

No-Reference Image Quality Assessment Using Local Binary Patterns: A Comprehensive Performance Evaluation

Zihan Zhou, Yong Xu, Xi Wan, Yuhui Quan, Jing Li, Patrick Le Callet

Abstract—One key in image quality assessment (IQA) is the design of image representations that can capture the changes in image structures caused by distortions. In the last decades, local binary patterns (LBP) have been proven to be a powerful tool as a statistical model for texture and local structure representation. LBP and its variants, as the texture descriptor with low computational complexity, have been applied widely and successfully in several specific applications, such as texture classification, face recognition and IQA. Generally, visible impairments alter the statistics of LBP descriptors, making it possible to measure degradation and then estimate image quality. However, only a few variants of LBP are applied in no-reference (NR) IQA and a few characteristics of LBP descriptors are explored for quality prediction, while some characteristics useful in the rest variants are ignored for IQA. Two previous studies give a review of 16/4 LBP operators and compare the performance with them separately in IQA application only on synthetic distorted data. To extend this work, we provide a review of LBP methodologies to assist the scientific community and new researchers, as well as explore more LBP descriptors in NR-IQA methods under various distortion conditions, particularly real-world cases. Specifically, we comprehensively review 30 widely-used or effective LBP-based operators, including recent variations. We then utilize a common framework for applying LBP descriptors in NR-IQA. The practicality of the reviewed descriptors is demonstrated and analyzed using experimental results under synthetic and authentic cases, indicating suitable LBP descriptors and characteristics for NR-IQA. Codes implementing the reviewed LBP operators will be available soon.

Index Terms—Local Binary Patterns, Variants, Texture Descriptors, No-Reference Image Quality Assessment

I. INTRODUCTION

DIGITAL images are ubiquitous in modern life and work for communication, entertainment and data analysis. However, the quality of images can be compromised by various distortions that occur during acquisition, processing, transmission, and display. These introduced distortions negatively impact image quality, which further limits the performance of applications relying on distorted images, such as object detection [1]–[4], face recognition [5]–[8] and video

Zihan Zhou is with the College of Mathematics and Informatics at South China Agricultural University, China. Yong Xu, Yuhui Quan and Xi Wan are with the School of Computer Science and Engineering at South China University of Technology, China. Yong Xu and Yuhui Quan are also with Pa Zhou Lab, Guangzhou, China. Yong Xu is also with Peng Cheng Lab, Shenzhen, China. Jing Li is with the Moku Lab, Alibaba Group, Beijing, China. Patrick Le Callet is with Nantes Universite, Ecole Centrale Nantes, CAPACITES SAS, CNRS, LS2N, UMR 6004, F-44000 Nantes, France. Email: zhousizhan@scau.edu.cn, yxu@scut.edu.cn, csxwan@mail.scut.edu.cn, csyhquan@scut.edu.cn, lj225205@alibaba-inc.com, patrick.lecallet@univ-nantes.fr.

stream recognition systems. Consequently, there is growing interest in developing objective image quality assessment (IQA) algorithms to automatically evaluate perceptual quality. IQA has numerous potential applications, including instant feedback generation for image collection systems, automatic optimization of camera settings or post-processing parameters, the guidance of designing image restoration models [9], [10].

There are three categories in objective IQA: full-reference (FR) IQA, reduced-reference (RR) IQA and no-reference (NR) IQA. The FR approaches (*e.g.* [11]–[14]) are designed for the scenarios where the reference image of very high quality is given. Such scenarios can be found in image compression, image watermarking, as well as the training stages of learning-based image processing methods. The RR approaches (*e.g.* [15]–[17]) assume that only partial information (in the form of features) of the reference image can be accessed. RR-IQA is particularly suitable for image transmission, where receivers lack the reference images. Instead, reference features are sent and compared to transmitted image features (see [15] for an example). In contrast, NR-IQA (*i.e.*, Blind) is more challenging since blind approaches (*e.g.* [18]–[21]) estimate quality without reference versions. Due to their relaxed requirements, blind IQA methods are more flexible for real-world use than FR and RR, better handling cases like image recovery and identifying low-quality images, especially for wild user-generated content.

The key in NR-IQA is the design of image representations that can capture the changes of image structures caused by distortions—that is, being discriminative to diverse distortions and various image contents. Several non-deep-learning techniques have been applied successfully to the feature extraction of degraded images, such as natural scene statistics [22]–[25], independent component analysis [26], [27], sparse coding [28], [29] and local binary pattern description [30]–[32]. Among these, Local Binary Pattern (LBP) has emerged as a powerful statistical model for texture/structure representation [33]. With only fast binary operations, LBP has flexible real-world NR-IQA applicability on low-power-consumption devices and in real-time scenarios. Furthermore, distortions often damage image textures and statistics, enabling LBP descriptors to quantify quality and proving useful for IQA.

The original local binary pattern (LBP) is a non-parametric descriptor that efficiently summarizes local image structures by comparing each pixel value to its surrounding neighbor pixel values. LBP converts an image into an array or map, where each pixel is labeled with a decimal number representing

the LBP value. LBP values are derived from binary codes of pixel comparisons. The LBP maps are further converted into histograms used for specific applications. The LBP family has been developed until now. Many LBP variants have been developed, overcoming limitations and improving performance for specific applications like statistic/dynamic texture classification [34]–[36], face recognition [37]–[39], object detection [40], [41], image quality assessment [42], [43], etc. Table I summarizes the applications of LBP operators reviewed in this paper. For NR-IQA, although some variants help characterize structural degradation, many operators and characteristics are unexplored, especially for real-world images with a huge diversity of distortion and a large variation of image contents.

TABLE I: The applications of the reviewed LBP operators.

Application	Method
Texture Classification	LBP [33], Rotation Invariant LBP [33], Uniform LBP [33], Averaged LBP [44], Median LBP [45], Complete LBP [46], Dominant LBP [47], Multi-Scale LBP [48], Pyramid LBP [49], Opponent Color LBP [50], Adjacent Evaluation LBP [51], Path Integration-based LBP [52], Decorrelated LBP [53], Adjacent Evaluation LTP [51], Local Phase Quantization [54], Local Binary Count [55], Completed Local Binary Count [55], Completed Local Derivative Patterns [56], Improved Local Quinary Patterns [57]
Face Recognition	LTP [58], Local Gabor Binary Patterns [59], Local Phase Quantization [60], Local Gradient Patterns [61], Eight Local Directional Patterns [62]
Image Quality Assessment	LBP [63], LTP [64], Multi-Scale LBP [65], Wavelet Domain LBP [66], Salient LBP [67], Multi-Scale Salient LBP [68], Local Phase Quantization [69], Local Variance Patterns [42], Orthogonal Color Planes Patterns [70], Opponent Color [71], Complete LBP [69]
Image Retrieval	Multi-Scale LBP [72]
Writer Identification	Wavelet Domain LBP [66]
Fingerprint Liveness Detection	Weber Local Binary Descriptor [73]

Although successful in various computer vision applications, basic LBP operators have some limitations [74]. For example, using the central pixel as a threshold makes them sensitive to noise. To address the limitations of conventional LBP, several variants have been proposed in the literature. Some variants are designed to increase description capability, *e.g.* Complete LBP [46], Averaged LBP [44]. Some aim to increase robustness to noise, *e.g.* Local Ternary Pattern (LTP) [58], Adjacent evaluation LBP [51], Improved Local Quinary Patterns [57]. Some aim to robust to blur, *e.g.*, Local Phase Quantization [54]. Some determine other types of neighboring pixels to utilize more information, *e.g.* Pyramid LBP [49], and Multi-scale LBP [65]. Some calculate LBP codes in other domain with specific transforms such as wavelet transform, Gabor transform, see [59], [66] for example.

Obviously, it would be impossible to consider all (*i.e.*, more than 100) variants of LBP have been proposed in the literature.

Therefore, we selected a representative set of popular, widely-used, and recent LBP variants from typical applications like static/dynamic texture analysis, face recognition, and image quality assessment. Since our focus is on image quality assessment rather than video, we limited the scope to 2D image variants. While not an exhaustive list, the chosen variants provide a broad overview of key LBP developments. Table II briefly introduces the basic LBPs and reviewed variants, including common abbreviations, with details provided in the following sections and supplementary material ordered by the table.

This work expands upon previous studies by Freitas et al. [69], [75], which compared 16 and 4 LBP-based operators separately for IQA using synthetically distorted data. We extend them by going beyond synthetic distortions to focus more on real-world degradation scenarios. Additionally, we review more LBP variants used in broader computer vision tasks. This paper aims to familiarize the scientific community and support new researchers with past and current local binary pattern methodologies used in different fields, as well as explore more LBP descriptors for NR-IQA. In this paper, we first give a review of four basic LBP operators and 26 variants. We then apply these LBP variants to an NR-IQA framework, analyzing their pros and cons. Due to page limitations, we focus on key ideas of LBPs, with references for further details.

To apply LBP and its variants to NR-IQA, we utilize a two-step process: 1) Extracting visual image features using LBPs, then 2) applying learning-based regression models, *i.e.*, support machine regression (SVR) or random forest (RF), to derive quality scores from extracted features. See Figs. 1 for the whole framework which includes training and testing stages of LBPs-based NR-IQA methods.

Our contributions in this work are two-fold:

- This work provides a comprehensive review of 30 LBP descriptors to familiarize the scientific community and support new researchers with past and current local binary pattern methodologies.
- We implement/reuse and apply these LBP variants to an NR-IQA framework. Experiments on synthetic and authentic data reveal suitable LBP descriptors and characteristics for NR-IQA.

The remainder of this paper is organized as follows. Section II as well as supplementary material presents a brief review of basic LBP operators and variants. Section III describes the experimental setup, experimental results on three synthetic and three real-world IQA databases, and a discussion of these results. Section IV concludes the paper.

II. LBP DESCRIPTORS

LBP is a statistical method that summarizes the local structure of the image [76]. It is regarded as an effective local texture descriptor. Texture is a fundamental attribute of images, but there is no consensus on its definition. In this paper, local structures such as textures refer to regional characteristics perceived as combinations of basic image patterns, following the definition in [75]. These basic image patterns exhibit a certain regularity that can be obtained through statistical measurement, which are then utilized for IQA.

TABLE II: Brief descriptions of LBP variants

LBP Variants	Abbreviation	Description	Ref.
Local Binary Patterns	LBP	Encode binary codes via intensity comparison in a local neighborhood	[33]
Rotation Invariant LBP	RILBP	Clockwise shift the LBP code bit-wisely to get the smallest decimal value	[33]
Uniform LBP	ULBP	Count the number of bitwise 0/1 changes of LBP codes	[33]
Rotated Uniform LBP	RIULBP	Combine rotation-invariant LBP and uniform LBP	[33]
Averaged LBP	ALBP	Use the mean of local neighbourhood as the threshold	[44]
Median LBP	MLBP	Use the median value over a local neighborhood as the threshold	[45]
Complete LBP	CLBP	Consider the sign and magnitude of the local differences, and the global intensity	[46]
Dominant LBP	DLBP	Make use of the most frequently occurred patterns of LBP	[47]
Multi-Scale LBP	MSLBP	Encode LBP codes across multiple circular regions at different scales with different sampling points	[65]
Pyramid LBP	PLBP	Cascade the LBP information of the hierarchical spatial pyramid	[49]
Opponent Color LBP	OCLBP	Calculate intra-channel and inter-channel LBP codes for color images	[50]
Wavelet Domain LBP	WDLBP	Calculate LBP maps on wavelet coefficients of different sub-bands	[66]
Local Gabor Binary Patterns	LGBP	Calculate LBP maps on multi-scale and multi-orientation Gabor magnitude maps	[59]
Adjacent Evaluation LBP	AELBP	Set the averages in adjacent evaluation windows as the neighbors of the neighborhood center for binary code encoding	[51]
Path Integration-based LBP	pi-LBP	Filter and encode pixels to binary values at different scales along a particular path	[52]
Decorrelated LBP	dLBP	Operate discrete cosine transform on local differences of different scales	[53]
Salient LBP	SLBP	Weight the LBP map with the saliency map	[67]
Multi-Scale Salient LBP	MS-SLBP	Utilize the saliency map as weights for the LBP maps at multiple scales	[68]
Local Ternary Patterns	LTP	Encode three-digit codes by intensity comparison with a user-specific threshold	[58]
Adjacent Evaluation LTP	AELTP	Consider the means in adjacent evaluation windows as the neighbors of the neighborhood center for three-value code encoding	[51]
Local Phase Quantization	LPQ	Quantize phase of the discrete Fourier transform on images computed in local neighborhoods	[54]
Local Gradient Patterns	LGP	Encode binary codes by comparing absolute values of the intensity difference with their average	[61]
Local Binary Count	LBC	Count the number of value 1's in the binary neighborhood	[55]
Completed Local Binary Count	CLBC	Combine the LBC with the measures of local intensity difference and central gray level	[55]
Eight Local Directional Patterns	ELDP	Encode the local neighborhood filtered by eight Kirsch compass masks	[62]
Local Variance Patterns	LVP	Compute the spread of the texture local energy based on the square of intensities	[42]
Completed Local Derivative Patterns	CLDP	Complement the CLBP operator with the local directional derivative information	[56]
Orthogonal Color Planes Patterns	OCPP	Consider three two-dimensional planes decomposed from the tri-dimensional (XYZ) space for color images	[70]
Improved Local Quinary Patterns	ILQP	Encode five-digit codes by intensity comparison with two definitive thresholds	[57]
Weber Local Binary Descriptor	WLBD	Combine the LBP under Weber's law and compute gradients from center-symmetric pixel pairs	[73]

This section only describes 4 basic LBP descriptors and 3 variants due to page limitations. The remaining LBP variants are demonstrated in the supplementary material. Table III and Table IV show the classification of the reviewed LBP variants in terms of the behavior for improving standard LBP and design objectives, respectively.

