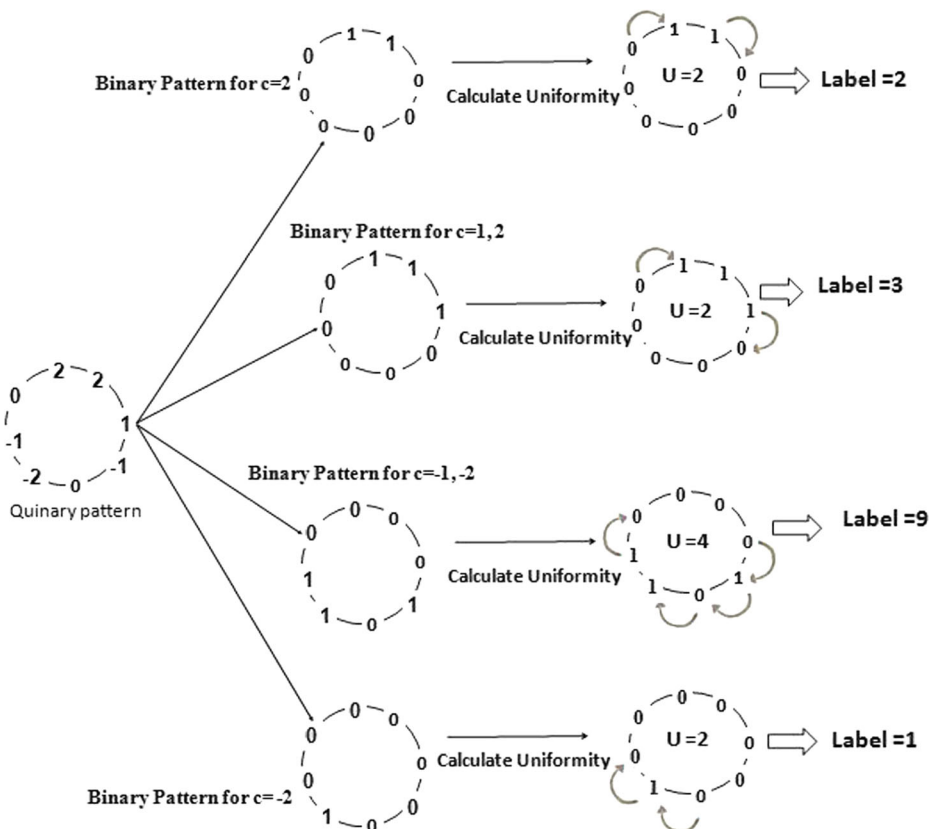


Fig. 6 An example of dividing quinary pattern and label assigning process in proposed ILQP

3.2 Proposed algorithm to divide the Quinary code into binary patterns

In LQP the quinary pattern is divided into four local binary patterns according to Eq.6. In the LQP method, the second and third binary codes are created by observing 1 and -1 . The resulted patterns cannot express meaningful differences. Also, in the second binary patterns, no distinction is made between 2, -1 and -2 . As we know, 2 indicates greater difference between intensity values, but the -1 and -2 indicates lower or extremely lower difference between intensity values. Therefore, the resulted patterns are not acceptable and we change the division way here. In our method, the division process of the quinary pattern into four local binary patterns is as follows.

- To create the first binary pattern (strongly positive pattern): the division takes place observing code 2 in the quinary pattern. In this pattern, the difference between the intensity values of the neighboring pixels and center is greater than τ_2 .