Throughout the paper, unless specified, for calculating various LBPs, R is the sampling radius of the neighborhood of the central pixel, P is the total number of neighbors sampled with a distance R . And g_c is gray (*i.e.*, intensity) value of the center pixel and g_p is gray value of the neighbors. $s(x)$ is the

function what equals to 0 when x is less than 0 and equals to 1 otherwise. $U(\cdot)$ is a measure to count the number of spatial transitions (*i.e.*, 0-1, 1-0 jumps) in the binary code.

A. Basic Local Binary Patterns

Generally, the LBP operator marks the pixels of an image with decimal numbers, called Local Binary Patterns, which encode the local structure around each pixel. In this section, 4 basic LBP operators (*i.e.*, original LBP, rotated LBP, uniform LBP and rotated-uniform LBP) are introduced. The original LBP was first proposed by [33]. It records a 0/1 sequence by

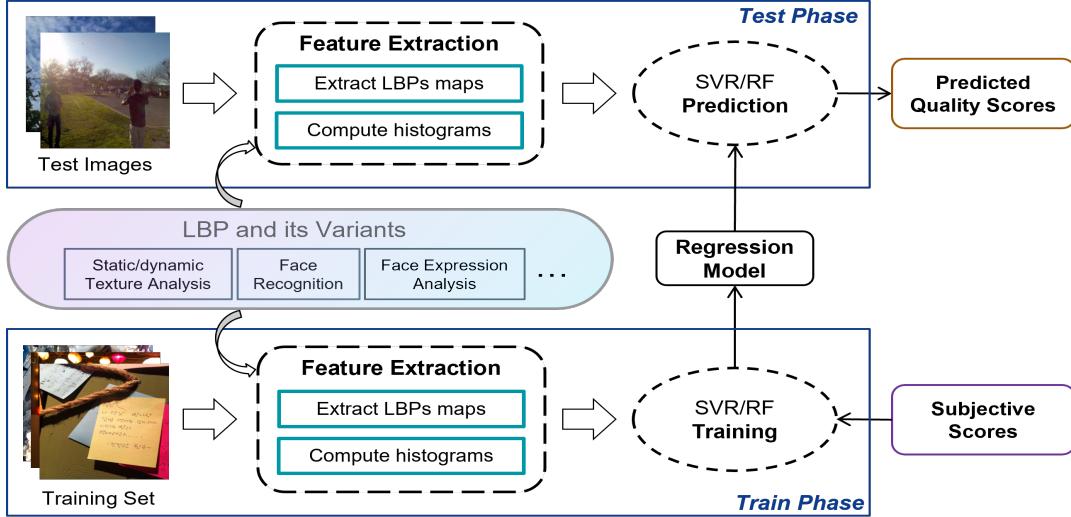


Fig. 1: The framework of NR-IQA used in this paper. Two phases are included: 1) training the quality estimation model and 2) predicting quality scores.

TABLE III: Classification for LBP variants based on behavior

No.	Behavior	Method
1	Refining the encoding strategy	Rotation Invariant LBP, Uniform LBP, Dominant LBP, Local Ternary Patterns, Local Binary Count, Local Variance Patterns, Improved Local Quinary Patterns, Weber Local Binary Descriptor
2	Changing the threshold to be compared	Averaged LBP, Median LBP
3	Refining the neighbors of the neighborhood center	Adjacent Evaluation LBP, Path Integration-based LBP, Adjacent Evaluation LTP, Local Gradient Patterns, Eight Local Directional Patterns
4	Considering more information in the spatial domain	Complete LBP, Multi-Scale LBP, Pyramid LBP, Salient LBP, Multi-Scale Salient LBP, Completed Local Binary Count, Completed Local Derivative Patterns
5	Encoding in the frequency domain	Wavelet Domain LBP, Local Gabor Binary Patterns, Decorrelated LBP, Local Phase Quantization
6	Extension to color images	Opponent Color LBP, Orthogonal Color Planes Patterns

comparing each pixel value with all surrounding neighboring pixel values and calculates a decimal number of binary strings. The traditional LBP operator is defined as follows:

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

where R is the radius of neighborhood of the central pixel, P is the total number of neighbors sampled with a distance R , g_c is gray (*i.e.*, intensity) value of the center pixel and g_p is gray value of neighboring pixel. An example to calculate LBP

TABLE IV: Classification for LBP variants based on objective

Objective	Method
Improving the robustness to noise	Uniform LBP, Averaged LBP, Median LBP, Pyramid LBP, Adjacent Evaluation LBP, Local Ternary Patterns, Adjacent Evaluation LTP, Improved Local Quinary Patterns
Improving the robustness to local intensity variation	Local Gradient Patterns, Eight Local Directional Patterns, Weber Local Binary Descriptor
Improving the sensitivity to distortions	Salient LBP, Multi-Scale Salient LBP, Local Variance Patterns
Increasing the description capability	Complete LBP, Dominant LBP, Multi-Scale LBP, Opponent Color LBP, Wavelet Domain LBP, Local Gabor Binary Patterns, Path Integration-based LBP, Decorrelated LBP, Completed Local Binary Count, Orthogonal Color Planes Patterns

binary code and LBP decimal value for one center pixel value with $P = 8, R = 3$ is shown in Fig. 2. The LBP feature of images is combining all LBP values (*i.e.*, decimal numbers) of all pixels. To reduce feature dimension, histograms can be calculated from LBP feature vectors/maps as feature vectors of images.

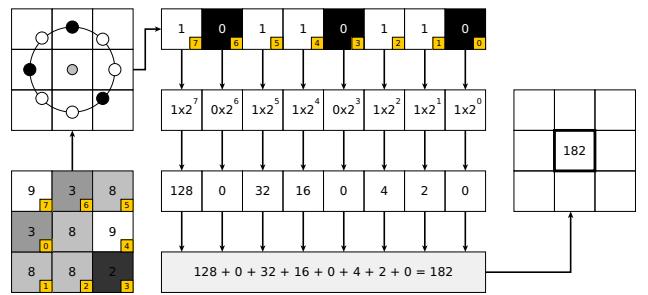


Fig. 2: An illustration of the original LBP [33].

Such LBP is sensitive to image rotation. To address this issue, rotation invariant LBP ("ri" strategy) is made in which the LBP binary code is rotated one turn bit by bit to take the smallest decimal value among all candidate rotated binary codes. It is usually defined as:

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

where $ROR(x, i)$ is the clockwise bit-wise shift operator that shifts the binary code x by i times.

The number of spatial transitions (bitwise 0/1 changes) in the pattern is defined as follows:

$$\begin{aligned} U(LBP_{P,R}) &= |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\ &+ \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \end{aligned} \quad (3)$$

The uniform pattern means that bitwise 0/1 changes (*i.e.*, 0 to 1 and 1 to 0 jumps) are not more than 2. Uniformity is important because it characterizes the patches that contain primitive structural information such as edges and corners. Based on the observation that more than 80 percent of binary patterns are uniform and many of the remaining ones contain essentially noise, Uniform LBP [33], are defined to reduce the number of local binary patterns. The non-uniform patterns are grouped into one mode (*i.e.*, single bin), typically without losing too much information. Combining rotation-invariant LBP and Uniform LBP can retain the minimum binary code with no more than two spatial transitions is called rotation-invariant uniform LBP ("riu2" strategy).

$$LBP_{P,R}^{riu2} = \begin{cases} LBP_{P,R}, & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1, & \text{else.} \end{cases} \quad (4)$$

B. Averaged Local Binary Patterns

One obvious drawback of the basic LBP operator is the sensitivity to noise and non-monotonic intensity changes because the neighborhood pixels are compared with a single pixel (*i.e.*, the center pixel). To overcome this, Averaged LBP (ALBP) [44] was proposed to turn the threshold value in LBP into the average pixel value of all pixels in a neighborhood including the center. Since the average is computed over all pixels in the neighborhood, changing the central pixel has less impact on the threshold value and in turn the resulting binary codes. After encoding, a unique scalar value is assigned to each pattern, and the feature descriptor is obtained by calculating the 2^P bins histogram of ALBP codes, which measures the spatial distribution of the patterns over the image.

C. Median Local Binary Patterns

To be more robust to noise, Median LBP (MLBP) [45] improves the LBP by changing the threshold value for texture classification. MLBP seeks to derive the localized binary pattern by thresholding the pixels against their median value over a local neighborhood. A typical snapshot is shown in Fig. ???. Note that the central pixel is included in this filtering process in MLBP therefore more possible patterns can be obtained in terms of the same sampling number P to LBP. The MLBP operator is invariant to monotonic gray-scale changes since the threshold does not depend on the amount of intensity.

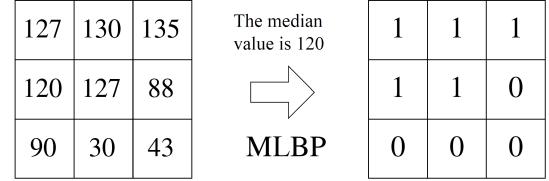


Fig. 3: An illustration of MLBP [45].

D. Complete Local Binary Patterns

To preserve more information than LBP, Complete LBP (CLBP) [46] considers both the sign s_p and magnitude m_p of the local differences and the original intensity value of the central pixel. The CLBP operator consists of three LBP type features (CLBP_S, CLBP_M and CLBP_C) and the framework of computing CLBP descriptor is shown in Fig. 4.

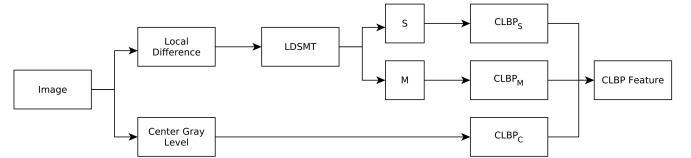


Fig. 4: Framework of CLBP descriptor [46].

Express the sign and magnitude of each neighbor of the center point as:

$$s_p = s(g_p - g_c), \quad m_p = |g_p - g_c|, \quad (5)$$

where g_c is gray (*i.e.*, intensity) value of the center pixel and g_p is gray value of neighboring pixel. CLBP_S is actually the traditional LBP using s_p . CLBP_M compares the absolute difference between a center pixel and its neighbors with a threshold c to be determined adaptively. It is defined as:

$$CLBP_M_{P,R} = \sum_{p=0}^{P-1} s'(m_p, c) 2^p, \quad s'(x, c) = \begin{cases} 1, & x \geq c \\ 0, & x < c. \end{cases} \quad (6)$$

CLBP_C is to compare the center pixels (*i.e.*, image gray g_c) with the average gray value c_I of the whole image with the definition below.

$$CLBP_C_{P,R} = s'(g_c, c_I). \quad (7)$$

It provides global image information which is complementary to local texture features.

Features from CLBP_S, CLBP_M and CLBP_C operators can be combined jointly or hybrid to obtain 2D/3D joint histograms. For joint combination, similar to the 2D joint histogram, a 3D joint histogram of them is built, denoted by "CLBP_S/M/C". For hybrid combination, a 2D joint histogram such as "CLBP_S/C" or "CLBP_M/C" is built first, and then the histogram is converted to a 1D histogram followed by being concatenated with CLBP_M or CLBP_S to generate a joint histogram, denoted by "CLBP_M_S/C" or "CLBP_S_M/C". Note that the uniform and rotation invariance version of CLBP (*i.e.*, applying "riu2" strategy) can be developed easily for applications, see *e.g.* [46].

E. Dominant Local Binary Patterns

The uniform LBPs effectively capture the fundamental information of textures, which mostly consist of straight edges or low curvature edges. Since the $P+2$ uniform patterns extracted from texture images are not necessarily to be the patterns having dominating proportions, performing texture classification based on uniform LBPs on textures having more complicated shapes such as high curvature edges, crossing boundaries is possibly problematic. To capture sufficient and descriptive textural information as well as avoiding to consider all the possible patterns, the Dominant LBP (DLBP) method [47] makes use of the most frequently occurred rotation invariant patterns which are less sensitive to histogram equalization and noise. DLBP is reasonable since the frequency of occurrence of different patterns varies widely, and some patterns rarely appear in texture images.

Given a training image set, DLBP first determines the number (*i.e.*, $K_{80\%}$) of dominant patterns required to occupy 80% of all rotation invariant LBP pattern occurrences. The number of patterns for 80% pattern occurrences of each training image is found via sorting the occupancy in descending order then calculating the occupancy ratio iteratively, then these numbers are averaged to obtain $K_{80\%}$. Then, the DLBP feature vectors are extracted from the input image in three steps: 1) Construct a histogram of patterns considering all patterns; 2) Sort the histogram boxes in non-increasing order; 3) Take the top $K_{80\%}$ histogram boxes as the final feature vectors.

F. Multi-Scale Local Binary Patterns

Multi-Scale LBP (MSLBP) [48], [65], [72] improves the discrimination of LBP without much increment of feature dimension via properly enlarging the size of the feature pool. It encodes LBP codes across multiple circular regions at different scales with different sampling points. Using circular neighborhoods and interpolating the pixel values allow any radius and number of pixels in the neighborhood. MSLBP concatenates LBP histograms with different radii R and different numbers of neighbors P to obtain final feature vectors. Fig.5 gives an example of the encoding process.