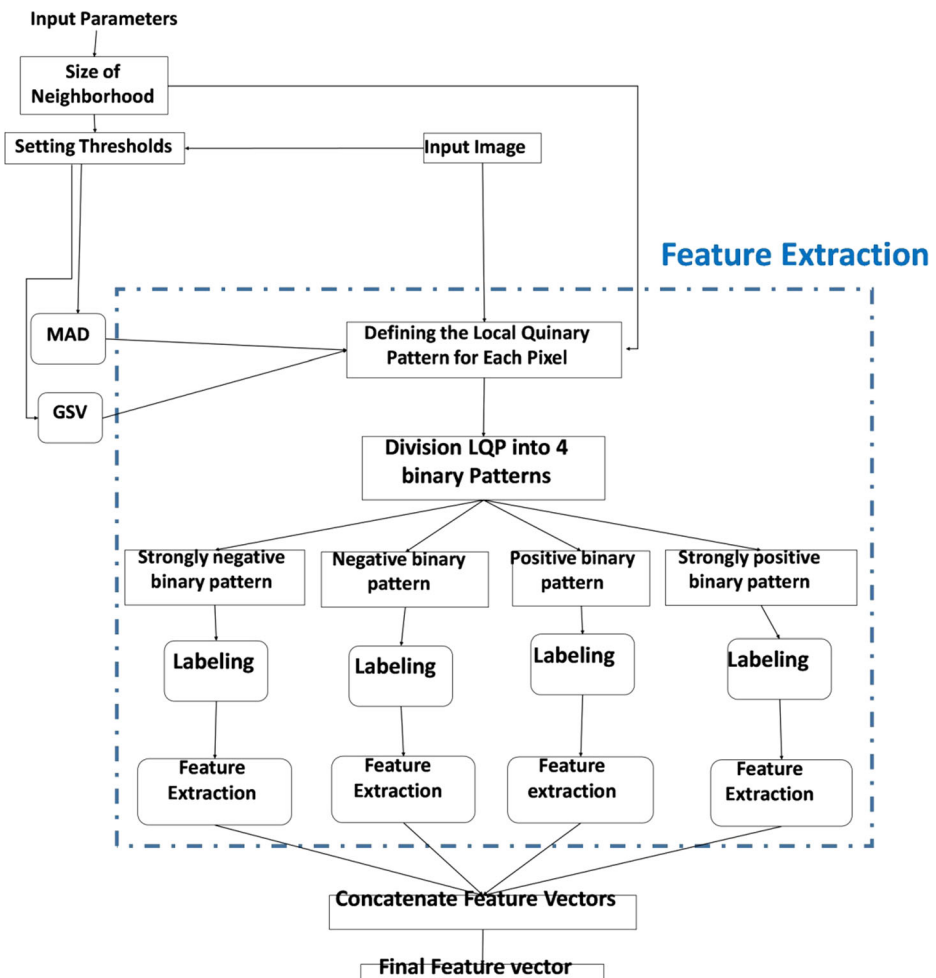


Fig. 7 General block diagram of the proposed texture analysis method ILQP

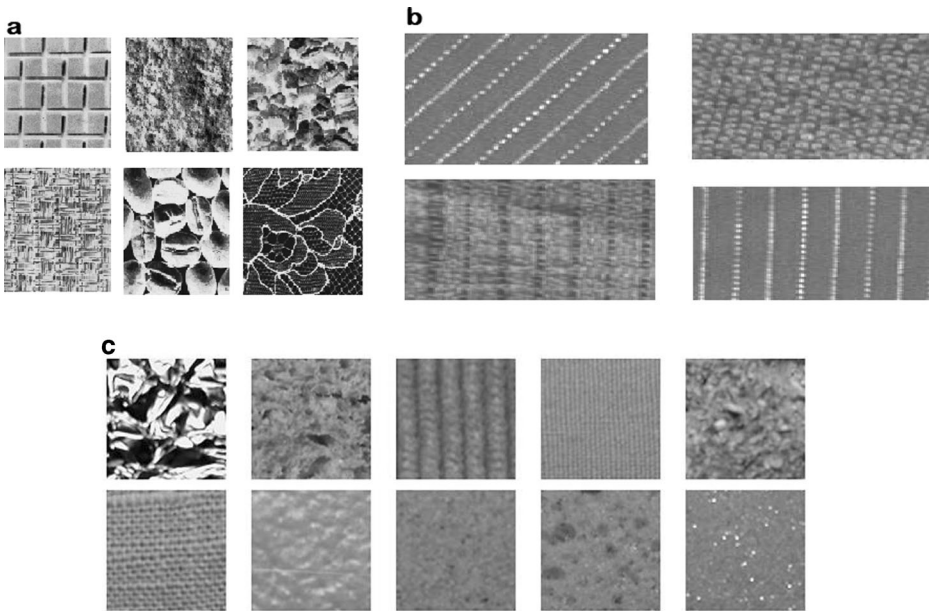


Fig. 8 (a) Some examples of Brodatz database (b) Some examples of Outex database (c) Some example of KTH_TIPS

- To create the second binary code (positive pattern), we consider the values of 2 and 1 in the quinary pattern. If the quinary code observes 1 and 2 for each of these two values, then code 1 is assigned to them, otherwise we will assign a value of 0. This binary pattern means that the difference between the intensity values of the neighbors and center is greater than the threshold τ_1 .
- To create the third binary code (negative pattern) we consider -1 and -2 together; this binary pattern shows that the difference between the intensity value of the neighbors and center is lower than the threshold τ_1 .
- The fourth binary pattern (strongly negative) is based on the observation of code -2 in the LQP; this binary pattern shows a much smaller difference between the intensity of the neighbors and center.

Fig. 5 shows an example of a quinary pattern division process in our proposed method. As you can see in the strongly positive binary pattern (0000011), the first and second neighboring pixels have the greatest difference with the neighboring center; and also for the positive pattern

Table 1 Classification accuracy (%) of the proposed method on the Outex TC_000012 database with 10-fold validation

Classifier Operation	1NN	3 NN	5 NN	Naive Bayes	J48 Tree
ILQP _{8,1}	98.81	98.72	98.41	94.20	92.37
ILQP _{16,2}	99.33	99.29	99.25	96.78	93.20
ILQP _{24,3}	99.50	99.22	99.12	96.08	92.79
ILQP _{8,1} +ILQP _{16,2}	99.54	99.52	99.43	97.20	93.66

Bold values shows the highest provided accuracy

Table 2 Classification accuracy (%) of the proposed method on the Outex TC_000010 database with 10-fold validation

Classifier Operation	1NN	3 NN	5 NN	Naive Bayes	J48 Tree
ILQP _{8,1}	99.00	98.97	98.44	91.55	89.72
ILQP _{16,2}	99.32	99.25	99.16	95.53	93.58
ILQP _{24,3}	99.58	99.74	99.58	98.49	94.60
ILQP _{8,1+} ILQP _{16,2}	99.56	99.49	99.39	97.36	94.05