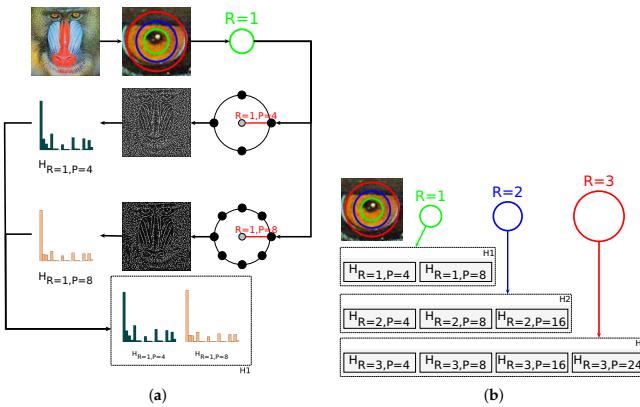


Fig. 5: Feature extraction steps of MSLBP [65]. (a) Multipoint LBP sampling. (b) Multiple histogram generation from LBP.

G. Pyramid Local Binary Patterns

Pyramid LBP (PLBP) [49] extends the conventional LBP to the pyramidal transform domain. Different from MSLBP descriptors obtained from the same image with fixed resolution, the PLBP descriptor takes into account the variation of texture resolution by cascading the LBP information of the hierarchical spatial pyramid. Fig.6 shows a diagram for pyramid transform spatial pyramid sampling.

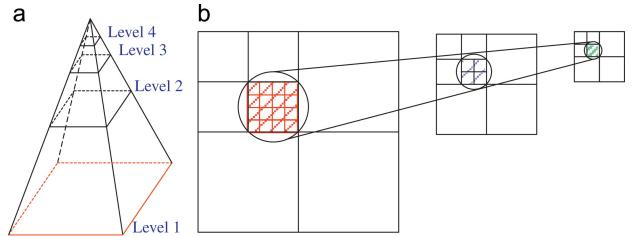


Fig. 6: Diagram of pyramid transform and spatial pyramid sampling in PLBP [49].

For pyramid generation, each pixel in the low spatial pyramid is obtained by down-sampling from its adjacent low-pass filtered high-resolution image. As a result, a pixel in the low-resolution images corresponds to a region in its high-resolutions. Images in the pyramid can be constructed recursively as follows: Given $G_1(x, y) = f(x, y)$ for pyramid level $l = 1$, then for the image of adjacent resolution can be determined as $G_l(x, y) = \sum_m \sum_n \kappa(m, n) G_{l-1}(R_x x + m, R_y y + n)$ for pyramid level $l > 1$, where R_x and R_y are the down-sampling ratios in x and y directions, respectively. And $\kappa(x, y)$ is a low-pass filter. It can be Gaussian filters with various standard deviations and low-pass filters of wavelet bases. The final PLBP descriptor combines LBP histograms calculated on N spatial pyramid LBP maps.

H. Opposite Color Local Binary Patterns

Although the LBP descriptor is effective for describing grayscale textures, it is not sensitive to certain types of degradations, such as contrast distortion or chromatic aberrations. To combine texture and color information into a joint descriptor, the Opponent Color LBP (OCLBP) operator was proposed [50]. There are two approaches to calculating OCLBP. First, the LBP operator can be applied to each color channel individually. This method is called "intra-channel" because the central pixel and the corresponding sampled neighboring points belong to the same color channel. For the second "inter-channel" approach, the central pixel belongs to one color channel and its corresponding sampled neighbors belong to another color channel. For a three-channel color space, *e.g.*, RGB, there are six possible channel combinations: $OCLBP_{RG}$, $OCLBP_{GB}$, $OCLBP_{BG}$, $OCLBP_{GB}$, of OCLBP and $OCLBP_{BR}$. For an $OCLBP_{MN}$ operator, the central pixel is positioned in the channel M , while the neighborhood is sampled in the channel N . Fig. 7 depicts the sampling approach of OCL when the central pixel is sampled in R channel. Based on the observation that the symmetric

opposing pairs are very redundant (*e.g.*, $OCLBP_{RG}$ is equivalent to $OCLBP_{GR}$), only three more descriptive inter-channel pairs are used, *i.e.*, $OCLBP_{RG}$, $OCLBP_{GB}$, $OCLBP_{BG}$. Finally, the three resulting intra-channel and three resulting inter-channel LBP codes can be concatenated separately and then used as feature vectors.

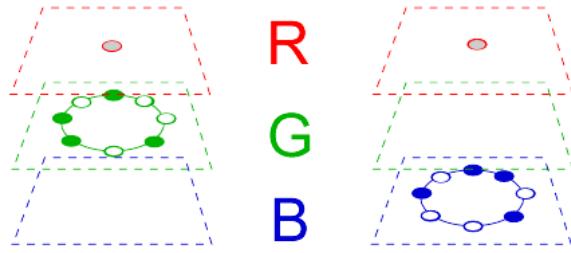


Fig. 7: Sampling scheme of OCLBP for the $OCLBP_{RG}$ and $OCLBP_{RB}$ descriptor.

I. Wavelet Domain Local Binary Patterns

Considering that local structure information in wavelet domain can be more discriminative than spatial domain [77], Wavelet Domain LBP (WDLBP) [66], [77] was proposed. The feature extraction of WDLBP consists of two steps: 1) Wavelet decomposition of the image; 2) Calculation of local binary pattern in wavelet domain. In [66], WDLBP performs a two-dimensional Haar wavelet transform (WT) on the input image, and thus the image is decomposed into four subbands (*i.e.*, subimages) for low-frequency and high-frequency information. Then four wavelet decomposed subimages composed by wavelet coefficients are described separately by LBP histograms with $P = 4$, $R = 1$ and then are concatenated jointly. Fig. 8 illustrates the general frame of feature extraction in WDLBP.

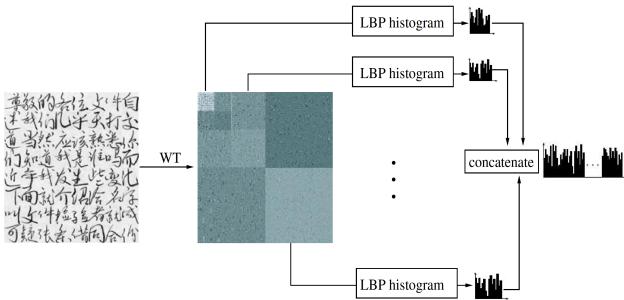


Fig. 8: The diagram of feature extraction in WDLBP [77].

J. Local Gabor Binary Patterns

Local Gabor Binary Patterns (LGBP) [59] was proposed with more discriminating power than the standard LBP, which uses multi-scale and multi-orientation Gabor filters to decompose images and the LBP operator. It consists of two steps: 1) Normalize and transform the input image by Gabor filters with five scales and eight orientations to obtain multiple

Gabor Magnitude Pictures (GMPs) in the frequency domain; 2) Convert each GMP to LGBP map by LBP operator. Each LGBP map can be further divided into non-overlapping rectangle regions, and the histogram is computed for each region. Finally, the final histogram sequence as the feature of the input image can be formed by concatenating the LGBP histograms of all the LGBP maps, see [59] for more details.

K. Adjacent Evaluation Local Binary Patterns

LBP is sensitive to noise because the values of the neighbors are easily changed by random noise, or the change of the center pixel value (*i.e.*, the threshold) alters the binary codes. To address this problem, Adjacent Evaluation LBP (AELBP) [51] is proposed via constructing the adjacent evaluation window which is around the neighbor. The main difference between LBP and AELBP is that the neighbors of the neighborhood center g_c are set as the evaluation center a_p . That is, AELBP is defined as follows:

$$AELBP_{P,R} = \sum_{p=0}^{P-1} s(a_p - g_c) 2^p, \text{ where } s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (8)$$

Excluding the pixel value of the evaluation center, the value of a_p is obtained by calculating the average of the remaining values in the p^{th} evaluation window. Fig. 9 gives an example of this process. The local binary codes are obtained by comparing the value of a_p with the value of the neighborhood center g_c .

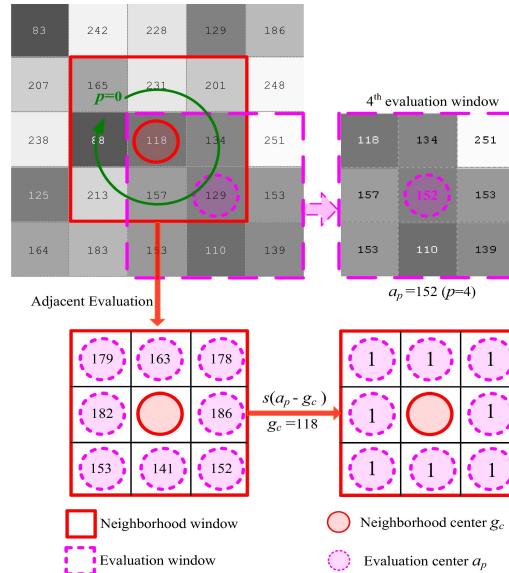


Fig. 9: An illustration of AELBP [51].

L. Path Integral Local Binary Patterns

Multi-scale LBP has been explored successfully by combining local image structures of different scales. However, cross-scale correlation is underutilized in MSLBP since it encodes local patterns in each scale individually. Path integration-based LBP (pi-LBP) [52] is designed to compensate for this

drawback, in which pixels at different scales along a particular path are filtered and then encoded. By taking different paths and filters, pi-LBP can effectively encode the cross-scale correlation and provide a better local-image-structures descriptor for texture analysis.

The pi-LBP is defined by

$$pi - LBP_{P,G_0,f} = \sum_{p=0}^{P-1} s \left(\sum_{i=1}^k f(i) g_{p,i} \right) 2^p, \quad (9)$$

where G_0 is a path of the image starting from the pixel g_c . Specifically, G_0 is a vector whose elements $\{g_{0,1}, \dots, g_{0,k}\}$ are image pixels with $g_{0,1} = g_c$. The new path $G_p = (g_{p,1}, \dots, g_{p,k})$ is obtained by rotating G_0 by $(2\pi)/P$ degree in an anti-clockwise direction from the origin g_c . $f = (f(1), \dots, f(k))$ is a filter satisfying $\sum_{i=1}^k f(i) = 0$. $s(x)$ is a sign function and k is the number of points sampled on each path. Thus, the weighted sum $\sum_{i=1}^k f(i) g_{p,i}$ is an integral along a path. For example, by taking $G_0 = (c, u, v)$ and $f = (-2, 1, 1)$, we have $\sum_{i=1}^k f(i) g_{0,i} = u + v - 2c$.

Briefly, the pixel values along the path are weighted and summed by filters of the same length as parameters for generating the pi-LBP code. When $G_0 = (c, x_0)$ and $f = (-1, 1)$, pi-LBP is actually the basic LBP, so pi-LBP can be viewed as a generalization of the basic LBP. For a better illustration, an example of pi-LBP is shown in Fig. 10. In the figure, $P = 8$, G_0 is a path containing three pixels, and the filter $f = (1, 1, -2)$. Finally, the "riu2" strategy is adopted in pi-LBP for feature dimension reduction, and the pi-LBP features can be derived as

$$pi - LBP_{P,G_0,f}^{riu2} = \begin{cases} s \left(\sum_{i=1}^k f(i) g_{p,i} \right) 2^p, & \text{if } U(pi - LBP_{P,G_0,f}^r) \leq 2, \\ P + 1, & \text{otherwise.} \end{cases} \quad (10)$$

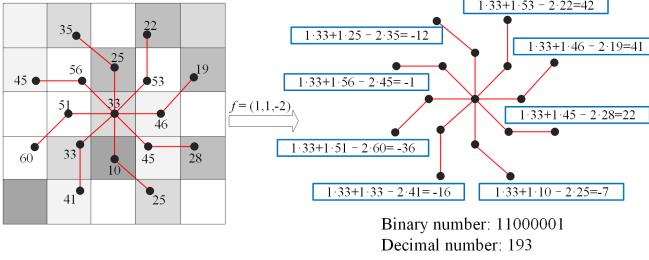


Fig. 10: An illustration of pi-LBP [52].

M. Decorrelated Local Binary Patterns

In MSLBP, concatenating LBP histograms of different scales assumes that LBP features on different scales are independent. However, different scales LBP are in fact correlated in MSLBP based on statistics [53]. To deal with the independence assumption and to better exploit multi-scale information, Decorrelated LBP (dLBP) [53] is proposed via operating discrete cosine transform (DCT) on local differences of different scales. dLBP consists of three descriptors, denoted

as dLBP_S, dLBP_M, and dLBP_C. dLBP_S considers the transformed local differences obtained by operating DCT on signs of differences $g_p - g_c$ in different local neighborhoods. For dLBP_M, the transformed local differences magnitudes obtained by operating DCT on magnitudes of differences $|g_p - g_c|$ are considered. The encoding procedures of dLBP_S and dLBP_M in one scale are same to rotation-invariant uniform version of CLBP_S and CLBP_M, see Subsection II-D for more information. LBP features on different scales are concatenated followed by binarizing the transformed differences with an image-content adaptive threshold to derive local patterns of dLBP_S and dLBP_M operators. dLBP_C is same with CLBP_C to capture global information. The global feature dLBP_C can be combined with the local features dLBP_S and dLBP_M to get final feature of the whole image.

N. Salient Local Binary Patterns

Salient LBP (SLBP) [67] is an extension designed for better IQA by combining the saliency map and the LBP map of "riu2" strategy. The SM produced by visual saliency models provides a measure of the perceptual importance of a region, which allows the IQA metric to weight distortion degree according to the content. Given the input image I , the $LBP_{P,R}^{riu2}$ map with decimal numbers can be obtained by the uniform and rotation invariant LBP, and the saliency map W is calculated by the boolean map saliency model [67]. $W(i,j)$ corresponds to the probability that the pixel $I(i,j)$ attracts the attention of a human observer. Then the SLBP map $SLBP_{P,R}^{riu2}$ is a weighted LBP map, i.e., each pixel $SLBP_{P,R}^{riu2}(i,j)$ equals to $W(i,j) \cdot LBP_{P,R}^{riu2}(i,j)$.

O. Multi-Scale Salient Local Binary Patterns

Multiscale Salient LBP (MS-SLBP) [68] can be regarded as the extension of SLBP, it utilizes saliency map as weights for the LBP maps at multiple scales. MS-SLBP descriptor connects SLBP histograms with different radii R and different number of neighbors P to obtain feature vectors. Fig.11 gives an example of this process.

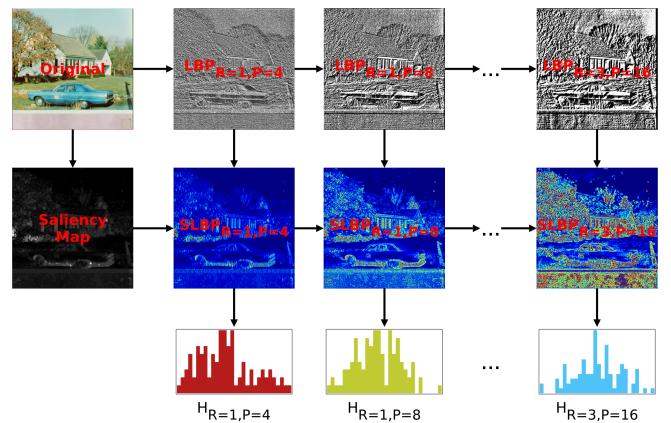


Fig. 11: Multiple histogram generation of MS-SLBP [68].

P. Local Ternary Patterns

LBP tends to be sensitive to noise because the threshold is at exactly the value of the central pixel, particularly in near-uniform image regions. Local Ternary Patterns (LTP) [58] is a version with 3-value codes to quantizes difference between a current pixel and its neighbors into three levels with a user-specific threshold. This make LTP is more discriminant and less sensitive to noise in uniform regions over LBP. For LTP, the step function is changed as:

$$s'(g_p, g_c, \tau) = \begin{cases} 1, & g_p \geq g_c + \tau, \\ 0, & |g_p - g_c| < \tau, \\ -1, & g_p \leq g_c - \tau, \end{cases} \quad (11)$$

in which pixel values in a zone of width $\pm\tau$ around g_c are quantized to zero, ones above this are quantized to $+1$ and ones below it to -1 . Here τ is a user-specified threshold making LTP codes can be more resistant to noise, but more sensitive to monotonic gray-level transformations. Then each resulting ternary LTP code is split into two LBP codes (*i.e.*, upper pattern and lower pattern) corresponding to its positive and negative halves. Specifically, first, the upper mode is created by converting the negative value -1 to zero. Next, the lower mode is created by setting the positive value to zero and converting the negative value -1 to a positive value $+1$. Thus the LTP descriptor generates two separate channels of LBP descriptors, *i.e.*, two texture information maps. Fig. 12 illustrates the basic feature extraction procedure of the LTP descriptor with $\tau = 5$ for a single pixel. Two separate histograms can be computed and concatenated to generate the feature vector of an input image.

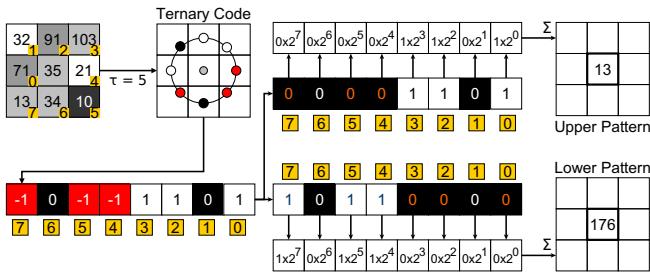


Fig. 12: Illustration of the LTP descriptor [75].

Q. Adjacent Evaluation Local Ternary Patterns

The adjacent evaluation strategy aims to be robust to noise by constructing the adjacent evaluation window which is around the neighbor. Adjacent Evaluation LTP (AELTP) [51] is proposed to apply this strategy on LTP. AELTP and step function are defined as follows:

$$AELTP_{P,R} = \sum_{p=0}^{P-1} s'(a_p, g_c, t) 2^p, \quad (12)$$

$$s'(a_p, g_c, t) = \begin{cases} 1, & a_p \geq g_c + t \\ 0, & |a_p - g_c| < t \\ -1, & a_p \leq g_c - t, \end{cases}$$

where t is a user-specified threshold, a_p is the adjacent evaluation center calculated by the average of the remaining

values in the p^{th} evaluation window. As is operated in the LTP operator, a coding scheme then is used to split each ternary pattern into two parts of LBP, and then two histograms corresponding to two LBP features are built and concatenated into one feature vector.

R. Local Phase Quantization

To tackle the limitation that the traditional LBP is sensitive to blur, Local phase quantization (LPQ) [54], [60] was proposed. The LPQ operator is based on quantized phase of the discrete Fourier transform (DFT) computed in local neighborhoods.

First, the conditions under which the DFT phase is invariant to blur is introduced. Given $G(u, v)$ and $F(u, v)$ are the DFT coefficients of the blurred image $g(x, y)$ and the clean image $f(x, y)$ individually, which has relationship below:

$$G(u, v) = F(u, v) \cdot H(u, v), \quad (13)$$

where $H(u, v)$ is DFT coefficients of the point spread function (PSF) of the blur $h(x, y)$ that describes the response of an imaging system to a point source, \cdot means point-wise multiplication. The phase information can be given by:

$$\angle G(u, v) = \angle F(u, v) \cdot \angle H(u, v), \quad (14)$$

Assume that $h(x, y)$ is centrally symmetric, then its DFT is always real-valued, and thus its phase is only a two-valued function:

$$\angle H(u, v) = \begin{cases} 0, & H(u, v) \geq 0 \\ \pi, & \text{otherwise.} \end{cases} \quad (15)$$

This means that phase invariance is achieved between $\angle G(u, v)$ and $\angle F(u, v)$ for all $\angle H(u, v) \geq 0$.

For the LPQ descriptor, the phase is examined in a rectangular M -by- M neighborhood N at each pixel position (l_1, l_2) of the blurred image $g(x, y)$. The local spectrum is computed with the following equation:

$$G(u, v, l_1, l_2) = \sum_{x, y \in N} g(l_1 - x, l_2 - y) \cdot e^{-j2\pi(ux+vy)/M}, \quad (16)$$

which is a short-term Fourier transform. In LPQ, only four frequencies are used to describe the phase information for each pixel position, *i.e.*:

$$G(\omega) = [G(\omega_1), G(\omega_2), G(\omega_3), G(\omega_4)], \quad (17)$$

where $\omega_1 = [a, 0]^T$, $\omega_2 = [0, a]^T$, $\omega_3 = [a, a]^T$, and $\omega_4 = [a, -a]^T$. In these cases, a is sufficiently small to satisfy $H(\omega_i) > 0$. Experimentally, a is set to $1/M$. The results are decomposed as:

$$\mathbf{W} = [\text{Re}(G(\omega)), \text{Im}(G(\omega))]^\top, \quad (18)$$

where $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ indicate the real and imaginary parts of the complex-valued number. To maximally preserve information in scalar quantization, \mathbf{W} is decorrelated by the singular value decomposition to obtain \mathbf{W}' of the same size. The phase of the Fourier coefficients is the textural information described

LQP and is given by encoding/qantized the elements in \mathbf{W}' as:

$$LPQ_{P,R} = \sum_{j=0}^{P-1} s(\mathbf{W}'_j)2^j. \quad (19)$$

Fig.13 gives a simplified procedure for computing a LPQ pattern. The histogram of the resulting LPQ codes at all pixel locations is created and used as the feature of the input image.

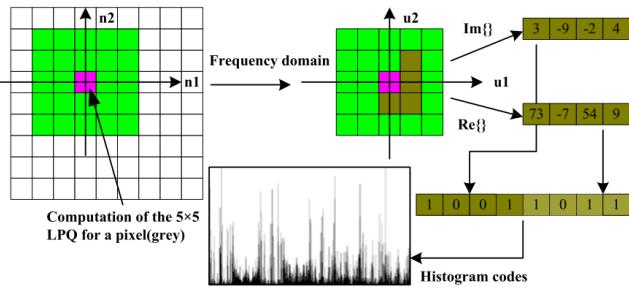


Fig. 13: An illustration for computing LPQ [78].

S. Local Gradient Patterns

Although LBP representations are invariant to monotonic global intensity variations, they are sensitive to local intensity variations along the edge components that occur commonly such as eyes and mouths. Local Gradient Patterns (LGP) [61], having a higher discriminant power than LBP, is designed to overcome this problem. It is assigned the value 1 if the neighboring gradient of a given pixel is greater than the average of that of P neighboring gradients, and 0 otherwise. The gradient values of P neighbors of a center pixel are computed as the absolute value of intensity difference between the center and its neighboring pixels. The average of the gradient values of the P neighboring pixels is used as the threshold value for LGP encoding. Given the intensity (*i.e.*, gray-value) of center pixel g_c and its neighboring pixel value g_p , ($p = 1, 2, \dots, P$), LGP operator is defined as follows:

$$LGP_{P,R}(x, y) = \sum_{p=0}^{P-1} s(i_p - i_\mu)2^p, \quad (2)$$

where $i_p = |g_p - g_c|$ is the defined gradient, $i_\mu = \frac{1}{P} \sum_{p=0}^{P-1} i_p$. Fig.14 shows an example for computing LGP descriptor.

5	9	1	1	5	3	0	1	1
4	4	6	0	-	2	0	-	0
7	2	3	3	2	1	1	0	0

$g_\mu = 2.125$

Fig. 14: An illustration of LGP [61].

T. Completed Local Binary Count

Local Binary Count (LBC) and Completed LBC (CLBC) operators were proposed in [55]. The LBC operator is more rotation invariant than $LBP_{P,R}^{ri}$ for texture classification since it discards the local structural information from the LBP

operator and just extracts the local binary grayscale difference information.

After assigning each pixel in the local neighborhood a binary value by comparing it with the central pixel, the LBC operator only count the number of value 1's in the binary neighborhood instead of encoding them. Fig.15 gives an example of the working principle of LBC with $R = 1, P = 8$. And the computing process for LBC is defined as follows:

$$LBC_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c), \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \quad (20)$$

The LBC focuses on the local binary grayscale difference information since only that how many pixels have comparatively higher intensity than the central one in local region is involved in the LBC. Although the LBC values do not represent visual microstructure, the statistics of LBC features is used to represent the local texture effectively.

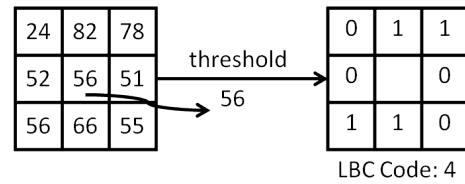


Fig. 15: An illustration of LBC [55].

Motivated by CLBP, CLBC is proposed to obtain more discriminative representation power by combining the LBC with the measures of local intensity (*i.e.*, gray level) difference and central gray level. Similar to CLBP, CLBC contains three operators: CLBC-Sign (CLBC_S), CLBC-Magnitude (CLBC_M), and CLBC-Center (CLBC_C). CLBC_S is just the original LBC described above. CLBC_M counts how many neighbors have comparatively much higher difference of intensity between them and the central pixel than the threshold. It is defined as:

$$CLBC_M_{P,R} = \sum_{p=0}^{P-1} s(m_p - c), \quad m_p = |g_p - g_c|, \quad (21)$$

where c denotes the mean value of m_p in the whole image. CLBC_C is defined as the same as CLBP_C. Please refer to Subsection II-D for more information. Finally, three operators can be combined jointly to get the feature of the input image.

U. Eight Local Directional Patterns

Eight Local Directional Patterns (ELDP) [62], operating in the gradient domain by utilizing Kirsch compass masks, is designed for an illumination insensitive and noise robust representation. ELDP can be viewed as a special filter-based texture operator. ELDP code scheme uses Kirsch compass masks in eight directions to compute the edge responses of a pixel's 3×3 neighborhood. The masks are shown in Fig. 16. For each pixel in the input image, there are eight directional numbers used to assign an associated 8-bit binary code. Different from the conventional LBP, ELDP uses the

$$\begin{array}{cccc}
 M_0 & M_1 & M_2 & M_3 \\
 \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 M_4 & M_5 & M_6 & M_7 \\
 \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
 \end{array}$$

Fig. 16: Kirsch masks in eight directions [79].

positive or negative edge response information of the eight directional edge images to generate the 8-bit code.

In detail, given the input image I , the smoothed image I' by computing the convolution of the original Image I and the Gaussian filter. Then the eight edge response images are calculated by convolving the smoothed image I' with the Kirsch compass masks

$$\mathbb{I}_i = I' * M_i; i = 0, 1, \dots, 7, \quad (22)$$

where M_i is the i^{th} Kirsch compass mask, $*$ means convolution. Then for each pixel at location (x, y) in \mathbb{I} , an 8-bit binary code B is assigned by

$$B_i = \begin{cases} 1, & \mathbb{I}_i(x, y) > 0 \\ 0, & \text{otherwise,} \end{cases} \quad (23)$$

where $i = 1, 2, \dots, 8$. Then the binary code B is converted to decimal value in the feature map. Finally, the feature descriptor is obtained by calculating the histogram of the feature map.