Bold values shows the highest provided accuracy

(00000111), the illumination intensity of the first, second and third neighboring pixels are greater than τ_1 .

$$ILQP_{ch} = \begin{cases} ILQP_1 = \begin{cases} 1 & \text{if } c = 2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_2 = \begin{cases} 1 & \text{if } c = 1 \text{ or } c = 2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_3 = \begin{cases} 1 & \text{if } c = -1 \text{ or } c = -2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_4 = \begin{cases} 1 & \text{if } c = -2 \\ 0 & \text{otherwise} \end{cases} \end{cases} \quad \text{where } c \in \{2, 1, 0, -1, -2\} \tag{12}$$

3.3 Feature extraction in the proposed approach

According to [25], the uniformity shape is more appropriate than other methods in the LBP family, both in terms of efficiency and performance. In addition, this method is insensitive to rotation. In [34], it is shown that if in the definition of the LBP the uniformity threshold be equal to P/4, only a negligible percentage of the texture patterns will be labeled as P + 1 (non-uniform pattern). Therefore, in this paper, the threshold of uniformity (homogeneity) is considered as P/4. As shown in Eqs. 13 and 14, we use the uniformity method for extracting the features for each of the four binary patterns. At the end, after extracting the features of each binary pattern, they are merged, and as a result, a features vector with $4 \times (P + 2)$ dimensions is provided (Eq. 15).

$$U(ILQP_{ch}) = |(g'_{P-1, ch}) - (g'_{0, ch})| + \sum_{p=0}^{P-1} |(g'_{p, ch}) - (g'_{p+1, ch})| \tag{13}$$

Table 3 Classification accuracy (%) of the proposed method on the Brodatz database with 10-fold validation

Classifier Operation	1NN	3 NN	5 NN	Naive Bayes	J48 Tree
ILQP _{8,1}	94.82	93.39	92.67	91.78	75.29
ILQP _{16,2}	94.88	94.16	91.48	90.65	73.09
ILQP _{24,3}	94.22	93.03	91.19	90.71	72.55
ILQP _{8,1+} ILQP _{16,2}	95.89	94.82	93.80	93.03	77.32

Bold values shows the highest provided accuracy

Table 4 Classification accuracy (%) of the proposed method on the KTH_TIPS database with 10-fold validation

Classifier Operation	1NN	3 NN	5 NN	Naive Bayes	J48 Tree
ILQP _{8,1}	91.2	88	85.65	69.48	76.88
ILQP _{16,2}	93.30	88.98	87.50	66.40	76.64
ILQP _{24,3}	93.80	92.07	88.74	68.61	79.48
ILQP _{8,1+} ILQP _{16,2}	94.04	92.07	89.11	69.97	80.46

Bold values shows the highest provided accuracy

$$ILQP_{P,R,Ch}^{riu2} = \begin{cases} |(g'_{P-1ch})-(g'_{0ch})| + \sum_{p=0}^{P-1} |(g'_{pch})-(g'_{p+1ch})| & \text{if } U(LBP_ILQP_{ch}) \leq P/4 \\ P + 1 & \text{otherwise} \end{cases} \tag{14}$$

Where, P represents the number of neighbors and g'_p is the value of the p^{th} neighbor in the extracted binary pattern g . The ch index corresponds to each of the four patterns (strongly positive, positive, negative, and strongly negative) (Fig. 6).

Fig. 7 shows a block diagram of the general method of the proposed method. As it is shown in the Eq. 14, a label between 0 to $P + 1$ is assigned to each binary pattern. After applying the proposed $ILQP_{ch}$ on entire image, for each pixel a label would be computed. So, a feature vector with notation $F_{ILQP_{ch}}$ can be computed as follows.

$$F_{ILQP_{ch}} = \langle F_0, F_1, \dots, F_{P+1} \rangle \tag{15}$$

Where, F_0 shows the occurrence probability of label 0 in the whole image after applying $ILQP_{P,R,ch}$. As it is described above, each quinary pattern should be divided to four binary patterns with different index number $ch = 1, 2, 3, 4$ (strongly positive, positive, negative, and strongly negative). So, four different binary patterns can be extract in each local neighborhoods. Finally a feature vector can be extracted as follow.

$$Final\ Feature\ Vector_{ILQP} = \langle ILQP_{P,R,1}^{riu2}, ILQP_{P,R,2}^{riu2}, ILQP_{P,R,3}^{riu2}, ILQP_{P,R,4}^{riu2} \rangle \tag{16}$$

Table 5 Classification accuracy (%)of the proposed method on the Outex TC_000012 database with train percentage validation

Train-size Classifier	30%	40%	50%	60%	70%	80%
1NN	97.5	97.91	98.33	98.54	98.61	98.85
3NN	97.35	97.56	98	98.28	98.40	98.75
5NN	96.75	97.46	97.87	98.02	9840	98.75
Naive Bayes	93.39	93.75	94.37	94.49	94.51	94.73
J48 Tree	88.51	89.47	90.08	91.11	91.43	92.70

Table 6 Classification accuracy (%) of the proposed method on the Outex TC_000010 database with train percentage validation

Train-size Classifier	30%	40%	50%	60%	70%	80%
1NN	97.75	98.26	98.24	98.66	98.91	99.30
3NN	96.52	97.87	98.24	98.41	98.37	98.61
5NN	96.52	96.91	98.05	98.37	98.61	98.68
Naive Bayes	89.46	90.58	90.74	91.06	91.12	91.26
J48 Tree	83.30	85.06	87.73	87.88	89.12	90.16

Below the pseudo code of our proposed texture analysis algorithm is shown in a step by step manner.