V. Local Variance Patterns

Local Variance Patterns (LVP) [42] is an extension of the LBP operator, which is designed for IQA task specifically. It computes the spread of the texture's local energy using the formula below:

$$LVP_{R,P}^{riu2}(g_c) = \left\lfloor \frac{P \cdot V_{R,P}(g_c) - [LBP_{R,P}(g_c)]^2}{P^2} \right\rfloor, \quad (24)$$

where

$$V_{R,P}(g_c) = \sum_{p=0}^{P-1} [s(g_p - g_c) \cdot 2^p]^2. \quad (25)$$

Given $R = 1, P = 8$, Fig. 17 shows a comparison of steps of texture information extraction using the LBP and LVP descriptors. The numbers in the yellow blocks represent the order of neighbors to be computed. The final LVP feature of the input image is used for histogram statistics, and the histogram bins are averaged into 10 bins in order to reduce the dimensionality of its feature vector in [42]. For IQA, the effect that specific impairments have on the texture can be estimated by the LVP operator. For example, a Gaussian blurring impairment decreases the local texture energy, while a noise impairment increases it.

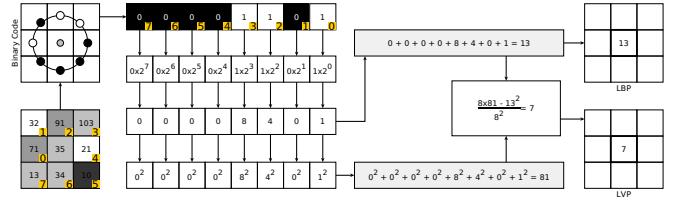


Fig. 17: Pattern extraction process for a given pixel using LBP and LVP operators [42].

W. Completed Local Derivative Patterns

Completed Local Derivative Patterns (CLDP) [56] is an extension of CLBP to represent local texture features. The CLBP operator only involves local differences, CLDP operator proposes a complementary component to local patterns in CLBP by encoding the directional variation of the local differences of two scales. Four types of patterns are utilized in CLDP to represent the local sign, local magnitude, and local directional derivative information in differences and the global intensity.

CLDP_S, CLDP_M and CLDP_C are defined identically to CLBP_S, CLBP_M and CLBP_S respectively, which are demonstrated detailedly in Subsection II-D. Then the "riu2" strategy are applied to CLDP_S, CLDP_M to get uniform and rotation invariant operators CLDP_S^{riu2} and CLDP_M^{riu2}.

The new component Local Directional Derivative Pattern (CLDP_D) is proposed in CLDP to characterize the directional smoothness of local textures. CLDP_D first defines two adjacent neighboring circles with radius R and $R - 1$. Then sign components from two neighboring circles are compared and encoded. CLDP_D is defined as follows:

$$CLDP_D_{P,R} = \sum_{p=0}^{P-1} (s(g_{p,R} - g_c) \oplus s(g_{p,R-1} - g_c)) \cdot 2^p, \quad (3)$$

where $g_{p,R}$ and $g_{p,R-1}$ are intensities of two neighbors on two adjacent circles with radius R and $R - 1$ in the same direction, in terms of the same central pixel value g_c . \oplus is a bitwise exclusive OR (XOR) operation between the sign components of two adjacent circles in the same direction. $s(g_{p,R} - g_c) \oplus s(g_{p,R-1} - g_c) = 1$ indicates that two local differences in one direction have different sign components, and it is likely that the intensity values of two adjacent pixels vary significantly. Conversely, if the value is 0, then two local differences have the same sign component, representing a certain smoothness. The "riu2" patterns CLDP_D^{riu2} of CLDP_D is used as final operator. The simplified diagram of the CLDP operator is shown in Fig. 18. Four patterns are combined using histogram-based manners to generate feature vectors of input images.

X. Orthogonal Color Planes Patterns

Orthogonal Color Planes Patterns (OCPP) [70] is an extension of the LBP operator that incorporates color information. For IQA, encoding both spatial and color information simultaneously makes it more sensitive to different types of degradation (e.g., color and contrast degradation). The basic

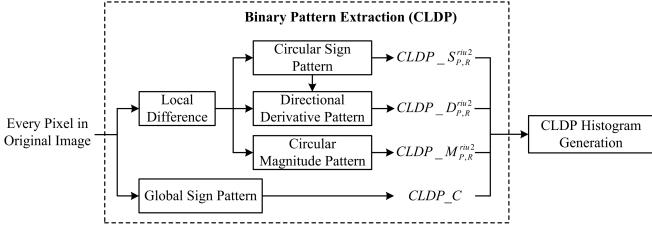


Fig. 18: The diagram of CLDP [56].

LBP descriptor operates on a bi-dimensional XY space for gray-scale images, while OCPP descriptor considers three two-dimensional planes decomposed from the tri-dimensional (XYZ) space for color images. For the color image, it can be decomposed into a set of individual XY planes stacked along the Z axis, a set of YZ planes stacked along the X axis, or a set of XZ planes stacked along the Y axis. Then for each pixel of three-dimensional coordinates as a center, three LBP decimal values corresponding to the XY, XZ, and YZ planes are calculated by the basic LBP operator independently and connected to build OCPP texture descriptor. The OCPP descriptor of the input image is built by concatenating LBP descriptors of all pixels as centers. Since the spatial dimensions of the XY, XZ, and YZ planes are generally different, the radius R_X , R_Y , and R_Z and the number of sampled points P_{XY} , P_{XZ} , and P_{YZ} corresponding to each of the LBP maps can vary. See Fig. 19 for an example where $R_Z = 1$ and $R_X = R_Y = R$, (b)-(d) illustrate how each of the XY, XZ, and YZ planes are sampled.

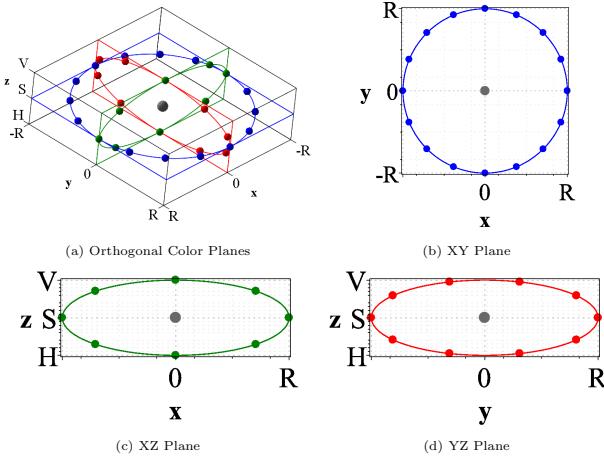


Fig. 19: General view (a) and three planes for sampling (b)-(d) of OCPP [70].

Y. Improved Local Quinary Patterns

Local Quinary Patterns (LQP) [80] was proposed to reduce LBP sensitivity to noise in near-homogeneous regions of the image. The LQP operator considers another encoding way for the evaluation of the local gray-scale difference. It creates a five-digit code based on the intensity differences between the center of the neighborhood and the surrounding neighbors,

and two fixed and user-specific thresholds (τ_1 and τ_2). The encoding procedure is shown as follows.

$$\tilde{s}(g_c, g_p, \tau_1, \tau_2) = \begin{cases} 2, & g_p \geq g_c + \tau_2 \\ 1, & g_c + \tau_1 \leq g_p < g_c + \tau_2 \\ 0, & g_c - \tau_1 \leq g_p < g_c + \tau_1 \\ -1, & g_c - \tau_2 \leq g_p < g_c - \tau_1 \\ -2, & \text{otherwise.} \end{cases} \quad (26)$$

Motivated by the LTP which splits the three-valued code into two binary patterns according to its positive and negative components, the quinary pattern is split into four binary patterns considering the following binary function $b_c(x)$, $c \in \{-2, -1, 1, 2\}$:

$$b_c(x) = \begin{cases} 1, & x = c \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

Take the first mode as an example, it is generated by converting 2 to 1 and the rest to zero, and so on considering $c = 1$, $c = -1$ and $c = -2$ to obtain the rest three binary modes, respectively. An example of the dividing process is shown in Fig. 20. Finally, histograms calculated from these four binary modes are concatenated to form the feature vector.

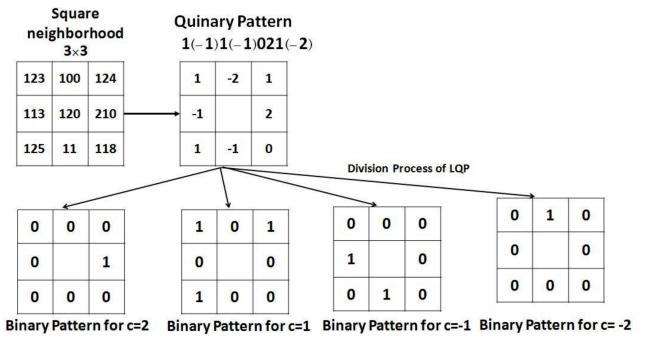


Fig. 20: An illustration of the quinary pattern division process of LQP [57].

Since the accuracy of the LQP analysis is highly sensitive to the threshold and the definition of the division of the quinary code into four binary patterns is ambiguous to represent any particular occurrence in the image, Improved Local Quinary Patterns (ILQP) [57] was proposed to overcome these drawbacks. The ILQP operator is improved by defining a new function to divide local quinary patterns into four binary definitive patterns. Besides, in ILQP, two thresholds are selected automatically using the two criteria of median absolute deviation (MAD) and the global significant value (GSV) instead of being user-defined. The first threshold τ_1 (i.e., the total MAD) in ILQP is calculated over the local MAD for all local neighborhoods. The local MAD is defined as follows:

$$\text{LocalMAD}(g_c) = \text{median}(|G - \text{median}(G)|), \quad (28)$$

where $G = \{g_p | p = 0, 1, \dots, P - 1\}$,

where G a set of grey values defined in the local neighborhood with the center value g_c . Then τ_1 is calculated as follows:

$$\tau_1 = \text{MAD}(I) = \text{median}(|l - \text{median}(l)|), \quad (29)$$

where $l = \{\text{LocalMAD}(g_k) | k = 1, 2, \dots, M \times N\}$,

where M and N are the size of input image I . MAD is a measure for the accurate measurement of how a set of data is developed in terms of median absolute deviation. In this case, the existence of outliers has negligible effect on τ_1 .

To calculate the second threshold τ_2 , the GSV criterion is adopted. For GSV, the local absolute value mean, called the local significant value (LSV), is first obtained from the intensity difference between the center and neighboring pixels in a local region. The LSV for each neighborhood are calculated in the whole image as follows:

$$LSV_c = \frac{1}{P} \sum_{p=0}^{P-1} (|g_c - g_p|), \quad (30)$$

where LSV_c shows LSV of the neighborhood with center pixel of gray value g_c at c^{th} location. Then τ_2 is the average of LSV values:

$$\tau_2 = \frac{1}{M \times N} \sum_{i=1}^{M \times N} LSV_c. \quad (31)$$

Same as LQP, ILQP get five-digit codes via Equ. (26). Then ILQP redefines the process of dividing the quintic mode into four local binary modes: the first binary mode (strong positive mode) is created by converting 2 to 1 and the rest to zero, in which the difference between the intensity values of neighboring pixels and the center is greater than τ_2 . The second binary code (positive mode) is created by converting 2 and 1 to 1 and the rest to zero. A third binary code (negative mode) is created by converting -1 and -2 to 1 and the rest to zero. Finally, a fourth binary mode (strong negative mode) is created by converting -2 to 1 and the rest to zero. Fig.21 gives an example of this process. Given $c \in \{2, 1, 0, -1, -2\}$, four meaningful local binary modes are generated as follows:

$$ILQP_{ch} = \begin{cases} ILQP_1 = \begin{cases} 1 & \text{if } c = 2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_2 = \begin{cases} 1 & \text{if } c = 1 \text{ or } c = 2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_3 = \begin{cases} 1 & \text{if } c = -1 \text{ or } c = -2 \\ 0 & \text{otherwise} \end{cases} \\ ILQP_4 = \begin{cases} 1 & \text{if } c = -2 \\ 0 & \text{otherwise.} \end{cases} \end{cases} \quad (32)$$

Finally, feature extraction is based on uniformity labeling of all binary patterns in $ILQP_{ch}$ to increase discriminatively of the extracted feature vector, *i.e.*, applying the "riu2" strategy on $ILQP_{ch}$ then concatenating four resulting feature vectors.

Z. Weber Local Binary Descriptor

Weber Local Binary Descriptor (WLBD) [73] is a LBP-based operator with good discriminate power. It has powerful texture representation ability and is expected to be robust to noise and illustration changes when the difference and division operations are adopted. The WLBD operator consists of two components: LBDE, the local binary differential excitation component that extracts intensity-variance features; and LBGO, the local binary gradient orientation component that

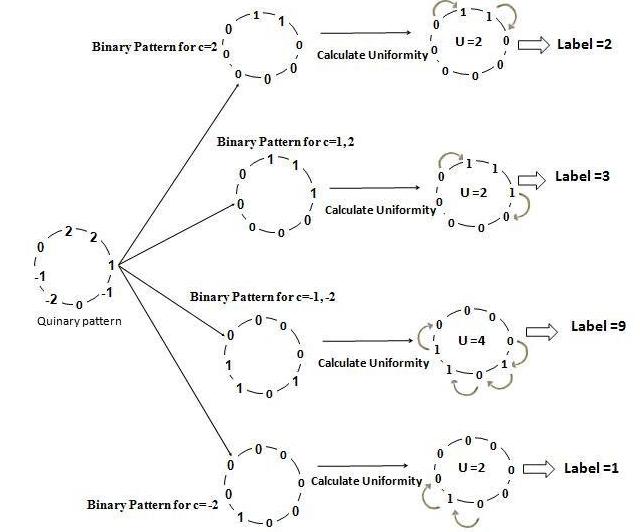


Fig. 21: An example of the division algorithm of ILQP [57].

extracts orientation features. The differential excitation and gradient orientation are both calculated using the computing model in LBP.

The conventional LBP descriptor just compares the gray level of center pixel and neighbors, without considering the background intensity that is important the weber law. LBDE considers both the intensity increase and the background and it is constructed by combining the conventional LBP with Weber's law. The LBDE value is defined as:

$$LBDE_{P,R,K}(g_c) = \sum_{p=0}^{P-1} \hat{s}\left(\frac{|g_p - g_c|}{g_c}\right) \times 2^p, \quad (33)$$

where $\hat{s}(x) = 0$ if $x < K$; else, $\hat{s}(x) = 1$ and K is an empirical threshold. The uniform version of LBDE is calculated as:

$$LBDE_{P,K}^{u2} = \begin{cases} LBDE_{P,R,K}, & \text{if } U(LBDE_{P,R,K}) \leq 2 \\ (P-1)P+2, & \text{else.} \end{cases} \quad (34)$$

LBDE component analyzes the amplitude change among the neighborhood pixels, while LBGO considers the orientation which is important for gradient information. The LBGO is designed to extract gradients from center-symmetric pixel pairs. In other words, LBGO compares the intensity of surrounding pixels on the diagonal in a local area. If the neighbor is larger than its diagonal pixel value, the neighbor is marked as 1, otherwise it is 0. The LBGO is defined as:

$$LBGO_{P,R}(g_c) = \sum_{p=0}^{P/2-1} s(g_p - g_{i+(P/2)}) \times 2^p, \quad (35)$$

where $g_{i+(P/2)}$ and g_i are the gray values of center-symmetric pair of pixels.

For each pixel as a center in a local region, the pair of patterns from LBDE and LBGO operators are calculated, the calculation process is illustrated in Fig. 22. All pairs of patterns for the input image then are combined jointly to form a 2-D histogram, which is defined as the final WLBD.

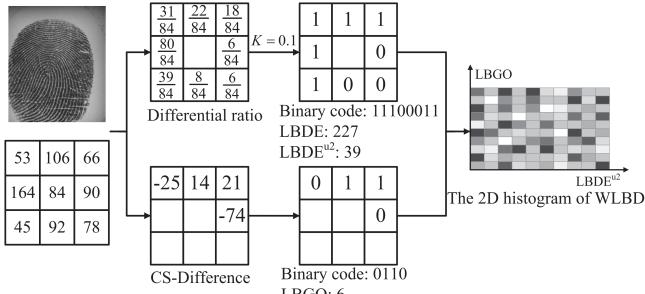


Fig. 22: An illustration of WLBD [73].

III. EXPERIMENT

In this section, we investigate how each reviewed LBP descriptor affects no-reference image quality evaluation. We present performance comparisons to determine which descriptors are better suited for NR-IQA under synthetic and real-world cases. Additionally, we analyze the relationship between descriptor type and IQA method accuracy.

A. Experimental Setups

1) *Databases*: Six publicly available natural image quality databases are used for experimental evaluation, including (i) three artificially-distorted sets: LIVE [81], CSIQ [82] and TID2013 [83]; and (ii) three realistically-distorted sets: LIVE-C [84], KonIQ-10k [85] and SPAQ [86]. See below for their details.

- LIVE: 29 pristine images each of which is degraded by 5 distortions at four to five different levels of distortion, resulting in 982 distorted images.
- CSIQ: 30 pristine images and 866 distorted images with 6 distortion types at 4 to 5 distortion levels.
- TID2013: 25 pristine images and a total of 3,000 distorted images with 17 distortion types at 4 degradation levels.
- LIVE-C: 1,162 realistically natural pictures with resized resolution of 500×500 pixels.
- KonIQ-10k: 10,073 realistically and complexly distorted images with resolution of 1024×768 pixels.
- SPAQ: 11,125 images captured by 66 smartphones with diverse resolutions.

2) *Evaluation Metrics*: Three commonly-used evaluation metrics are adopted for performance comparison: Spearman Rank Order Correlation Coefficient (SROCC) measuring prediction monotonicity; Pearson Linear Correlation Coefficient (PLCC) measuring linear correlation; and Root Mean Square Error (RMSE) measuring prediction accuracy. An effective IQA metric should yield high PLCC and SROCC values along with low RMSE values.

3) *Implementation Details*: The overall framework including training and testing stages of LBPs-based NR-IQA methods are described in Fig. 1. Following [75], the random forest is used as the regressor. It is implemented by TreeBagger of MATLAB with 50 trees and the optimization search method is not used. The experimental results are generated from the laptop with Intel (R) Xeon (R) e5-2690 V4 processor at 2.60 GHz with MATLAB. LBP codes are implemented in this paper

except for LBP, CLBP, LPQ, AECLBP and AELTP whose codes are provided by the authors. For LBP variants designed for the gray images, we expand it to three-channel version for color images by concatenating LBP features for taking three images of one channel as input separately.

Following [87], [88], we randomly sample 80% of the images in each database for training and leave the rest for testing. Specifically, for synthetically distorted datasets, we split the training and test sets according to the pristine images such that the content is not intersected between the two sets. For performance comparison, the average values of evaluation metrics across one hundred sessions on five-fold division are reported for reducing the randomness of data partition.

For parameter settings for LBP variants used in IQA methods, the individual default values in the original papers are adopted if they are provided, otherwise the radius and the number of neighbor points are taken as 1 and 8 experimentally. Concretely, for example, the radius of LBP is 1 and the number of neighbors is 8. The threshold of LTP is 5. The window size of LPQ is set to 3×3 . MSLBP sets the radius to 1 and the number of neighbors to 4,8,8, the radius to 2 and the number of neighbors to 4,8,16, the radius to 3 and the number of neighbors to 4,8,16,24, and 9 histograms using the "riu2" strategy are concatenated. MS-SLBP combines 9 SLBP histograms with the "riu2" strategy into a feature vector by concatenating the radius 1 with 4,8,8 neighbors, radius 2 with 4,8,16 neighbors, and radius 3 with 4,8,16,24 neighbors. Pi-LBP chooses the number of samples on the path to be 3. The standard deviation σ of the PLBP is set to 0.5 for the low-pass filter, and a downsampling rate of 2, and a 4-stage pyramid were used. In LGBP, five scales and eight orientations Gabor filters are used. In the experimental tables, LBP means basic operator using "riu2" strategy.

B. Performance Evaluation on Individual Databases

Two well-known traditional handcrafted NR-IQA models NIQE [89] and ILNIQE [90] are selected for performance comparison. We first conduct evaluation on individual databases including both synthetic and authentic distortions. The results on synthetically-distorted datasets (*i.e.* LIVE, CSIQ and TID2013) are reported in Table V. The results show almost all LBP variants can be consistently and successfully applied to IQA on the LIVE database, except dLBP which performed poorer than other operators with mean SROCC below 0.6. LVP based on local energy achieved the best performance on LIVE. Most operators also demonstrated acceptable performance on CSIQ compared to NIQE, with WLBD using Weber's law performing best. However, on the more challenging TID2013 dataset, with complex image contents and distortion types, almost all operators struggled to perform consistently well. pi-LBP with strong descriptive capacity performed best on TID2013. dLBP performed worst on CSIQ and TID2013 since the decorrelation reduces distortion sensitivity. Notably, multiscale strategies and gradient computation improved performance on CSIQ and TID2013.

The results on authentically-distorted databases (*i.e.* LIVE-C, KonIQ-10k and SPAQ) are reported in Table VI. The results

TABLE V: Performance comparison on synthetically-distorted databases.

Database	LIVE		CSIQ		TID2013	
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
NIQE	0.9072	0.9060	0.7268	0.6232	0.5432	0.5787
ILNIQE	0.9123	0.9071	0.8736	0.8325	0.5982	0.5704
LBP	0.9225	0.9239	0.8235	0.7742	0.6237	0.5277
ALBP	0.8692	0.8828	0.7748	0.6998	0.6321	0.5483
MLBP	0.8332	0.8545	0.8244	0.7732	0.6270	0.5256
CLBP	0.9318	0.9384	0.8546	0.7996	0.6894	0.6244
DLBP	0.8165	0.8417	0.7751	0.7073	0.5296	0.3659
MSLBP	0.9196	0.9208	0.8311	0.7814	0.6922	0.6060
PLBP	0.9228	0.9232	0.8398	0.7934	0.6534	0.5514
OCLBP	0.9289	0.9406	0.8760	0.8379	0.6881	0.6133
WDLBP	0.9158	0.9220	0.8207	0.7620	0.6904	0.6015
LGBP	0.8879	0.8999	0.7854	0.7297	0.5971	0.5174
AELBP	0.8743	0.8943	0.8223	0.7421	0.6520	0.5721
pi-LBP	0.9280	0.9302	0.8557	0.8142	0.7154	0.6590
dLBP	0.5630	0.5565	0.4805	0.3755	0.4689	0.3657
SLBP	0.9386	0.9425	0.8316	0.7846	0.6851	0.5957
MS-SLBP	0.8946	0.9073	0.8063	0.7472	0.6679	0.5694
LTP	0.8429	0.8591	0.7970	0.7364	0.6266	0.5328
AELTP	0.8064	0.8269	0.7940	0.7190	0.6215	0.5307
LPQ	0.9032	0.9192	0.7935	0.7376	0.6723	0.6052
LGP	0.9258	0.9349	0.8226	0.7757	0.6312	0.5473
CLBC	0.9134	0.9174	0.8444	0.7959	0.6409	0.5419
ELDP	0.9067	0.9180	0.8748	0.8347	0.6829	0.6041
LVP	0.9380	0.9448	0.8763	0.8424	0.7119	0.6242
CLDP	0.9215	0.9262	0.8214	0.7621	0.6616	0.5587
OCPP	0.8383	0.8442	0.7908	0.7608	0.6099	0.5191
ILQP	0.9145	0.9205	0.8616	0.8216	0.6841	0.6008
WLBD	0.9302	0.9350	0.8833	0.8547	0.6757	0.6065

indicate all variants struggled with the authentic data, likely due to the extremely complex distortions and diverse image contents. However, all variants still outperformed standard LBP on real-world data. In this case, saliency-based and adjacent evaluation strategies proved helpful for assessment, while gradient computation did not improve performance as much. Additionally, CLBP demonstrated superior performance over all other operators on the three datasets.

C. Cross-database Evaluation

We performed the cross evaluation to investigate the generality of LBP variants on IQA tasks. This evaluation uses all images from one database to train the methods and to test all images in other databases. Table VII presents the SROCC values for synthetic data and authentic data. We only show some LBP variants with good performance when testing on six datasets individually. From the results, in terms of SROCC and PLCC, we can see that WLBD and CLBP outperform other methods on LIVE and LIVE-C, respectively. However, the best results are not satisfied due to the limitation of the representation power of LBP and the regression power of the random forest.

D. Comparison on Diverse Distortion Types

It is important to analyze LBP-based IQA metric performance across different distortion types. For this purpose, we

TABLE VI: Performance comparison on authentically-distorted databases.

Database	LIVE-C		Koniq-10k		SPAQ	
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
NIQE	0.5147	0.4643	0.5975	0.6014	0.7123	0.7032
ILNIQE	0.5080	0.4320	0.5230	0.5070	0.7214	0.7141
LBP	0.4369	0.3760	0.3941	0.3890	0.6989	0.6984
ALBP	0.6165	0.5710	0.7228	0.7138	0.7633	0.7603
MLBP	0.4900	0.4578	0.5932	0.5937	0.7422	0.7403
CLBP	0.6215	0.6019	0.7852	0.7742	0.8216	0.8172
DLBP	0.3482	0.3217	0.3568	0.3540	0.6499	0.6491
MSLBP	0.5656	0.5340	0.6321	0.6239	0.7614	0.7599
PLBP	0.5780	0.5505	0.6726	0.6599	0.7866	0.7841
OCLBP	0.5992	0.5796	0.6983	0.7061	0.7745	0.7740
WDLBP	0.5890	0.5491	0.6787	0.6653	0.7589	0.7571
LGBP	0.5382	0.5183	0.6851	0.6516	0.8065	0.8008
AELBP	0.6302	0.5874	0.7810	0.7615	0.8174	0.8129
pi-LBP	0.5835	0.5593	0.6948	0.6898	0.8026	0.7843
dLBP	0.4775	0.4645	0.7009	0.6715	0.7759	0.7615
SLBP	0.6158	0.5969	0.7054	0.7032	0.8105	0.7963
MS-SLBP	0.5691	0.5513	0.6975	0.6832	0.8042	0.7846
LTP	0.5529	0.5324	0.7108	0.7013	0.7497	0.7471
AELTP	0.5628	0.5203	0.7107	0.6890	0.7778	0.7748
LPQ	0.5562	0.5293	0.6918	0.6853	0.7644	0.7631
LGP	0.5127	0.4577	0.6069	0.5977	0.7482	0.7464
CLBC	0.5626	0.5426	0.6876	0.6820	0.7420	0.7416
ELDP	0.5477	0.5398	0.6048	0.6122	0.7047	0.7053
LVP	0.5972	0.5883	0.6771	0.6822	0.7586	0.7571
CLDP	0.5837	0.5560	0.7098	0.6973	0.7901	0.7859
OCPP	0.4522	0.4382	0.6377	0.6267	0.8015	0.8004
ILQP	0.6218	0.6084	0.7209	0.7224	0.7567	0.7546
WLBD	0.5811	0.5656	0.6757	0.6799	0.7504	0.7484

TABLE VII: Cross-database Validation.