Algorithm

1. Load the grayscale image
2. Define neighborhood radius (R) and number of neighboring points (p)
3. Create (GSV and MAD) as thresholds
For each pixel **do**
4. Make a quinary pattern
5. Divide the quinary code into four binary patterns (strongly positive, positive, negative, and strongly negative) as in Eq.(12)
6. Tag a label between 0 to P+1 using the uniformity method for each one of the binary patterns using Eq.(14)
end
7. Extract four feature vectors based on the occurrence probability of each label for each four pattern as it is shown in Eq. (15)
8. Construct the final feature vector: concatenate four provided feature vectors (strongly positive, positive, negative, and strongly negative) as it is shown in Eq. (16)

4 Experimental results

In this section, several tests are performed to measure the performance of the proposed approach. All experiments are performed with the MATLAB 2014a and Weka 3.8 under the Windows 10 Enterprise operating system on a Core i3 and 2.20 GHz computer with an 8 GB RAM.

There are some well known texture databases that were used in experiments on texture classification and which are freely available for comparison of texture analysis algorithms, e.g. the Outex dataset, Brodatz, KTH-TIPS, UMD, UIUC dataset etc.

Table 7 Classification accuracy (%) of the proposed method on the Brodatz database with train percentage validation

Train-size Classifier	30%	40%	50%	60%	70%	80%
1NN	89.54	90.07	91.78	93	93.65	93.75
3NN	82.05	85.41	87.97	89.28	90.27	91.36
5NN	75.51	81.84	84.46	86.90	88.09	90.77
Naive Bayes	78.99	83.23	85.47	85.69	85.91	86.90
J48 Tree	54.67	60.71	64.52	69.49	71.42	72.61

Table 8 Classification accuracy (%) of the proposed method on the KTH_TIPS database with train percentage validation

Train-size Classifier	30%	40%	50%	60%	70%	80%
1NN	84.47	87.63	89.83	92.90	92.16	93.82
3NN	77.95	79.62	88.18	88.88	89.71	93.15
5NN	75.48	76.33	79.50	83.64	87.66	89.85
Naive Bayes	68.02	69.75	71.97	70.06	72.83	71.73
J48 Tree	62.25	68.31	73.90	75.30	79.83	72.46

We used Four databases called KTH-TIPS [36], Brodatz [4], Outex TC000010 [26], and Outex TC000012 [26] to measure the accuracy of texture classification. The Brodatz database [4] contains images of natural textures. This database contains 112 classes; each class contains only one sample with the size of 640*640 pixels. To test, it is necessary to divide the images into a number of smaller images. In this paper, we split each of the images into 16 sub-images with the size of 128*128 pixels and there was no overlap. Therefore, for each class, there are 16 samples. The KTH-TIPS [36], contains 10 texture class images, captured at 9 different scales and 9 different illumination conditions that is, 81 Images per class.

The Outex database [26] was developed by a group of researchers at the University of Oulu in 2002. The database consists of two types of textural and natural images. So far, different classifications have been produced, among which OutexTC_000010 and OutexTC_000012 are the most widely used ones. Each of the OutexTC_000010 and OutexTC_000012 contain 24 classes, and each class has 20 textural samples in 9 directions (0, 5, 10, 15, 30, 45, 60, 75, 90 degrees), with Inca illumination intensity for OutexTC_000010 and ‘Horizon’, ‘Inca’, ‘TL84’ illumination intensity for OutexTC_000012. The size of all samples is 128*128 pixels. Since the images in these datasets (OutexTC_000010 and OutexTC_000012) are in different directions, we consider circular neighborhood to eliminate the sensitivity of the method to the rotation (Fig. 8).

The most important criterion for the measurement of the texture classification methods is classification Accuracy. This criterion is the most popular and generally used measure to calculate the performance of texture image classification methods, which shows that to what extent the designed classifier accurately classifies the entire set of test records. Results of implementation of the proposed method with different neighboring radius $R = 1, 2, 3, 1 + 2$ on

Table 9 Comparison results of different texture classification methods on three datasets in terms of accuracy (%)

Dataset Operator	Outex TC_000012	Outex TC_000010	Brodatz	KTH_TIPS
GLCM [17]	93.22	92.41	75.89	72.15
LBP _{8,1} [8]	90.74	97.68	87.05	68.74
MLBP _{8,1} [15]	93.43	94.73	86.90	69.8
CLBP _{8,1} [22]	98.28	97.91	92.89	73.1
LTP _{8,1} [23]	97.60	97.70	91.81	71.3
LQP _{8,1} [16]	97.39	98	92.46	81.07
DMD [21]	97.14	96.25	89.07	78.50
PPU + SF [9]	91.06	90.87	85.47	71.28
1DLBP [33]	97.9	95.46	90.26	79.75
ILQP _{8,1} (Proposed approach)	98.81	99	94.82	91.2