	Train	Test	PLCC	SROCC	RMSE
LBP	TID2013	LIVE	0.4561	0.2859	39.5935
CLBP	TID2013	LIVE	0.5610	0.5620	39.5422
MSLBP	TID2013	LIVE	0.2186	0.0183	39.8800
PLBP	TID2013	LIVE	0.4965	0.4428	39.7700
Pi-LBP	TID2013	LIVE	0.5632	0.5990	39.4966
OCLBP	TID2013	LIVE	0.4849	0.4939	39.5074
LVP	TID2013	LIVE	0.5533	0.5757	39.3279
WLBD	TID2013	LIVE	0.6777	0.6886	39.3439
LBP	KoniQ_10k	LIVE_C	0.2741	0.2606	2.4938
CLBP	KoniQ_10k	LIVE_C	0.5411	0.5043	2.6038
MSLBP	KoniQ_10k	LIVE_C	0.3155	0.2940	2.4682
PLBP	KoniQ_10k	LIVE_C	0.3388	0.3117	2.3620
Pi-LBP	KoniQ_10k	LIVE_C	0.3956	0.3486	2.7598
OCLBP	KoniQ_10k	LIVE_C	0.4875	0.4537	2.6769
LVP	KoniQ_10k	LIVE_C	0.4960	0.4588	2.6753
WLBD	KoniQ_10k	LIVE_C	0.5205	0.4876	2.7012

evaluated the performance of each distortion type individually using the same protocol applied to the full datasets. Table VIII summarizes the results on the CSIQ database, which contains common degradation types. The per-distortion CSIQ results show WDLBP performed best for AWGN, pi-LBP for BLUR, OCPP for CONTRAST, OCLBP for FNOISE and JPEG, and LVP for JPEG2k. As expected, dLBP struggled on most types due to decorrelation losing distortion sensitivity. Overall, most variants succeeded on AWGN and BLUR, while JPEG and JPEG2k proved less challenging for operators. Gradient-

based variants performed poorly on contrast due to descriptor invariance. Considering performance across datasets and CSIQ distortions, we evaluated standard LBP plus top variants (OCLBP, pi-LBP, WLBD) on 24 TID2013 degradations. As Table IX shows, the OCLBP, pi-LBP and WLBD descriptors presented superior performance on several distortion types.

E. Computational Cost

The computational complexity is evaluated by comparing their running time. The testing time for LBP-based NR-IQA methods is important when facilitating their use in real-time image quality prediction. The time complexity of general NR-IQA methods includes the time consumed by feature extraction and score prediction, where the former costs the most time. In this experiment, we mainly consider the time consumed by extracting local statistical features by LBP operators. We only calculate the time (in seconds) consumed from reading the image to generating the image feature vector. We randomly chose 100 test images from the LIVE-C dataset with size 500×500 and recorded the average time over images. Then the average time over 100 runs is reported in Table X. It can be seen that the execution speeds of MS-LBP, pi-LBP, dLBP, MS-SLBP and LGBP are slower than those of other texture descriptors. This is mainly because of special time-consuming operations such as the DLBP decorrelation operation, the multi-path calculation of pi-LBP and the filtering of 40 Gabor filters of LGBP. Other than that, the rest of the texture descriptors take little time to execute and are all suitable for real-time applications in terms of time performance.

F. Basic Parameters Analysis

We test some basic but key parameters used mostly in LBP variants for assessing the qualities of real-world images. One is the radius length R of sampling. R indicates the texture scale that the texture descriptor can describe, and too large or too small radius may lead to distortion fluctuations that cannot be detected in the image, so it can affect the performance of IQA methods. Fig. 23 shows the SROCC values on different sampling radii for the LBP^{riu2} operator with $P = 8$ on the LIVE-C database. A total of 12 radius values from 1 to 12 are tested. From the results, we can see that the LBP operator with a radius of 1 has the highest performance, probably because the distortion of the image is more sensitive to feature variations with more complete information for the local region. When the P is fixed, smaller R means more concentrated and more completed information can be obtained. As a result, the parameter setting of $R = 1$ is used in the performance comparison experiment for LBP variants without default parameter settings.

Another is the number of sampled neighbors P . P represents the number of times when the central point of LBP needs to be compared with the surrounding points, and it decides the length of the resulting feature vector. If this value is small, the descriptive power of the LBP operators will be reduced and it is likely that the distortion will not be detected and quantized apparently. Bigger P means utilizing more information in each local region but leading to redundant information and

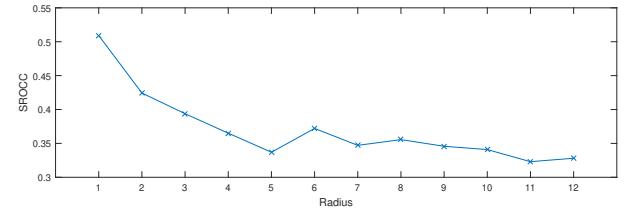


Fig. 23: Performance of the LBP operator with the different sampling radius on LIVE-C.

length of features. Fig. 24 shows the SROCC values of the LBP^{riu2} operator with $R = 1$ on the LIVE-C database for different number of sampled neighborhood points. A total of 13 sampling point numbers from 4 to 16 are tested. From the results, we can see that the performance has a tend to increase as P . Considering the trade-off between the description ability of the operator and the feature length, P is set to 8 in the performance comparison experiment for LBP variants without default parameter settings.

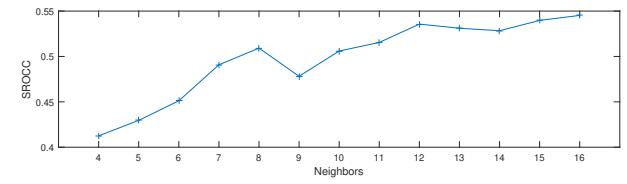


Fig. 24: Performance of LBP algorithms with different radius on LIVE-C.

IV. CONCLUSION

In this paper, we first provide a brief review of 30 variants of the basic LBP as texture descriptors. We then investigate their performance when applied to NR-IQA under synthetic and authentic distortion cases. From the results, we verify whether LBP variants can serve as effective feature descriptors for IQA applications and determine which characteristics of LBP operators are most suitable for IQA. Results demonstrate that multiscale approaches and gradient-based methods substantially improve quality prediction performance. This work aims to familiarize the scientific community and support new researchers with past and current local binary pattern methodologies used in different fields. Moving forward, we will integrate the characteristics most beneficial to IQA to design an improved LBP operator for quality prediction in real-world cases.

REFERENCES

- [1] H. K. Hsu, C. H. Yao, Y. H. Tsai, W. C. Hung, H. Y. Tseng, M. Singh, and M. H. Yang, "Progressive domain adaptation for object detection," 2020.
- [2] L. Kong, A. Ikusan, R. Dai, and J. Zhu, "Blind image quality prediction for object detection," 2019.
- [3] L. Kong, A. Ikusan, R. Dai, J. Zhu, and D. Ros, "A no-reference image quality model for object detection on embedded cameras," *International Journal of Multimedia Data Engineering and Management*, vol. 10, 2019.
- [4] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, 2020.

TABLE VIII: SROCC comparison of NR-IQA methods based on different LBP variants on individual distortion type of CSIQ.

Distortion Type	LBP	ALBP	MLBP	CLBP	DLBP	MSLBP	PLBP	OCLBP	WDLBP	LGBP	AELBP	pi-LBP	dLBP
AWGN	0.8473	0.8777	0.8909	0.9208	0.8169	0.8725	0.9209	0.9196	0.9443	0.8196	0.9298	0.9420	0.8452
BLUR	0.9086	0.8865	0.8187	0.9020	0.8849	0.9071	0.8973	0.9184	0.8822	0.8797	0.8970	0.9318	0.7871
CONTRAST	0.5025	0.6789	0.5755	0.5892	0.6600	0.4796	0.6175	0.5851	0.6119	0.5999	0.2945	0.5472	0.5300
FNOISE	0.7373	0.6172	0.7537	0.8594	0.6388	0.7817	0.8537	0.9067	0.8763	0.7277	0.7488	0.8380	0.3512
JPEG	0.8828	0.8258	0.8626	0.9253	0.8286	0.8995	0.8894	0.9458	0.8916	0.9306	0.8899	0.9168	0.4997
JPEG2k	0.8600	0.8110	0.7678	0.8914	0.7954	0.8455	0.8492	0.9035	0.8467	0.8703	0.8542	0.8850	0.6546
Distortion Type	SLBP	MS-SLBP	LTP	AELTP	LPQ	LGP	CLBC	ELDP	LVP	CLDP	OCPP	ILQP	WLBD
AWGN	0.8931	0.8351	0.7553	0.9453	0.7956	0.9006	0.8881	0.7915	0.9053	0.8816	0.8043	0.8805	0.9332
BLUR	0.9117	0.8912	0.7735	0.8747	0.9032	0.9012	0.8926	0.9103	0.9284	0.8975	0.6774	0.9086	0.9129
CONTRAST	0.6217	0.5229	0.6957	0.6157	0.5428	0.4254	0.4845	0.4972	0.5874	0.5330	0.7478	0.5997	0.6049
FNOISE	0.8044	0.7194	0.6848	0.8271	0.6906	0.6923	0.8200	0.8250	0.8937	0.7919	0.6194	0.8568	0.8932
JPEG	0.8987	0.8867	0.6700	0.7748	0.9397	0.8884	0.8863	0.9355	0.9434	0.9110	0.7398	0.8937	0.9384
JPEG2k	0.8758	0.8503	0.7106	0.7894	0.8704	0.8496	0.8750	0.9076	0.9079	0.8714	0.7210	0.8676	0.8876

TABLE IX: SROCC comparison of NR-IQA methods based on LBP variants on individual distortion type of TID2013.

Distortion Type	LBP	CLBP	MSLBP	PLBP	OCLBP	pi-LBP	LVP	WLBD
Additive Gaussian noise	0.7254	0.7691	0.7458	0.8746	0.8297	0.9293	0.8525	0.8465
Noise in color comp.	0.6340	0.7851	0.7907	0.8536	0.8665	0.8634	0.7730	0.8014
Spatially corr. noise	0.7032	0.8206	0.7592	0.8158	0.8869	0.8882	0.8674	0.8462
Masked noise	0.7739	0.7139	0.6923	0.7504	0.8312	0.8124	0.7762	0.7892
High frequency noise	0.7574	0.8917	0.8780	0.9101	0.9028	0.9206	0.8398	0.9187
Impulse noise	0.5853	0.8652	0.7896	0.8509	0.8473	0.8109	0.7060	0.8219
Quantization noise	0.8549	0.8947	0.8163	0.8472	0.8400	0.8293	0.8386	0.8420
Gaussian blur	0.9239	0.8840	0.9152	0.9147	0.8949	0.9043	0.9486	0.9106
Image denoising	0.6630	0.8566	0.7822	0.7408	0.7873	0.7556	0.8163	0.7440
JPEG compression	0.8098	0.8997	0.8108	0.8600	0.9210	0.8973	0.9200	0.9318
JPEG2000 compression	0.8803	0.8883	0.8799	0.8897	0.9211	0.9104	0.9414	0.8843
JPEG trans. errors	0.5531	0.5803	0.5920	0.5563	0.5130	0.4991	0.5772	0.5295
JPEG2000 trans.errors	0.6287	0.7152	0.6873	0.7583	0.7778	0.7709	0.7662	0.7438
Non ecc. patt. noise	0.1137	0.2083	0.1163	0.1222	0.1729	0.2380	0.1538	0.0909
Local block-wise dist.	0.2167	0.1473	0.1745	0.2907	0.2430	0.3736	0.2417	0.4806
Mean shift	0.4427	0.3711	0.4866	0.4098	0.4142	0.2924	0.3658	0.5121
Contrast change	0.4777	0.4127	0.5633	0.5389	0.4394	0.7423	0.4154	0.6630
Color saturation change	0.2290	0.3223	0.3576	0.1532	0.7478	0.2152	0.2535	0.3232
Multiplicative Gaussian noise	0.6334	0.7280	0.7223	0.8102	0.7587	0.8420	0.7685	0.8677
Comfort noise	0.6589	0.7218	0.7690	0.7551	0.7331	0.6889	0.7395	0.7505
Lossy com. of noisy images	0.7903	0.7917	0.8124	0.8332	0.8651	0.8353	0.8446	0.8643
Color quantization with dither	0.8319	0.8505	0.8342	0.8453	0.8258	0.8762	0.8090	0.8804
Chromatic aberrations	0.6139	0.6321	0.6610	0.6007	0.7144	0.7024	0.7064	0.7937
Sparse samp. and recons.	0.8418	0.9023	0.8536	0.8448	0.9041	0.9050	0.9010	0.9040

TABLE X: Computational time (in seconds) comparison

Operator	Time	Operator	Time	Operator	Time
LBP	0.0330	LGBP	0.3194	LGP	0.2799
ALBP	0.0905	AELBP	0.0552	CLBC	0.0385
MLBP	0.0917	pi-LBP	0.9757	ELDP	0.0351
CLBP	0.0982	dLBP	0.3308	LVP	0.0409
DLBP	0.0291	SLBP	0.0084	CLDP	0.1341
MSLBP	0.3841	MS-SLBP	0.2281	OCPP	0.0637
PLBP	0.0371	LTP	0.0574	ILQP	0.2021
OCLBP	0.0337	AELTP	0.0716	WLBD	0.0679
WDLBP	0.0090	LPQ	0.0181		

- [5] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, “Magface: A universal representation for face recognition and quality assessment,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 225–14 234.
- [6] A. Khodabakhsh, M. Pedersen, and C. Busch, “Subjective versus objective face image quality evaluation for face recognition,” 2019.
- [7] K. Chen, Y. Wu, Z. Li, Y. Wu, and D. Liang, “Face image quality