Bold values shows the highest provided accuracy

Table 10 Comparison results (%) of the proposed method and other approaches with 10-fold validation in the presence impulse noise on the Outex TC_000012 database

Noise Type	Without Noise	Speckle	Salt & pepper			
			5%	10%	15%	25%
Operator						
LBP _{8,1} [8]	90.88	28.60	86.35	83.02	79.71	63.77
MLBP _{8,1} [15]	93.04	23.27	90.22	85.58	83.04	67.81
CLBP _{8,1} [22]	98.41	49.18	97.08	92.06	80.54	80.14
LTP _{8,1} [23]	98	32.68	97.04	95.03	91.45	82.20
LQP _{8,1} [16]	97.72	26	96.83	94.84	92.52	83.29
ILQP _{8,1} (Proposed approach)	98.81	65.97	98.70	98.06	97.56	96.48

Bold values shows the highest provided accuracy

the Outex C_000012, Outex TC_000010, Brodatz and KTH_TIPS databases based on 10-fold validation are presented in Tables 1, 2, 3 and 4.

According to the results in the above tables, we will have better results with increasing radius and number of neighbors. As you can see in Tables 1 and 2, the proposed descriptor on the Outex TC_000012 database with an accuracy of 99.54 and with the 1NN provides the best results. The best result with 99.74% accuracy is obtained on Outex TC_00000010 using 3NN with neighboring radius 3. According to the Table 3, the best result was obtained with 95.89% accuracy on the Brodatz database. Table 4 shows that the proposed descriptor as combination of Radius = 1 and 2, on the KTH_TIPS dataset using 1-NN provides highest accuracy.

In another experiment, the Train-Test percentage validation has been used to examine the proposed method in terms of under-fitting and over-fitting problems. The results are presented in Tables 4, 5, 6 and 7. Results indicate that the ILQP is highly resistant to under-fitting and over-fitting problems. In other words, the proposed method, despite the low number of training data, has been able to provide good accuracy, and also, despite the fact that a high percentage of these samples have been trained, the classification accuracy has not dropped. The results of the implementation of the ILQP on the Outex TC_000012, Outex TC_000010, Brodatz and KTH_TIPS datasets for radius 1 and 8 neighbors are considered in Tables 4, 5, 6, 7, 8 and 9.

4.1 Comparisons with state-of-the-are methods

The main objective of this paper is to provide a approach for increasing the accuracy of image texture classification. For this purpose, we compare the proposed texture descriptor with some of

Table 11 Comparison results (%) of the proposed method with other approaches with 10-fold validation in the presence of impulse noise on the Outex TC_000010 database

Noise Type	Without Noise	Speckle	Salt & pepper			
			5%	10%	15%	25%
Operator						
LBP _{8,1} [8]	97.66	26.45	95.25	92.26	89.49	81.22
MLBP _{8,1} [15]	95.85	26.50	91.06	88.81	79.05	61.96
CLBP _{8,1} [22]	98.79	48.88	97.98	94.37	85.32	71.11
LTP _{8,1} [23]	98.72	27.43	97.70	95.64	93.81	83.44
LQP _{8,1} [16]	99	27.33	97.96	96.80	95.13	88
ILQP _{8,1} (Proposed approach)	99.00	40.04	98.93	97.5	96	94.37

Table 12 Comparison results (%) of the proposed method with other approaches with 10-fold validation in the presence of impulse noise on the Brodatz database

Noise Type	Without Noise	Speckle	Salt & pepper			
			5%	10%	15%	25%
Operator						
LBP _{8,1} [8]	89.10	72.85	84.88	84.22	85.17	83.09
MLBP _{8,1} [15]	90.41	74.82	87.32	82.61	80.77	72.38
CLBP _{8,1} [22]	94.05	85	94.04	92.20	90.89	85.89
LTP _{8,1} [23]	94.16	81.96	93.86	91.96	89.88	85.23
LQP _{8,1} [16]	93.84	81.72	93.21	92.91	89.34	85.91
ILQP _{8,1} (Proposed approach)	94.82	88.86	93.92	92.97	91.48	91.01

state-of-the-art methods in this area. In order to be able to fairly comparison, the same validation conditions and datasets are considered. In this experiment, we evaluated various classifiers (K-NN, Naive Bayes, J48 Tree), which the best result are provided by 1NN for all three datasets. The comparison results demonstrate that the proposed approach provides higher classification accuracy than most of the methods in this area.

The results of the implementation of LBP, MLBP, CLBP, LTP, LQP and the proposed ILQP method are calculated only on the radius $R = 1$ and the number of neighbors $P = 8$; other parameters of the methods are studied and the best of them are selected. The GLCM method is also considered with a 1-pixel movement in four directions (0, 45, 90, 135 degree). In [21], a texture representation is proposed based on a combination of patch-based features and feature encoding algorithm. The Dense micro-block difference (DMD) includes four steps as follows: Local structure extracting using dense micro blocks, Dimensionality reduction, encoding local features using GMM, and feature vector normalization. In [9], an innovative approach is proposed for stone texture analysis based on combination of some primitive pattern units and haralick statistical features, which is known as PPU + SF. In this experiment, we evaluated PPU + SF and 1-NN is used as classifier, same with other experiments. As can be seen, the proposed method is superior to many of the other efficient methods in this field.

4.2 Texture classification in the presence of noise

To evaluate the resistance of the proposed method to noise, we first apply two type of impulse noises (salt & pepper and speckle) on texture images and extract the texture features using the proposed method. Then, we recalculate the accuracy factor of the classification and finally compare the results

Table 13 Comparison results (%) of the proposed method and other approaches with 10-fold validation in the presence impulse noise on the KTH_TIPS database

Noise Type	Without Noise	Speckle	Salt & pepper			
			5%	10%	15%	25%
Operator						
LBP _{8,1} [8]	68.74	53.76	68.2	67.32	65.94	62.81
MLBP _{8,1} [15]	69.8	54.63	68.93	68.03	66.27	63.02
CLBP _{8,1} [22]	73.1	56.92	72.74	70.95	68.43	66.04
LTP _{8,1} [23]	71.3	55.82	71.23	69.03	66.97	62.95
LQP _{8,1} [16]	81.7	64.12	81.3	80.96	78.34	76.53
ILQP _{8,1} (Proposed approach)	91.20	74.69	91.20	90.59	88.24	85.40

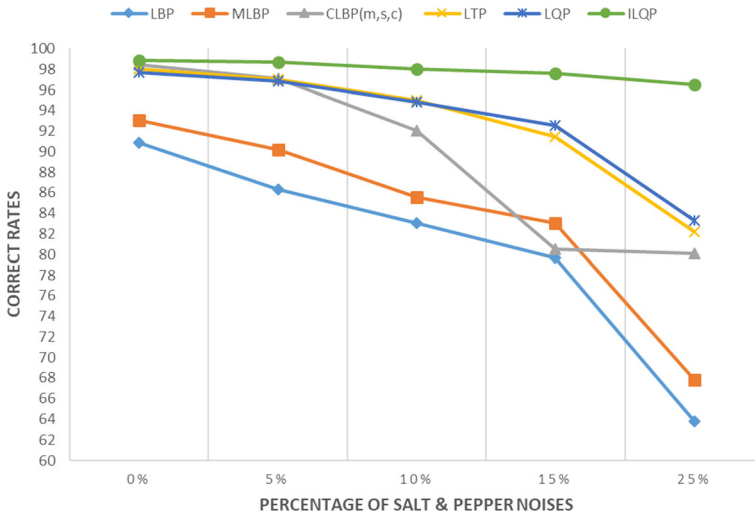


Fig. 9 Precision classification by 1NN on the Outex database TC_000012 by applying salt and pepper noise with different ratios

with the related literature; for the measurement criterion, we use the K-NN and consider the value of K to be 1. In the following tables, the speckle noise level with the variance of 0.04 and salt & pepper noise with a density ratio of 5%, 10%, 15%, and 25% are applied to texture images. Comparison of the proposed method with the other efficient methods in this domain is carried out with 10-fold validation in the presence of a variety of noise on the standard databases Outex TC_000012, Outex TC_000010, Brodatz and KTH_TIPS. All parameters in the concerned methods are the same as previous experiments (Tables 10, 11, 12 and 13).

The following Figs. 9, 10 and 11, show the comparison of the classification accuracy of the proposed method with other efficient methods in this area when the effects of salt & pepper noises are applied with different ratios on the three standard datasets. 1NN classifier has

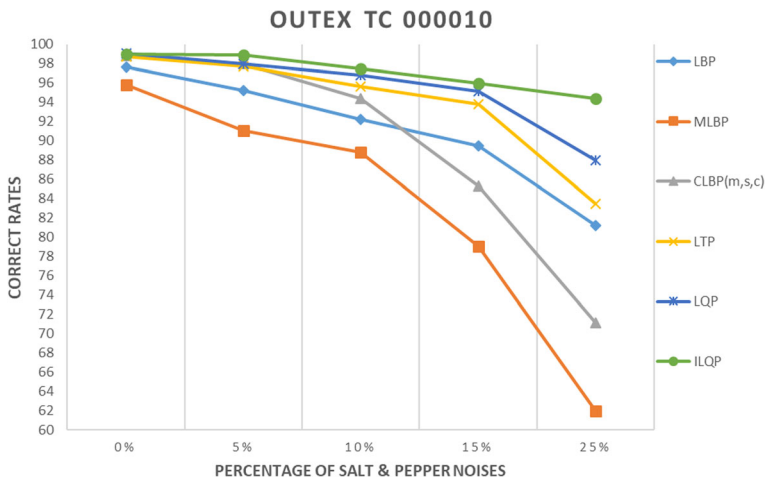


Fig. 10 Precision classification by 1NN on the Outex database TC_000010 by applying salt and pepper noise with different ratios