- assessment for model and human perception,” 2020.
- [8] E. Fourati, W. Elloumi, and A. Chetouani, “Anti-spoofing in face recognition-based biometric authentication using image quality assessment,” *Multimedia Tools and Applications*, vol. 79, 2020.
- [9] X. Liu, D. Zhai, J. Zhou, S. Wang, D. Zhao, and H. Gao, “Sparsity-based image error concealment via adaptive dual dictionary learning and regularization,” *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 782–796, 2017.
- [10] K. Gu, D. Tao, J.-F. Qiao, and W. Lin, “Learning a no-reference quality assessment model of enhanced images with big data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1301–1313, 2018.
- [11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–12, 2004.
- [12] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “Fsim: A feature similarity index for image quality assessment,” *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [13] H. W. Chang, H. Yang, Y. Gan, and M. H. Wang, “Sparse feature fidelity for perceptual image quality assessment,” *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4007–18, 2013.
- [14] Z. Zhou, J. Li, Y. Quan, and R. Xu, “Image quality assessment using kernel sparse coding,” *IEEE Trans. on Multimedia*, vol. 23, pp. 1592–

- 1604, 2020.
- [15] L. Ma, S. Li, and K. N. Ngan, "Reduced-reference video quality assessment of compressed video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 10, pp. 1441–1456, 2012.
- [16] J. Wu, W. Lin, G. Shi, L. Li, and Y. Fang, "Orientation selectivity based visual pattern for reduced-reference image quality assessment," *Information Sciences*, vol. 351, pp. 18–29, 2016.
- [17] Y. Liu, G. Zhai, K. Gu, X. Liu, D. Zhao, and W. Gao, "Reduced-reference image quality assessment in free-energy principle and sparse representation," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 379–391, 2018.
- [18] Z. Wang, A. C. Bovik, and B. L. Evans, "Blind measurement of blocking artifacts in images," in *Proc. Int. Conf. Image Process.*, 2000, pp. 981–984 vol.3.
- [19] H. R. Sheikh, A. C. Bovik, and L. Cormack, "No-reference quality assessment using natural scene statistics: Jpeg2000," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1918–1927, 2005.
- [20] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3350–3364, 2011.
- [21] X. Gao, F. Gao, D. Tao, and X. Li, "Universal blind image quality assessment metrics via natural scene statistics and multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, 2013.
- [22] I. F. Nizami, M. ur Rehman, M. Majid, and S. M. Anwar, "Natural scene statistics model independent no-reference image quality assessment using patch based discrete cosine transform," *Multimedia Tools and Applications*, 2020.
- [23] F. Z. Ou, Y. G. Wang, and G. Zhu, "A novel blind image quality assessment method based on refined natural scene statistics," vol. 2019-September, 2019.
- [24] Z. Wan, K. Gu, and D. Zhao, "Reduced reference stereoscopic image quality assessment using sparse representation and natural scene statistics," *IEEE Transactions on Multimedia*, vol. 22, 2020.
- [25] A. Sadiq, I. F. Nizami, S. M. Anwar, and M. Majid, "Blind image quality assessment using natural scene statistics of stationary wavelet transform," *Optik*, vol. 205, 2020.
- [26] Y. Huang and K. Yao, "Multi-exposure image fusion method based on independent component analysis," 2020.
- [27] C. Zhang, J. Xu, X. Huang, and S. H. Park, "No-reference image quality assessment using independent component analysis and convolutional neural network," *Journal of Electrical Engineering and Technology*, vol. 14, 2019.
- [28] Z. Zhou, J. Li, Y. Quan, and R. Xu, "Image quality assessment using kernel sparse coding," *IEEE Transactions on Multimedia*, vol. 23, 2021.
- [29] X. Yang, Q. Sun, and T. Wang, "No-reference image quality assessment based on sparse representation," *Neural Computing and Applications*, vol. 31, 2019.
- [30] S. Chakraborty and A. S. Jalal, "A novel local binary pattern based blind feature image steganography," *Multimedia Tools and Applications*, vol. 79, 2020.
- [31] M. Hassaballah, H. A. Alshazly, and A. A. Ali, "Ear recognition using local binary patterns: A comparative experimental study," *Expert Systems with Applications*, vol. 118, 2019.
- [32] C. Qin, Y. Hu, H. Yao, X. Duan, and L. Gao, "Perceptual image hashing based on weber local binary pattern and color angle representation," *IEEE Access*, vol. 7, 2019.
- [33] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002.
- [34] T. Song, Y. Han, J. Feng, Y. Wang, and C. Gao, "First- and second-order sorted local binary pattern features for grayscale-inversion and rotation invariant texture classification," 2020.
- [35] A. R. Masrat and Kaur, "A novel rotation invariant descriptor for texture classification with local binary patterns," V. Kamakshi, W. Jiacun, R. K. T. V. R. V. Sivakumar, and Prasad, Eds. Springer Singapore, 2021, pp. 385–396.
- [36] Z. Pan, S. Hu, X. Wu, and P. Wang, "Adaptive center pixel selection strategy in local binary pattern for texture classification," *Expert Systems with Applications*, vol. 180, 2021.
- [37] S. Karanwal and M. Diwakar, "Od-lbp: Orthogonal difference-local binary pattern for face recognition," *Digital Signal Processing: A Review Journal*, vol. 110, 2021.
- [38] D. G. R. Kola and S. K. Samayamantula, "A novel approach for facial expression recognition using local binary pattern with adaptive window," *Multimedia Tools and Applications*, vol. 80, 2021.
- [39] B. Ahuja and V. P. Vishwakarma, "Local binary pattern based elm for face identification," vol. 1164, 2021.
- [40] R. R. Isnanto, A. F. Rochim, D. Eridani, and G. D. Cahyono, "Multi-object face recognition using local binary pattern histogram and haar cascade classifier on low-resolution images," *International Journal of Engineering and Technology Innovation*, vol. 11, 2021.
- [41] H. Yin, Y. Chen, J. Xiong, R. Xia, J. Xie, and K. Yang, "An improved local binary pattern method for pollen image classification and recognition," *Computers and Electrical Engineering*, vol. 90, 2021.
- [42] P. G. Freitas, W. Y. L. Akamine, and M. C. Q. D. Farias, "Blind image quality assessment using local variant patterns," vol. 2018-January, 2017.
- [43] W. Zhou, L. Yu, W. Qiu, Y. Zhou, and M. Wu, "Local gradient patterns (lgp): An effective local-statistical-feature extraction scheme for no-reference image quality assessment," *Information Sciences*, vol. 397–398, 2017.
- [44] M. Hassaballah, H. A. Alshazly, and A. A. Ali, "Ear recognition using local binary patterns: A comparative experimental study," *Expert Systems with Applications*, vol. 118, pp. 182–200, 2019.
- [45] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," vol. 4633 LNCS, 2007.
- [46] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, 2010.
- [47] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Transactions on Image Processing*, vol. 18, pp. 1107–1118, 2009.
- [48] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [49] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "Plbp: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, pp. 2502–2515, 2011, semi-Supervised Learning for Visual Content Analysis and Understanding.
- [50] T. Mäenpää, "The local binary pattern approach to texture analysis: Extensions and applications," *Oulu Yliopisto: Oulu, Finland*, 2003.
- [51] K. Song, Y. Yan, Y. Zhao, and C. Liu, "Adjacent evaluation of local binary pattern for texture classification," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 323–339, 2015.
- [52] Q. Lin and W. Qi, "Multi-scale local binary patterns based on path integral for texture classification," 2015, pp. 26–30.
- [53] R. Hu, X. Li, and Z. Guo, "Decorrelated local binary patterns for efficient texture classification," *Multimedia Tools and Applications*, vol. 77, pp. 6863–6882, 2018.
- [54] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," vol. 5099 LNCS, 2008.
- [55] Y. Zhao, D.-S. Huang, and W. Jia, "Completed local binary count for rotation invariant texture classification," *IEEE Transactions on Image Processing*, vol. 21, pp. 4492–4497, 2012.
- [56] Y. Hu, Z. Long, and G. AlRegib, "Completed local derivative pattern for rotation invariant texture classification," 2016, pp. 3548–3552.
- [57] L. Armi and S. Fekri-Ershad, "Texture image classification based on improved local quinary patterns," *Multimedia Tools and Applications*, vol. 78, pp. 18 995–19 018, 2019. [Online]. Available: <https://doi.org/10.1007/s11042-019-7207-2>
- [58] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, 2010.
- [59] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition," vol. 1, 2005, pp. 786–791 Vol. 1.
- [60] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using local phase quantization," in *International conference on pattern recognition*. IEEE, 2008, pp. 1–4.
- [61] B. Jun and D. Kim, "Robust face detection using local gradient patterns and evidence accumulation," *Pattern Recognition*, vol. 45, no. 9, pp. 3304–3316, 2012.
- [62] M. R. Faraji and X. Qi, "Face recognition under illumination variations based on eight local directional patterns," *IET Biometrics*, vol. 4, 2015.
- [63] I. Nenakhov, V. Khryashchev, and A. Priorov, "No-reference image quality assessment based on local binary patterns," 2016.
- [64] P. G. Freitas, W. Y. Akamine, and M. C. Farias, "No-reference image quality assessment based on statistics of local ternary pattern," 2016.
- [65] ———, "Blind image quality assessment using multiscale local binary patterns," *Journal of Imaging Science and Technology*, vol. 60, no. 6, pp. 60 405–1, 2016.

- [66] F. Rezaie, M. S. Helfroush, and H. Danyali, “No-reference image quality assessment using local binary pattern in the wavelet domain,” *Multimedia Tools and Applications*, vol. 77, pp. 2529–2541, 2018.
- [67] P. G. Freitas, W. Y. L. Akamine, and M. C. Q. D. Farias, “No-reference image quality assessment using salient local binary patterns.” Society for Imaging Science and Technology, 2018.
- [68] P. G. Freitas, S. Alamgeer, W. Y. Akamine, and M. C. Farias, “Blind image quality assessment based on multiscale salient local binary patterns,” 2018.
- [69] P. G. Freitas, L. P. da Eira, S. S. Santos, and M. C. Farias, “Image quality assessment using bsif, clbp, lcp, and lpq operators,” *Theoretical Computer Science*, vol. 805, 2020.
- [70] P. G. Freitas, W. Y. L. Akamine, and M. C. Q. Farias, “No-reference image quality assessment using orthogonal color planes patterns,” *IEEE Transactions on Multimedia*, vol. 20, pp. 3353–3360, 2018.
- [71] P. G. Freitas and M. C. Q. D. Farias, “On the performance of visual semantics for improving texture-based blind image quality assessment,” 2017, pp. 330–337.
- [72] P. Srivastava and A. Khare, “Utilizing multiscale local binary pattern for content-based image retrieval,” *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12 377–12 403, 2018.
- [73] Z. Xia, C. Yuan, R. Lv, X. Sun, N. N. Xiong, and Y.-Q. Shi, “A novel weber local binary descriptor for fingerprint liveness detection,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, pp. 1526–1536, 2020.
- [74] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local binary patterns and its application to facial image analysis: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 765–781, 2011.
- [75] P. G. Freitas, L. P. Da Eira, S. S. Santos, and M. C. Q. de Farias, “On the application lbp texture descriptors and its variants for no-reference image quality assessment,” *Journal of Imaging*, vol. 4, no. 10, 2018.
- [76] A. Ramola, A. K. Shakya, and D. Van Pham, “Study of statistical methods for texture analysis and their modern evolutions,” *Engineering Reports*, vol. 2, no. 4, p. e12149, 2020.
- [77] L. Du, X. You, H. Xu, Z. Gao, and Y. Tang, “Wavelet domain local binary pattern features for writer identification,” in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 3691–3694.
- [78] M. Belahcene, M. Laid, A. Chouchane, A. Ouamane, and S. Bourennane, “Local descriptors and tensor local preserving projection in face recognition,” in *European workshop on visual information processing*. IEEE, 2016, pp. 1–6.
- [79] F. Zhong and J. Zhang, “Face recognition with enhanced local directional patterns,” *Neurocomputing*, vol. 119, pp. 375–384, 2013.
- [80] L. Nanni, A. Lumini, and S. Brahnam, “Local binary patterns variants as texture descriptors for medical image analysis,” *Artificial Intelligence in Medicine*, vol. 49, 2010.
- [81] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik, “Live image quality assessment database release 2,” [Http://Www.Live.Ece.Utexas.Edu/Research/Quality](http://Www.Live.Ece.Utexas.Edu/Research/Quality), 2019.
- [82] D. M. Chandler, “Most apparent distortion: full-reference image quality assessment and the role of strategy,” *Journal of Electronic Imaging*, vol. 19, 2010.
- [83] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C. C. J. Kuo, “Image database tid2013: Peculiarities, results and perspectives,” *Signal Processing: Image Communication*, vol. 30, 2015.
- [84] D. Ghadiyaram and A. C. Bovik, “Massive online crowdsourced study of subjective and objective picture quality,” *IEEE Transactions on Image Processing*, vol. 25, 2016.
- [85] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe, “Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment,” *IEEE Transactions on Image Processing*, vol. 29, 2020.
- [86] Y. Fang, H. Zhu, Y. Zeng, K. Ma, and Z. Wang, “Perceptual quality assessment of smartphone photography,” 2020.
- [87] H. Talebi and P. Milanfar, “NIMA: Neural image assessment,” *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3998–4011, 2018.
- [88] Y. Su and J. Korhonen, “Blind natural image quality prediction using convolutional neural networks and weighted spatial pooling,” in *Proc. Int. Conf. Image Process.*, 2020, pp. 191–195.
- [89] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Process. Letters*, vol. 20, no. 3, pp. 209–212, 2012.
- [90] L. Zhang, L. Zhang, and A. C. Bovik, “A feature-enriched completely blind image quality evaluator,” *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2579–2591, 2015.