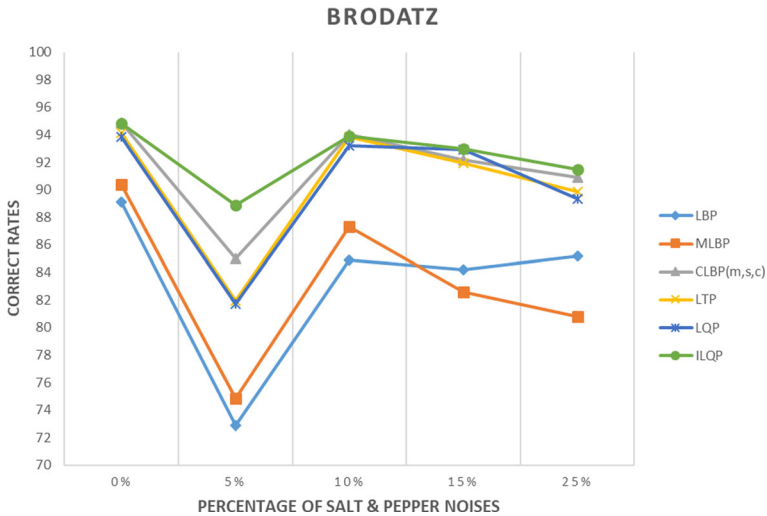


Fig. 11 Precision classification by INN on the Brodatz by applying salt and pepper noise with different ratios

provided the results. As can be seen, the accuracy of the proposed method has not been significantly reduced by increasing the percentage of noise, while the accuracy of other compared methods decreases significantly with increasing the percentage of salt & pepper noises.

5 Conclusion

The main aim of this paper was to propose an approach for texture image classification. In this respect, improved version of local quinary patterns is proposed with notation ILQP. Presenting a new algorithm to divide extracted local quinary pattern to four significant binary patterns is the main step of the proposed approach. The proposed dividing algorithm provides meaningful binary patterns. Also, a method is proposed in ILQP to consider dynamic thresholds in dividing process of quinary patterns. Dynamic thresholds increase generality of the ILQP in different applications. Experimental results demonstrated higher classification accuracy of the proposed approach in comparison with state-of-the-art texture classification methods. Rotation invariant is another advantage of the proposed method because of considering uniform class labeling algorithm in feature extraction step. Also, experimental results show the low noise sensitivity of the proposed approach because of using dynamic thresholds. High usability is another advantage of the proposed ILQP. The proposed texture analysis ILQP is a general descriptor which provides discriminative feature vector. Hence, it can be used in many applications for feature extraction step to analysis texture.

6 Future works

It is worth mentioning that in this paper, the descriptor ILQP proposed only for texture analysis. It is recommended that, ILQP can be used in some other image processing applications which dealing with texture information, such as: image retrieval, texture segmentation,

texture synthesis, inspection systems, etc. It is also suggested to use color information in joint of ILQP texture features.

Other suggestions can be Instead of using the circular neighborhoods; some other topologies can be used for neighborhood definition, such as ellipse, parabola, hyperbola, etc.

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

References

1. Ahonen T, Pietikäinen M (2007) Soft histograms for local binary patterns. In Proc. of Finnish Signal Processing Symposium
2. Al-Sumaidae S, Abdullah M, Al-Nima R, Dlay S, Chambers J (2017) Multi-gradient features and elongated quinary pattern encoding for image-based facial expression recognition. *Pattern Recogn* 71: 249–263
3. Arivazhagan S, Ganesan L, Kumar TS (2006) Texture classification using ridgelet transform. *Pattern Recogn Lett* 27(16):1875–1883
4. Brodatz P (1996) *Textures: A Photographic Album for Artists and Designers*. Dover Publications, Newyork
5. Bu H, Wang J, Huang XB (2009) Fabric defect detection based on multiple fractal features and support vector data description. *Eng Appl Artif Intell* 22(2):224–235
6. Chen J, Jain AK (1988) A structural approach to identify defects in textured images. In Proc. of IEEE International Conference on systems, manufacturing and cybernetics 29–32
7. Deep G, Kaur L, Gupta S (2018) Local quantized Extrema quinary pattern: a new descriptor for biomedical image indexing and retrieval. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 6(6):687–703
8. Eichkitz CG, Davies J, Amtmann J, Schreilechner MG, DeGroot P (2015) Grey level co-occurrence matrix and its application to seismic data. *First Break* 33(3):71–77
9. Fekri-Ershad S (2011) Color texture classification approach based on combination of primitive pattern units and statistical features. *International Journal of Multimedia and its applications* 3(3):1–13
10. Fekri-Ershad S (2012) Texture classification approach based on energy variation. *International Journal of Multimedia Technology* 2(2):52–55
11. Fekri-Ershad Sh (2012) Texture classification approach based on combination of edge & co-occurrence and local binary pattern. In proc. of Int'l Conference on Image Processing, Computer Vision and Pattern Recognition, 626–629
12. Fekri-Ershad S, Tajeripour F (2017) Color texture classification based on proposed impulse-noise resistant color local binary patterns and significant points selection algorithm. *Sens Rev* 37(1):33–42
13. Fekri-Ershad S, Tajeripour F (2017) Impulse-noise resistant color-texture classification approach using hybrid color local binary patterns and kullback-leibler divergence. *Comput J* 60(11):1633–1648
14. Fogel I, Sagi D (1989) Gabor filters as texture discriminator. *Biol Cybern* 61(2):103–113
15. Guo Z, Zhang L, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. *IEEE Trans Image Process* 19(6):1657–1663
16. Hafiane A, Seetharaman G, Zavidovique B (2007) Median binary pattern for textures classification. In proc. of international conference on image analysis and recognition. *Lect Notes Comput Sci* 4633:387–398
17. Haralick RM, Shanmugam K (1973) Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics* 6:610–621
18. Iakovidis DK, Keramidas EG, Maroulis D (2008) Fuzzy local binary patterns for ultrasound texture characterization. In proc. of international conference on image Analsysi and recognition. *Lect Notes Comput Sci* 5112:750–759
19. Jin H, Liu Q, Lu H, Tong X (2004) Face detection using improved LBP under bayesian framework. In Proc. of Third International conference on Image and Graphics 306–309
20. Manjunath BS, Ma WY (1996) Texture features for browsing and retrieval of image data. *IEEE Trans Pattern Anal Mach Intell* 18(8):837–842

21. Mehta R, Egiazarian K (2016) Texture classification using dense micro-block difference. *IEEE Trans Image Process* 25(40):1604–1616
22. Nanni L, Lumini A, Brahnam S (2010) Local binary patterns variants as texture descriptors for medical image analysis. *Artif Intell Med* 49(2):117–125
23. Nanni L, Brahnam S, Lumini A (2010) A local approach based on a local binary patterns variant texture descriptor for classifying pain states. *Expert Syst Appl* 37(12):7888–7894
24. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. *Pattern Recogn* 29(1):51–59
25. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987
26. Ojala T, Maenppa T, Pietikainen M, Viertola J, Kyllonen J, Huovinen S (2002) Outex-new framework for empirical evaluation of texture analysis algorithms. In *Proc. of 16th International Conference on Pattern Recognition* 701–706 Downloadable: <http://www.outex.oulu.fi/index.php?page=classification>
27. Pietikäinen M, Ojala T, Xu Z (2000) Rotation-invariant texture classification using feature distributions. *Pattern Recogn* 33(1):43–52
28. Rajesh R, Veerappan J, Sujitha S, Kumar EA (2012) Classification and retrieval of images using texture features. In *Proc. of Third International Conference on Computing Communication and Networking Technologies* 1–5
29. Rampun A, Morrow P, Scotney B, Winder J (2017) Breast density classification using multiresolution local quinary patterns in mammograms. In *Proc. of Annual Conference on Medical Image Understanding and Analysis* 365–376
30. Ren J, Jiang X, Yuan J (2013) Noise-resistant local binary pattern with an embedded error-correction mechanism. *IEEE Trans Image Process* 22(10):4049–4060
31. Shakoor MH, Tajeripour F (2017) Noise robust and rotation invariant entropy features for texture classification. *Multimed Tools Appl* 76(60):8031–8066
32. Shanmugavadivu P, Sivakumar V (2012) Fractal dimension based texture analysis of digital images. *Procedia Engineering* 38:2981–2986
33. Tajeripour F, Fekri-Ershad S (2014) Developing a novel approach for stone porosity computing using modified local binary patterns and single scale retinex. *Arab J Sci Eng* 39(2):875–889
34. Tajeripour F, Kabir E, Sheikhi A (2008) Fabric defect detection using modified local binary patterns. *EURASIP Journal on Advances in Signal Processing* 08(1):783–789
35. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transaction on Image Processing* 19(6):1635–1650
36. The KTH-TIPS and KTH-TIPS2 image databases (2006) Downloadable: <http://www.nada.kth.se/cvap/databases/kth-tips/>
37. Tuceryan M, Jain AK (1993) *Texture analysis handbook of pattern recognition and computer vision*, World Scientific Publishing Company(2nd Edition):207–248
38. Vipparthi SK, Nagar SK (2014) Color directional local quinary patterns for content based indexing and retrieval. *Human-centric computing and. Inf Sci* 4(1):1–13
39. Wang T, Dong Y, Yang C, Wang L, Liang L, Zheng L, Pu J (2018) Jumping and refined local pattern for texture classification. *IEEE Access* 6:4416–64426
40. Wen W, Xia A (1999) Verifying edges for visual inspection purposes. *Pattern Recogn Lett* 20(3):315–328
41. Yuan J, Zhu H, Gan Y, Shang L (2014) Enhanced local ternary pattern for texture classification. In *proc. of international conference on intelligent computing theory*. *Lect Notes Comput Sci* 8588:443–448



Laleh Armi received her M.Sc. majored in computer engineering-artificial intelligence from najafabad branch, Islamic azad university in 2017. Her research interests are image processing and computer vision applications includes texture analysis, texture classification, image descriptors, etc.



Shervan Fekri-Ershad received his B.Sc. degree in computer hardware engineering from Islamic Azad University of Najaf-Abad, Iran in 2009. He received his M.Sc. & Ph.D. degrees from International Shiraz University, Iran in 2012–2016, majored in Artificial Intelligence. He joined the faculty of computer engineering at najafabad branch, Islamic azad university, najafabad, Iran as a staff member (assistant professor) in 2012. His research interests are image processing applications includes visual inspection systems, visual classification, texture analysis, surface defect detection and etc.